Lecture Notes in Computer Science 3619

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

Xicheng Lu Wei Zhao (Eds.)

Networking and Mobile Computing

Third International Conference, ICCNMC 2005
Zhangjiajie, China, August 2-4, 2005
Proceedings

Springer

Volume Editors

Xicheng Lu
National University of Defense Technology
Changsha, Hunan 410073, China
E-mail: xclu@nudt.edu.cn

Wei Zhao
National Science Foundation
Computer and Network Systems Division
CISE, Room 1175.01, 4201 Wilson Boulevard, Arlington, VA 22230, USA
E-mail: wzhao@nsf.gov

Library of Congress Control Number: 2005929607

CR Subject Classification (1998): C.2, D.4.4, D.2, H.3.5, H.4, K.4.4, K.6.5

ISSN 0302-9743
ISBN-10 3-540-28102-9 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-28102-3 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11534310 06/3142 5 4 3 2 1 0

Preface

Welcome to Zhangjiajie for the 3rd International Conference on Computer Network and Mobile Computing (ICCNMC 2005).

We are currently witnessing a proliferation in mobile/wireless technologies and applications. However, these new technologies have ushered in unprecedented challenges for the research community across the range of networking, mobile computing, network security and wireless web applications, and optical network topics.

ICCNMC 2005 was sponsored by the China Computer Federation, in cooperation with the Institute for Electrical and Electronics Engineers (IEEE) Computer Society. The objective of this conference was to address and capture highly innovative and stateof-the-art research and work in the networks and mobile computing industries. ICCNMC 2005 allowed sharing of the underlying theories and applications, and the establishment of new and long-term collaborative channels aimed at developing innovative concepts and solutions geared to future markets.

The highly positive response to ICCNMC 2001 and ICCNMC 2003, held in Beijing and Shanghai, respectively, encouraged us to continue this international event. In its third year, ICCNMC 2005 continued to provide a forum for researchers, professionals, and industrial practitioners from around the world to report on new advances in computer network and mobile computing, as well as to identify issues and directions for research and development in the new era of evolving technologies.

ICCNMC 2005 was the result of the hard work and planning of a large group of renowned researchers from around the world, who served on the technical Program Committee and the Organizing Committee. Their invaluable efforts in developing this technical program are most gratefully acknowledged. We also would like to take this opportunity to thank our keynote speakers and panelists.

We would like to thank the Program Co-chairs, Prof. Xicheng Lu and Prof. Wei Zhao, for their devotion to ICCNMC 2005. We strongly feel that the interaction between the two working groups in the USA and China was especially important to the success of the conference. To help lay a foundation for a continuing dialogue, three keynote speakers were invited to provide perspectives on different aspects of the challenges we all face.

We would like to express our special gratitude to the National Natural Science Foundation of China. Last but not least, we would also like to take this opportunity to thank our industrial sponsors. Without their extensive and generous supports for both the technical program and the local arrangements, we would not have been able to hold a successful conference at all.

We hope that all of our participants found the conference both stimulating and enjoyable.

July 2005 Chita Das and Hequan Wu

Message from the Program Co-chairs

Welcome to the proceedings of the 2005 International Conference on Computer Networks and Mobile Computing (ICCNMC 2005). This year's conference was the third conference in its series aimed at stimulating technical exchange in the emerging and important fields of mobile, wireless, optical communications networks, and mobile computing.

ICCNMC 2005 followed in the footsteps of its previous conferences in that it addressed in-depth, highly innovative and state-of-the-art research and work in the networks and mobile computing industries. This year's technical program was extremely strong and diverse, with contributions in both established and evolving areas of research. The conference featured three keynote lectures by distinguished academic and industrial leaders and a panel discussion organized by outstanding computer scientists. Furthermore, a total of 662 papers came from over 28 different countries, representing a truly "wide area network" of research activity. The Program Committee engaged in a thorough and careful selection process. Due to the space constraints, only 133 papers were selected as normal papers, and 13 were selected as short presentations. Thus, we produced an excellent conference program that included a wide range of technical challenges in view of the growing interest in network architectures, protocol design and analysis, mobile computing, routing and scheduling, congestion management, quality of service, admission control, Internet and Web applications, multimedia systems, network security, and optical communication technologies.

We would like to express our sincere gratitude to all those individuals whose contributions helped to make ICCNMC 2005 a successful and valuable conference. We were delighted to present Outstanding Service awards to Jiannong Cao, Xiuzhen Cheng, Jinshu Su, Jie Wu, and Ming Xu for their tireless efforts and significant contributions towards organizing the conference. Special thanks are due to Ming T. Liu, our Honorary Chair, whose guidance was always extremely valuable. We also wish to thank the General Co-chairs, Chita Das and Hequan Wu, for their support and contributions. We would like to express our appreciation to all authors for their contributions, to the Program Committee members, and to the external reviewers for their hard work in evaluating submitted papers. Finally, several prestigious organizations, including the China Computer Federation, IEEE Computer Society Beijing Center, IEEE Technical Committee of Distributed Processing, and the Hunan Computer Society, provided valuable endorsement and sponsorship of ICCNMC 2005. We are truly grateful for their contributions.

July 2005 Xicheng Lu and Wei Zhao

Organization

Honorary Chair

Ming T. (Mike) Liu, Ohio State Univ., USA

General Co-chairs

Chita Das, Pennsylvania State Univ., USA
Hequan Wu, Chinese Academy of Engineering, China

Program Co-chairs

Xicheng Lu, National Univ. of Defense Technology, China
Wei Zhao, Texas A&M Univ., USA

Program Vice Co-chairs

Bo Li, Hong Kong Univ. of Science & Technology, Hong Kong, China
Jinshu Su, National Univ. of Defense Technology, China
Jie Wu, Florida Atlantic Univ., USA

Program Committee Members

Giuseppe Anastasi, Univ. of Pisa, Italy
Guohong Cao, Pennsylvania State Univ., USA
Jianer Chen, Texas A&M Univ., USA
Sajal K. Das, Univ. of Texas at Arlington, USA
Alois Ferscha, Univ. of Linz, Austria
Chuanshan Gao, Fudan Univ., China
Zhenghu Gong, National Univ. of Defense Technology, China
Weijia Jia, Hong Kong City Univ., Hong Kong, China
Jie Li, Univ. of Tsukuba, Japan
Xiaoming Li, Peking Univ., China
Prasant Mohapatra, Univ. of California at Davis, USA
Stephan Olariu, Old Dominion Univ., USA
Depei Qian, Xi'an Jiaotong Univ., China
Hualin Qian, Chinese Academy of Sciences, China
Mukesh Singhal, Univ. of Kentucky, USA
Bala Srinivasan, Monash Univ., Australia
Ivan Stojmenovic, Univ. of Ottawa, Canada
Chengzheng Sun, Griffith Univ., Australia
Jianping Wu, Tsinghua Univ., China
Li Xiao, Michigan State Univ., USA
Yuanyuan Yang, State Univ. of New York at Stony Brook, USA
Xiuzhen Cheng, George Washington University, USA

Steering Committee Chair
Benjamin W. Wah, Univ. of Illinois, USA

Publication Chair
Jiannong Cao, Hong Kong Polytechnic Univ., Hong Kong, China
Zhongzhi Shi, Chinese Academy of Sciences, China

Publicity Chair
Cho-li Wang, Univ. of Hong Kong, Hong Kong, China

Awards Chair
Wenhua Dou, National Univ. of Defense Technology, China

Organizing Chair
Laurence T. Yang, St. Francis Xavier Univ., Canada
Ming Xu, National Univ. of Defense Technology, China

IEEE Beijing Section, Director
Zhiwei Xu, Chinese Academy of Sciences, China

Conference Secretary
Shengquan Wang, Texas A&M Univ., USA
Jianjun Bai, National Univ. of Defense Technology, China

Reviewers

Ali Asqar Razavi	Juan luo	Wan Zhang
Aram valartha bharathi	Jun Jiang	Wanjun Huang
Baosheng Wang	Jun Lai	Wei Dong
Beatrice Cynthia	Jun Shen	Wei Feng
Dhinaharan	Kai Lu	Wei Peng
Bin Zhou	Kai Zheng	Wei Ren
Binod Vaidya	Kaiyu Cai	WeiChuan Lin
Binyu Zang	Ke Xu	Weifeng Ma
Bo Liu	KeeYoung Yoo	Weihong Han
Bole Shi	Keqiu Li	Weiping Zhu
Boon-Hee Kim	Kihong Kim	Weiqin Tong
ByoungSeob Park	Kun Zhang	Wenbin Hu
ByoungSon Choi	Kyungjun Kim	Wenguo Wei
Caiping Liu	Laibin Yan	Wenhua Dou
Caixia Liu	Layuan Li	Wenjun Xiao
Celia Li	Lee Hoseung	Wenyong Wang
Changda Wang	Lee Seoung Hyeon	WonSik Yoon
Changjun Jiang	Lei Shi	WooHun Kim
Chao Li	Lei Xuan	WoongChul Choi
ChengYuan Ho	Li Xu	Xian Liu

ChiaCheng Hu	Libao Zhang	Xiang Li
Chi-Hung Chi	Lidong Lin	Xianghui Liu
ChinChen Chang	Lihua Song	Xiangquan Shi
Chuandong Huang	Lin Chen	Xiangyu Wang
Chuanhe Huang	Lin Weiwei	Xiaoan Huang
Chuanshan Gao	Liqiang Zhao	Xiaodong Wang
Chunlin Li	Liquan Xiao	Xiaofeng Hu
Cungang Yang	Liran Ma	Xiaogang Qi
D Manivannan	Llhyung Jung	Xiaolin Lu
Dan Meng	Lu Ruan	Xiaomei Chen
Danli Li	Lu Yan	Xiaoming Li
Danlin Yao	Ma Changshe	Xiaoming Zhang
Dasheng Zhao	Magdy Koutb	Xiaoqiang Xiao
David Liu	Mahdi Jalili	Xiaoshe Dong
Depei Qian	Mahdi Jalilian	Xiaoya Fan
Dhinaharan Nagamalai	ManChing Yuen	Xin Lin
Dianxi shi	Marcin Matuszewski	Xin Xu
Ding-Jyh Tsaur	Marko Hassinen	Xingshe Zhou
Dong Seong Kim	Meng Wang	Xingwei Wang
Dong Xuan	Miae Woo	Xingye Tan
Donghoi Kim	Miao Liu	Xinsheng Xia
Dongsheng Li	Min Song	Xinwen Jiang
Duc Nguyen	Ming Liu	Xiuli Ren
Eric Hsiaokuang Wu	Ming Xu	Xiuzhen Cheng
Eun-Jun Yoon	MingI Hsieh	Xuehui Wang
Feng Gao	Mingjie Zhang	Xuhui Xiong
Feng Wang	Minglu Li	Xukai Zou
Feng Zhao	Mingmei Li	Yajun Guo
Francesco Palmieri	Mingming Lu	Yang Bo
Fuwen Liu	Mingqiao Wu	Yang Chen
Gang Wu	Mingwu Yao	Yang Panlong
Gao Shu	Mingyang Su	Yanhuang Jiang
Gobu Ezhumalai	Minte Sun	Yanmin Zhu
Gooyoun Hwang	Mohammed Houssaini	Yanxiang He
Guangwei Xu	Sqalli	Yaping Liu
Guohong Cao	Mohd Izani Zainal Abidin	Yaya Wei
Haibo Sun	Murugaiyan Aramudhan	Yeonkwon Jeong
Haitao Chen	MyongSoon Park	Yi Sun
Haiying Liang	Na Wang	Yijie Wang
Hengchang Liu	Namje Park	Yimin Chen
Heying Zhang	Ningning Han	Yin zhaolin
Hong Wang	Nirmal Kumar Gunaseelan	Yingchi Mao
Hong-Yong Yang	Nizamuddin Channa	Yiping Yao
Hosang Park	Nong Xiao	Yong Dou
Hsiao-Hong Tsai	Peide Liu	Yong Tang
Hsien-Chou Liao	Peidong Zhu	YongJin Lee
Hu Chen	Peng Hong	Yongjun Wang

Hua Wang	Peng Zhang	Yonglong Luo
Huadong Ma	Pengy Yue	Yongqiang Liu
Hualin Qian	Qianbing Zheng	Yongxian Jin
Huaping Hu	Qiang Dou	Yongxue Wang
Hui Li	Qicheng Liu	Yongzhe Zhao
Hui-Feng Huang	Qingchun Ren	Yonhzhuang Wei
Huizhang Zhao	Qingzhang Chen	Youchan Zhu
Hwashin Moon	Qiuxi Zhong	YoungChul Shim
HyangDuck Cho	Qixing Xu	YounHee Han
Hyun Kyung Cho	Quan Zhou	Yu Fei
Hyung Su Lee	R. Manoharan	Yuan Yang
Hyung-Jin Lim	Ruidong Li	Yuan Zhang
Hyunju Lee	Ruixuan Li	Yuan Zhou
Ivan Stojmenović	Sangseon Byun	Yuanbo Guo
Jae Kwang Lee	Sankaranarayanan Suresh	Yuancheng Wu
Jae-Won Choi	SeungSeob Park	Yuanhong Gu
Javier Paris	Shenghui Su	Yue Chen
Jayesh Seshadri	Shengquan Wang	Yuelong Zhao
Jianer Chen	Shengyi Wu	Yueshan Xiong
Jianfeng Ma	Shijinn Horng	Yugang Mao
Jiang Liu	Shoubao Yang	Yukwong Kwok
Jiangang Shen	Shoubin Dong	Yuxing Peng
Jiangping Yin	Shuangbao Wang	Zexin Lu
Jianhua Yang	Shyhfang Huang	Zhang Xihuang
Jianjun Bai	Siwei Luo	Zhanhei Li
Jiannong Cao	Songqiao Chen	Zhaohui Wu
Jie Chen	Sudha Ramachandra	Zhenghu Gong
Jie Li	Sumi Helal	Zhentao Shi
Jie Ma	SungJe Woo	Zhenyan Ji
Jie Wu	Sungjune hong	Zhenzhou Ji
Jiju M J	Sung-Won Moon	Zhi Jin
Jiman Hong	Sunhun Lee	Zhigang Chen
Jin Wu	Sunil Kim	Zhigang Jin
Jinghui Gao	Sunitha L	Zhigang Luo
JinHee Choi	Syed M.S. Islam	Zhigang Sun
Jinhui Xu	Taeseok Lee	Zhiqiang Zhang
Jinkeun hong	Tao Xie	Zhiying Yao
Jinshu Su	Tarek Guesmi	Ziqiang Wang
Jinsong Han	Teng Lv	Zuhui Yue
Jitae Shin	Tieming Chen	Zunguo Huang
Johannes Karlsson	Toshihiko Yamakami	
Jookyong Lee	Vedat Coskun	

Table of Contents

Session 3: Peer-to-Peer Systems

Session 4: Caching and Routing

Session 5: Wireless Networks

Session 6: Multicast I

Session 7: Ad Hoc Networks I

Session 8: Algorithms I

Session 9: Security I

Session 10: Peer-to-Peer Systems and Web Service

Session 11: Multicast II

Session 12: Traffic and Network Management

Session 13: QoS I

Session 14: Ad Hoc Networks II

Session 15: Routing

Session 16: Algorithms II

Session 17: Security II

Session 18: Internet Application

Session 19: QoS II

Session 20: Security III

Session 21: TCP/IP and Measurement

Session 22: Algorithms III

Session 23: Sensor Networks II

Session 24: Design and Performance Analysis

Session 25: Traffic and Network Management II

Session 26: Agent-Based Algorithms

Session 27: Security Algorithms

Self-organizing Wireless Sensor Networks in Action

John A. Stankovic

Department of Computer Science, University of Virginia, USA
jas9f@virginia.edu

Abstract. Wireless sensor networks (WSN), composed of a large numbers of small devices that self-organize, are being investigated for a wide variety of applications. Two key advantages of these networks over more traditional sensor networks are that they can be dynamically and quickly deployed, and that they can provide fine-grained sensing. Applications, such as emergency response to natural or manmade disasters, detection and tracking, and fine grained sensing of the environment are key examples of applications that can benefit from these types of WSNs. Current research for these systems is widespread. However, many of the proposed solutions are developed with simplifying assumptions about wireless communication and the environment, even though the realities of wireless communication and environmental sensing are well known. Many of the solutions are evaluated only by simulation. In this talk I describe a fully implemented system consisting of a suite of more than 30 synthesized protocols. The system supports a power aware surveillance and tracking application running on 203 motes and evaluated in a realistic, large-area environment. Technical details and evaluations are presented for power management, dynamic group management, and for various system implementation issues. Several illustrations of how real world environments render some previous solutions unusable will also be given.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, p. 1, 2005.

The Internet Control Architecture: Successes and Challenges

Don Towsley

Department of Computer Science, University of Massachusetts, USA
towsley@cs.umass.edu

Abstract. The Internet has evolved into a very robust system that is integral part of our lives today. In large part, this is due to the clever development and engineering of routing algorithms and congestion controllers. In this talk we explore how this came about, focusing on the major changes that have occurred in the Internet control architecture over the years. We also examine the recent development of formal modeling and control frameworks within which to study these problems. These frameworks make us better able to appreciate earlier decisions made during the 80s. At the same time, they also allow us to identify shortcomings in the current architecture. In particular, the current control architecture separates congestion control from routing. We present the development of a new architecture that resolves these shortcomings as a challenge. The talk concludes with some preliminary ideas for such an architecture.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, p. 2, 2005.

Distributed Localization Refinements for Mobile Sensor Networks*

Yanmin Zhu, Min Gao, and Lionel M. Ni

Department of Computer Science,
Hong Kong University of Science and Technology,
Clear Water Bay, Kowloon, Hong Kong
{zhuym, mgao, ni}@cs.ust.hk

Abstract. Location information is crucial for many applications of sensor networks to fulfill their functions. A mobile sensor network is comprised of both mobile and stationary sensors. So far little work has been done to tackle the mobility of sensors in localization for sensor networks. In this paper, we propose the QoL-guided distributed refinements for anchor-free localization in wireless mobile sensor networks. Accuracy is the core concern for localization. We introduce the important concept of Quality of Localization (QoL) to indicate the accuracy of a computed location for a specific sensor node. Our approach is divided into two phases. In Phase one, we propose the algorithm QoL-guided spreading localization with refinements to compute locations for sensor nodes right after the deployment of the sensor network when the mobile sensors are required to stay static temporarily. In Phase two, the non-movement restriction is released and we propose the mobile location self-updating algorithm to update locations of mobile sensors regularly or on demand. Extensive simulations are conducted, which demonstrate that our approach is a promising technique for localization in wireless mobile sensor networks.

1 Introduction

Recent years have witnessed the rapid development of wireless sensor networks (WSN), which promises to revolutionize the way we monitor environments of interest. Many attractive applications, such as habitat monitoring [1], will greatly benefit from WSNs. To achieve the appealing potential, however, location information of sensor nodes is very crucial for many applications of sensor networks. The term *localization* refers to the process of determining the physical location of every sensor node in a sensor network. If sensor nodes fail to obtain their locations, many applications would become infeasible. For instance, for event reporting applications, whenever an event is captured, the corresponding sensor node has to enclose its location in the event to be routed back to the sink; otherwise, the operator has no way to identify where the event occurred. And, many novel routing protocols for sensor networks [2]

* This research was supported in part by Hong Kong RGC Grants HKUST6264/04E and AoE/E-01/99.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 3 – 12, 2005.

all presume that sensor nodes have the knowledge of their locations; otherwise, these protocols would be useless. Therefore, localization is really an important building block in sensor networks.

A lot of approaches have been proposed to provide per-node location information. In general, these approaches can be divided into two categories: anchor-based and anchor-free. An anchor (also known as beacon) in a sensor network is a node which has the priori knowledge of its absolute location, via GPS or manual configuration. Anchor-based approaches require that initially a number of anchors are deployed across the sensor network. The goal is to compute the locations of other sensor nodes by referencing to the anchors. Whereas, anchor-free approaches make no assumption about the availability or validity of anchors, and aim to determine the relative locations of sensor nodes. Although anchor-based approaches can provide absolute location information, they do have many limitations. Introducing GPS receivers for sensor nodes is not feasible due to the extra power consumption and the line-of-sight restriction posed by satellites.

This paper targets mobile wireless sensor networks, and focuses on the anchor-free solution, with the goal of determining the relative locations of sensor nodes. A mobile sensor network comprises both mobile and stationary sensors. To the best of our knowledge, so far little research has been conducted to tackle the mobility of sensors in localization for sensor networks. In this paper, we propose QoL-guided distributed approach for anchor-free localization in mobile sensor networks. Accuracy is the core concern for localization. We introduce the important concept of Quality of Localization (QoL) to indicate the accuracy of a computed location for a specific sensor node. The guidance of QoL throughout the process of localization is very advantageous for accurate localization. Our approach is divided into two phases. In Phase one, we propose the algorithm QoL-guided spreading localization with refinements to compute locations for sensor nodes. This phase takes place right after the deployment of the sensor network when the mobile sensors are required to stay static temporarily. In Phase two, the non-movement restriction is released and the mobile sensors are allowed to move freely. We propose the mobile location self-updating algorithm to update locations of mobile sensors regularly or on demand.

The remainder of the paper is organized as follows. Section 2 discusses the previous work in literature for localization in sensor networks. Section 3 describes the proposed approach in detail. The simulation results are represented in Section 4. Finally, we conclude the paper in Section 5.

2 Related Work

A lot of anchor-based algorithms have been proposed. Bulusu et al. [3] proposed the GPS-less approach, in which an idealized radio model is assumed, and a fixed number of nodes in the network with overlapping regions of coverage are placed as reference points. A connectivity-based localization method was proposed for localization. This approach relies on the availability of the ideal radio model, but the radio signals in real environments are highly dynamic, which reveals the inapplicability. In the DV-hop method [4], initially each anchor floods its location to all nodes in the sensor network. If an unknown node collects the locations of at least three anchors and the

corresponding hop distances to them, this node can compute its location. DV-hop only works well with dense and uniformly distributed sensor networks. Doherty et al. [5] used the connectivity between nodes to formulate a set of geometric constraints and then solved it using convex optimization to compute locations of sensor nodes. One pitfall is that the optimization is performed in a centralized node. Robust Positioning Algorithm [6] proposed a refinement algorithm after sensor nodes get initial location estimates. However, the refinement only utilizes the direct neighbors.

So far only a few anchor-free approaches have been proposed. Capkun, et al. [7] proposed a distributed algorithm for the localization in an ad hoc network without the support of GPS. The algorithm first establishes a local coordinate for each sensor node. These local coordinate systems are then combined to form a relative global coordinate system. Since this algorithm is originally designed for mobile ad hoc networks, other than sensor networks, communication overhead and power consumption were not a concern in the solution. The algorithm makes the first step to anchor-free localization; however, the computation of the local coordinate system is coarse-grained, and some node may fail to be localized if the density is not so high. A cluster-based approach for anchor-free localization proposed in [8] made some improvements over the method presented in [7]. Less communication overhead is introduced and shorter convergence time is needed. Priyantha et al. [9] proposed a decentralized anchor-free algorithm AFL, in which nodes start from a random initial coordinate assignment and converge to a consistent solution. The key idea is fold-freedom, where nodes first configure into a topology that resembles a scaled and unfolded version of the true configuration, and then run a force-based relaxation procedure. This approach is complicated and introduces too much computation.

3 Distributed QoL-Guided Localization

The whole process can be divided into two phases. In Phase one, the sensor network is just deployed, and we require that all the mobile sensors do not move around. The QoL-guided spreading localization with refinements is proposed to compute locations for sensor nodes. In Phase two, the mobile sensors are allowed to move unrestrictedly in the sensor network. The mobile location self-updating algorithm is proposed to update locations of mobile sensors regularly or on demand.

Before performing the localization for the static sensor network, we need an initialization for each sensor node. Initially, each sensor node maintains a list of its neighbors with the corresponding distance estimates to them. To facilitate the following operations, every sensor node is further required to obtain the lists maintained by each of its neighbors so that it is able to be aware of the nodes and the corresponding distance estimates within two hops. To this end, each sensor node exchanges its neighbor list with all its neighbors.

The basic technique for locating a sensor node is multilateration. By referencing to three or more other nodes, the location of a node can be uniquely determined. It is intuitive that more reference nodes can result in more accurate location estimation. Because of ranging errors, however, for a sensor node a computed location is probably a certain distance away from the real location. And the Euclidean distance between the computed location and the real location reflects the accuracy of the computed location for the sensor node, and hence is defined as the *localization error*.

In our approach, we introduce the concept of Quality of Localization (QoL) to indicate the accuracy of a computed location for a specific sensor node. Any computed location is associated with a QoL. A better QoL means that the computed location is much closer to the real location (less localization error). To quantitatively reflect the QoL of a computed location, we represent a computed location with a circle (referred as location circle). The center of the circle is considered as the location estimation. And the real location of the sensor node is guaranteed to reside within the circle. It is intuitive that a longer radius infers that the computed location is less accurate (i.e., a lower QoL).

Extensive research has been conducted on the distance measurement via RF signal strength. Through the statistical technique [10], ranging errors could be effectively restricted. In this paper, we assume that the ranging error (e) of a distance estimate (d) is below β percent of d, i.e., $|e| \leq \beta\ \% \times d$. It follows that

$$d(1 - \beta\%) \leq d_0 \leq d(1 + \beta\%),$$

where d_0 is the real distance. Later, we use d_{AB} to denote the distance between nodes A and B measured by node A, and d_{BA} to denote the distance between nodes A and B measured by node B. It is not necessary that d_{AB} is equal to d_{BA} because of the dynamics of RF signals.

3.1 Annulus Intersection Based Multilateration

We propose the annulus intersection based multilateration to locate a given node. Given a measured distance between two nodes, if one end node has been located, we can predict the area where the other end node possibly shows up. As shown in Fig. 1, suppose that node P is a neighbor of node A, and the distance d_{PA} has been measured. Provided that the location of node A has been computed and represented as a circle with the radius R_A, we can conclude that P must be within the shadowed annulus. Next, we explain how the annulus intersection based multilateration works to locate a node. For simplicity while without losing generality, we illustrate trilateration in Fig. 2. Nodes A, B and C are located already, and the distances d_{PA}, d_{PB} and d_{PC} are known. We illustrate how node P is located. Nodes A, B and C form the annuluses, respectively. As P must be within each of the annuluses, P must fall into the intersection area of the three annuluses. The smallest circle which fully contains the intersection area is taken as the location circle for P.

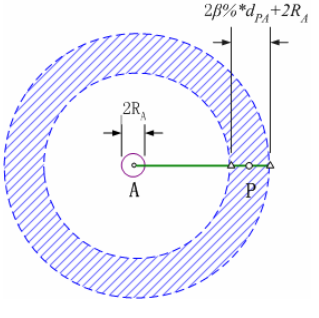


Fig. 1. Annulus area

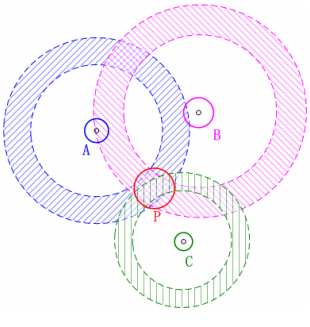


Fig. 2. Trilateration

3.2 Determination of the Coordinate System

Before we start to compute locations for sensor nodes, a unique coordinate system for the sensor network must be settled. The sink node is responsible for determining the unique coordinate system. In Fig. **3** the settlement of the coordinate system is illustrated. Node S is the sink node, and the solid circle is the approximate communication coverage of the sink node. Determining the coordinate system is to fix both the x-axis and the y-axis. To determine the *x*-axis, the sink selects the first node, say node *A*, from its neighbors. Then the sink sets the ray $\overrightarrow{SA}$ as the *x*-axis of the coordinate system. To determine the *y*-axis, the sink selects the second node from the shared neighbors of *S* and *A*, and locates it based on the two measured distances d_{SB} and d_{BA}. Since at least three reference nodes are needed to uniquely locate a node, there are two candidate locations for *B*. By choosing one of the two candidate locations for *B*, we can determine the direction of the *y*-axis. We require that the positive part of *y*-axis and *B* are on the same side of the *x*-axis. Thus, the coordinate system is uniquely defined by the sink, the *x*-axis and *y*-axis.

3.3 QoL-Guided Distributed Refinements

We proposed the QoL-guided spreading localization with refinements for localization in the temporarily static sensor networks. The localization process is spread outward from the sink to the edge of the sensor network like a water wave does. In the following we describe the algorithm in detail.

Till now, two nodes (i.e., Nodes *A* and *B*) have been computed locations. Thus, besides the sink, three nodes have been computed locations. Next, the sink tries to compute the locations of its remaining neighbors incrementally. Each time the sink selects a node with more than three neighbors that have been computed locations, and then computes the location for the node. The sink does not terminate the process until all its neighbors are computed locations. After the process, the sink forms a location update message (*LUM*) which contains the list of all its neighbors and the sink itself with the corresponding locations computed by it. Next, the sink starts the spreading localization process by broadcasting the *LUM* to its neighbors. The localization process is then

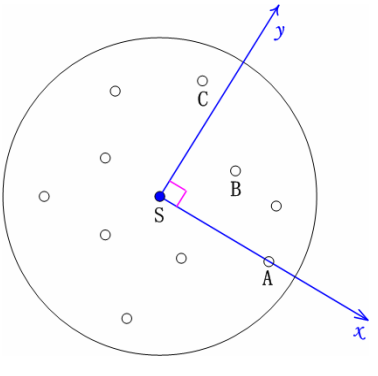


Fig. 3. Settlement of coordinate system

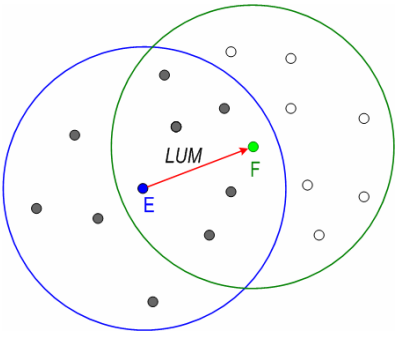


Fig. 4. Spreading localization

spread outward from the sink. On receiving a *LUM*, a node in turn computes the locations of all its neighbors, forms a *LUM*, and then broadcasts it to all its neighbors.

We employ the hop distance technique to control the localization spreading. Each sensor node is required to obtain the shortest hop distance between the sink and itself. The hop distance of the sink is zero and a greater hop distance generally means that the node is much farther from the sink. Many existing algorithms such as Gradient [11] can easily compute the hop distance for every sensor node. We set the restriction that a node only makes use of *LUM*s from those nodes with less hop distances. This helps to prevent localization vibration and guarantee the eventual termination of the localization process. It is also reasonable because of the intuitive observation that the QoL of a node with a smaller hop distance is usually higher than that of a node with a greater hop distance.

For a large-scale sensor network, rapid accumulation of localization errors is really a serious problem. It is apparent that for those sensor nodes further from the sink node, the accumulation of errors is more significant; however, we believe that QoLs of these sensor nodes can be further improved by taking into account all location instances received. Therefore, we propose the QoL-guided refinements to effectively reduce the error accumulation. The basic idea is to determine the location of a sensor node by referencing to as many other nodes as possible. To this end, we propose the refining technique which is to be described shortly.

To present the algorithm, we take the following case as an example. Suppose that node *F* is a neighbor of node *E*, and *F* received a *LUM* from *E* (as shown in Fig. **4**). Each sensor node locally maintains its location, and tries to refine its maintained location using those location instances obtained from the *LUM*s received. If *F* receives a *LUM* from a neighbor with a greater hop distance, it does nothing but drop the *LUM* received. Otherwise, there are two cases for *F*. Case 1: *F* has not determined its location yet, and therefore its locally maintained location is still empty. Then *F* simply sets the location instance obtained from the *LUM* sent by *E* as its location. Case 2: *F* has determined its location. Then *F* tries to refine its location using this location instance.

The proposed refinement technique is introduced here. Suppose that a node, say *P*, locally maintains a location circle *PL*, and it receives a location instance represented by circle *PN*. Now the problem is that given these two circles representing locations of the same node respectively, how to compute a new location circle for the node which has a better QoL (i.e., a shorter radius). In the following, we explain how the location of *P* is refined using our refining technique. Since a location circle guarantees to contain the real location of the sensor node, it follows that the real location of a sensor node must fall into the intersection area of the two circles. Thus, the location of *P* is adjusted to a new circle, which completely contains the intersected area.

3.4 Self-updating Locations of Mobile Nodes

After the spreading localization process over the static sensor network is done, the non-movement restriction is released and those mobile sensor nodes are allowed to move around freely. Now the problem is how to update the locations of those mobile nodes when they are moving from place to place. We propose the mobile location self-updating algorithm. Now the situation is that the whole sensors can be divided

into two categories: mobile sensors and stationary sensors. A stationary sensor remains to stay the place where it was deployed and its location will not be changed. While, a mobile node may be moving from place to place, and its location must be updated from time to time to the right location where it is momently. Because of the movement, a new problem arises that a sensor must update its neighbor list and the distance estimates to these neighbors in real-time.

The proposed mobile location self-updating algorithm is described as follows. It is the mobile sensor itself that is responsible for updating its own location. Before updating the location, a mobile sensor broadcasts a location informing request, expecting that each neighbor sensor responds with sending back its respective current location. Only those stationary sensors will respond by sending back their locations on receiving such a request. When a mobile sensor receives a location informing request, it simply drops it since because of the mobility, its inaccurate location will contribute little to the multilateration of the neighbor. Once a mobile sensor collects the answers from its stationary neighbors, it performs the annulus intersection based multilateration to compute its new location and hence updates its location.

4 Performance Evaluation

In this section we design various simulation experiments to evaluate the performance of our proposed approach. The simulations are conducted on a sensor network which is deployed over a rectangle region. The sensor nodes are randomly distributed across the rectangle. The sink node is deployed at the center of the rectangle. The error of each distance estimate is randomly generated. In Phase one, the error is less than $\beta\%$ of the real distance. In Phase two, for a mobile sensor, the error of any distance estimate is less than $\xi\%$ of the real distance. The statistical technique is less helpful In Phase two because of the node movement, so ξ is much greater than β. The mobile sensors are not allowed to move outside the rectangle region.

We design the first experiment to study the localization coverage achieved by our algorithm in Phase one. The localization coverage is defined as the ratio of the number of nodes which finally got locations to the total number of nodes in the sensor network. The coverage is examined with respect to different node densities. The node degree of a sensor node is the number of its immediate neighbors. The average node degree reflects the node density. As is shown in the Fig. 5, with the increasing average node degree the localization coverage increases rapidly. When the node degree is nine, the coverage is as high as 90%. If the node degree is too low, some sensor nodes may lack enough reference nodes and therefore cannot be computed the location. The proposed distributed spreading localization scheme significantly alleviates the high node density requirements commonly needed by other localization approaches due to the novel spreading technique.

The second experiment is designed to study the localization accuracy achieved by our algorithm in Phase one. The localization error is normalized to the average communication range of sensor nodes. In this experiment, three different ranging error parameters (i.e., β=2, 5, and 10) are studied, respectively. As shown in Fig. 6, the localization errors are decreasing with the increasing average node degree, which is reasonable because a high node density always leads to more refinements. When the

ranging error parameter β is relatively smaller, the resulting localization error is smaller too. It can be concluded from the figure that the final localization error will converge to a certain value which is greater than β. After the average node degree reaches fourteen, the improvement due to the increase of node degrees becomes less and less. The converging localization errors, however, are very small indeed with respect to the given ranging error parameters.

The next experiment is to study the variance of localization errors after mobile sensors are allowed to move. In this experiment, β is set to 5, and ξ is set to 10 which is double of β. In the sensor network, 10% are mobile sensors and the remaining are stationary sensors. We examine the averaged localization errors of mobile sensors at the moment when they stay static in Phase one and the moment when they are moving around in Phase two. The variance is illustrated in Fig. 7. As can be seen that the localization errors incurred when the mobile sensors are moving around are slightly greater than the one incurred when these mobile sensors are static.

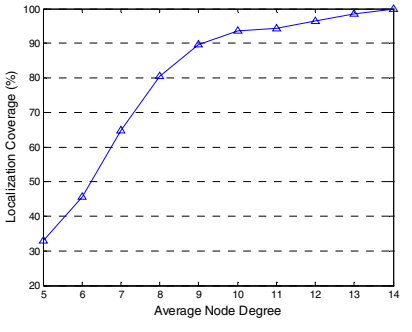


Fig. 5. Localization coverage after Phase one

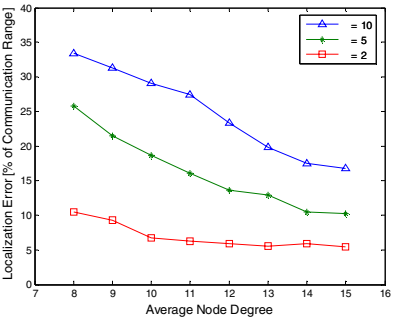


Fig. 6. Localization error after Phase one

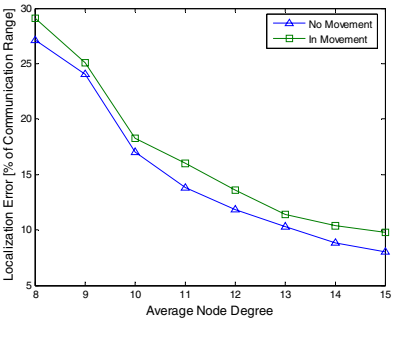


Fig. 7. Variance of localization errors of mobile sensors with and without movement

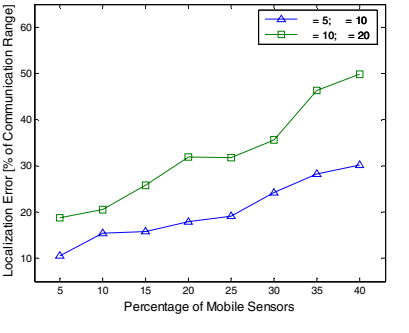


Fig. 8. Impact of percentage of mobile sensors on localization error

We design the final experiment to study the impact of the relative quantity of mobile sensors on the localization error. In this experiment, the average degree is set to 10, and the percentage of mobile sensors is increasing from 5% all the way to 40%.

Two configurations are examined: one is $\beta = 5$ and $\xi = 10$, and the other one is $\beta = 10$ and $\xi = 20$. As is shown in Fig. 8, with the increasing percentage of mobile sensors, the localization error increases rapidly. This is because when the percentage of mobile sensors is higher, the number of stationary neighbors that a mobile sensor can reference to become less, which leads to a lower quality of multilateration.

5 Conclusion

In this paper, we have proposed the QoL-guided localization refinements for anchor-free localization in mobile sensor networks. We made the first step to tackle the sensor mobility problem. Our contributions include: first, we introduced the novel and helpful concept of QoL, and represented a computed location with a circle which quantitatively reflects the QoL of the location by its radius. The location circle representation is very convenient for multilateration in accordance with the framework of QoL. Second, the proposed refinement technique based on the location circle representation effectively improved the accuracy of resulting locations. Third, the proposed mobile location self-updating algorithm provides each mobile sensor with distributed and robust capability to update its location on-demand by itself. Detailed simulation results demonstrate that our approach achieves high localization coverage even in face of relative low node densities, good localization accuracy and small accuracy degradation in face of random movements of mobile sensors. Therefore, the proposed approach is a very promising localization technique for wireless mobile sensor networks.

References

[1] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," presented at the 1st ACM international workshop on Wireless sensor networks and applications, 2002.

[2] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed Energy Conservation for Ad Hoc Routing," presented at MOBICOM '01, Rome, Italy, 2001.

[3] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," IEEE Personal Communications, pp. 28-34, 2000.

[4] D. Niculescu and B. Nath, "DV Based Positioning in Ad hoc Networks," Journal of Telecommunications Systems, 2003.

[5] L. Doherty, K. Pister, and L. E. Ghaoui, "Convex position estimation in wireless sensor networks," presented at IEEE Infocom, Anchorage, AK, 2001.

[6] C. Savarese, K. Langendoen, and J. Rabaey, "Robust positioning algorithms for distributed ad-hoc wireless sensor networks," presented at USENIX Technical Annual Conference, Monterey, CA, 2002.

[7] S. Capkun, M. Hamdi, and J. P. Hubaux, "GPS-Free Positioning in Mobile Ad-Hoc Networks," presented at 34th Hawaii International Conference on System Sciences (HICCSS '01), Maui, Hawaii, 2001.

[8] R. Iyengar and B. Sikdar, "Scalable and distributed GPS free positioning for sensor networks," presented at ICC 2003, 2003.

[9] N. B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, "Anchor-Free Distributed Localization in Sensor Networks," MIT Loboratory for Computer Science, Technical Report April 8 2003.

[10] S. Klemmer, S. Waterson, and K. Whitehouse, "Towards a Location-Based Context-Aware Sensor Infrastructure," CS Division, EECS Department, University of California at Berkeley 2003.

[11] R. D. Poor, "Gradient Routing in Ad Hoc Networks," Media Laboratory , Massachusetts Institute of Technology Cambridge, MA 02139.

Cooperative Target Localization Method for Heterogeneous Sensor Networks

Qing Yang, Lu Su, Quanlong Li, and Xiaofei Xu

Department of Computer Science and Engineering,
Harbin Institute of Technology, Heilongjiang, P.R.C, 150001
{yangqing,suluhit,liquanlong,xiaofei}@hit.edu.cn

Abstract. Based on the binary sensor model, a novel method for target localization in heterogeneous sensor networks is presented. With the binary information reported by nodes, target's position is locked into the intersection area of sensing areas of all nodes which detect the same target, and then the estimated position is computed by geometric means. The proposed method adapts to heterogeneous sensor networks, moreover, it can integrate with other target localization approaches easily. Simulation results demonstrate that, in sensor networks composed of the same type of sensors, our method lead to a decrease in average localization errors compared with the traditional method; in heterogeneous sensor networks, the method renders more accurate estimate of the target's location.

1 Introduction

Advances in the fabrication and integration of sensing and communication technologies have facilitated the deployment of large scale sensor networks. A wireless sensor network consists of tiny sensing devices, deployed in a region of interest. Each device has processing and wireless communication capabilities, which enable it to gather information from the environment and to generate and deliver report messages to the remote base station (remote user). The base station aggregates and analyzes the report messages received and decides whether there is an unusual or concerned event occurrence in the deployed area [1].

Because of its spatial coverage and multiplicity in sensing aspect and modality, a sensor network is ideally suited for a set of applications: biomedicine, hazardous environment exploration, environmental monitoring and military tracking. Target localization is the foundation of many sensor networks' applications, so research about target localization in sensor networks has recently attracted much attention. For example, Time of Arrival (TOA) technology is commonly used as a means of obtaining range information via signal propagation time; Maximum Likelihood testing (ML) [2] and minimum square estimation [3], are applied to compute the target's position at one node which in charge of collecting the data captured by other sensors. Some other methods estimated the target location at one sensor by successively computing on the current measurement and the past history at other sensors [4, 5, 6]. With hardware limitations and the inherent energy constraints of sensor devices, all the signal proc-

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 13 – 22, 2005.

essing technologies present a costly solution for localization in wireless sensor networks. Unlike these approaches, our cooperative target localization method requires only that a sensor be able to determine whether an object is somewhere within its maximum detection ranger. Our proposed method is similar to the algorithm mentioned in [7] which considers the average x and y coordinates of all reporting nodes as the target location. However, our algorithm can render more accurate target location estimation without losing the briefness and efficiency.

This paper makes three major contributions to the target localization problem in sensor networks. First, though many methods [2, 3, 4, 5, 6] have been proposed to solve this problem, none of them has considered the networks composed of heterogeneous sensors. This paper provides a realistic and detailed algorithm to determine the target's location in heterogeneous sensor networks. Second, compared with the prior algorithm such as that mentioned in [7], the proposed method renders more accurate estimate of target's location. Third, the presented approach can guarantee that the target must be in a small intersection area X which our algorithm works out; that means other methods do not need to search the whole area but only X.

The organization of the rest of this paper is as follows. Section 2 gives brief description of the binary sensor model, preliminaries and assumptions. In Section 3, we present details of the target localization model and VSB (Valid Sensing Border) updating method. Section 4 designs and analyses the Cooperative Target Localization (CTL) Algorithm. In Section 5, we present simulation results, comparing CTL with the traditional method mentioned in [7]. Section 6 concludes the paper and outlines the direction for future work.

2 The Binary Sensor Network

Suppose a set of m different kinds of sensors $S = \{s_1, s_2, s_3, \ldots, s_m\}$ are deployed within a bounded 2-dimensional area, these sensors compose a binary sensor network. In this binary sensor network, each sensor's result is converted reliably to one bit of information only. This binary information may have different meanings, for example, it means whether an object is approaching or moving away from sensors in [8]. In this paper, we define it as whether an object is somewhere within the maximum detection range of sensors.

Nevertheless, in heterogeneous sensor networks, detection ranges of sensors are different from one to another. For example, sensors with infrared or ultrasound sensing devices have a circle-like sensing area as illustrated in Fig.1 (a); image sensors have a sector-like sensing area, as illustrated in Fig.1 (b); some other sensors' sensing areas may be irregular as illustrated in Fig.1 (c). It is difficult to use these sensing areas directly, so we uniformly define the sensing border of every sensor as a circle.

Definition 1. Sensing Radius: The sensing radius of one sensor $s_i \in S$ is defined as $Max|s_ip|$, where p is one point of set Q which consists of all the points that can be detected by sensor s_i. We denote s_i's sensing radius as $s_i.R$.

Definition 2. Sensing Border: Consider any sensor $s_i \in S$, the circle centered at this node with radius $s_i.R$ is s_i's sensing border, denoted as $s_i.C$.

Definition 3. Sensing Area: Consider any sensor $s_i \in S$, its sensing border and the inside area are s_i's sensing area, denoted as $s_i.A$.

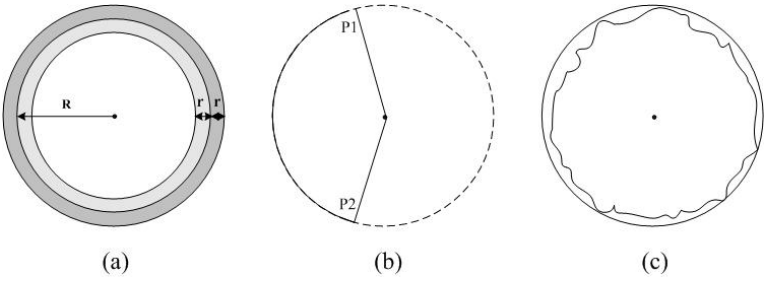


Fig. 1. Sensing area of different kinds of sensors

Consider the sensing area shown in Fig.1 (a), based on the probability-based sensor detection model [9], this kind of sensor's sensing radius is $R+r$. For the image sensor, valid sensing border is not a circle but an arc, such as inner arc p_1p_2 illustrated in Fig.1 (b). We will define the sensor's valid sensing border in later sections.

In heterogeneous sensor networks, sensors have different sensing radiuses, which may be caused by two reasons. First, sensors have different sensing radiuses initially. Second, sensor's sensing radiuses may change during its lifetime. For example, the power level may have an impact on sensor's sensing range. In this paper, we make the assumption that sensing radiuses of every sensor are known and will not change during the whole lifetime. Second, we suppose that each node knows its own location and nodes are not moving. The node's location information does not need to be precise because we are using conservative sensing area to calculate target's location.

Based on the above assumptions, for any sensor $s_i \in S$, one bit of information '1' will be sent if the distance between a target and itself is less than $s_i.R$. If this distance is no less than s_i's sensing radius, the binary information is '0' and nothing will be sent.

3 Cooperative Target Localization Method in Heterogeneous Sensor Networks

3.1 Target Localization Model for Binary Sensor Networks

In binary sensor networks, target localization problem can be formulated as follows. Within a bounded 2-dimensional area, m binary sensors $S = \{s_1, s_2, s_3, \ldots, s_m\}$ are deployed. Assume a target T moves into the area and is located at (x_0, y_0), there will be a set of sensors $D = \{d_1, d_2, d_3, \ldots, d_n\}$ ($D \subseteq S$) detect a target appearing (for example, 6 nodes detect one target as illustrated in Fig.2) by signal processing approaches such as LRT [6]. These nodes in D then send binary information '1' to base station or Cluster Heads which analyzes the report results and estimates the target's location. With the location information of each node and reported binary results, target can be locked into

the intersection area $X = d_1.A \cap d_2.A \cap d_3.A \cap \ldots \cap d_n.A$; then the average x and y coordinates of all vertexes of X will be regarded as the target's location.

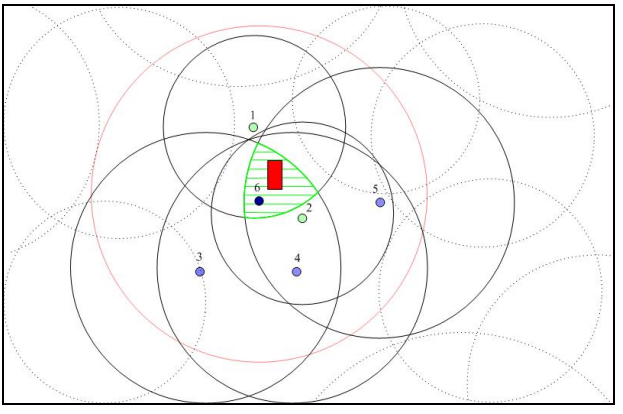


Fig. 2. Detection of one target by multi-sensors

Consider any two nodes d_i and d_j in D, suppose they are located at o_i and o_j respectively. Since they detected the same target, sensing borders of d_i and d_j must intersect. Let $d_i.C$ and $d_j.C$ touch at point p_1 and p_2, then the target must appear in the area enveloped by arc $o_i p_1 p_2$ and arc $o_j p_1 p_2$. Because sensors' locations are known, the arcs generated by intersections of all nodes' sensing borders will be the most import information to compute area X.

Definition 4. Valid Sensing Border (VSB): Consider any sensor $d_i \in D$, its valid sensing border is defined as all of the arcs which satisfy that for any point p on these arcs and any sensor $d_j \in D$ ($i \neq j$), p must be in the sensing area of d_j. If no such arc exists, then the valid sensing border of d_i is null. We denote d_i's valid sensing border as $d_i.arc$.

Theorem 1. Suppose the area enveloped by all sensors' valid sensing borders is R, the intersection area of all sensors' sensing areas is X, then $R = X$.

Proof: (1) We will prove that $R \subseteq X$.

Consider any sensor $d_i \in D$ whose VSB is not null. For any point p on $d_i.arc$, from Definition 4, p must be in the area $X = d_1.A \cap d_2.A \cap d_3.A \cap \ldots \cap d_n.A$. Since sensor d_i and point p are randomly selected, that means for each $i = 1, 2, \ldots, n$, if $d_i.arc$ is not null, it must be in the area X. Moreover, area R is enveloped by VSBs of all sensors, so for any point q in the area R, $q \in X$ must holds. That is to say $R \subseteq X$.

(2) We will show $X \subseteq R$.

Assume by contradiction that the claim is false. This implies that there exists at least one point p, and p is in the area X but not in R. If p is outside R, then p must not be on $d_i.arc$ (for each $i = 1, 2, \ldots, n$). From Definition 4, there must be at least one

sensor d_j which satisfies that p is not in $d_j.A$. That means point p is not in X, this is contradictory to our hypothesis. Thus, the expression $X \subseteq R$ must holds.

Combining (1) and (2) completes the proof.

Based on Theorem 1, we can use VSBs of all nodes in D to compute the target's location instead of using the intersection area of all nodes' sensing areas directly. In this way, as Fig.1 (b) shows, we can initialize the VSB of image sensor as arc p_1p_2. In the next section, we will discuss in detail how to update VSB of each sensor. After getting every node's VSB, we regard the average x and y coordinates of vertexes of all sensors' VSBs as the target's location.

3.2 VSB Updating Method

Suppose every node in D has the same sensing radius, the case with different sensing radius will be discussed later. Consider any sensor $d_i \in D$, assume its VSB is arc p_1p_2. Since the sensing radius of each sensor is identical, the arc p_1p_2 must be an inner arc. In the following, if we do not indicate specially, the word 'arc' means inner arc. All nodes in D detected the same target, so $d_i.C$ must intersect other sensors' sensing borders. Suppose $d_i.C$ touches with $d_j.C$ (another node's sensing border) at point p_3 and p_4, and then four cases will appear as shown in Fig.3 (a, b, c, d).

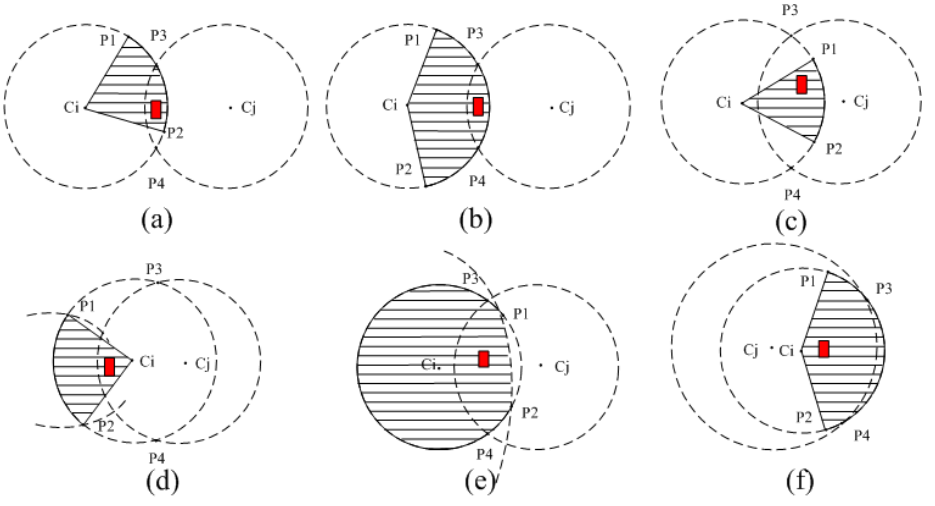


Fig. 3. One sensor's VSB intersects another's sensing border. The arc drawn by real line is d_i's VSB and the rectangle denotes the target

Case 1: As Fig.3 (a) illustrated, if only one point of p_3 and p_4 is on arc p_1p_2 (without loss of generality, suppose p_3 is on arc p_1p_2), then we must have only one point of $p_1$and p_2 is on arc p_3p_4 (suppose p_2 is on arc p_3p_4). Thus, the new VSB of d_i is arc p_2p_3 on $d_i.C$.

Case 2: As Fig.3 (b) illustrated, if both p_3 and p_4 are on arc p_1p_2, at the same time, both p_1 and p_2 are not on arc p_3p_4, then the new VSB of d_i is arc p_3p_4.
Case 3: As Fig.3 (c) illustrated, if both p_3 and p_4 are not on arc p_1p_2, both p_1 and p_2 are on arc p_3p_4, then d_i's VSB is still arc p_1p_2.
Case 4: As Fig.3 (d) illustrated, if both p_3 and p_4 are not on arc p_1p_2, at the same time, both p_1 and p_2 are also not on arc p_3p_4, based on Theorem 2, VSB of d_i is null.

Theorem 2: Consider any two sensors d_i and d_j in the D, suppose $d_i.arc$ is arc p_1p_2 and $d_i.C$ intesects $d_j.C$ at point p_3 and p_4. Thus, if both p_3 and p_4 are not on arc p_1p_2, both p_1 and p_2 are not on arc p_3p_4, then VSB of d_i is null.

Proof: Assume by contradiction that VSB of d_i is not null. Then, from Definition 4, for any point p on $d_i.arc$ and any node $d_k \in D$ ($i \neq k$), p must be in the sensing area of d_k. If we use p_1 and d_j to replace p and d_k, then p_1 must be in $d_j.A$. Since p_1 is on circle $d_i.C$, that means p_1 is in the $d_i.A \cap d_j.A$. Because $d_i.C$ and $d_j.C$ touch at p_3 and p_4, p_1 is on $d_i.C$, then p_1 must be on arc p_3p_4 of $d_i.C$. On the other hand, p_1 is not on arc p_3p_4, which is contradictory; thus, the claim $d_i.arc$ is null holds.

All the cases discussed above will happen if each node in D has the same sensing radius. If sensing radiuses of sensors in D are different, there will be some new cases. Firstly, we will define the distance of two sensors in S. Consider any two sensor s_i and s_j in S, suppose they are located at (x_i, y_i) and (x_j, y_j) respectively; then the distance between them is defined as:

$$dis(s_i, s_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{1}$$

Since sensing radiuses of s_i and s_j are different, s_i's sensing border will completely contains $s_j.C$, which happens whenever $s_j.R + dis(s_i, s_j) < s_i.R$ holds.

Case 5: If $d_j.R + dis(d_i, d_j) < d_i.R$, without further calculation, VSB of d_i is null and $d_j.arc$ will not change. This is because target T can only be in the sensing area of d_j, and then there will be no VSB on $d_i.C$.

Notice that if sensing radiuses are different from one to another, VSB of each sensor will not always be an inner arc. If one sensor's VSB is an outer arc, the updating method mentioned above will still work for Case 1, 3 and 4. But for Case 2, the original VSB will be cut into two pieces.

Case 2a: As Fig.3 (e, f) illustrated, if both p_3 and p_4 are on arc p_1p_2, both p_1 and p_2 are on arc p_3p_4, then the new VSB of d_i are arc p_1p_3 and arc p_2p_4.

In this case, the updating method mentioned above need some modifications since $d_i.arc$ has more than one arc. Suppose $d_i.arc$ is composed of arc_1, arc_2, arc_3, . . . , arc_k (later we will proof that k is less than n). Obviously, all these k arcs are inner arcs. We firstly consider $d_i.arc$ has only one arc such as arc_1, and then update this VSB according to the method mentioned above. If Case 4 or Case 5 occurs, then wipe arc_1 out from the original VSB of d_i; otherwise, new VSB will replace arc_1. In succession, let $d_j.arc$ is arc_2, arc_3, . . . , arc_k, and then update the VSB.

Theorem 3: Every sensor in set D has at most $n - 1$ valid sensing borders.

Proof: Consider any node d_i in D, $d_i.C$ will intersect at most $n - 1$ sensing borders of other sensors in D, this can produce at most $2n - 2$ points on the sensing border of d_i.

Since every VSB of sensor d_i must be the arc between two points produced by sensing borders' intersection, these $2n - 2$ points can build at most $n - 1$ arcs. That means $d_i.arc$ has at most $n - 1$ arcs, so we complete the proof.

4 Cooperative Target Localization (CTL) Algorithm

Based on the above analysis, it is very important to find out VSBs of all node in D. For the image sensor, as shown in Fig1 (b), we initialize its VSB as the arc p_1p_2; for the other types, as Fig.1 (a, c) illustrated, their initial VSBs are null. Consider any sensor $d_i \in D$, our algorithm aims to calculate its new VSB after $d_i.C$ intersects the sensing border of every other node d_j ($i \neq j$) in D. If $d_i.arc$ is null, the arc produced by the intersection of $d_i.C$ and $dj.C$ is d_i's new VSB; if $d_i.arc$ is not null, we use the VSB updating method introduced in section 3.2 to get d_i's the new VSB. In some unusual cases, sensing borders of d_i and d_j may touch at one point. That means the target is located at this point, so we need no more calculations. After getting the new VSB of d_i, we apply the same method to d_j and then update $d_j.arc$. Based on the VSB of each node in D, we use formula 2 to calculate the target's location.

$$\begin{cases} x = \sum_{i=1}^{n'} \sum_{j=1}^{ki} (d'_i.arc_j.p_1.x + d'_i.arc_j.p_2.x) \Big/ 2(k1 + k2 + \cdots + kn') \\ y = \sum_{i=1}^{n'} \sum_{j=1}^{ki} (d'_i.arc_j.p_1.y + d'_i.arc_j.p_2.y) \Big/ 2(k1 + k2 + \cdots + kn') \end{cases} \quad (2)$$

In sensor set D, these sensors whose valid sensing borders are not null compose a new set, we denote it as D'. For any node $d'_i \in D'$, assume its VSB has ki pieces of arcs ($1 \le ki \le n - 1$). Then we can use $d'_i.arc_j.p_1$ and $d'_i.arc_j.p_2$ ($1 \le i \le n'$, $1 \le j \le ki$) to denote two vertexes of the jth valid sensing border of d'_i. Formula 2 aims to calculate the average x and y coordinates of all vertexes of X.

```
Procedure CTL(D)
 1: for i = 1 to n do
 2:  for j = i + 1 to n do
 3:   if d_i and d_j cross then
 4:    p_3, p_4 ←The points of intersection between d_i and d_j;
 5:    for every arc arc_k of d_i.arc do
 6:     if only one of p_3 and p_4 is on arc_k then          /*Case 1*/
 7:      Wipe arc_k out; Add the overlapped part of arc_k and arc p_3p_4 to d_i.arc;
 8:     end if
 9:     if both p_3 and p_4 are on arc_k then
10:      if both arc_k.p_1 and arc_k.p_2 are not on arc p_3p_4 then  /*Case 2*/
11:       arc_k.p_1 ← p_3 and arc_k.p_2 ← p_4;
12:      end if
13:      if both arc_k.p_1 and arc_k.p_2 are on arc p_3p_4 then      /*Case 2a*/
14:       Wipe arc_k out; Add arc p_1p_3 and arc p_2 p_4 to d_i.arc;
15:      end if
```

```
16:       end if
17:       if both p3 and p4 are not on arck and                    /*Case 4*/
18:         both arck.p1 and arck.p2 are also not on arc p3p4 then
19:         Wipe arck out from di.arc;
20:       end if
21:     end for
22:   else                                                         /*Case 5*/
23:     if di.R > dj.R then di.arc ← null;
24:     end if
25:   end if
26:   update dj.arc using the same method;
27:  end for
28: end for
29: for i = 1 to n do            /*average x and y coordinates of all vertexes*/
30:  while di.arc != null do   /* The initial values of x, y and num are 0*/
31:   x ← di.p1.x + di.p2.x + x;
32:   y ← di.p1.y + di.p2.y + y;
33:   num ← num + 1;
34:  end while
35: end for
36: x ← x/(2×num); y ← y/(2×num);
37: return (x, y)
```

Based on Theorem 1, the region enveloped by all VSBs must contain the target; so our CTL algorithm is right. In the following, we will discuss the running time of this algorithm. Line 4, 6-8, 9-16, 17-20, 23-24 can be performed in O (1) time. Because one sensor's VSB has at most $n - 1$ arcs, line 3-25 is executed at most O (n) time. The "for" loop in 2-3 requires O (n^2) time, then line 1-28 takes at most O (n^3) running time. Line 29-35 contributes O (n^2) to the running time. Thus, the total running time of this algorithm is at most O (n^3). Obviously, running time only depends on the number of sensors which detected the same target.

5 Simulation

We implemented a simulator for CTL in order to examine the accuracy of estimates. Let networks cover a 1000m×1000m rectangle area which was divided into 1m×1m grids. Suppose the target is located on each grid, we record the distance between the estimated and real target's position. AvgXY denotes the method mentioned in [7].

5.1 Results

(1) We firstly define the Node Density (ND) as the average number of nodes per R×R area where R is the sensing radius. If all sensors have the same sensing radius, let R = 20m, then Fig. 4 explores the localization estimation accuracy of two methods. From this picture, we can easily find that the average localization errors of CTL are about 1 meter less than those of AvgXY.

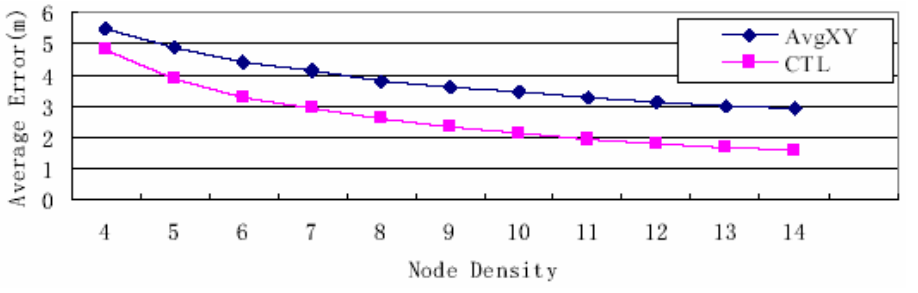


Fig. 4. Average localization errors of two methods with the same sensing radius

(2) In heterogeneous sensor networks composed by 10*m* and 30*m* sensors, we define the ND as the average number of nodes per 20*m*×20*m* area. Fig5 shows that the average localization errors of CTL are 2 meters less than AvgXY.

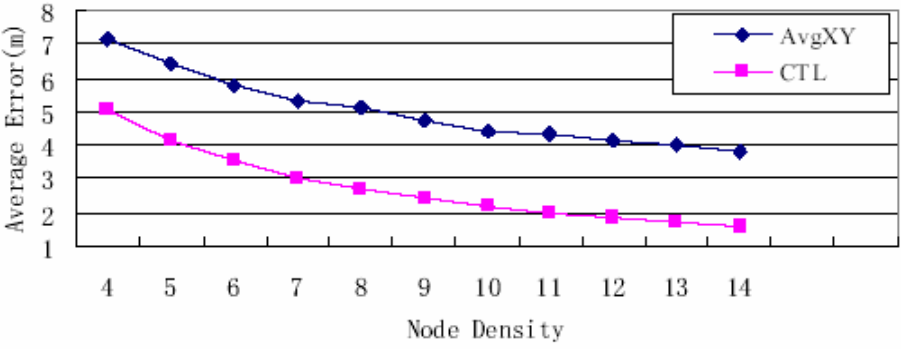


Fig. 5. Average localization errors of two methods in heterogeneous sensor networks

(3) Given ND being a constant (assume ND = 6), Fig. 6 shows that estimation errors of two methods increase as sensing radius become larger. However, CTL always renders a less localization error than AvgXY .

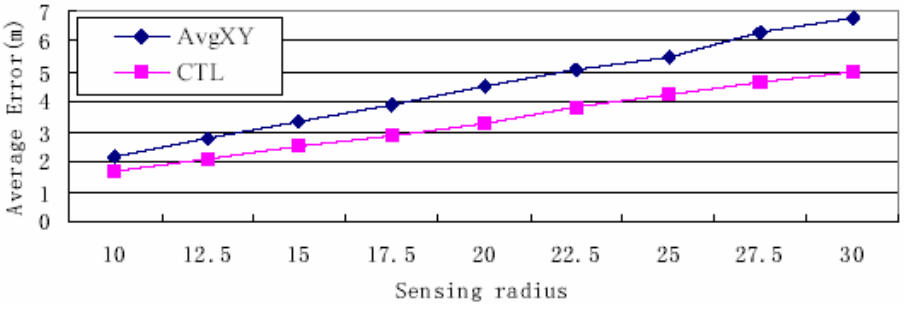


Fig. 6. Average localization errors varying sensing radiuses

6 Conclusion

In this paper, we described a cooperative target localization algorithm for heterogeneous sensor networks. Based on the binary sensor model, we presented the definition of sensing radius, sensing border, sensing area and valid sensing border; then give the target localization model for heterogeneous sensor networks. Simulation results have demonstrated that, not only in sensor networks consist of same types of sensors but also in heterogeneous sensor networks, the proposed method lead to a decrease in average localization errors compared with the traditional method. In addition, the proposed approach can guarantee that the target is in a small region; this implies that other target localization methods need only to consider this region instead of the whole area. Moreover, if distances between the target and sensors are added into our method, the estimation accuracy will be improved. If consider the target classification information, we can implement the multiple targets localization method.

References

1. Tian, Di, Georganas, Nicolas D: "A Coverage-Preserving Node Scheduling Scheme for Large Wireless Sensor Networks" *Proceedings of the ACM International Workshop on Wireless Sensor Networks and Applications*, 2002, p 32-41
2. X. Sheng, and Y-H Hu: "Energy Based Acoustic Source Localization" *Proc. of 2nd Workshop on Information Processing in Sensor Networks (IPSN'03)*, April 2003.
3. D. Li, K.Wong, Y. Hu and A. Sayeed: "Detection, Classification, Tracking of Targets" *IEEE Signal Processing Magazine*, pp. 17-29, March 2002
4. F. Zhao, J. Shin and J. Reich: "Information-Driven Dynamic Sensor Collaboration for Tracking Applications," *IEEE Signal Processing Magazine*, March 2002, 19(2):61-72
5. M. Chu, H. Haussecker and F. Zhao: "Scalable Information-driven Sensor Querying and Routing for Ad Hoc Heterogeneous Sensor Networks," *Int'l J. High Performance Computing Applications*, vol. 16, no. 3, Fall 2002.
6. J. Liu, J. Liu, J. Reich, P. Cheung, and F. Zhao: "Distributed Group Management for Track Initiation and Maintenance in Target Localization Applications," *Proc. 2nd Workshop on Information Processing in Sensor Networks (IPSN'03)*, April 2003:113-128.
7. Tian He, Chengdu Huang, Brian M. Blum, John A. Stankovic, Tarek Abdelzaher: "Range-free localization schemes for large scale sensor networks" *MobiCom '03,*September 2003, San Diego, CA, SA:81-95
8. Javed Aslam, Zack Butler, Florin Constantin, Valentino Crespi, George Cybenko, Daniela Rus: "Tracking a Moving Object with a Binary Sensor Network" SenSys'03, November 5–7, 2003, Los Angeles, California, USA.
9. Yi Zou and Krishnendu Chakrabarty: "Energy-Aware Target Localization in Wireless Sensor Networks" Proceedings of the First IEEE International Conference on Pervasive Computing and Communications 2003:60-67

Sensor Network Configuration Under Physical Attacks

Xun Wang, Wenjun Gu, Kurt Schosek, Sriram Chellappan, and Dong Xuan

The Department of Computer Science and Engineering,
The Ohio State University,
Columbus, Ohio 43210, USA
{wangxu, gu, schosek, chellapp, xuan}@cse.ohio-state.edu

Abstract. Sensor networks typically operate in hostile outdoor environments. In such environments, sensor networks are highly susceptible to physical attacks that can result in physical node destructions. In this paper, we study the impacts of physical attacks on sensor network configuration. Lifetime is an important metric during configuration for many sensor applications. While lifetime is constrained by limited energies and has been addressed before, prior results cannot be directly applied in the presence of physical attacks. In this paper, we define a practical lifetime problem in sensor networks under a representative physical attack model that we define. We develop an anlytical approach to derive the minimum number and deployment plan of sensors to meet lifetime requirement under physical attacks. We make several observations in this paper. One of our important observations is the high sensitivity of lifetime to physical attacks highlighting the significance of our study.

1 Introduction

Sensor networks are typically expected to operate in hostile and inaccessible environments. Instances are battlefields, seismic/volcanic areas, forests etc. Attackers can "physically destroy" sensor nodes due to small sizes of the sensors and the distributed nature of their deployment. We term such attacks as *Physical attacks*. Physical attacks are patent and potent in sensor networks. Attacks can range from a simple and low cost brute force destruction of sensor nodes like bombs, missiles, grenades, moving tanks/vehicles etc. to more intelligent attacks. The end result of physical attacks can be *fatal*. The backbone of the sensor network (the sensor nodes themselves) can be destroyed resulting in severe performance degradation. While much attention has been paid to other types of attacks [1, 2] in sensor networks, to the best of our knowledge threats due to physical attacks is still unaddressed. We believe that viability of sensor networks in the future is closely intertwined with their ability to resist physical attacks.

In this paper, we study the impacts of physical attacks on sensor network configuration. Specifically the problem we study here is: Given a desired lifetime for which the sensor network must be operational, determine the minimum number of nodes and how they must be deployed in order to achieve the desired lifetime when the network is subjected to physical attacks. While there are other variations of physical attacks, in this paper we study physical attacks in the form of bombs targeted at a sensor network

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 23 – 32, 2005.

with the intention of destroying the sensors. The problem is significant and practical. Lifetime is one of the most important metrics during sensor network configuration [3, 4, 5, 6]. This is mainly due to the low energy availabilities in today's sensors that constrain their lifetimes. Sensor networks are typically expected to last for a specific duration to sense desired events and the resources have to be procured and deployed accordingly to meet the lifetime objective [7, 8]. Physical attacks are inevitable in sensor networks, and as such the problem we study is significant. While a body of work has appeared in studying lifetime, their results cannot be directly applied in physically hostile environments primarily due to their not considering the threats of physical attacks.

The output of our solution is the minimum number of nodes needed and the deployment plan, which depend on several factors including the nodes deployment, routing strategies, power availability etc. The presence of physical attack introduces randomness along with the above factors, which make the problem more challenging. We propose an analytical approach to solve this problem. The key idea is to determine and deploy the nodes taking into account both energy minimization and lifetime requirement. We conduct both analysis and simulations to validate our approach. Our data show that results obtained through our analysis matches well with simulation. Our data also show that the lifetime of sensor network is indeed sensitive to physical attacks, which further highlight the significance of our work.

2 System Model and Problem Setup

2.1 Sensor Network Model

We consider a 2-tier hierarchical network model here. The sensor network consists of n^s uniformly deployed sensor nodes. Each sensor node initially has e^s joules of energy. Sensor nodes that sense the data use a set of nodes called *forwarder nodes* as relays to continuously transmit their data to the BS. The forwarder nodes do not generate data. They just relay data using other forwarder nodes progressively closer to the BS. The data transmission from a sensor node to its nearest forwarder node is one hop, while the data from the forwarder node to the BS requires one hop or many hops through other forwarder nodes to the BS. Each forwarder node initially has e^f joules of energy.

The effectiveness of the sensor network is measured by the overall throughput in bits per second received by the BS. Our analysis in this paper is not constrained by the shape of the area of deployment. However, for ease of understanding of the corresponding derivations, we assume the sensors are uniformly deployed over a circular area of radius D, with the area of the network being $\pi \cdot D^2$. The Base Station (BS) is located at the center of the sensor field. All notations, their definitions and standard values are given in Table 1[1].

[1] Empty fields in Column 3 imply that the corresponding parameters are variables in performance evaluation.

Table 1. Notations, Definitions and Standard Values

Notation	Definition	Value	Notation	Definition	Value
α_1	Receiver constant	180nJ/bit	$C(t)$	Throughput at time t	
α_2	Transmitter constant	10pJ/bit/m^2	C^*	Desired throughput	
n	Path loss factor	2	λ	Attack arrival rate	
e^s	Initial power of sensor node [2]	2200J	A	The radius of the area destroyed per attack instance	
e^f	Initial power of forwarder node [3]	18400J	n^s	Number of sensor nodes	
r	The sending rate	2kbps	n^f	Number of forwarder nodes	
d_{char}	Characteristic distance	134.16 Meters	β_d	Density of forwarder nodes at distance d from BS	
T	Desired lifetime		D	Sensor network radius	
$C(0)$	Initial throughput	n^s *r	cf	Confidence	

In the radio model [3], the power expended in relaying (receiving then transmitting) a traffic flow with data rate r to a receiver located at distance d is given by,

$$\overline{p(d)} = r(\alpha_1 + \alpha_2 d^n). \tag{1}$$

Assuming a $1/d^n$ path loss [3], α_1 includes the energy/bit consumed by the transmitter electronics (including energy costs of imperfect duty cycling due to finite startup time) and the energy/bit consumed by the receiver electronics, and α_2 accounts for energy dissipated in the transmit op-amp (including op-amp inefficiencies). Standard values of α_1, α_2, n are given in Table 1. Forwarder nodes have more energy and can increase their transmission range at the cost of more energy dissipation according to (1).

2.2 Attack Model

In this paper we study physical attacks in the form of bombs targeted at a sensor network with the intention of destroying the sensors. Attack events occur in the sensor field of interest. Each event destroys an area in the field. Nodes (sensor nodes and forwarder nodes) located within this area are physically destroyed. Each attack event destroys a circular region of radius A. In this paper we assume attack events follow a Poisson distribution in time. The probability of k attacks in a time interval t, with a mean arrival rate λ is given by,

$$\Pr[N = k] = e^{\lambda \cdot t} \cdot (\lambda \cdot t)^k / k!. \tag{2}$$

[2] Initial power for sensor node is based on 500mA-hr, 1.3V battery.

[3] Initial power for forwarder node is based on 1700mA-hr, 3V battery which is similar to the PicoNodes used in [9].

The attack events are assumed to be uniformly geographically distributed over the sensor field. While the sensor and the forwarder nodes can be destroyed due to attacks, we assume here that the BS will not be destroyed during attacks.

2.3 Problem Setup

The problem we address is: Given a sensor network consisting of n^s uniformly distributed sensor nodes that continuously send data to a BS and given a desired lifetime T for which the network must maintain a minimum throughput C^* with a confidence, cf, determine the minimum number of forwarder nodes n^f and the optimal geographic deployment of these nodes in the sensor field such that the lifetime is guaranteed under physical attacks. More specifically, the inputs to our problem are n^s, D, C^*,T, A, λ. We solve the problem by calculating the optimal number of forwarder nodes at distance d away from the BS under physical attacks. We denote the density of forwarder nodes d away from the BS as β_d. The forwarder nodes in β_d are distributed uniformly in a ring at a distance d from the BS. In this case, d ranges between $(0, D)$, where D is the radius of the sensor field. The integration of β_d is the total number of needed forwarder nodes, n^f.

3 Problem Solution

We now discuss how to determine β_d and deployment plan of the forwarder nodes To solve our problem, we need to derive formulas to compute total traffic throughput to BS and power consumption of each forwarder node as follows.

3.1 Throughput and Power Consumption Rate Computation

In this subsection, we discuss how to compute the sensor network throughput and then describe the derivation of the power consumption rate for each forwarder node. The definitions for notations used here are provided in Table 1.

The sensor network throughput, $C(t)$, changes over time. To compute $C(t)$, we need to know the total number of sensor nodes which send traffic to the BS. The number of sensor nodes whose traffic can reach the BS without considering physical attacks is:

$$S(t) = \alpha \cdot \int_{u=0}^{d_{\min}} 2 \cdot \pi \cdot u \cdot \prod_{i=1}^{H(u,t)} f^{f}_{u-\sum_{k=1}^{i} d_m(k,u,t)}(t) \cdot du. \tag{3}$$

In (3), d_{min} is the radius of the area centered at the BS within which the traffic from the sensor nodes is required to be forwarded to guarantee the throughput requirement; $f_u^f(t)$ is an indicator that shows whether the forwarder nodes u distance away from the BS are out of power (with value 0) or are active (with value 1) at time t; $H(u,t)$ is the number of forwarder nodes needed by a sensor node that are at a distance u away from the BS at time t to send traffic to the BS; $m(t)$ is the number of physical attacks that are expected to arrive in a time period t; $d_m(k,u,t)$ is the average hop routing distance of the k^{th} hop for the sensor nodes that are at a distance u away from the BS at

time *t*. Due to space limitation, we do not discuss the detail derivations of *S(t)* and d_{min}, $f_u^f(t)$, *H(u,t)*, *m(t)* and $d_m(k,u,t)$. Interested readers can refer to [10].

Clearly $(\pi \cdot D^2 - \pi \cdot A^2)/(\pi \cdot D^2)$ is the ratio of remaining sensor or forwarder nodes to the total initial number of sensor or forwarder nodes after one instance of physical attack. Hence, the number of sensor nodes whose traffic can reach the BS at time *t* under physical attacks is:

$$S^*(t) = \alpha \cdot \int_{u=0}^{d_{\min}} 2 \cdot \pi \cdot u \cdot \prod_{i=1}^{H(u,t)} f^f_{u-\sum_{k=1}^{i} d_m(k,u,t)}(t) \cdot du \cdot \left((\pi \cdot D^2 - \pi \cdot A^2)/(\pi \cdot D^2)\right)^{m(t)}. \quad (4)$$

It is now simple to calculate the overall network throughput. The network throughput at time *t* is $S^*(t) \cdot r$, where *r* is the sending rate of the sensor nodes. Thus the throughput in the sensor network subject to physical attacks is given by,

$$C(t) = \int_{u=0}^{d_{\min}} 2 \cdot \pi \cdot u \cdot \prod_{i=1}^{H(u,t)} f^f_{u-\sum_{k=1}^{i} d_m(k,u,t)}(t) \cdot du \cdot \alpha \cdot \left((\pi \cdot D^2 - \pi \cdot A^2)/(\pi \cdot D^2)\right)^{m(t)} \cdot r. \quad (5)$$

The power consumption rate changes over time and each forwarder node has a different power consumption rate. However, the sensor network we are studying is a circle, the BS is at the center of the network, and the sensor nodes are uniformly distributed throughout the network area. Thus forwarder nodes with the same distance to the BS have the same power consumption rate. We denote the power consumption rate for a forwarder node at a distance *d* away from the BS at time *t* as $p_d^f(t)$. To compute $p_d^f(t)$ we need to compute the traffic forwarding rate of each forwarder node *d* away from the BS and the next hop distance. The traffic load of a forwarder node at distance *d* and time *t*, denoted by $w_d^f(t)$, is given by,

$$w_d^f(t) = \frac{\int_{u=u'}^{d_{\min}} 2 \cdot \pi \cdot u \cdot f_u^s(t) \cdot du \cdot \alpha \cdot ((D^2 - A^2)/D^2)^{m(t)} \cdot r}{\int_{u=d-d'/2}^{u=d+d'/2} 2 \cdot \pi \cdot u \cdot \beta_u \cdot ((D^2 - A^2)/D^2)^{m(t)} \cdot du}, \quad (6)$$

where β_u is the density of forwarder nodes at distance *u* away from BS.

For the forwarder nodes whose distance from the BS, *d,* is less than $d_m(1,d,t)$, their next transmission distance is always *d*. However, for other nodes, their next transmission distance will be $d_m(1,d,t)$. Thus $p_d^f(t)$ can be given by the following general formula:

$$p_d^f(t) = \begin{cases} \dfrac{[d_{\min}^2 - (d + d_m(1,d,t)/2)^2] \cdot \alpha \cdot r}{2 \cdot d \cdot d_m(1,d,t) \cdot \beta_d} \cdot (\alpha_1 + \alpha_2 \cdot d_m(1,d,t)^n), \; if \; d \geq d_m(1,d,t) \\ \dfrac{[d_{\min}^2 - (d_m(1,d,t))^2] \cdot \alpha \cdot r}{2 \cdot d^2 \cdot \beta_d} \cdot (\alpha_1 + \alpha_2 d^n), \; if \; d < d_m(1,d,t). \end{cases} \quad (7)$$

The overall power consumption of a forwarder node that is at a distance d away from the BS is given by $\int_{t=0}^{T} p_d^f(t) \cdot dt$. The total number of forwarder nodes in the sensor network can be calculated by,

$$n^f = \int_{u=0}^{D} 2 \cdot \pi \cdot u \cdot \beta_d \cdot du \tag{8}$$

Due to space limitation, we do not give detail derivations of throughput *C(t)*, traffic load of a forwarder node $w_d^f(t)$, and power consumption rate $p_d^f(t)$. Interested readers can refer to [10].

3.2 Our Solution

Having derived the formulas to compute *C(t)* and $p_d^f(t)$, our problem can be expressed as in Figure 1. The intuitive way to solve this problem is to deploy forwarder nodes in such way that the energy spent by the forwarding nodes is minimized with the intention of minimizing the total number forwarding nodes. However, we will see this is not always the case.

Objective: Minimize n^f

Constraints:

$\int_{t=0}^{T} p_d^f(t) \cdot dt \le e^f$ (9), $p_d^f(t)$ is given in (7)

$$C(t) = [\int_{u=0}^{d_{\min}} 2\pi \cdot u \cdot \prod_{i=1}^{H(u,t)} f^{\,f}_{\,u-\sum_{k=1}^{i} d_m(k,u,t)}(t) \cdot du] \cdot \alpha \cdot \left((\pi D^2 - \pi A^2)/(\pi D^2) \right)^{m(t)} \cdot r \ge C^* \tag{10}$$

Fig. 1. Restated problem description

Energy consumption is determined by the routing policy. The routing policy includes the number of intermediate forwarder nodes and the transmission distance. In [3], if each forwarder node's transmission distance is equal to the d_{char} in (11), the energy consumption is minimum. In (11), denoting α_1, α_2, and n as the receive, transmit amplifier, and path loss constants, we have,

$$d_{char} = \sqrt[n]{\alpha_1 / (\alpha_2 (n-1))}\ . \tag{11}$$

To guarantee a routing distance of d_{char}, a certain density of forwarder nodes needs to be deployed so that the average distance between two neighboring forwarder nodes towards the BS, $\underline{d}$, should be *less than or equal to* d_{char}. Our solution gives a *lower bound* of the required forwarder nodes number given desired lifetime. Thus, we need a function to relate $\underline{d}$ with the *lower bound* of forwarder node density. We denote the

function mapping the network forwarder node density β and $\underline{d}$ as $G(.)$. A reasonable $G(.)$ is $\underline{d} = \sqrt{1/\beta}$ or $\beta = 1/\underline{d}^2$. For detailed explanation, refer to [10]. We denote the *lower bound* of the network density which can guarantee d_{char} as β_{char}. In order to guarantee d_{char} under physical attack over a time period t, the initial node density β_{char} should be *greater than or equal to* $1/(d_{char}^2 \cdot (\pi \cdot D^2 / (\pi \cdot D^2 - \pi \cdot A^2))^{m(t)})$.

With the above routing arrangement, enough forwarder nodes will be available for routing through the entire lifetime to guarantee d_{char}. Formula (10) can be simplified as follows,

$$C(t) = \pi \cdot d_{\min}^2 \cdot \alpha \cdot \left((D^2 - A^2) / D^2 \right)^{m(t)} \cdot r \geq C^*. \tag{12}$$

We can determine the density of forwarder nodes based on the requirement of routing over a distance of d_{char}. In order to meet the lifetime requirement under attack, assuming the routing distance d_{char}, we can also derive another minimum network density requirement, denoted as β_d^{power}. β_d^{power} can be computed from (7), (9) and (12) as following.

Given the routing distance is always d_{char}, $d_m(k,u,t)$, the average routing distance of the first next hop, is d_{char}. Once $d_m(k,u,t)$ is determined, d_{min} can be calculated based on (12), and then β_d^{power} can be computed from (7) and (9). Note that in general cases d_{min} is less than D, the radius of the sensor network. However, in special cases, where, for instance, C^* is so big that the number of present sensor nodes cannot provide enough traffic, d_{min} is larger than D. Under this situation, the network is not deployable.

If $\beta_d^{power} >= \beta_{char}$, our assumption that d_{char} can be guaranteed holds. Otherwise, the forwarder node density of β_d^{power} does not guarantee d_{char}. But the problem is: do we have to guarantee d_{char}? The answer is no. Consider a simple case where each forwarder node has enough power to handle all forwarding tasks. In this case only a few or even one forwarder node is enough to meet the lifetime requirement. This in turn means that the density of forwarder nodes is extremely small and routing distance need not necessarily be d_{char} and optimal energy routing is not necessary here.

In the case when $\beta_{char} > \beta_d^{power}$, we do not deploy nodes with the intention of guaranteeing β_{char}. Instead we only need to deploy a minimal number of nodes to meet the lifetime requirement. However, if we decrease the density to be smaller than β_{char}, d_{char} cannot be guaranteed, and optimum energy routing cannot be achieved. Consequently, β_d^{power}, which is calculated assuming a routing distance of d_{char}, may need to be increased due to the actual hop distance being larger than d_{char}. In order to get the optimum, i.e. the optimal nodes density β_d (and the corresponding hop distance) at the distance d away from the BS, we design an iterative procedure to get the minimum density which can satisfy (7), (9) and (12). Thus we obtain the optimum β_d, lying between β_{char} , which gives an upper bound and β_d^{power} , which gives the lower bound of the network density when $\beta_{char} > \beta_d^{power}$.

With our solution, the routing distance cannot be always guaranteed to be d_{char}. In fact,

$$d_m(1,u,t) = \max(d_{ch}(u,t), d_{char}), \tag{13}$$

where $d_{ch}(u,t)$ is the actual average one hop distance for node that is at a distance u away from the BS at time t, which is given by $d_{ch}(u,t) = \sqrt{1/\beta_u(t)}$ (according to $G(.)$). Here $\beta_u(t)$ stands for the forwarder nodes density in the area that is at a distance u away from the BS at time t. The density at initial time is $\beta_u(0) = \beta_u$.

4 Performance Evaluation

In this section, we report our performance data based on the analysis in Section 3. We reiterate that our sensor network is a circular region of radius, D=1000 *meters* and BS is located at the center of the region. Attack events follow a Poisson distribution with a rate λ. Each event destroys a circular region of radius A and attacks are uniformly geographically distributed. Throughout our performance evaluation, the desired throughput C^* is set at 60% of the initial throughput $C(0)$, cf = 95%.

Fig. 2 shows the sensitivity of n^f to λ with different lifetimes when the radius of one attack destruction area (A) is fixed as 20 meters. We make the following observations: First, the required number of forwarder nodes, n^f, is sensitive to the physical attack rate, λ. When λ is big, the attack occurs more frequently. More forwarder nodes are needed in this case to meet the desired network lifetime.

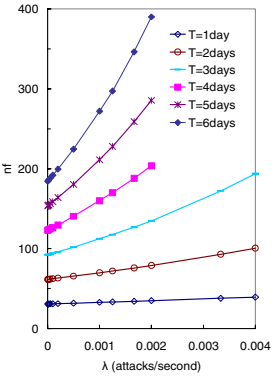


Fig. 2. Sensitivity of n^f to λ

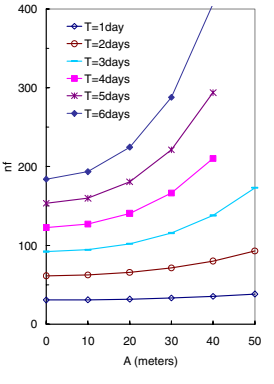


Fig. 3. The sensitivity of n^f to A

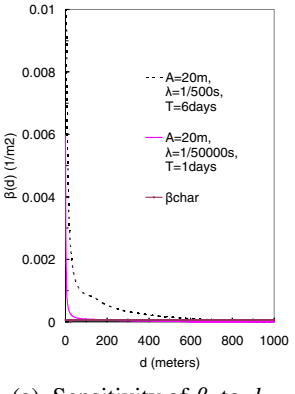
Second, the sensitivity of n^f to λ is more pronounced with larger λ. When λ is very big, the attacks come in very frequently. Here, a little increase in λ can increase the attack intensity significantly. This change greatly increases the required n^f. However, when λ is small, the attacks occur infrequently. In this case, n^f is not too sensitive to λ. This is because when the physical attack comes in very infrequently, fewer nodes are destroyed over a certain period of time. In such cases, n^f is mainly decided by the power consumption of the forwarder nodes. The impact of the physical attacks is not

the deciding factor when the attacks are infrequent. Third, n^f is sensitive to sensor network lifetime, T. When the network lifetime increases, the sensitivity of n^f to attack rate increases. The reason is that the number of nodes destroyed by the physical attacks increases over time. Fourth, when λ is too large, long lifetimes cannot be achieved no matter how we deploy the forwarder nodes. As shown in Fig. 2, when λ is larger than 0.002/s, the lifetime, T, of more than 3 days cannot be guaranteed.

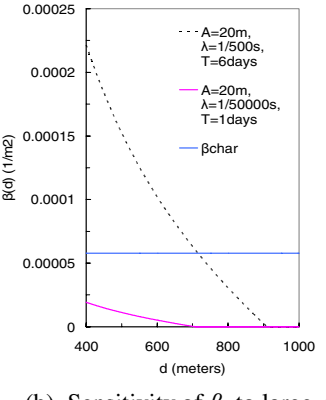
Fig. 3 shows the sensitivity of n^f to A, with different lifetime T, and a fixed λ of 1/2000s. The figure shows that n^f increases with increasing attack size, A. The reason is that, the larger the attack size, the bigger the impact of each physical attack. This, in turn, requires more forwarder nodes be deployed initially to maintain the forwarding task.

Fig. 4(a) shows the density of forwarder nodes and the sensitivity of β_d (deployment) to the distance from the BS under different attack environments and lifetime requirements. The density of required forwarder nodes decreases rapidly with distance, d. This is because there must be a larger number of forwarder nodes near the BS (with small d) to forward the large volume of traffic destined for the BS. Also, the area which these forwarder nodes occupy is very small. When d is large (far away from the BS), the forwarding overhead on each forwarder node is small. Therefore the necessary forwarder node density is small in the areas farther away from the BS.

In Fig. 4(b), we plot β_d with respect to longer distances (d) away from the BS. We enlarge the right hand part of Fig. 4(a) to plot Fig. 4(b). Across most of the network in an infrequent attack and short lifetime environment the optimal forwarder node deployment has a small node density and does not guarantee a hop distance of d_{char} between nodes sending and forwarding packets. The density is low because this optimal deployment only uses the necessary number of forwarder nodes in order to maintain the required throughput for the required lifetime. The lower curve in Fig. 4(b) is an example of this fact. On the other hand, when physical attacks are frequent and the required lifetime is long, many forwarder nodes are deployed. This guarantees d_{char} for most areas in the network and is depicted by the upper curve in Fig. 4(b).



(a). Sensitivity of β_d to d



(b). Sensitivity of β_d to large d

Fig. 4. The optimal forwarder node deployment β_d

We developed a deployment algorithm for findings of this paper to be practically applied. The basic idea is to separate the entire circular area, whose radius is D, into many homocentric rings with small widths. Forwarder nodes based on β_d are randomly, uniformly deployed in each ring. Interested readers can refer to [10] for the details of the algorithm.

5 Final Remarks

Physical attacks are a patent and potent threat in sensor networks. Physical destruction of small size sensors in hostile environments is inevitable. In this paper we studied lifetime of sensor networks under physical attacks. We conducted a detained analysis on how many nodes to deploy and their detailed deployment plan to achieve desired lifetime objectives. Our analysis data matches quite well with simulations, highlighting the fidelity of our analysis. There are several potential directions to extend our study. One of our current focuses is effective counter measuring strategies against physical attacks to enhance the security of the network from physical attacks. We also plan to study impacts due to other forms of physical attacks. Attacks can be intelligent in that they can target nodes to destroy with more sophistication and intelligence raising a host of interesting issues left to be addressed.

References

1. C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeaures," *IEEE International Workshop on Sensor Networks*, May 2003.
2. A. Wood and J. Stankovic, "Denial of service in sensor networks," *IEEE Computer*, pp. 54-62, 2002.
3. M. Bhardwaj, A. Chandrakasan, and T. Garnett, "Upper bounds on the lifetime of sensor networks," *Proc. IEEE ICC '01,* pp. 785-790, 2001.
4. M. Bhardwaj and A. Chandrakasan, "Bounding the lifetime of sensor networks via optimal role assignment," *Proc. IEEE Infocom '02,* pp. 1587-1596, 2002.
5. Z. Hu and B. Li, "On the fundamental capacity and lifetime of energy-constrained wireless sensor networks," *Proc. IEEE RTAS '04*, pp. 38-47, 2004.
6. Z. Hu and B. Li, "Fundamental performance limits of wireless sensor networks," to appear in Ad Hoc and Sensor Networks, Yang Xian and Yi Pan, Editors, Nova Science Publishers, 2004.
7. W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," International Conference on System Sciences, January 2000.
8. M. Kochal, L. Schwiebert, and S. Gupta, "Role-based Hierarchical Self Organization for Wireless Ad hoc Sensor Networks," Proc. ACM WSNA '03, pp. 98-107, 2003.
9. J. Reason and J Rabaey, "A study of energy consumption and reliability in a multi-hop sensor network," ACM SIGMOBILE Mobile Computing and Communications Review, vol. 8, num. 1, pp. 84-97, January 2004.
10. X. Wang, W. Gu, K. Schosek, S. Chellappan and D. Xuan, "Sensor Network Configuration under Physical Attacks", Tech. Report (OSU-CISRC-7/04-TR45), Department of CSE, The Ohio State University, November 2004.

TPSS: A Time-Based Positioning Scheme for Sensor Networks with Short Range Beacons*

Fang Liu[1], Xiuzhen Cheng[1], Dong Hua[1], and Dechang Chen[2]

[1] Department of Computer Science, The George Washington University,
801 22nd St. NW, Washington, DC 20052, USA
{fliu, cheng, gwuhua}@gwu.edu
[2] Uniformed Services University of the Health Sciences,
4301 Jones Bridge Road, Bethesda, MD 20814, USA
dchen@usuhs.mil

Abstract. Location discovery is a challenging problem in sensor networks. However, many sensor network applications require the availability of the physical sensor positions. In this paper, we present TPSS, a time-based positioning scheme for sensor networks when a number of short-range beacons[1] are randomly and uniformly deployed. By measuring the Time Difference of Arrivals (TDoAs) of signals from nearby beacons, nodes can use TPSS to effectively estimate their locations based on the range differences through trilateration. TPSS requires no long-range beacons to cover the entire network, an essential difference compared to TPS [2] and iTPS [15]. Features of TPSS include high scalability, low communication and computation overheads, no requirement for time synchronization, etc. Simulation results indicate that TPSS is an effective and efficient self-positioning scheme for sensor networks with short range beacons.

1 Introduction

A wireless sensor network is composed of a large number of small and inexpensive smart sensors for monitoring, surveillance and control [4,12]. Such a network is expected to be deployed in unattended environments or hostile physical locations.

Almost all sensor network applications require sensors to be aware of their physical locations. For example, the physical positions should be reported together with the corresponding observations in wildlife tracking, weather monitoring, location-based authentication, etc [7,11,17]. Location information can also be used to facilitate network functions such as packet routing [3,10] and

* The research of Dr. Xiuzhen Cheng is supported by NSF CAREER Award No. CNS-0347674; The research of Dr. Dechang Chen is supported by NSF grant CCR-0311252.

[1] In this paper we refer *beacons* to nodes being capable of self-positioning, while *sensors* denote nodes with unknown positions. A beacon node could be a typical sensor equipped with a GPS (Global Positioning System) receiver.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 33–42, 2005.

collaborative signal processing [6], by which the complexity and processing overhead can be substantially reduced. Further, each node can be uniquely identified with its position, thus exempting the difficulty of assigning a unique ID before deployment [19].

However, many challenges exist in designing an effective and efficient self-positioning scheme for sensor networks. First, a localization algorithm must scale well to large sensor networks. Further, the location discovery scheme should not aggravate the communication and computation overheads of the network, since the low-cost sensors have limited resource budget such as battery supply, CPU, memory, etc. What's more, the localization scheme should not raise the construction cost of sensor nodes. Finally, the positioning scheme should be robust enough to provide high precision even under noisy environments. In this paper, we present TPSS, a time-based scheme that meets many of the requirements mentioned above.

TPSS is different from TPS [2] and iTPS [15], even though all three rely on TDoA measurements to calculate a sensor position through trilateration. The beauty of TPSS lies in that there is no requirement for base stations to cover the entire network by powerful long-range beacons. Only a number of short-range beacon nodes with known positions need to be deployed. A beacon node could be a typical sensor with GPS. Recall that TPS (iTPS) requires three (four) long-range beacon stations with each being able to cover the entire network. TPSS releases this restriction while retaining many nice features of the other two. For example, all these three schemes require no time synchronization among sensors and beacons. In TPSS, each sensor listens passively for signals from the beacons in its neighborhood. A sensor computes the range differences to at least three beacons and then combines them through trilateration to obtain its position estimate. This procedure contains only simple algebraic operations over scalar values, thus incurs low computation overhead. Since a beacon signal is transmitted within a short range only, the communication overhead is low, too. Whenever a sensor resolves its own position, it can work as a beacon and help other nodes on location computation. Simulation results indicate that TPSS is an effective self-positioning scheme for sensor networks with short range beacons.

This paper is organized as follows. Section 2 summarizes the current research on location discovery. The new positioning scheme, TPSS, is proposed in Section 3. Simulation results are reported in Section 4. And we conclude our paper in Section 5.

2 Related Work

2.1 Current Location Detection Schemes

The majority of the current location detection systems first measure the distances or angles from sensors to base stations, then obtain location estimation through techniques such as *triangulation*, *trilateration*, *multilateration*, etc. In outdoor sensor networks, GPS is the most popular localization system. However,

it is not practical to install GPS on each sensor due to the cost, form factors, power consumption, antenna requirements, etc. Hence, extensive research has been directed to designing GPS-less localization systems with either long-range or short-range beacons.

Systems with long-range base stations [1,2,13] have a fixed set of powerful beacons, whose transmission range can cover the entire network. Usually these base stations are manually deployed, are time-synchronized, and are equipped with special instruments such as directional antennas. These systems shift the design complexity from sensors to beacon stations. In systems with short-range beacons [8,9,17,18], a small percentage of sensors with known positions are randomly deployed amongst with other ordinary sensors. Some of them relies on transmitting both RF and ultrasound signals at the same time [5,17,18], where the RF is used for time-synchronizing the sender and the receiver. Connectivity-based location discovery schemes [14,16,20] require either long-range beacons or short-range beacons, but these schemes have poor scalability due to the use of global flooding. TPSS exploits local connectivity information among beacon nodes and requires no time synchronization. Therefore, it has better scalability.

2.2 TPS, iTPS, and TPSS

TPS [2] and iTPS [15] rely on the transmission of RF signals from beacon stations for location discovery. Such schemes require no time synchronization in the network and minimal extra hardware in sensor construction. TPS and iTPS are localized algorithms, thus scale well to large networks. Since sensors just listen passively to beacon signals, no extra communication overhead is introduced. As the location detection algorithm involves only some simple algebraic operations, the computation overhead is also low. TPSS retains the above nice features of TPS and iTPS, but requires no powerful long-range beacons to cover the entire network. With only a number of short-range beacons deployed, sensors can compute their positions easily. TPSS can be applied to large-scale sensor networks where the deployment of powerful long-range beacons are too expensive or not practical.

3 TPSS: A Time-Based Positioning Scheme with Short Range Beacons

3.1 Network Model

In this paper, we consider a sensor network deployed over a two-dimensional monitored area. Actually, our TPSS scheme can be easily extended to a higher-dimensional space. In this model, each sensor has limited resources (battery, CPU, etc.), and is equipped with an omni-directional antenna. Some sensors, called *beacons*, have the ability to position themselves. They are deployed together with typical sensors whose positions are to be computed with the TPSS. The beacon nodes will broadcast beacon signals periodically to assist other sensors with location discovery. Note that the only difference between a beacon and

a sensor is whether the location is known. Whenever a sensor gets localized using the TPSS algorithm, it will broadcast its own location and help other sensors for position detection. In other words, it can work as a beacon node.

3.2 A Time-Based Location Detection Scheme with Short Range Beacons

In this section, we propose TPSS, a time-based positioning scheme for sensor networks with short range beacons. TPSS consists of three steps. In the first step, a sensor collects all the signals from the neighboring beacons, and groups them according to the sources of the signals. The next two steps work on the signals belonging to the same group: the range differences from beacon nodes to the sensor are computed and then the coordinates are resolved.

Step 1: Signal Collection
Assume each beacon node initiates a beacon signal once every T seconds. This signal contains the beacon's location and a TTL (Time To Live) field with an initial value ≥ 3. The format of the message is demonstrated in Fig. 1. A beacon node hearing a beacon signal with $TTL > 0$ will broadcast it again after decreasing the TTL value by 1 and after attaching both its own location and the time difference between when the signal is received and when it is re-broadcasted. This is indicated by the *relay* and *delay* fields in the message format shown in Fig. 1. Each sensor with unknown location listens passively for the beacon signals and group them according to the initiators of the messages. If a sensor receives the same signal (originated from the same beacon) at least three times, the location of the sensor can be readily determined by the following two steps.

src	*TTL*	$relay_1$	$delay_1$	$relay_2$	$delay_2$	.....

src: location of the node generating the message
TTL: time to live
$relay_i$: location of the i-th node relaying the message
$delay_i$: time bw. the msg is received and re-broadcasted by the i-th relay

Fig. 1. Format of the Message Transferred

Step 2: Range Detection
We only consider groups containing at least three messages originated from the same beacon node. In each group, select three where the involved beacons are *non-collinear*.

We first assume the beacon signal is relayed without loss, that is, the signal from the initiator as well as from all the intermediate relay nodes can successively reach the sensor S. Fig. 2 shows one such example. Beacon A starts a message $M = (A, 3, -, -)$ which arrives S and beacon B at time t_1 and t_b, respectively. B modifies M to get $M' = (A, 2, B, \Delta t_b)$ and re-broadcasts it at time t_b', where $t_b' = t_b + \Delta t_b$. M' arrives at S and beacon C at time t_2 and t_c, respectively.

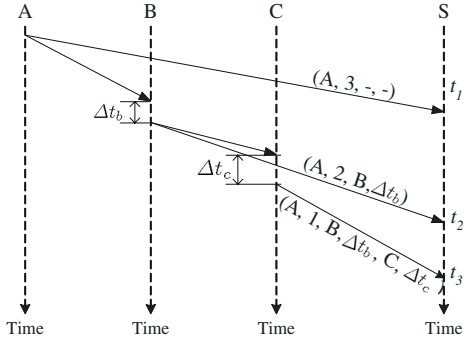


Fig. 2. Range Detection: Signal is Relayed Without Loss

C modifies $M^{'}$ to get $M^{''}$=$(A, 1, B, \Delta t_b, C, \Delta t_c)$ and broadcasts $M^{''}$at time $t_c^{'}$, where $t_c^{'} = t_c + \Delta t_c$. Finally, $M^{''}$arrives at S at time t_3. Assume all the nodes transfer the signals at the same speed v. Let d_{sa}, d_{sb}, d_{sc} represent the distance between sensor S to beacons A, B, C. Let d_{ab}, d_{ac} denote the distances between beacons A and B, A and C, respectively. We have

$$\frac{d_{ab}}{v} + \Delta t_b + \frac{d_{sb}}{v} - \frac{d_{sa}}{v} = t_2 - t_1 \tag{1}$$

$$\frac{d_{bc}}{v} + \Delta t_c + \frac{d_{sc}}{v} - \frac{d_{sb}}{v} = t_3 - t_2 \tag{2}$$

which gives

$$d_{sa} = d_{sb} + k_1, \quad \text{where } k_1 = d_{ab} - v \cdot (t_2 - t_1 - \Delta t_b) \tag{3}$$

$$d_{sc} = d_{sb} + k_2, \quad \text{where } k_2 = -d_{bc} + v \cdot (t_3 - t_2 - \Delta t_c) \tag{4}$$

Eqs. (3)(4) show that k_1, k_2 can be obtained by measuring t_1, t_2, t_3 with S's local timer, learning the positions of A, B, C and time differences $\Delta t_b, \Delta t_c$ from the beacon signals. We are going to apply trilateration with k_1, k_2 to compute coordinates (x, y) for sensor S in Step 3.

Note that TPSS can still work if some beacon signals get lost during the transmission from the initiator or any intermediate relay nodes. As long as a sensor S receives one signal from three different relay beacons, S's location can be computed with TPSS. For example (Fig. 3), M is a beacon signal travelling along beacons 1, 2, 3, 4 and 5. The messages relayed by beacons 1 and 4 are lost or destroyed during the transmission. S receives M only from beacons 2, 3, 5 at time t_0, t_1, t_2, respectively. Let $d_{ij}(d_{sj})$ denote the distance between node $i(s)$ and j, and Δt_i be the time difference information conveyed by beacon node i. We have:

$$\frac{d_{23}}{v} + \Delta t_3 + \frac{d_{s3}}{v} - \frac{d_{s2}}{v} = t_1 - t_0 \tag{5}$$

$$\frac{d_{34}}{v} + \Delta t_4 + \frac{d_{45}}{v} + \Delta t_5 + \frac{d_{s5}}{v} - \frac{d_{s3}}{v} = t_2 - t_1 \tag{6}$$

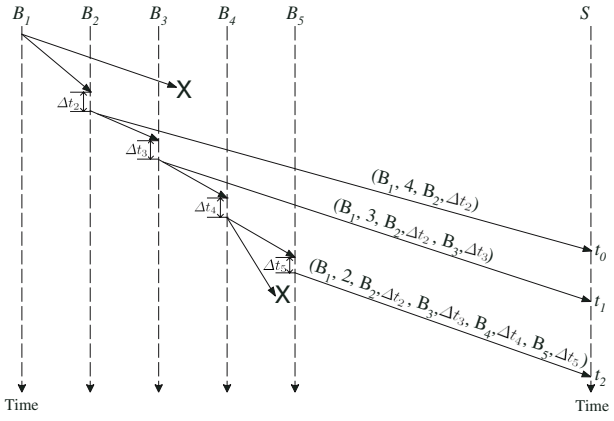


Fig. 3. Range Detection: Signal is Relayed With Loss

It follows that,

$$d_{s2} = d_{s3} + k_1, \quad \text{where } k_1 = d_{23} - v \cdot (t_1 - t_0 - \Delta t_3) \tag{7}$$

$$d_{s5} = d_{s3} + k_2, \quad \text{where } k_2 = -(d_{34} + d_{45}) + v \cdot (t_2 - t_1 - \Delta t_4 - \Delta t_5) \tag{8}$$

Comparing Eqs. (3)(4) with (7)(8), we can summarize the result of range detection as following:

$$d_{sa} = d_{sb} + k_1 \tag{9}$$

$$d_{sc} = d_{sb} + k_2 \tag{10}$$

where A, B, C are the three relay nodes in the same group that convey messages originated from the same source and are sorted according to the order of relaying the signal.

Remarks:
(*i*) All times are estimated locally. For example, the arrival times of the signals (t_1, t_2, etc.) are measured at sensor S's local timer; the time differences at relay nodes ($\Delta t_b, \Delta t_c$, etc.) are computed by beacon's local timer and known system delays.
(*ii*) For each sensor S, range detection is conducted on each group that contains messages from the same initiator. Corresponding location computation is taken in the next step. Averaging all the results computed for S, we obtain the final coordinates of node S.
(*iii*) For each group, there may exist multiple methods to select the three messages. Consider a signal travelling along beacons 1 to 4, and assume all the relayed signals arrive at S finally. We have $d_{s,i} = d_{s,i-1} + k_{i-1}$, where $k_i = v \cdot (t_{i+1} - t_i - \Delta t_{i+1}) - d_{i,i+1}$, $d_{ij}(d_{sj})$ is the distance between node $i(s)$ and j, Δt_i is the time difference at the relay node i, and t_i is the time S receives the

message from beacon i, for $i = 2, 3$, and 4. The three equations can be divided into two overlapping groups. Group I contains $d_{s2} = d_{s1} + k_1$, $d_{s3} = d_{s2} + k_2$; while group II contains $d_{s3} = d_{s2} + k_2$, $d_{s4} = d_{s3} + k_3$. Each group can be used to compute S's coordinates in the next step independently.

Step 3: Location Computation
From Eqs. (9)(10), $d_{sa} = d_{sb} + k_1$, $d_{sc} = d_{sb} + k_2$, we get the following three equations with three unknowns x, y and d_{sb} based on trilateration:

$$(x - x_b)^2 + (y - y_b)^2 = d_{sb}^2 \tag{11}$$
$$(x - x_a)^2 + (y - y_a)^2 = (d_{sb} + k_1)^2 \tag{12}$$
$$(x - x_c)^2 + (y - y_c)^2 = (d_{sb} + k_2)^2 \tag{13}$$

As proposed in [2], we can solve these equations in two steps: First, transform the coordinates into a system where A, B, C reside at (x_1,0), (0,0) and (x_2, y_2), respectively; Second, solve the equations with the efficient method proposed in [2]. Since the positions at the original coordinate system can always be obtained through rotation and translation, the solution provided by [2] can be treated as a general one:

$$x = \frac{-2k_1 d_{sb} - k_1^2 + x_1^2}{2x_1} \tag{14}$$
$$y = \frac{(2k_1 x_2 - 2k_2 x_1) d_{sb}}{2x_1 y_2} + \frac{k_1^2 x_2 - k_2^2 x_1 + x_2^2 x_1 + y_2^2 x_1 - x_1^2 x_2}{2x_1 y_2} \tag{15}$$

where d_{sb} is the root of $\alpha d_{sb}^2 + \beta d_{sb} + \gamma = 0$, with

$$\alpha = 4[k_1^2 y_2^2 + (k_1 x_2 - k_2 x_1)^2 - x_1^2 y_2^2], \tag{16}$$
$$\beta = 4[k_1(k_1^2 - x_1^2) y_2^2 + (k_1 x_2 - k_2 x_1)(k_1^2 x_2 - k_2^2 x_1 + x_2^2 x_1 + y_2^2 x_1 - x_1^2 x_2)], \tag{17}$$
$$\gamma = (k_1^2 - x_1^2)^2 y_2^2 + (k_1^2 x_2 - k_2^2 x_1 + x_2^2 x_1 + y_2^2 x_1 - x_1^2 x_2)^2. \tag{18}$$

Remarks:
Steps 2 and 3 are repeated on all triple messages within each group and all valid groups that can help S estimate its position. The final coordinates (x, y) are obtained by averaging all the results. Once S's position is known, it will become a beacon and help other sensors on location estimation. The iteration of such process can help more and more sensors get localized, as shown by our simulation results in Section 4.

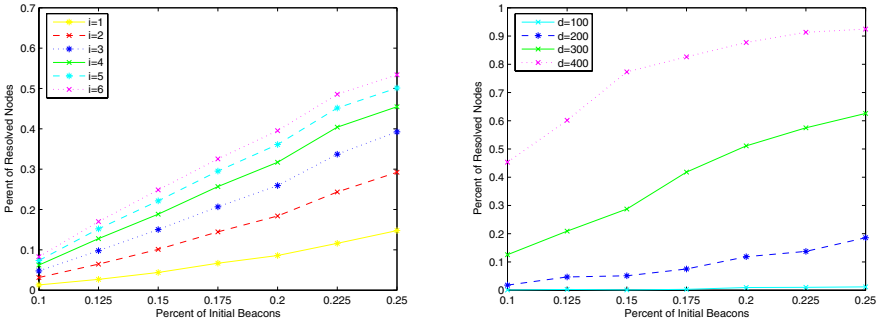
4 Simulation

We consider a sensor network deployed over a field of 100 by 100. The transmission range of sensors and beacons is fixed to 10. We assume each sensor can correctly receive from all the beacons within its transmission range. Each beacon

initiates a beacon signal once per epoch. A sensor becomes a beacon node after its position is resolved. Since MATLAB provides procedures to randomly deploy sensors and beacons, it is selected to perform all the simulations.

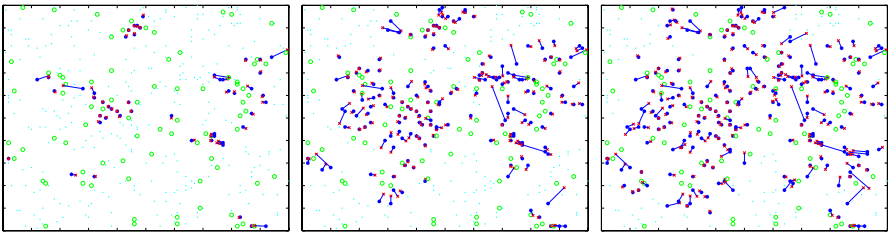
According to Eqs. (3)(4) and (7)(8), the coordinates (x, y) are obtained from the measurements of t_i's, Δt_i's. The accuracy of t_i's depends on the local timers of the sensor nodes, whose measuring errors are affected by the TDoA timer drift, the signal arrival time correlation error, and the reception delays, etc. In the beacon node, Δt_i is computed based on the beacon's local timer and the known system delay, whose inaccuracy is determined by the reception and transmission delays, the time-stamping inaccuracies, and the turn-around delay measurement errors, etc. In our simulation study, we only consider the inaccuracy of the TDoA measurement at the sensors (t_i's), since Δt_i's play the same role. Such inaccuracy is modeled as a normal distribution in the simulation.

We will evaluate the effectiveness of TPSS. First, we want to study the percentage of sensors whose locations can be resolved while varying the percentage of beacons. We consider a network with 300 nodes. Fig. 4(a) reports the results for the first 6 epochs. We can tell that the percentage of resolved nodes increases as the percentage of the initial beacons increases. This also holds true as the number of epochs increases. Second, we test the impact of network density on the localization process. Fig. 4(b) illustrates the percentage of resolved sensors when the percentage of the initial beacon nodes varies under different network density. The number of epochs is set to 10. It shows that as the network density increases, more and more sensors get localized. This is reasonable. As the network density increases, the number of beacons increases if the beacon percentage is fixed. Therefore the probability that a sensor can be reached by three beacons will also increase, since the network is of fixed size. All the results are the average of 100 runs. We obtain two observations from Fig. 4. First, the more beacons deployed, the more sensors get localized. Second, once more and more sensors resolve their positions, more and more sensors get localized. Thus we can expect that with only a small number of short-range beacons, many sensors can be localized using our TPSS scheme.



(a) the first 6 epochs (b) with different network density

Fig. 4. Percentage of Resolved Nodes vs. Percentage of Initial Beacons



(a) t=1, 15.94% resolved (b) t=3, 51.56% resolved (c) t=5, 69.69% resolved

Fig. 5. Illustration of TPSS in terms of variant epochs (t) and resolved percentage. The measuring errors are normally distributed w.r.t. N(0, 0.05). In each figure, "o" represents a beacon, "x" represents the estimated location of a sensor which is linked to the real position (denoted by *), and "·" represents a node whose location is not resolved yet

A snapshot of applying TPSS over a network with 400 nodes and 20% initial beacons is shown in Fig.5. We observe that as the epoch (t) increases, the position error tends to increase. This trend shows the effect of cumulative errors. Recall that once a sensor gets localized, it will use its computed position to help others on position estimation. Considering the unavoidable measuring errors, such a process makes it possible to "pass" computation errors from resolved sensors to others, though it does help in reducing the number of beacons necessary for location discovery. As more sensors get localized, larger computation errors are introduced, that is, the inaccuracy gets cumulated. However, as indicated by Fig 5, such an error cumulation is quite slowly in TPSS. For most of the resolved sensors, the localization error is still tolerable comparing with the transmission range.

5 Conclusion

In this paper, we present TPSS, a time-based localization scheme that uses only short-range beacons. While retaining most of the nice features that TPS and iTPS have, TPSS releases the strict requirement that the beacon stations should be able to reach all the sensor nodes in the network. Simulation results show that TPSS is a simple, effective and practical location discovery scheme.

References

1. Bulusu,N., Heidemann,J., Estrin,D.: GPS-less low cost outdoor localization for very small devices. IEEE Personal Communications 7(5), Oct. 2000, 28-34.
2. Cheng,X., Thaeler, A.,Xue,G., Chen,D.: TPS: A time-based positioning scheme for outdoor wireless sensor networks. IEEE INFOCOM, 2004.
3. De,S., Qiao,C., Wu,H.: Meshed multipath routing: an efficient strategy in wireless sensor networks. Computer Networks, Special Issue on Wireless Sensor Networks, 2003.

4. Fang,Q., Zhao,F., Guibas,L.: Lightweight sensing and communication protocols for target enumeration and aggregation. ACM MOBIHOC, 2003, 165-176.
5. Girod,L., Estrin,D.: Robust range estimation using acoustic and multimodal sensing. International Conference on Intelligent Robots and Systems, Oct. 2001.
6. Heidemann,J., Bulusu,N.: Using geospatial information in sensor networks. ACM MOBICOM, 2000.
7. Intanagonwiwat,C., Govindan,R., Estrin,D.: Directed diffision: a scalable and robust communication paradigm for sensor networks. ACM MOBICOM, 2000, 56-67.
8. Koushanfar,F., Slijepcevic,S., Potkonjak,M., Sangiovanni-Vincentelli,A.: Location discovery in ad-hoc wireless sensor networks. X. Cheng, X. Huang and D.-Z. Du (Eds.), Ad Hoc Wireless Networking, Kluwer Academic Publisher, 2003, 137-173.
9. Langendoen,K., Reijers,N.: Distributed localization in wireless sensor networks: a quantitative comparison. The International Journal of Computer and Telecommunications Networking, 43(4), Special issue on Wireless sensor networks (November 2003) 499-518.
10. Li,J., Jannotti,J., DeCouto,D.S.J., Karger,D.R., Morris,R.: A scalable location service for geographic ad hoc routing. ACM MOBICOM, 2000.
11. Madden,S., Franklin,J.M., Hellerstein,J.M., Hong,W.: TAG: a tiny aggregeation service for ad-hoc sensor networks. OSDI, 2002.
12. Mainwaring,A., Polastre,J., Szewczyk,R., Culler,D.: Wireless sensor networks for habitat monitoring. ACM Workshop on Sensor Netowrks and Applications, 2002.
13. Nasipuri,A., Li,K.: A directionality based location discovery scheme for wireless sensor networks. ACM WSNA'02, 2002, 105-111.
14. Niculescu,D., Nath,B.: Ad hoc positioning system (APS). IEEE GlobeCom, 2001.
15. Thaeler,A., Ding,M., Cheng,X.: iTPS: An Improved Location Discovery Scheme for Sensor Networks with Long Range Beacons. Journal of Parallel and Distributed Computing, Special Issue on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks, 2004.
16. Savarese,C., Rabaey,J., Langendoen,K.: Robust positioning algorithms for distributed ad-hoc wireless sensor networks. USENIX technical annual conference, Monterey, CA, 2002, 317-328.
17. Savvides,A., Han,C.-C., Srivastava,M.B.: Dynamic fine-grained localization in ad-hoc networks of sensors. ACM MOBICOM, 2001, 166-179.
18. Savvides,A., Park,H., Srivastava,M.: The bits and flops of the N-hop multilateration primitive for node localization problems. ACM WSNA'02, Atlanta, GA, 2002, 112-121.
19. Schurgers,C., Kulkarni,G., Srivastava,M.B.: Distributed on-demand address assignment in wireless sensor networks. IEEE Transactions on Parallel and Distributed Systems, 13(10) (2002) 1056-1065.
20. Shang,Y., Ruml,W., Zhang,Y., Fromherz,M.: Localization from mere connectivity. ACM MOBIHOC, 2003.

Energy-Efficient Connected Coverage of Discrete Targets in Wireless Sensor Networks*

Mingming Lu[1], Jie Wu[1], Mihaela Cardei[1], and Minglu Li[2]

[1] Department of Computer Science and Engineering,
Florida Atlantic University, USA
[2] Department of Computer Science and Engineering,
Shanghai Jiao Tong University, China

Abstract. A major concern in wireless sensor networks is to maximize network lifetime (in terms of rounds) while maintaining a high quality of services (QoS) at each round such as target coverage and network connectivity. Due to the power scarcity of sensors, a mechanism that can efficiently utilize energy has a great impact on extending network lifetime. Most existing works concentrate on scheduling sensors between sleep and active modes to maximize network lifetime while maintaining target/area coverage and network connectivity. This paper generalizes the sleep/active mode by adjusting sensing range to maximize total number of rounds and presents a distributed heuristic to address this problem.

1 Introduction

The paramount concern in wireless sensor networks (WSNs) is power scarcity, driven partially by battery size and weight limitations. Mechanisms that optimize sensor energy utilization have a great impact on extending network lifetime. Power saving techniques can generally be classified in two categories: scheduling sensors to alternate between active and sleep mode, or adjusting their sensing ranges. In this paper, we combine both methods by dynamic management of node duty cycles in a high target density environment. In this approach, any sensor schedules its sensing ranges from 0 to its maximum range, where range 0 corresponds to sleep mode.

Target coverage characterizes the monitoring quality of WSNs. The general requirement of target coverage is that each target should be covered by at least one sensor. The energy consumption of target coverage is the total energies consumed by all sensors. The problem of the single sensing range is that there exists a lot of targets covered by several active sensors together, which causes redundancy in energy consumption. Adjustable sensing ranges [16] allow sensors more choices to reduce their energy consumption, and thus prolong WSNs' lifetime.

However, target coverage is not the only responsibility of WSNs. To reduce network overhead and energy consumption, WSNs should also provide satisfactory network connectivity so that sensors can communicate for data gathering or data fusion.

* The work was supported in part by NSF grants ANI 0083836, CCR 0329741, CNS 0422762, CNS 0434533, EIA 0130806, NSFC (No. 60473092), and Program for New Century Excellent Talents in University (No. NCET-04-0392). Email: {mlu2@, jie@cse., mihaela@cse.}fau.edu, mlli@sjtu.edu.cn

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 43–52, 2005.

In this paper, we study the problem of maximizing network lifetime (in terms of rounds) in WSNs, where in each round, sensor-target coverage and sensor connectivity are maintained. Unlike the traditional approaches [12], [14] in area coverage where the connectivity is trivialized by assuming that the transmission range is at least twice of the sensing range, we focus on a more generic connectivity condition that can be used even when the transmission range is less than twice the sensing range.

Although maximizing the lifetime of WSNs by scheduling sensors' activity is not a new problem, none of existing algorithms deal with the case of scheduling sensors' activity by self-configuring sensing ranges, in the environment where both discrete target coverage and network connectivity are satisfied.

The main contributions of this paper are: 1) to introduce the adjustable sensing range connected sensor cover (ASR-CSC) problem, where target coverage and connectivity are maintained, 2) to present a generic connectivity condition, 3) to design efficient distributed heuristics to solve the ASR-CSC problem, 4) to demonstrate the performance of our approach through simulations.

The rest of the paper is organized as follows. In section 2 we present related works on coverage and connectivity problems. Section 3 formulates the ASR-CSC problem and section 4 presents our heuristic contributions. In section 5 we present the simulation results and section 6 concludes our paper.

2 Related Work

The general target coverage problem is introduced in [1], where the problem is modelled as finding maximal number of disjoint set covers, such that every cover completely monitors all targets. The general problem is NP-complete [1]. This problem is extended further in [2], where sensors are not restricted to participation in only disjoint sets, i.e. a sensor can be active in more than one set.

Authors in [15] study area coverage and connectivity in an unreliable wireless sensor grid network, and present a necessary and sufficient condition for coverage and connectivity. In [14], a sufficient condition, the transmission range being larger than twice the sensing range, under which coverage implies connectivity, is given. A similar sufficient condition is considered in [12] in the environment that requires target coverage and connectivity of active sensors in a large scale WSN. Although the connectivity can be relatively easy to specify in the environment with area coverage and uniform sensing range, such a condition will be hard to specify in the environment with with adjustable sensing range and discrete target coverage. In this paper, we present a generic way to address this problem.

The work most relevant to our approach is [3], which extends [2] with adjustable sensing range in point coverage (where target are discrete). Compared with [3], we are also concerned with maintaining network connectivity for the ASR-CSC problem. We analyze the impact of connectivity on energy efficient management sensors, present a generic connectivity condition, and design a distributed heuristic algorithm to maximize the lifetime of WSNs.

3 Problem Formulation

We have two important assumptions in this paper: 1) all sensors in WSNs are connected; 2)any target should be located in the maximal sensing range of at least one sensor. In this paper, we compute the sensor-target coverage and sensor-sensor connection relationship based on Euclidean distance, i.e., a sensor covers a target with sensing range r_k if the Euclidean distance between them is no greater than r_k, and sensor i is connected to sensor j if their Euclidean distance is no greater than transmission range r_c. In this paper, we adopt a fixed transmission range r_c and adjustable sensing ranges $R = \{r_0, r_1, \cdots, r_k, \cdots, r_P\}$, in which r_k is the k-th sensing range. In particular, $r_0 = 0$ is 0-th sensing range, corresponding to sleep mode, r_1, the minimum sensing range in active mode, is the 1-st sensing range, and r_P the maximum sensing range, is the P-th sensing range. For convenience, we index sensor i's selected sensing range by $p(i)$, and $p(i) = k$ means sensor i's current sensing range is the kth range r_k. For consistance, we use R_c to denote the transmission range set, i.e., $R_c = \{r_c\}$. We denote S, T to be the set of sensors and the set of targets respectively, in which $s_i \in S$ means sensor i, and $t_j \in T$ represents target j. Finally, we define $S(i)$ the sensors within s_i's transmission range.

Upon above notations, we model our problem on graph $G_U \cup G_D$, where $G_U = (S, R_c, E_S)$ is the sensor communication graph, and $G_D = (S \cup T, R, E_D)$ is the sensor-target coverage graph. G_U is undirected since sensors' communication ranges are the same, and G_D is directed since different sensors can set different sensing ranges. $E_S = \{(s_i, s_j)| \ |s_i s_j| \leq r_c\}$ is a subset of $S \times S$, which characterizes the direct connection between any two sensors. $E_D = \{(s_i, r_{p(i)}, t_j)| \ |s_i t_j| \leq r_{p(i)}\}$ is a subset of $S \times R \times T$, which represents the sensor-target coverage relationship. Triple $(s_i, r_{p(i)}, t_j)$ means sensor s_i with sensing range $r_{p(i)}$ covering target t_j. Let $S_a = \{s_i | p(i) > 0, \forall s_i \in S\}$ be the active sensors in each round. **Target coverage** can be defined: at any given time during the lifetime of WSNs, $\forall t_j \in T$, $\exists s_i \in S_a$ such that $(s_i, r_{p(i)}, t_j) \in E_D$. WSNs' connectivity depends on the connectivity of its communication graph G_U, thus we can adopt the following definition, **network connectivity**: $\forall s_i, s_j \in S_a$, $\exists s_{i_1}, s_{i_2}, \cdots, s_{i_m} \in S_a$, such that (s_i, s_{i_1}), (s_{i_1}, s_{i_2}), $\cdots$, $(s_{i_m}, s_{i_j}) \in E_S$. Thus, our problem can be formally defined as follows:

Definition 1. (ASR-CSC Problem) *Given a set of targets and a set of sensors with adjustable sensing ranges in a WSN, schedule sensors' sensing ranges, such that the WSN's lifetime is maximized, under the conditions that both target coverage and network connectivity are satisfied, and each sensor's energy consumption should be no more than initial energy E.*

There are two energy models in this paper. The first model is linear model, in which energy consumption is a linear function of the sensing range. The second model is quadratic model, in which energy consumption is a quadratic function of the sensing range. We do not consider the energy consumption caused by transmission. We denote $e_k = f(r_k)$ the energy consumption under sensing range r_k, in which f can be linear or quadratic. A comparison of these two models is illustrated in section 5.

Since AR-SC problem [3] is a special case of the ASR-CSC problem by assuming the communication graph G_U to be a complete graph, according to restriction method [6], the ASR-CSC problem is NP-complete.

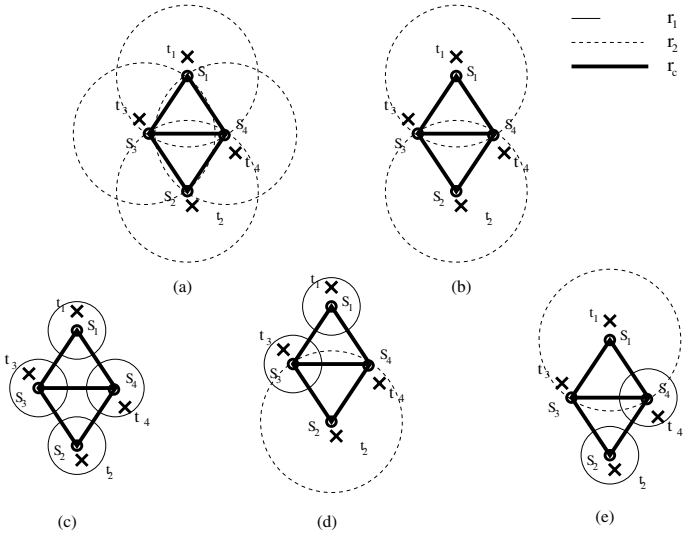


Fig. 1. Example of connected sensor covers

Figure 1 shows an example with four sensors s_1, s_2, s_3, s_4 and four targets t_1, t_2, t_3, t_4. In this example we assume a sensor's sensing area is a disk centered at the sensor, with a radius equal to the sensing range. Each sensor has two sensing ranges r_1, r_2 with $r_1 < r_2$. We use circles with solid lines to denote sensing area with range r_1, circles with dotted lines for area with range r_2, and heavy solid lines for transmission range r_c. The sensor-target coverage relationships are illustrated in Figure 1 (a), (c). Figure 1 (c) shows the targets covered by each sensor with range r_1 : $(s_1, r_1) = \{t_1\}$, $(s_2, r_1) = \{t_2\}$, $(s_3, r_1) = \{t_3\}$, and $(s_4, r_1) = \{t_4\}$. Figure 1 (a) shows the targets covered by each sensor with range r_2: $(s_1, r_2) = \{t_1, t_3\}$, $(s_2, r_2) = \{t_2, t_4\}$, $(s_3, r_2) = \{t_3\}$, and $(s_4, r_2) = \{t_4\}$. The sensors' connection relationships are presented in solid lines: $S(s_1) = \{s_3, s_4\}$, $S(s_2) = \{s_3, s_4\}$, $S(s_3) = \{s_1, s_2, s_4\}$, $S(s_4) = \{s_1, s_2, s_3\}$.

All possible connected sensor covers C_1, C_2, C_3 are illustrated in Figure 1 (c), (d), and (e) respectively, where $C_1 = \{(s_1, r_1), (s_2, r_1), (s_3, r_1), (s_4, r_1)\}$, $C_2 = \{(s_1, r_1), (s_2, r_2), (s_3, r_1)\}$, and $C_3 = \{(s_1, r_2), (s_2, r_1), (s_4, r_1)\}$. Figure 1 (b) shows an sensor cover which doesn't meet the connectivity requirement.

In this example, we assume $E = 2$, $e_1 = 0.5$, and $e_2 = 1$. Each set cover is active for a unit time of 1. The optimal solution has the following sequence of sensor covers: C_1, C_1, C_1, C_1 with maximum lifetime 4. After that, all sensors run out of energy.

If sensors do not have adjustable sensing ranges and the sensing range equal to r_2, then all sensors should be active. The reason is that s_1 and s_2 have to be active to cover t_1 and t_2, and one of s_3 and s_4 has to be active to maintain connectivity. Sensors can be organized in two distinct set covers, i.e., $C_4 = \{s_1, s_2, s_3\}$ and $C_5 = \{s_1, s_2, s_4\}$. But no matter how we schedule the set of sensors, the life time can be no more than 2. Therefore, this example shows a 100% lifetime increase when adopting adjustable sensing ranges.

4 Solution for the ASR-CSC Problem

In this section, a distributed and localized algorithm is given to solve the ASR-CSC problem. In the traditional area coverage, the connectivity is ensured if $r_c \geq 2 \cdot r_k$ for the case of uniform sensing range r_k. However, this result does not apply to point coverage even when $r_k = r_P$. A simple illustration is shown in Figure 2, where heavy solid lines represent transmission range r_c and circles with with light dotted lines denote sensing area with the minimal sensing range r_1. Two sensors i and j with sensing ranges $r_{p(i)}$ and $r_{p(j)}$ respectively take the responsibility of covering discrete targets. However, i and j are so far apart that a range r_c ($\geq 2 \cdot r_1$) cannot connect i and j. Therefore, we have to select some sensors not for target coverage but for connecting i and j. In this case, three other sensors have to be active just for connectivity. The sensing ranges of the three interconnected sensors are r_1 in order to save energy while maintaining connectivity. In fact, r_1 can be considered the minimal energy consumption of an active sensor.

Instead of narrowing our efforts on the relationship between target coverage and network connectivity, we focus on finding a generic way to satisfy both discrete target coverage and network connectivity. We build a virtual backbone first to satisfy network connectivity, and ensure coverage based on that backbone.

We first give a high level view of the whole algorithm. Our algorithm works in rounds, at the beginning of each round the following steps execute: 1) Construct a virtual backbone for the WSN; 2) For each sensor in the virtual backbone, set its sensing range to be the minimal range r_1; 3) All remaining sensors with range r_0 (dominatees) together with sensors with range r_1 (dominators) iteratively adjust their sensing ranges based on contribution (the ratio of the number of covered targets to $e_{p(i)}$, corresponding to $r_{p(i)}$) until a full coverage is found; 4) Each active sensor i reduces $e_{p(i)}$ from its residual energy.

In providing such a virtual backbone in our algorithm, we first construct a connected dominating set and prune redundant sensors by applying Rule-k in [13]. Since it is a distributed and localized method, to ensure network connectivity, we have to assume that the sensors in a given area are dense enough so that all sensors in that area are connected. However, target need not to be dense.

In this method, each sensor determines its status (active/sleep) by applying an eligibility rule. If it meets the rule's requirement, then it decides to sleep; otherwise, it chooses to work for the rest of the round. We formally define the rule : let $S_h(i)$ be

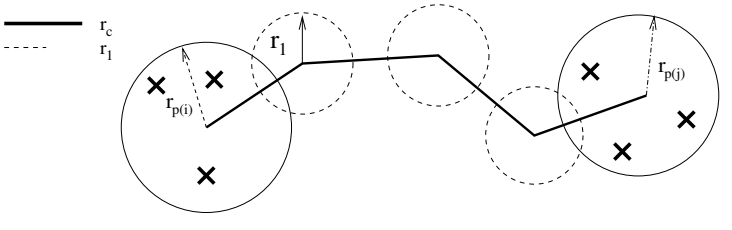


Fig. 2. Sensors contribute only for connectivity

the sensors in $S(i)$ (Note $S(i)$ is i's neighbor sensors) with higher priority than i's. i is able to sleep if and only if the following conditions are satisfied: 1) Sensors in $S_h(i)$ are connected. 2) Sensor i's low priority neighbor $S(i) - S_h(i)$ are covered by sensors in $S_h(i)$.

The result of this connectivity initialization phase is the set of connected active sensors (dominators). The sensing range of those sensors will be set to r_1 in order to save energy. Since dominators alone cannot guarantee target coverage, all dominators together with all still inactive sensors (dominatees) will participate in a coverage initialization phase to ensure target coverage. The reason for active sensors participating in the coverage initialization phase is that dominators can contribute more than dominatees. Since some targets can be located in a distant location such that no dominators can cover those targets, so dominatees should participate the coverage initialization phase.

We present the connectivity initialization phase. This phase is run by each individual sensor before the coverage initialization phase.

Connectivity Initialization

start a timer $t_i \leftarrow \frac{W}{b(i)}$
if receiving message from s_j before t_i expires **then**
  $S_h(i) \leftarrow S_h(i) \cup j$;
  Construct subgraph $(S(i), E_{S(i)})$;
  if $S_h(i)$ is connected and covers $S(i) - S_h(i)$ **then**
    $p(i) \leftarrow 0$;
  end if
end if
$p(i) \leftarrow 1$

In the above algorithm, $b(i)$ denotes the residual energy of sensor i, $S_h(i)$ represents sensor i's neighbor sensors, which have higher residual energy than that of i or have higher ID when residual energies are equal, and W is the longest back-off time. Assigning higher priority to higher residual energy sensors is to balance energy consumption among sensors in the virtual backbone.

In forming the virtual backbone, each sensor i self determines its responsibility by testing Rule-k. If it is satisfied, i decides to sleep; otherwise, it chooses to work. After the connectivity initialization phase, all dominators will be active for the rest of the round. But r_1 is not the final sensing ranges for dominators. The dominators can adjust their sensing range if more contributions can be obtained than other sensors'. After the connectivity initialization phase, a second phase is issued to guarantee target coverage. In the second phase, dominatees combined with dominators will jointly take the responsibility to ensure target coverage, and a sensor's sensing range is increased based on its contribution to target coverage. Once the second phase is done, the sensors whose sensing range greater than r_0 will form the connected sensor cover, while all other sensors will be off-duty in the current round.

To complete our algorithm, we informally describe the coverage initialization phase. For the coverage initialization phase, We use a distributed algorithm similar to the one in [4] to handle target coverage. For brevity, we just describe the main idea of the target coverage algorithm. In each round, each sensor i backs off a time in reverse propor-

tion to its maximal contribution. If before the back-off time is up, it receives messages from its neighbors, it reduces its uncovered target set, recalculates its contribution, and adjusts its back-off time. When the back-off time is up, it broadcasts $p(i)$ (that corresponds to the maximal contribution) and covered target set to its neighbors. At the end of this stage, all the targets will be covered.

5 Simulation Results

In this section, we give an evaluation of our distributed algorithm. Our simulations are based on a stationary network with sensor nodes and targets randomly located in a $100m \times 100m$ area. We assume sensors are homogeneous and initially have the same energy. In the simulation, we consider the following tunable parameters: 1) the number of sensor nodes N. In our experiments we vary it between 50 and 150; 2) the number of targets to be covered M. It varies it between 250 to 500; 3) the number of positive sensing ranges P. We vary it between 1 and 6, and the sensing range values between $10m$ and $60m$; 4) Time slot d, which shows the impact of the transfer delay on the performance of the distributed greedy heuristic. We vary d between 0 and 1 with increase 0.25.

In the first experiment in Figure 3(a), we study the impact of the number of adjustable sensing ranges on network lifetime. We consider 500 targets randomly dis-

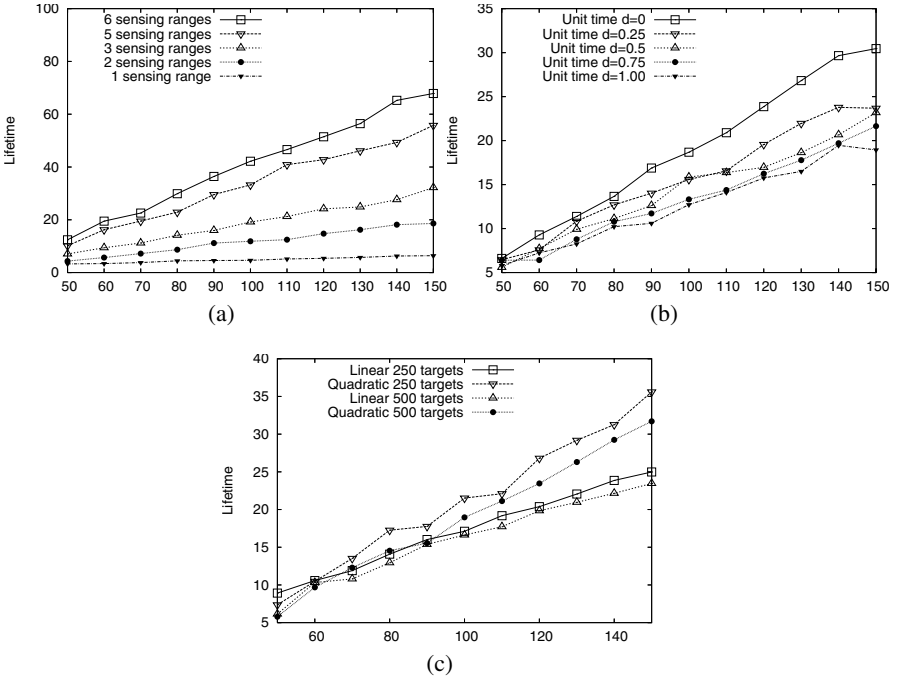


Fig. 3. Simulation results

tributed and we vary the number of sensors between 50 and 150 with an increment of 10. We let the largest sensing range be $30m$ for all cases. We observe the network lifetime when sensors support up to 6 sensing range adjustments: $r_1 = 5m$, $r_2 = 10m$, $r_3 = 15m$, $r_4 = 20m$, $r_5 = 25m$, and $r_6 = 30m$. A case with P positive sensing ranges, where $P = 1..6$, allows each sensor node to adjust $P + 1$ sensing ranges r_0, r_1, $r_2, \cdots, r_P$. Note that $P = 1$ is the case when all sensor nodes have a fixed sensing range with value $20m$. The other environment parameters include initial energy 20. Simulation results indicate that adjustable sensing ranges have great impact on network lifetime.

In Figure 3(b) we observe the network lifetime under different unit time assumptions. We measure the network lifetime when the number of sensors varies between 50 and 150 with an increment of 10 and the number of targets is 500. Each sensor has 3 sensing ranges with values $10m$, $20m$, and $30m$. The energy consumption model is

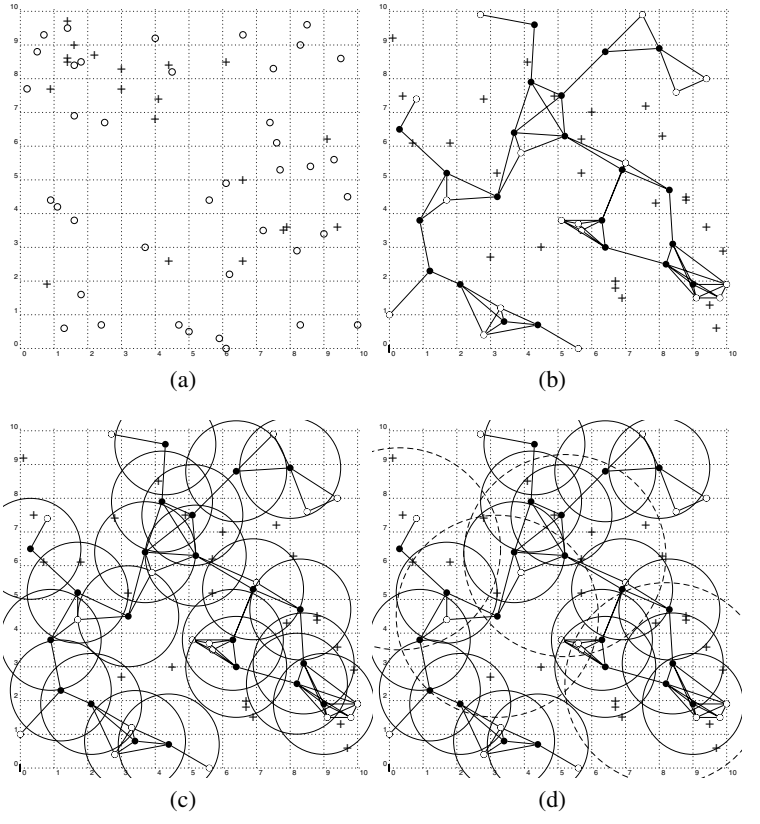


Fig. 4. Set covers example, where $''\circ''$ are sensors and $''+''$ are targets. (a) Sensors and targets deployment. (b) Connected dominating set (black nodes) selected by Connectivity Initialization. (c) Partial coverage when all sensors in the connected dominating set work in sensing range r_1. (d) Full coverage

quadratic. We change the length of the unit time d in the distributed greedy algorithm to $d = 0$, $d = 0.25, 0.5, 0.75$ and 1. Network lifetime produced by the algorithm with lower unit time is longer than those with higher unit time. This happens because, in the distributed heuristic, breaking a tie is at the expense of back-off time, and there is also no guarantee of avoid conflict. A conflict occurs the time between any two sensors' broadcast is less than d. Then, there might be sensors that work instead of going to the sleep state, even if the targets within their sensing ranges are already covered. As illustrated in Figure 3(b), the transfer delay also affects the network lifetime. The longer the transfer delay, the shorter the lifetime.

In Figure 3(c) we study the impact of two energy models on the network lifetime computed by the distributed greedy heuristic when we vary the number of sensors between 50 and 150, and the number of targets is 250 or 500. Each sensor has $P = 3$ sensing ranges with values $10m, 20m$, and $30m$. The two energy models are the linear model $e_p = c_1 * r_p$, and quadratic model $e_p = c_2 * r_p^2$. In this experiment we defined constants $c_1 = E/2(\sum_{r=1}^{P} r_p)$ and $c_2 = E/2(\sum_{r=1}^{P} r_p^2)$, where $E = 20$ is the sensor starting energy. For both energy models, the simulation results are consistent and indicate that network lifetime increases with the number of sensors and decreases as more targets have to be monitored.

In Figure 4, we give an example of active sensor set in a round. We assume a $100m \times 100m$ area, with 40 sensors and 25 targets. We use a linear energy model. The first graph represents the sensors' and targets' random deployment. The transmission range r_c is $25m$. If the distance between any two sensor nodes is no more than r_c, we connect these two sensors by a undirected link. Thus a connected graph is constructed, as shown in 4 (b). Notice that the active sensors are blackened. Each sensor has $P = 3$ sensing ranges with values $15m, 30m$, and $45m$. We use solid lines to represent $r_1 = 15m$, dashed lines for $r_2 = 30m$, and dotted lines for $r_3 = 45m$. Figure 4 (c) show a partial coverage when connected dominating sensors, which are selected in the connectivity initialization phase, keep sensing range r_1. Figure 4(d) shows the schedule satisfying both connectivity and coverage. Note the line type indicates the sensing range value.

6 Conclusions

In this paper, we study the problem to maximize WSN's lifetime (in terms of rounds) while maintaining both discrete target coverage and network connectivity. This not only provides satisfied quality of service in WSNs, but also presents more options and challenges to design an energy efficient sensor scheduling. We study the relationship between network connectivity and target coverage and introduce a generic condition to guarantee network connectivity. We design a round-based distributed algorithm to coordinately determine sensors' sensing range based on different relations between transmission range and maximal sensing range.

In the future, we will study the impact of the degree of coverage on network lifetime and its relationship with network connectivity. We will also take into account the communication cost and its impact on network lifetime.

References

1. M. Cardei, D.-Z. Du, Improving Wireless Sensor Network Lifetime through Power Aware Organization, ACM Wireless Networks, Vol 11, No 3, pg. 333-340, May 2005.
2. M. Cardei, M. Thai, Y. Li, and W. Wu, Energy-Efficient Target Coverage in Wireless Sensor Networks, IEEE INFOCOM 2005, Mar. 2005.
3. M. Cardei, J. Wu, M. Lu, and M. Pervaiz, Maximum Network Lifetime in Wireless Sensor Networks with Adjustable Sensing Ranges, IEEE WiMob2005, Aug. 2005.
4. M. Cardei, J. Wu, Energy-Efficient Coverage Problems in Wireless Ad Hoc Sensor Networks, accepted to appear in Computer Communications, special issue on Sensor Networks.
5. J. Carle and D. Simplot, Energy Efficient Area Monitoring by Sensor Networks, IEEE Computer, Vol 37, No 2, pg. 40-46, 2004.
6. M. R. Garey and D. S. Johnson, Computers and Intractability: A guide to the theory of NP-completeness, W. H. Freeman, 1979.
7. C.-F. Huang and Y.-C. Tseng, The Coverage Problem in a Wireless Sensor Network, ACM MobiCom'03, pg. 115-121, Sep. 2003.
8. D. Tian and N. D. Georganas, A Coverage-Preserving Node Scheduling Scheme for Large Wireless Sensor Networks, Proc. of the 1st ACM Workshop on Wireless Sensor Networks and Applications, pg. 32-41, 2002.
9. X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. D. Gill, Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks, First ACM Conference on Embedded Networked Sensor Systems, pg. 28-39, 2003.
10. J. Wu and S. Yang, Coverage and Connectivity in Sensor Networks with Adjustable Ranges, International Workshop on Mobile and Wireless Networking (MWN), Aug. 2004.
11. Y. Ye, An $o(n^3l)$ Potential Reduction Algorithm for Linear Programming, Mathematical Programming, Vol 50, pg. 239-258, 1991.
12. H. Zhang, J. C. Hou, Maintaining Coverage and Connectivity in Large Sensor Networks, The Wireless Ad Hoc and Sensor Networks: An International Journal, 2005
13. F. Dai and J. Wu, Distributed Dominant Pruning in Ad Hoc Networks, in Proceedings of the IEEE 2003 International Conference on Communications (ICC 2003), Vol. 1, pg. 353-357, May 2003 Anchorage, AK.
14. X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, Integrated coverage and connectivity configuration in wireless sensor networks. In SenSys '03: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, pg. 28-39, Los Angeles, California, USA.
15. S. Shakkottai, R. Srikant, and N. Shroff. Unreliable sensor grids: coverage, connectivity and diameter. In IEEE INFOCOM, pg.1073-1083, 2003.
16. http://www.migatron.com/products/rps-400-6/rps-400-6.htm

Coverage Algorithm and Protocol in Heterogeneous Sensor Networks

Lu Su, Qing Yang, Quanlong Li, and Xiaofei Xu

Department of Computer Science and Engineering,
Harbin Institute of Technology, Heilongjiang, P.R. China 150001
{suluhit, yangqing, liquanlong, xiaofei}@hit.edu.cn

Abstract. One fundamental issue in wireless sensor networks is the coverage problem. In heterogeneous sensor networks composed of different types of sensors, the difference of the sensing radius of nodes augments the computation difficulty of coverage degree. In this paper, we analyze the features of heterogeneous sensor networks and discuss the approaches to guarantee and calculate the coverage degree of the region deployed by heterogeneous sensor networks. Based on our analysis, a *Distributed Coverage Optimization Algorithm* by which each node in the network can determine whether it should be turn active/inactive is proposed. Simulation shows that our algorithm can make the extraneous nodes inactive and minimize the number of nodes need to remain active.

1 Introduction

Recently, the research of wireless sensor networks has attracted more and more attention due to the wide-range of potential applications that will be enabled by such networks. In wireless sensor network, energy efficiency is a key research problem because the battery power of an individual sensor node is severely limited and can not be replaced due to the remote and hazardous monitoring scenarios of sensor networks, such as ocean floor and battlefield. However, the system lifetime of sensor networks, which is measured by the time till all nodes have been drained out of their battery power or the network no longer provides an acceptable event detection ratio [1], is always expected relative long by many sensing applications.

Extending research and simulation have showed that significant energy savings can be achieved by dynamic management of node duty rounds in sensor networks of high node density. It is important for a sensor network to assign the extraneous nodes an off-duty operation mode and minimize the number of nodes on duty while still achieving acceptable quality of service, in particular, satisfying the sensing coverage requirements of applications. Different applications require different degrees of sensing coverage. For example, target surveillance may only require that every location in the sensing region be monitored by one node, while target localization and tracking require at least three coverage degrees [2] [3].Recent three years, the problem of sensing coverage has been investigated extensively, several coverage schemes and protocols have been addressed. In [4], adjustable probing range and wakeup rate of sensor nodes were adopted to control the degree of sensing coverage. Literature [5] proposed a distributed node-scheduling algorithm, in which each node arithmetically calculates

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 53 – 63, 2005.

the union of all the sectors covered by its neighbors and determines its working status according to the calculation result. In [6], a differentiated surveillance scheme was presented, the scheme schedules the sensing phase of each node and makes all the nodes work alternatively to achieve the energy balance of the network. Literature [7, 8] addressed how to combine consideration of coverage and connectivity maintenance in a single activity scheduling.

However, most of coverage schemes can only be applied to the homogeneous sensor networks. Based on the analysis of the heterogeneous sensor networks, we proposed a *Distributed Coverage Optimization Algorithm, DCOA* by which each node in the network can determine whether it should be turn active/inactive.

2 Characters of Heterogeneous Sensor Networks

2.1 Heterogeneous Sensing Model

Heterogeneous sensor network is such a network which consists of sensors with different functions and different sensing ranges. In this paper, we define A as the convex region where sensor nodes are deployed in, $Bond(A)$ as the boundary of region A. Assume each kind of nodes can do 360° observation, for any sensor s in the node set S, We define the boundary of s's coverage region as a circle $C(s)$, the radius of $C(s)$ as the sensing range of s, denoted by $Rs(s)$. We also define $Rc(s)$ as the maximal communication radius of s. In order to guarantee the connectivity of the network, we assume that for any pair of sensors $s_1,s_2 \in S$, $Rc(s_1) \geqslant 2Rs(s_2)$ [7,8].

Intuitively, for any point p in A, p is assumed to be covered by a node s if their Euclidian distance is less than the sensing range of s, i.e. $d(ps) < Rs(s)$. Similarly, we define the convex region A as having a coverage degree of K if every location inside A is covered by at least K nodes. In this paper, we denote the coverage degree of p and A as $Cov(p)$ and $Cov(A)$.

2.2 Unnecessary Sensor Node

Based on the above model, we begin to discuss the characters of heterogeneous sensor network. The difference of the sensing radius of sensor nodes in heterogeneous sensor networks augments the computational difficulty of coverage degree. For example, consider the scenario illustrated by Figure 1, the sensing area of s_2 and s_3 is entirely enclosed by the sensing circle of s_1.

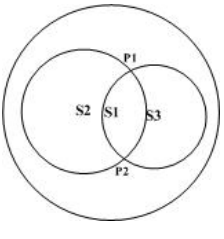


Fig. 1. Unnecessary Sensor Node

The phenomena of "enclose" in heterogeneous sensor networks create a new kind of nodes which we call *Unnecessary Sensor Node, USN*. Before giving the definition of *USN*, we introduce several notions as follows.

Definition 1: *Region Point*. The intersection or tangent point of the boundary of region A and the sensing circle of a sensor node, is called *Region Point*, denoted as *RP*.

Definition 2: *Sensor Point*. The intersection or tangent point of the sensing circles of two adjacent sensor nodes, is called *Sensor Point*, denoted as *SP*.

For a given node, there may exist two special kinds of points covered within its sensing area, defined as follows.

Definition 3: *Circle Point*. For any sensor $s \in S$, the *Region Point* or the *Sensor Point* on the sensing circle of s, is called *Circle Point*, denoted as *CP*.

Definition 4: *Interior Point*. For any sensor $s \in S$, the *Region Point* or the *Sensor Point* inside the sensing circle of s, is called *Interior Point*, denoted as *IP*.

In this paper, for any point p in region A, we define $Cro(p)$ as the number of the sensing circles that cross p. Generally, $Cro(p) = 2$ if p is a *Sensing Point* while $Cro(p) = 1$ if p is a *Region Point*. In some unusual cases, there may exist more than 2 circles crossing a point. Shown as Figure 1, point p_1 and p_2 are *Sensor Points*, they are the *Circle Points* of the nodes s_2 and s_3; Simultaneity, they are also the *Interior Points* of node s_1. On the other hand, *Region Point* also can be the *Circle Point* or *Interior Point* of a certain node in the sensing region.

Now, let's define the notion of *USN*.

Definition 5: *Unnecessary Sensor Node*. For any sensor node $s \in S$, if s has no *Circle Point* on its sensing circle, we call s an *Unnecessary Sensor Node*, denoted as *USN*.

USN has no *Circle Point*, but may have *Interior Point*. The *Interior Points* of *USN* are the intersection or tangent points of the sensing circles enclosed by the circle of *USN*, we call them *Unnecessary Points*, *UP*. On the other hand, the intersection or tangent points in region A which are outside the circles of *USN* are called *Necessary Point*, *NP*.

Theorem 1: Whether the *Unnecessary Sensor Node* is active does not affect the coverage degree of the deployed region.

Proof: Intuitively, the coverage degree of the deployed region is the degree of the location in this region which is monitored by the smallest number of sensor nodes. Therefore, to prove the theorem, we should prove that the location with the lowest coverage degree is outside the sensing circle of *USN*.

We prove by contradiction. Illustrated by Figure 2(a), suppose s_i is an *USN*, p is the point that has the lowest coverage degree K in region A, p_1 is a randomly selected point on $C(s_i)$. Join pp_1 and extend it until it intersects the sensing circle of a node (denoted as s_j) at p_2. Assume p_3 is a randomly selected point on p_1p_2, there are two possible cases.

Case 1: All the points including p_3 on p_1p_2 are outside $C(s_j)$. Clearly, $Cov(p_3)$ is smaller than $Cov(p)$ because the sensing area of s_i covers p, but does not cover p_3.

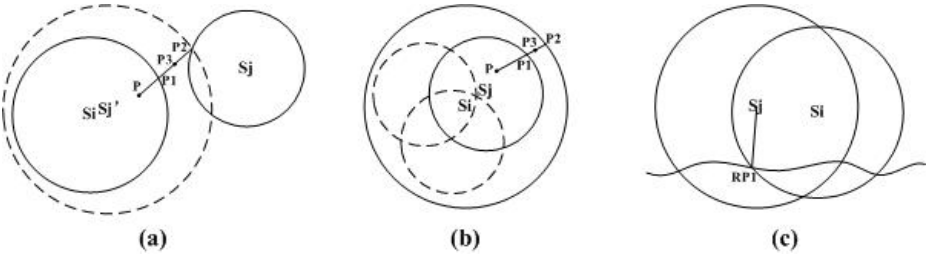


Fig. 2. Samples of Theorems about Unnecessary Sensor Node

Case 2: All the points including p_3 on p_1p_2 are inside $C(s_j)$. Since s_i is an *USN*, $C(s_i)$ does not have any intersection point with the rest nodes in region A, thus $C(s_j)$ must enclose $C(s_i)$, and the points p and p_3 are both inside the sensing area of s_j. Therefore, $Cov(p)>Cov(p_3)$ still holds.

From the proof of case 1 and case 2, we can come into the conclusion that p is not the point with the lowest coverage degree. This contradicts with the assumption, so the theorem is proved.

Theorem 2: Whether the node whose sensing circle is enclosed by the sensing circle of an *USN* is active doesn't affect the coverage degree of the deployed region.

Proof: Illustrated by Figure 2(b), similar to the proof of theorem 1, it is easy to prove that the point having the lowest coverage degree doesn't lie in the sensing area of node s_j whose sensing circle is enclosed by the sensing circle of s_i, an *Unnecessary Sensor Node*.

Theorem 3: All the *Region Points* are *Necessary Points*.

Proof: Illustrated by Figure 2(c), suppose rp_1 is a *Region Point* of node s_i and region A. For any node whose sensing area covers rp_1, such as s_j, since the length of segment s_jrp_1 is not longer than $Rs(s_j)$, i.e. $d(s_jrp_1)\leqslant Rs(s_j)$, s_j must intersects region A. Therefore, s_j is not an *USN*, and rp_1 is not covered by any *USN*. rp_1 is a *Necessary Point*.

Based on Theorem 3, only the *Sensor Points* in the deployed region are likely to be *USN*, thus we sort them into *Necessary Sensor Points* and *Unnecessary Sensor Points.*

3 Coverage Guarantee of Heterogeneous Sensor Networks

For a heterogeneous sensor network, it is important to find out whether the deployed region achieves the expected coverage degree. Obviously it is impossible to calculate the coverage degree of every location in the deployed region. Some papers [1, 5, 6, 7] have dealt with this problem, however, their strategies can not be applied to heterogeneous sensor networks. Based on the analysis of heterogeneous sensor network in previous paragraph, we propose a *Coverage Guarantee Theorem*.

Theorem 4: Suppose convex region A is deployed by a heterogeneous sensor network, A is K-covered if and only if all of the following conditions are satisfied:

Condition 1: There exist *Region Points* in region A
Condition 2: There exist *Necessary Sensor Points* in region A
Condition 3: For any *Region Point rp* in A, $Cov(rp) - Cro(rp) \geq K$
Condition 4: For any *Necessary Sensor Point sp* in A, $Cov(sp) - Cro(sp) \geq K$

Proof: Firstly, we try to prove the if part, i.e. prove that if region A is K-covered, the four conditions must be satisfied.

Condition 1: As illustrated in Figure 3(a), let p be any point on the boundary of region A, since A has a coverage degree of K, p is monitored by at least K sensors. Let node s be any of the sensors covering p, since $d(sp) \leqslant Rs(s)$, s must have intersection or tangent points with the boundary of A, thus there exist *Region Points* in region A.

Condition 2: We prove by contradiction, suppose there is no *Necessary Sensor Point* in region A. As illustrated in Figure 3(b-1) and (b-2), let s_i be the node that has the largest sensing radius; let s_j be the node whose circle is closest to the circle of s_i, i.e., for any $s \in S$, $d(s_is_j)-Rs(s_i)-Rs(s_j) \leqslant d(s_is)-Rs(s_i)-Rs(s)$. Join s_i and s_j, suppose p_1 is the intersection point of the segment s_is_j and $C(s_i)$, and p_2 is the intersection of segment s_is_j and $C(s_j)$. Let p be a randomly selected point on p_1p_2, since A has a coverage degree of K, p is monitored by at least K sensors. Let node s_k be any of the sensors covering p, Join s_i and s_k, suppose the line which joins s_i and s_k intersects two sensing circles at points p_3 and p_4 respectly. Draw a line tangent to $C(s_k)$ at point p_4 which intersects the segment s_is_j at point p_5. There are two possible cases: the sensing area of s_j is outside $C(s_k)$(illustrated by Figure 3(b-1)) or inside $C(s_k)$(illustrated by Figure 3(b-2)). In both cases, it is obvious that the length of right-angle side s_ip_4 is shorter than the length of slope side s_ip_5 in right triangle $s_ip_4p_5$, thus $d(p_3p_4)<d(p_1p_5)<d(p_1p_2)$. This implies that $C(s_k)$ is closer to $C(s_i)$ than $C(s_j)$, which contradicts with the assumption. Therefore s_k is inexistent. From the above analysis, we can draw the conclusion that the sensing circles of nodes s_i and s_j must have intersection or tangent points. Since $C(s_i)$ is the largest circle in region A, there exist *Sensor Points* in region A.

Condition 3: As illustrated in Figure 3(c), suppose rp_1 is any *Region Point* which is created by $C(s_i)$ and $Bond(A)$. Outside the sensing area of s_i, let rp_2 be the *Region Point* that has the shortest path to rp_1 along $Bond(A)$. There are two possible cases: rp_1 is outside $C(s_j)$ or inside $C(s_j)$. Suppose p is a randomly selected point on path rp_1rp_2, in both case1 and case2, it is obvious that all points on path rp_1rp_2 have the same coverage degree, thus $Cov(rp_1) - Cro(rp_1) = Cov(p)$. Since A has a coverage degree of K, p is monitored by at least K sensors. Therefore, $Cov(rp_1) - Cro(rp_1) = Cov(p) \geq K$.

Condition 4: Suppose sp_1 is any *Necessary Sensor Point* which is created by $C(s_i)$ and $C(s_j)$. Join sp_2sp_1 and extend it until it intersects a sensing circle (As illustrated in Figure 3(d-1)) or $Bond(A)$ (As illustrated in Figure 3(d-2)) at point p_1. Similar to the proof of condition 3, all points on segment p_1sp_1 have the same coverage degree, thus it is easy to prove that for any point p on the segment p_1sp_1, $Cov(sp_1) - Cro(sp_1) = Cov(p) \geq K$.

Therefore, the if part is proved.

Then, we try to prove the only if part, i.e. prove that region A must be K-covered, if the four conditions are satisfied.

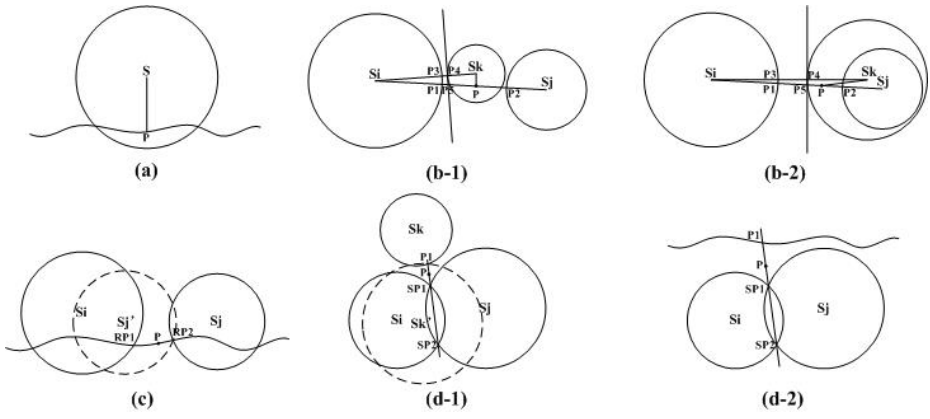


Fig. 3. Proofs of the if part

To prove that A is K-covered, we should prove that all the points in A are K-covered. We split our consideration into two cases. In case 1, we discuss the points inside *Bond(A)*, in case 2, we discuss the points on *Bond(A)*.

Case 1: As illustrated in Figure 4(a), suppose p is any point inside *Bond(A)*. Based on *Condition 2*, there exists *Necessary Sensor Point* in region A. Let sp_1 be the closest *Necessary Sensor Point* to p, i.e., for any sp in region A, $d(psp_1) \leqslant d(psp)$. Join psp_1, suppose p_1 is the closest intersection point to p on the segment psp_1, which is created by the intersection of psp_1 and the sensing circle of a node (denoted as s_k). If s_k is an *USN*, turn it off (by *Theorem 1*) and continue to search the closest intersection to p on psp_1, until the closest intersection of psp_1 and the sensing circle of a *non-USN* is found. If this intersection is sp_1, since all points on segment psp_1 have the same coverage degree, $Cov(p) = Cov(sp_1) - Cro(sp_1)$. Based on *Condition 4*, $Cov(sp_1) - Cro(sp_1) \geqslant K$, thus $Cov(p) \geqslant K$. If this intersection is not sp_1, denote it as p_m, suppose it is created by the intersection of segment psp_1 and the circle of *non-USN* s_m. Assume sp_2 is the closet *Sensor Point* to p_m along $C(s_m)$. Since all the points on segment pp_m have the same coverage degree and at the same time, all the points on arc $p_m sp_2$ have the same coverage degree, thus based on *Condition 4*, $Cov(p) = Cov(p_m) - Cro(p_m) = Cov(sp_2) - Cro(sp_2) \geqslant K$.

Case 2: As illustrated in Figure 4(b), suppose p is any point on *Bond(A)*, Based on *Condition 1*, let rp_1 be the *Region Point* that has the shortest path to p along *Bond(A)*. If p is located outside $C(s_i)$, then $Cov(p) = Cov(rp_1) - Cro(rp_1)$; If p is located inside $C(s_i)$, $Cov(p) > \mathrm{Cov}(rp_1) - Cro(rp_1)$. Based on *Condition 3*, $Cov(rp_1) - Cro(rp_1) \geqslant K$, thus $Cov(p) \geqslant K$.

Therefore, the only if part is proved. Now we can draw the conclusion that the theorem is true.

Based on the *Coverage Guarantee Theorem* of heterogeneous sensor network, a conclusion can be drawn that for any *Region Point* or *Sensor Point* p in the deployed region, if the value of $Cov(p) - Cro(p)$ is not lower than the given coverage degree,

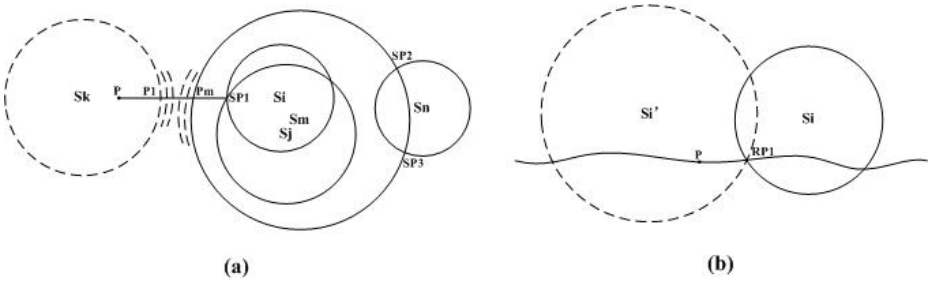


Fig. 4. Proofs of the only if part

the coverage of this region is guaranteed. Therefore, for a convex region *A*, the coverage degree of *A* can be calculated by the following formula.

$$Cov(A) = min(min(Cov(RP) - Cro(RP)), min(Cov(SP) - Cro(SP))) \quad (1)$$

4 Distributed Coverage Optimization Algorithm

4.1 Coverage Calculation of Sensor Node

If we regard the sensing area of a sensor node *s* as the deployed region, then the *Circle Point* of *s* can be considered as the *Region Point* of this circle region, and the *Interior Point* of *s* can be considered as the *Sensor Point*. Based on the *Coverage Calculation Formula* for the deployed region, we can derive a formula to calculate the coverage degree of *s*.

$$Cov(S) = min(min(Cov(CP) - Cro(CP)+1), min(Cov(IP) - Cro(IP))) \quad (2)$$

Different from *Formula (1)*, we should add *1* to the value of $Cov(CP) - Cro(CP)$ because the bound of this region, i.e., the sensing circle of *s*, is included in the count of *Cro(CP)*.

4.2 Design and Analysis of Algorithm

Based on *Formula (2)*, to determine whether a node *s* should be turn active/inactive, we must find all the *Circle Point*s and *Interior Points* of this node. Therefore, all the nodes having intersecting sensing area with *s* should be considered. We call such nodes the *Neighbor Nodes* of *s*, which compose the *Neighbor Set* of *s*.

Definition 6: *Neighbor Set*. Suppose node $s \in S$, for any other node $s_i \in S$, if $d(ss_i) \leqslant Rs(s)+Rs(s_i)$, then $s_i \in NS(s)$.

Now we introduce the *Distributed Coverage Optimization Algorithm*. For any given node *s*, the input is the expected coverage degree *K* and the output is a Boolean Variable *s.active* which denotes whether *s* should be turn active/inactive.

```
Procedure Coverage Optimization(s,K)
 /* Search Neighbor Sensors */
 1: for any s_i ∈ S do
 2:   if d(ss_i) ≤ Rs(s)+Rs(s_i)
 3:     then NS(s) ← s_i; end if end for
  /* Search Region Points of s and put them into Circle Point Set */
 4: CP(s) ← {p | p ∈ the intersection or tangent points of s and A};
  /* Neighbor Set */
 5: n ← NS(s).length;
 6: for i ← 1 to n do
  /* Search Sensor Points of s and put them into Circle Point Set */
 7:   if d(sns_i) ≥ |Rs(s)-Rs(ns_i)|
 8:     then CP(s) ← {p | p ∈ the intersection or tangent points of s and ns_i };
 9:   end if
 /* Search Region Points of Neighbor Sensors and put them into Interior Point Set */
10:   IP(s) ← {p | p ∈ the intersection or tangent points of ns_i and A
                  && d(sp) ≤ Rs(s)};
  /* Search Sensor Points of Neighbor Sensors and put them into Interior Point Set */
11:   for j ← i to n do
12:     if d(ns_i ns_j) ≤ Rs(ns_i)+Rs(ns_j) && d(ns_i ns_j) ≥ |Rs(ns_i)-Rs(ns_j)|
13:       then IP(s) ← {p | p ∈ the intersection or tangent points of ns_i and ns_j
                          && d(sp) ≤ Rs(s)};
14:     end if end for end for
  /* Calculate minimal coverage degree of Circle Points of s */
15: m ← CP(s).length;
16: int minCP ← ∞;            //initialize the minimal coverage degree with a large value
17: for i ← 1 to m do
18:   Cov(cp_i) ← 1;          //cp_i is at least covered by s
19:   for j ← 1 to n do
20:     if d(cp_i ns_j) < Rs(ns_j)    //subtract Cro(cp_i) from Cov(cpi) by excluding the case of "="
21:       then Cov(cp_i)++; end if end for
22:   if Cov(cp_i) < minCP
23:     then minCP ← Cov(cp_i); end if end for
  /* Calculate minimal coverage degree of Interior Points of s */
24: m ← IP(s).length;
25: int minIP ← ∞;            //initialize the minimal coverage degree with a large value
26: for i ← 1 to m do
27:   Cov(ip_i) ← 1;          //ip_i is at least covered by s
28:   for j ← 1 to n do
29:     if d(ip_i ns_j) < Rs(ns_j)    //subtract Cro(ip_i) from Cov(ip_i) by excluding the case of "="
30:       then Cov(ip_i)++; end if end for
31:   if Cov(ip_i) < minIP
32:     then minIP ← Cov(ip_i); end if end for
  /* Calculate coverage degree of s */
33: if minCP > minIP
34:   then Cov(s) ← minIP;
35:   else Cov(s) ← minCP; end if
  /* whether s is an Unnecessary Sensor Node */
```

```
36: if |CP(s)|=0;
37:   then s.active←false;
38:   else s.active←true; end if
   /* whether the coverage degree of s is higher than expected */
39: if Cov(s)>K;
40:   then s.active←false;
41:   else s.active←true; end if
   /* return the optimization result that s should be turned active/inactive */
42: return s.active;
```

5 Experimentation

In this section, we evaluate the *Distributed Coverage Optimization Algorithm* by simulation experiments. Suppose the test region is a *100m*×*100m* rectangular area, and we deploy three homogeneous sensor networks and one heterogeneous sensor network in the test region separately. Suppose the sensing radiuses of sensor nodes in three homogeneous sensor networks are *20m*, *25m* and *30m* respectly, and the heterogeneous sensor network is composed of same number of these three types of nodes.

We investigate the performances of coverage optimization using *DCOA* for these four sensor networks, and all the results in this section are based on at least fives runs with different random network topologies.

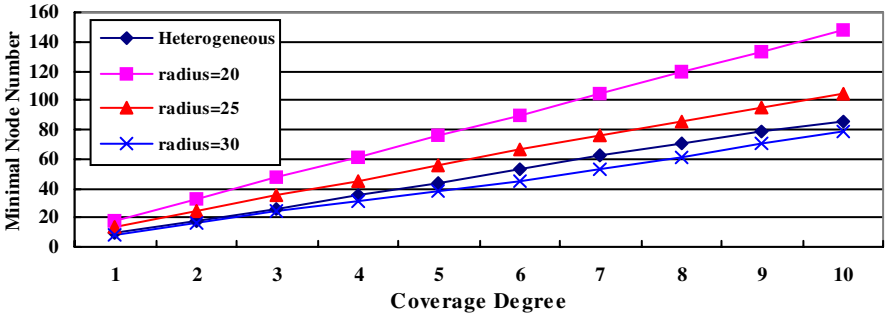


Fig. 5. The minimal numbers of nodes need to remain active

As illustrated in Figure 5, in homogeneous sensor networks, for any expected coverage degree from 1 to 10, the minimal number of nodes that needs to be deployed reduces with the increase of sensing radius. For the heterogeneous sensor network, the minimal number of nodes needs to be deployed in the test region is between the 25m-homogeneous sensor networks and the 30m-homogeneous sensor network.

As illustrated in Figure 6, in homogeneous sensor networks, for any expected coverage degree from 1 to 10, the average coverage degree in the test region increases with the augment of sensing radius. For the heterogeneous sensor network, the average coverage degree in the test region is lower than any of the three homogeneous sensor networks.

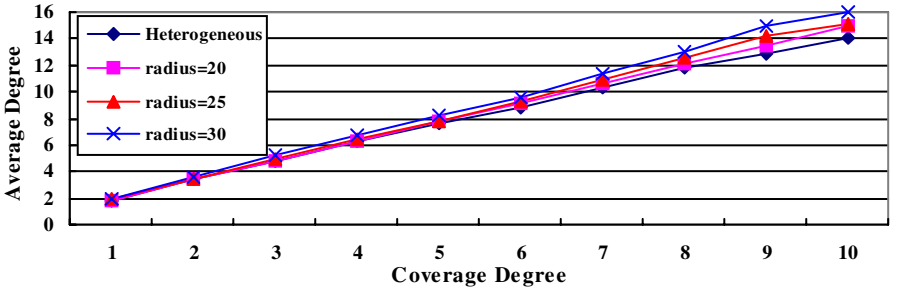


Fig. 6. The average degree of the deployed region

6 Conclusion

In this paper, we explore the problem of coverage in wireless sensor networks. Different from the previous schemes, the algorithm and protocol presented in this paper can be applied to not only homogeneous sensor networks, but also heterogeneous sensor networks. In heterogeneous sensor networks, the difference of sensing radius of sensor nodes should be taken into account because this difference can create *Unnecessary Sensor Nodes* which have no intersection or tangent points with the sensing circle of other nodes. The *Distributed Coverage Optimization Algorithm* proposed in this paper is a good solution for this problem. It can turnoff the abundant nodes including *Unnecessary Sensor Nodes* as well as guarantee the expected coverage degree of the deployed region.

References

1. D. Tian, N.D. Georganas. Location and Calculation Free Node-Scheduling Schemes in Large Wireless Sensor Networks. Ad Hoc Networks Journal, Elsevier Science, Vol.2, Issue 1, Jan. 2004, pp. 65-85
2. A. Savvides, C.-C. Han, and M. B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In ACM Int'l Conf. on Mobile Computing and Networking (MobiCom), pages 166–179, 2001.
3. D. Li, K. Wong, Y.H. Hu, A. Sayeed. Detection, Classification and Tracking of Targets in Distributed Sensor Networks. IEEE Signal Processing Magazine, Volume: 19 Issue: 2, Mar 2002.
4. F. Ye, G. Zhong, S. Lu, and L. Zhang. Energy Efficient Robust Sensing Coverage in Large Sensor Networks. UCLA Technical Report 2002.
5. D. Tian and N.D. Georganas. A Coverage-Preserving Node Scheduling Scheme for Large Wireless Sensor Networks. In processing of ACM wireless sensor network and application workshop (WSNA 2002), Atlanta, September 2002.
6. T.Yan, T.He, J.A.Stankovic. Differentiated Surveillance for Sensor Networks. In proceeding of the First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003), Los Angeles, November 2003.

7. X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, C.D. Gill. Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks. In proceeding of the First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003), Los Angeles, November 2003.
8. H. Zhang, J.C. Hou. Maintaining Sensing Coverage and Connectivity in Large Sensor Networks. Technical report UIUCDCS-R-2003-2351, June 2003.

Simplified Message Transformation for Optimization of Message Processing in 3G-324M Control Protocol*

Man-Ching Yuen [1], Ji Shen [2], Weijia Jia [3], and Bo Han [4]

Department of Computer Science, City University of Hong Kong,
83 Tat chee Avenue, Kowloon Tong, Kowloon, Hong Kong, China
[1] connie.yuen@alumni.cityu.edu.hk
[2] ji.shen@student.cityu.edu.hk
[3] itjia@cityu.edu.hk
[4] Bo.Han@student.cityu.edu.hk

Abstract. 3G-324M is a multimedia transmission protocol designed for 3G communication environment. Meanwhile H.245 standard is a control protocol in 3G-324M and gives specific descriptions about terminal information messages in H.245 control channel as well as the procedures using them. The message syntax is defined using an external data representation standard called Abstract Syntax Notation One (ASN.1). For transmission, ASN.1 formatted data is transformed into bit-stream based on an ASN.1 encoding standard called Packed Encoding Rules (PER). In order to meet the requirement of high speed data transfer in 3G communication, it is important to design the procedure of message processing as simple as possible. In this paper, we propose *Single-step Direct Message Transformation (SDMT)* for the optimization of tree-structured message processing in H.245 module. By testing in realistic environments in some China industries, performance evaluation shows that code redundancies in terms of file size and code size are reduced significantly.

1 Introduction

With wider bandwidth of third-generation networks (3G) and increasing number of multimedia service categories, the mobile communication market has grown at an explosive rate in recent years, especially 3G is launching in different places in the world. 3G wireless multimedia communications are particularly referred to as International Mobile Telecommunications 2000 (IMT-2000) that has been deployed and developed substantially. 3G-324M [1] is a standard umbrella protocol for supporting multimedia transmission using 3G technologies. In 3G communication environments, it consists of a signaling channel which is used for the exchange of capabilities and opening of video, audio and data channels between two different phones. The signaling channel is defined by H.245 protocol [2].

* The work is supported by Research Grant Council (RGC) Hong Kong, SAR China, under grant nos.: CityU 1055/00E and CityU 1039/02E and CityU Strategic grant nos. 7001709 and 7001587 and This paper is sponsored by 973 National Basic Research Program, Minister of Science and Technology of China under Grant No. 2003CB317003.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 64 – 73, 2005.

H.245 standard has been defined to be independent of the underlying transport mechanism, but is intended to be used with a reliable transport layer, which provides guaranteed delivery of correct data. H.245 specifies syntax and semantics of messages as well as the procedures for in-band negotiation at the start of or during communication. The message syntax is defined using Abstract Syntax Notation One (ASN.1) [3]. ASN.1 is a specification language for describing structured information using in communication protocols and allows a protocol designer to define parameters in Protocol Data Units (PDU) without concerning how they are encoded for transmission. The definition of data types in ASN.1 can be grouped into two categories: primitive type and constructive type. If the data value contains other data values, then it is defined as a constructive type, otherwise it is a primitive type. There are over 20 primitive data types in ASN.1 including BOOLEAN, INTEGER, ENUMBERATED, REAL, BIT STRING, OCTET STRING, NULL, ANY and OBJECT IDENTIFIER. Examples of constructive data types in ASN.1 are SET, SEQUENCE, SET OF, SEQUENCE OF and CHOICE. For transmission of messages in H.245 control channel, ASN.1 formatted data is transformed into bit-stream based on an ASN.1 encoding standard called Packed Encoding Rules (PER) [4]. PER is one of derivatives of Basic Encoding Rules (BER) [5] and especially designed for high speed data transfer. PER provides a much more compact encoding than BER and tries to represent data units using the minimum number of bits.

A BER encoding is highly structured [5] and comprised of a sequence of octets. In BER, ASN.1 uses Tag-Length-Value (TLV) format for encoding the data types. The TAG field consists of a tag which is uniquely associated with a specific ASN.1 data type. The LENGTH field indicates the length of the contents encoded in case of definite length encoding, while indefinite length encoding uses an end-of-contents (EOC) indicator to delimit the contents. The VALUE field contains either a value of a primitive type or values of different component types of a constructive type.

PER is not a TLV style of encoding, so tags are not encoded at all. Only data of some primitive types will have their length encoded. Data of constructive types do not have their length encoded explicitly; instead they rely on their components to define their length. The compactness of PER encodings requires that the decoder knows the complete original abstract syntax of the data structure to be decoded. In other words, PER encodings are not self-defining, thus less flexible than BER encodings. As a result, the implementation of PER is much complicated than BER.

With the control of H.245 module during a communication session, messages are generated and encoded into binary bits streams in ASN.1 syntax based on PER. After that, the encoded bits streams are delivered to transport layer to send to the peer communicators. In both initialization and the communication stages, H.245 module is first invoked, and then messages are generated and sent dynamically during the runtime. As a result, message processing of encoding and decoding may be invoked many times flexibly, and its efficiency will affect the whole performance of H.245 module.

Based on the above observations, in this paper, we propose a *Single-step Direct Message Transformation (SDMT)* for the optimization of tree-structured message processing in H.245 module. It increases the difficulty to design SDMT for PER which is a complicated encoding scheme. Our implementation has been tested in a realistic heterogeneous 3G communication environment in some China industries. It shows that the scheme of SDMT has lower code redundancy and higher efficiency because of the simplified encoding and decoding routines.

The rest of the paper is organized as follows. Section 2 describes the common tree-structured implementations of message processing in H.245 module. Section 3 presents our proposed Single-step Direct Message Transformation (SDMT) for message processing in H.245 module. Its implementation and performance evaluation are presented in Section 4. Section 5 concludes the paper.

2 Tree-Structured Message Processing in H.245 Module

To provide guaranteed delivery of correct data, H.245 specifies syntax and semantics of terminal information messages in H.245 control channel as well as the procedures for in-band negotiation at the start or during the communication. The messages cover receiving and transmitting capabilities as well as mode preference, logical channel signaling and control. In H.245 module, Signaling Entity (SE) is referred to as a procedure that is responsible for special functions. It is designed as state machine and changes its current state upon reaction to an event occurrence.

In H.245 module, messages are defined in tree-like structure. H.245 defines a general message type *MultimediaSystemControlMessage* (MSCM). Four types of special messages are further defined in MSCM as *request*, *response*, *command* and *indication*. A *request* message results in a specific action and requires an immediate response. A *response* message responds to a request message. A *command* message requires an action but no explicit response. An *indication* message contains information that does not require action or response. Messages with various types are transformed into MSCM for uniform processing and parameters in MSCM are set to distinguish different types of messages.

Each of the four types of special messages has a number of its own subtypes, and is further defined as one of them. The number of subtypes of request, response, command and indication is 16, 25, 13 and 24 respectively. A message defined with subtype consists of values of a number of elements of ASN.1 notation. These elements may be of primitive type or constructive type. As mentioned in Section 1, a constructive type is defined by a number of primitive types and constructive types. An illustrative example is shown in the following. For MasterSlaveDetermination type, it is defined as a RequestMessage in the first level of message definition, and then the definition is refined as a MasterSlaveDeterminationMessage in the second level of message definition. Finally, the definition of all elements of MasterSlaveDeterminationMessage is declared in the third level of message definition.

Message Definition in H.245 Specification

```
Level 1 Definition (Definition of MultimediaSystemControlMessage)
MultimediaSystemControlMessage       ::=CHOICE
{
    request   RequestMessage,
    response  ResponseMessage,
    command   CommandMessage,
    indicationIndicationMessage,
    ...
}
```

Level 2 Definition (Definition of RequestMessage of MultimediaSystemControlMessage)

```
RequestMessage::=CHOICE
{
    nonStandard NonStandardMessage,
    masterSlaveDetermination MasterSlaveDetermination,
    terminalCapabilitySet TerminalCapabilitySet,
    openLogicalChannel OpenLogicalChannel,
    closeLogicalChannel CloseLogicalChannel,
    requestChannelClose RequestChannelClose,
    multiplexEntrySend MultiplexEntrySend,
    requestMultiplexEntry RequestMultiplexEntry,
    requestMode RequestMode,
    roundTripDelayRequest RoundTripDelayRequest,
    maintenanceLoopRequest MaintenanceLoopRequest,
    ...,
    communicationModeRequest CommunicationModeRequest,
    conferenceRequestConferenceRequest,
    multilinkRequest MultilinkRequest,
    logicalChannelRateRequest LogicalChannelRateRequest,
    genericRequest GenericMessage
}
```

Level 3 Definition (Definition of MasterSlaveDeterminationMessage of RequestMessage of MultimediaSystemControlMessage)

```
MasterSlaveDetermination ::=SEQUENCE
{
    terminalType INTEGER (0..255),
    statusDeterminationNumber INTEGER (0..16777215),
    ...
}
```

Since H.245 defines messages in tree-structure, following the specification in H.245, the procedure of message processing is also in tree-structure. In tree-structured message processing approach, it contains numerous encoding/decoding routines and parsing routines. They are in parallel to corresponding data representations at specific message definition levels in H.245 protocol. Hence, the top-level encoding/decoding routines correspond to the definition of messages, and the bottom-level contains the encoding/decoding routines of each ASN.1 given data type. The encoding/decoding routines provide translations of ASN.1 data between abstract type and transfer type, while the parsing routines are designed for classification of elements of a message definition level in H.245 protocol.

Fig. 1 shows the flowchart of tree-structured message processing in H.245 module. In order to process messages in H.245 module, the top-level parsing routine calls the lower level parsing routines, and the parsing routines at different levels set values to classify their element types which are stored as intermediate data representations in buffer. The process of parsing continues and is complete when the lowest level of parsing is called. Once messages are classified, by using the intermediate data stored in buffer, the encoding process starts to transform the terminal information message

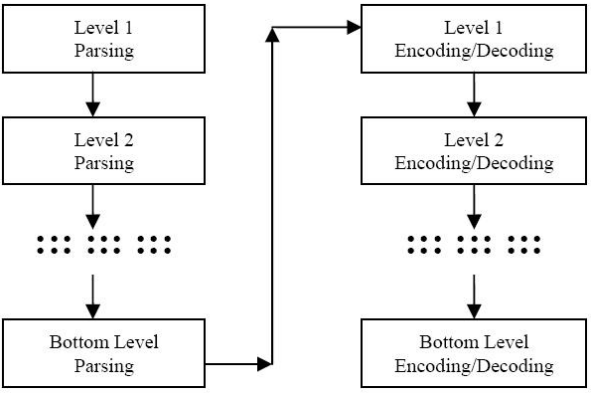


Fig. 1. Flowchart of Tree-structured message processing in H.245 module

into bit-stream. The top-level encoding routine calls the lower level encoding routines and the encoding routines at different levels set values to the corresponding items in a message. The processes go ahead until the messages are encoded into a bit-stream eventually. The similar work is done for the decoding process.

The tree-structure approach works efficiently in dealing with maintenance. However, as parsing process has to be completed before encoding/decoding process starts, the tree-structured definition in H.245 protocol has to be considered in parsing process and encoding/decoding process separately. The lengthy and complicated encoding rules are executed step by step along the tree-structure for twice. Note that the lower the level of definition of message is, the higher the complexity of message definition is. Thus, the message processing is still complex especially for a complicated encoding scheme such as PER and it results in a high code redundancy.

For existing implementation, codes of encoding and decoding are usually automatically generated by ASN.1 compilers. The ASN.1 compiler proposed in [6] called CASN.1 translates data without converting data into an intermediate form but is only based on the fundamental encoding scheme BER, while PER for high speed data transfer in 3G is much complicated and not suitable to use. Based on these observations, in this paper, we propose Single-step Direct Message Transformation (SDMT) in H.245 module. The methodology of SDMT will be described in detail in Section 3.

3 Single-Step Direct Message Processing (SDMT) in H.245 Module

For data transfer, terminal information messages have to be processed in H.245 module before transmission. In H.245 module, all messages are expected to be parsed into MSCM and then encoded by PER. Thus messages received from peer terminals should be parsed and decoded for further processing. The implementation of parsing and encoding/decoding involves some similar memory access and bitwise logical operations. To better utilize the limited resources of a terminal, we propose a simple and efficient approach on implementation of message processing in H.245 module

called *Single-step Direct Message Transformation* (SDMT), in which some procedures in parsing and encoding/decoding are compressed and integrated.

The differences of processing flow between tree-structured implementation and our single-step implementation are described as follows. In the tree-structured implementation, to transfer data, we need to parse a semantic terminal information message into MSCM in accordance with tree-structure specification, and then MSCM is encoded into a bit-stream by PER based on the tree-structured specification again. As the tree-structure specification has to be referred twice, the recursive callings of descend functions are executed twice, one for parsing and the other for encoding. Besides, the data received by peer terminals is also parsed and decoded for further processing. In our implementation, in order to simplify the process, we have combined the parsing and encoding procedures into one procedure by directly transforming the message (i.e., the leaves of the tree) into a bit-stream. Based on the same idea, the decoding procedures are also combined with the corresponding parsing procedures. As the message is not reduced into MSCM as defined in H.245 specification, the complexity of message transformation between semantic messages of ASN.1 notation and encoded messages with use of PER is greatly decreased.

Fig. 2 shows the flowchart of Single-step Direct Message Processing (SDMP) in H.245 module. The top-level encoding routine calls the lower level encoding routines. Unlike tree-structured message processing, SDMP combines the parsing and encoding routine of the same level into one integrated encoding routine. Thus the encoding routine at each level calls the corresponding parsing routine, classifies the structure of message at the view of the level, sets values to the corresponding items and then encodes them accordingly. The processes go ahead until messages are encoded into a bit-stream eventually. The similar work is done for the decoding process.

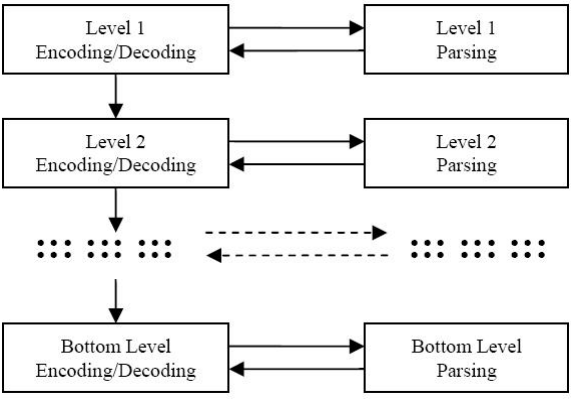


Fig. 2. Flowchart of Single-step Direct Message Processing in H.245 module

Our implementation has two main characteristics: (1) *Parsing procedures execute within the encoding/decoding procedures at each level.* It differs from tree-structured approach which all parsing procedures have to be completed before the encoding procedures start. The reason for the feasibility of our approach is described as follows.

In H.245 module, information messages are defined and presented in the form of a number of levels for increasing readability when implementing the protocol. SDMP translates data values directly between the ASN.1 formatted data structures and the PER transfer syntax thus eliminating the overhead in converting the data into an intermediate stage. In this way, it greatly reduces the encoding/decoding codes with same maintainability and simplifies their complexity of encoding/decoding operations.

(2) For high-level messages containing a number of ASN.1 data values, in our simplified implementation concepts, a high-level message and its ASN.1 data values can be viewed as a linked list and nodes connected by the linked list respectively. It makes the coding in encoding/decoding procedures still to be readable and maintainable. In abstract concept, each information message contains a number of elements that are ASN.1 data structure. An element in a message is taken as a node and the nodes in a message are linked together into a list that represents the message. It gives a quick mapping to the syntax of MSCM. All messages in a communication sessions are further linked together into a global list. The global list refers to the encoded bit stream, which is the output of bottom level encoding routine. In [4], PER rules are defined according to different data types, thus we have implemented the explicit encoding/decoding functions for each data type accordingly. When encoding/decoding a message, the nodes in the message list are encoded one by one using corresponding PER data type encoding functions. This approach is still easy to manage, and we have packed the procedures into our final implementation. As a result, based on this approach, the PER codec in our implementation is simplified without increasing the workload of encoding/decoding operations or modifying the syntax and semantics of the messages.

Without encoding tags indicating types of encoded data, PER is less flexible than BER. Thus the design of ASN.1 compiler with the use of PER is much complex than BER, and it is necessary to have a simpler design approach for PER codec compared with BER codec. Our PER codec implementation is simplified based on Single-step Direct Message Processing (SDMP) design approach compared with that of others. In SDMP, each encoded ASN.1 data structure can be self-descriptive and used in high-level encodings directly. In this way, without implementing the intermediate syntax in our PER codec, specific ASN.1 syntax specification information still can be included in encoded bit stream, thus enough information is obtained for decoding process in PER after data transfer. Moreover, our H.245 module is much simple in design and implementation.

4 Our Implementation

Our PER codec is written in programming language C. We select C language as our target language, because translation for most data types between ASN.1 and C can be achieved simply by direct mapping. As ASN.1 contains a richer set of types, some ASN.1 types require programmer defined C structures for the translation. However, all the ASN.1 types can be simply represented by using C structures. The major part of the ASN.1 standard is presently implemented in our PER codec.

In the implementation of our PER codec, the C files contain encoding and decoding routines for data types of both primitive type and constructive type. Note that in

SDMP, each encoded ASN.1 data structure can be self-descriptive and used in high-level encodings directly. We present the implementation issues in our Single-step Direct Message Processing (SDMP) in H.245 module in Section 4.1 and evaluate the performance of our approach comparing with the tree-structure message processing approach in Section 4.2.

4.1 Message Processing

In our PER codec, without implementing the intermediate syntax, specific ASN.1 syntax specification information still can be included in encoded bit stream, thus enough information is obtained for decoding process in PER after data transfer. Moreover, our H.245 module is much simple in design and implementation. In this part, the implementation of message processing of our PER codec is presented in detail. As the definition of MasterSlaveDetermination message is shown in Section 2, it is also used as an example to demonstrate our implementation. The following shows the pseudo code for encoding of MasterSlaveDetermination message.

Pseudo code for encoding of MasterSlaveDetermination message

```
EncodedBits* h245_encode_MasterSlaveDetermination(BYTE *bit_stream, int
*pos, int terminalType, int statusDeterminationNumber)
{
   EncodedBits *head,*tail,*newBits
   enc_init(&head,&tail))
    /* encode the choice in MultimediaSystemControlMessage */
   newBits = enc_choice(3,RequestMessage_chosen,4)
   encode_append_bits(head,&tail,newBits)
    /* encode the choice in RequestMessage */
   newBits = enc_choice(10,MasterSlaveDetermination_chosen,16)
   encode_append_bits(head,&tail,newBits)
    /* encode the extension marker in MasterSlaveDetermination message */
   newBits = enc_seq(0,1)
   encode_append_bits(head,&tail,newBits)
    /*** start encoding content of MasterSlaveDetermination message ***/
    /* encode the terminalType */
   newBits = enc_integer(0,255,terminalType)
   encode_append_bits(head,&tail,newBits)
   /* encode the statusDeterminationNumber */
   newBits = enc_integer(0,16777215,statusDeterminationNumber)
   encode_append_bits(head,&tail,newBits)
    /*** end of encoding content of MasterSlaveDetermination message ***/
    /* concatenate the encoded bit stream */
   enc_concatenate(head,&tail)
    return head
}
```

There are four input parameters: (1) *bit_stream* is a pointer pointing to the input bit stream data; (2) *pos* indicates the bit position of the first byte of the bit_stream; (3) *terminalType* is one of the two elements of MasterSlaveDetermination message and it is an integer between 0 to 255; (4) *statusDeterminationNumber* is another element of MasterSlaveDetermination message and it is an integer between 0 to 16777215. After

initialization of a pointer of bit stream, the message is encoded based on the first level message definition, which is MultimediaSystemControlMessage. It is CHOICE type with an extension marker after index 3 and has 4 elements totally. MasterSlaveDetermination message is defined as a RequestMessage in MultimediaSystemControlMessage definition, while *Request_chosen* is a constant representing the index of RequestMessage in the MultimediaSystemControlMessage definition. Note that, the encoded bits resulted from the encoding of each level message definition are appended to a bit stream. In next step, message encoding is based on the second level message definition, Request message. Request message is a CHOICE type with an extension marker after index 10 and has 16 elements totally. *MasterSlaveDetermination_chosen* is a constant representing the index of MasterSlaveDetermination message in the RequestMessage definition.

MasterSlaveDetermination message is of SEQUENCE type. For encoding of SEQUENCE type, it has an extension marker but without extension part, so that the first input parameter is 0. The bit map is 1 because the message has two elements and the index of the last element is 1. Note that the indexing starts from 0. The two INTEGER elements of the MasterSlaveDetermination message are then encoded and appended to the bit stream. Finally, the bit stream is concatenated in the octet string (a multiple of 8 bits) if aligned PER encoding is used.

The decoding procedure of MasterSlaveDetermination message is just the reverse of the encoding process of the message. Firstly the MasterSlaveDetermination message is decoded based on the first level message definition MultimediaSystemControlMessage, and the index of RequestMessage in the CHOICE is resulted. Then, the message is decoded based on the second level message definition RequestMessage, it results the index of MasterSlaveDetermination message in the CHOICE. Finally, the decoding of the bottom level message definition (decoding of the ASN.1 basic data types) carries out, and all the elements of MasterSlaveDetermination message, *terminalType* and *statusDeterminationNumber*, are decoded.

4.2 Performance Evaluation

In this section, we evaluate the performance of our Single-step Direct Message Processing (SDMP) approach comparing with the tree-structure message processing approach. Our implementation has been tested in a realistic heterogeneous 3G communication environment in some China industries. Applying Single-step Direct Message Transformation (SDMT) in H.245 module, the intermediate processes are skipped and the complexity of the overall process in H.245 is expected to be decreased. Two measurement criteria are used to evaluate the performance for code redundancies called *file size* and *code size*. The file size is the size of files containing the source code, while the code size is the size of files containing object code originated from compilation. Table 1 shows the performance comparison on H.245 module between two implementation approaches, and they are traditional tree-structured message processing and our Single-step Direct Message Transformation (SDMT).

Table 1. Performance Comparison on H.245 module between two implementation approaches

	File Size (Source code)	Code Size (Object generated from compilation)
Traditional approach	434 KB	261 KB
Our proposed SDMT	306 KB	221 KB
Reduction Rate of SDMT	**29.5%**	**15.3%**

The file and code sizes of the implementation of traditional design as the PER specification are 434 KB and 261 KB, respectively. In the implementation of Single-step Direct Message Transformation (SDMT), the file size is 306 KB and the code size is 221 KB. As a result, the reduction rates in file size and code size of our approach are 29.5% and 15.3% respectively. It shows a significantly reduction in code redundancies, which is very important for a mobile terminal with limited memory.

5 Conclusion

In this paper, we proposed *Single-step Direct Message Transformation (SDMT)* for the optimization of tree-structured message processing in H.245 module, which is the control module defined in an umbrella protocol 3G-324M for supporting multimedia communication in 3G environment. SDMT simplifies the encoding/decoding routine in H.245 message processing even using PER codec which can give high compression rate but also high implementation complexity. Performance evaluation shows that code redundancies in terms of file size and code size are reduced significantly.

References

1. ITU-T H.324, Terminal for low bit-rate multimedia communication, Int'l Telecommunication Union, 2002.
2. ITU-T H.245, Control protocol for multimedia communication, Int'l Telecommunication Union, 2003.
3. ITU-T ISO/IEC IS 8824: 1987, Information processing systems – Open Systems Interconnection – Specification of Abstract Syntax Notation One (ASN.1), Int'l Telecommunication Union, 1987.
4. ITU-T ISO/IEC IS 8825-2: 1995, Information Technology – ASN.1 encoding rules: Specification of Packed Encoding Rules (PER), Int'l Telecommunication Union, 1995.
5. ITU-T ISO/IEC IS 8825: 1987, Information processing systems – Open Systems Interconnection – Specification of Basic Encoding Rules for Abstract Syntax Notation ONE (ASN.1), Int'l Telecommunication Union, 1987.
6. G.W. Neufeld, Y. Yang, "The Design and Implementation of an ASN.1-C Compiler," IEEE Transactions on Software Engineering, vol. 16, no. 10, Oct. 1990, pp. 1209-1220.

Dynamic Packet Scheduling Based on Utility Optimization in OFDM Networks

Kunqi Guo[1], Shilou Jia[1], and Lixin Sun[2]

[1] Department of Communication Engineering, Harbin Institute of Technology,
65 Sidazhijie Street, Harbin, Heilongjiang, China
guokunqi2004@163.com, Shljia@hit.edu.cn
[2] Department of Communication Engineering,
Harbin University of Science and Technology,
65 Xuefu Street, Harbin, Heilongjiang, China
xinleo2004@163.com

Abstract. A scheduling scheme is proposed to dynamically allocate resources for the downlink data transmission in orthogonal frequency division multiplexing (OFDM) networks. In addition, an algorithm with linear complexity is also presented that is based on maximizing the utility function with respect to average waiting time to allocate subcarriers among users. The total utility is obtained by the algorithm taking advantage of multiuser diversity which includes the current channel conditions and queue length. Several feasible assumptions are allowed to achieve high efficiency while maintaining fairness, which involve frequency and time multiplexing, the information of channel condition being available to the scheduler by pilot signals and each user having a buffer with large capacity for its incoming packets. We demonstrate the effectiveness of our scheme through mathematical analysis and simulation.

Keywords: OFDM, utility function, packet scheduling, fairness, delay-sensitive.

1 Introduction

The unique characteristics of wireless channel created many technical issues for efficient resource scheduling. *Quality-of-Service* (*QoS*) challenges for high-speed bursty data traffic over wireless fading channels are provided due to limited bandwidth, time-varying fading channels, and resource competition among multiple users. Utilizing multiuser diversity, the base station can schedule transmissions to users when their channels are in good condition because of the delay tolerance of data traffic. However, many applications such as music, video streams are delay-sensitive. The gain of multiuser diversity that utilizes independent channel fluctuations is restricted by the relatively low latency tolerance of applications. Therefore, a cross-design is needed that balances delay *QoS* and efficient resource utilization by making use of information of channel and queuing states as well as user's subjective performance metrics.

Recently, adaptive resource management for multiuser orthogonal frequency division multiplexing (OFDM) systems has attracted enormous research interests. It

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 74 – 83, 2005.

was proved in [1], [2] that the system spectral efficiency can be greatly enhanced by adjusting the allocation of subcarrier, power and constellation sizes in accordance with the user's channel conditions. However, with the fast emergence of wireless packet-access services, new issues arise because packets arrive according to a random process. Thus, the resource allocation algorithms should be able to utilize the traffic variation as well as queue state information. In spite of system efficiency being crucial, fairness among the operating users is also an important factor. Therefore, resource allocation schemes should be able to provide fairness to all traffics admitted by the system since different subcarriers have different channel qualities and there are always some subcarriers with good conditions to be used for packets transmission. Much of work deals with modifications of fair queuing scheduling or earliest due date previously developed in wireline network [3], [4]. These modified schemes do not exploit multiuser diversity to enhance efficiency. *Proportionally fair* (*PF*) scheduling is studied in [5], [6]. Its objective is to maximize the long-term throughput of users with respect to their average channel conditions, however, it is not efficient for delay-sensitive applications. In [7], *Max delay utility* (*MDU*) is investigated that takes channel conditions and queue length into account, but does not consider *empty-after-service* (*EaS*) events.

In this paper, we focus on data services that are delay-sensitive and the scheduling for downlink data transmission in OFDM networks is investigated. It is verified in [8] that maximizing utility can automatically balance resource efficiency and fairness. Based on utility function relative to average waiting time, an algorithm is proposed to dynamically allocate subcarriers among users which considers the current subcarrier's channel conditions and queue length. A utility function, $U(W) = -W^r / r \ (r \geq 1)$ is introduced to guarantee fairness. Compared with *PF* and *MDU* algorithms, the proposed algorithm can provide better delay performance and enlarging the queue stable region. The effectiveness of our scheme is verified through complexity analysis and simulation.

2 System Model

OFDM provides a physical layer basis for multiple shared channels. The scheduler at base station (BS) simultaneously serves *M* users. To obtain high performance, in this paper, it is assumed that frequency and time multiplexing is allowed in the whole resource, the channel state information, *signal-to-noise* (*SNR*) is obtained by the scheduler by pilot signals and each user has a buffer with large capacity for its incoming packets. To predict channel conditions, the BS transmits the pilot signals to each user. Upon receiving the pilot signals, users estimate the channel state information, *signal-to-noise* (*SNR*) and feed the information back to BS by which the scheduler at BS can determine the achievable data transmission rate and allocate subcarriers for users. Having obtained the channel conditions and the required *BER* of each user, scheduler allocates subcarriers based on average waiting time. The granularity offered by OFDM is exploited in this scheme. Packets scheduling for downlink data transmission is achieved using the proposed algorithm. The scheduling model for downlink data transmission with *N* subcarriers and *M* users is shown in Fig. 1.

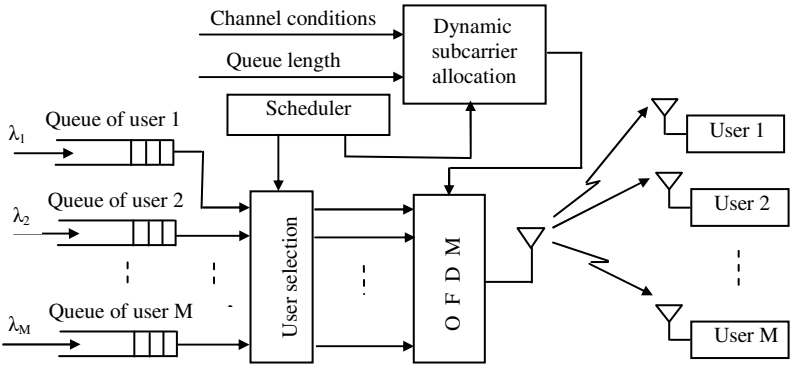


Fig. 1. Scheduling for downlink packet transmission in OFDM networks

3 Packet Scheduling

3.1 Problem Formulation

Assume that total bandwidth is B consisting of K subcarriers channels, hence each subcarriers has a bandwidth of $\Delta f = B / K$. Let S denote the subcarrier index set $S = \{1, 2, \ldots, K\}$. The lower the variance of the service transmission rate, the shorter queuing latency according to the queuing theory [9]. A high peak data transmission rate contributes to shortening the delay of bursty traffic. Our objective is to minimize queuing delay by dynamic allocating subcarriers in terms of the current queue length and channel quality. Considering delay-sensitive applications, when the traffic load is heavy, the violation probability, $Pr\{.\}$can be denoted by: $Pr\{T > T^{max}\} \approx \exp\{-T^{max} / W\}$ that can be utilized in $G1/G/1$ queues. T and T^{max} are a packet waiting time and the delay bound respectively. W is the expected value of T, $E[T]$. Hence, it is acquired that shortening the average waiting time is approximately equivalent to minimizing the delay violation when the traffic is heavy. It is reasonable to formulate optimization problems using the mean of delays since most delay violations are caused during heavy traffic.

Utility theory provides the exact means to formulate the relations between user experience and various network performance matrices. It relates network resource to user-perceived application performance [10]. In this paper, we denote utility function as $U(r)$ that relates effective serving rate to user's application requirement that is delay-sensitive by using average waiting time.

3.2 Utility Function

Assume that control mechanism is perfect which is typically a combination of link layer ARQ and advanced modulation schemes obtained by selecting the appropriate one, therefore, for delay-sensitive applications, $U(r)$ depends upon channel quality that is determined by the current channel *signal-to-noise ratio* (*SNR*) and the resources that are the subcarriers and buffers provided for users. In terms of the user's

perception of application performance, the required *bit-error-rate* (*BER*) is considered when formulating a utility function. Let the achievable data transmission rate per *Hz* for user *i* on subcarrier *k* during time slot n be $C_i[k,n]$. In a general way, $C_i[k,n]$ are determined by the current channel *SNR* and the bit-error-rate (*BER*) required by user. When continuous rate adaptation is exploited, the achievable transmission rate (per *Hz*) on subcarrier *k* for user *i* can be formulated as a function relative to the *SNR*, $\rho_i[k,n]$ [10]. $C_i[k,n] = log_2 (1 + \beta \rho_i[k,n])$, and β is a constant determined by the required *BER*, $\beta = -1.5 / ln (5BER)$.

Establishment of the Utility Function. Let $D_i^{(n)}$ denote the set subcarriers indices allocated to user *i* at time *n*. In OFDM systems, since each subcarriers cannot be shared by multiple users, we have:

$$D_i^{(n)} \cap D_j^{(n)} = \phi, \quad \forall \ i \neq j \tag{1}$$

With subcarrier allocation, when continuous rate adaptation is used, the data transmission rate of user *i* at time slot *n*, $r_i[n]$ can be expressed by [11]:

$$r_i[n] = \sum_{k \in D_i^{(n)}} c_i[k,n] \, \Delta f \tag{2}$$

For a given user service, the required *BER* is a constant, which means the corresponding β is fixed. Therefore, $c_i[k,n]$ is only determined by $\rho_i[k,n]$. In a OFDM system, Δf is fixed, hence, we can express $r_i[n]$ as a function with respect function $\rho_i[k,n]$. For delay-sensitive applications, according to: $P_r\{ T > T^{max}\} \approx \exp\{-T^{max} / W\}$, when keeping T^{max} constant, the delay violation in the heavy-traffic scenario can be minimized by shortening the average waiting time. Without loss of generality, most delay violations occur during heavy traffic load, thus we consider it reasonable to formulate optimization problems by exploiting the mean of delays. Denote the average arrival bit rate of user *i* as λ_i, which is defined to be:

$$\lambda_i = \frac{1}{T_s} \lim_{n \to \infty} \frac{A_i[n]}{n} \tag{3}$$

where, $A_i[n]$ represents the total amount of bits arriving during $(0, nT_s)$ and T_s is the length of each time slot in OFDM system where signaling is time-sloted. We denote $Q_i[n]$ as the amount of bits in the queue of user *i* at time nT_s. Assuming that $Q_i[n]$ is ergodic, with Little's law, we express the average waiting of time for user *i* as W_i, $W_i = Q_i/\lambda_i$ and have:

$$Q_i = \lim_{N \to \infty} \left(\sum_{n=o}^{N-1} Q_i[n] \right) \Big/ N \tag{4}$$

The scheduler serves user *i* at rate $r_i[n]$ during time slot *n*. Let $a_i[n]$ denote the amount of arrival bits during time slot *n*, the queue length of user *i* at time $(n+1)T_S$, $Q_i[n+1]$ can be expressed as, $Q_i[n+1] = Q_i[n] - r_i[n]T_s + a_i[n]$. Let T_w denote a time

window length. When $\rho_w = T_s / T_w$, the average queue length over the time window of user i at time nT_s, $\bar{Q}_i[n]$ is derived from:

$$\bar{Q}_i[n] = (1-\rho_w)\bar{Q}_i[n-1] + \rho_w Q_i[n] \tag{5}$$

Let the average waiting time over time window at time nT_s be $W_i[n] = \bar{Q}_i[n] / \lambda_i$, since $Q_i[n]$ is ergodic, the predicted average waiting time at the end of time slot n is computed through:

$$\hat{W}_i[n+1] = \bar{Q}_i[n+1] / \lambda_i \tag{6}$$

Since the expectation of $a_i[n]$ is equal to its average value, using $E\{a_i[n]\}=\lambda_i T_s$ and according to (4),(5),(6), we have:

$$\hat{W}_i[n+1] = (1-\rho_w)W_i[n] + \rho_w Q_i[n]/\lambda_i + \rho_w T_s - \rho_w/(\lambda_i T_s r_i[n]) \tag{7}$$

Optimization Objective. It is obtained from (7) that the predicted average waiting time at time $(n+1)T_s$ is a function of the service rate during time slot n, $r_i[n]$. The optimization objective is to maximize the total utility function with respect with the average waiting time at each time slot, $U_i(\hat{W}[n+1])$, which is expressed as $\max_{r_i[n], i\in\{1,2...M\}} \sum_{i=1}^{M} U_i(\hat{W}[n+1])$. According to differential calculus theory, it can be derived through: $\partial U / \partial r_i = (-\partial U_i / \partial r_i)(\rho_w T_s / \lambda_i)$ when given the arrival process. Since $r_i[n-1]$ is constant at time slot n, when ρ_w is small enough, the optimization can be obtained by maximizing a linear function with respect to $r_i[n]$:

$$\max_{D_i^{(n)}, i\in E^n} \sum_{i=1}^{M} \left(\left| U_i'(W_i[n]) \right| \hat{\lambda}_i \right) / r_i[n] \tag{8}$$

where, $\hat{\lambda}_i$ that can be estimated with Poisson distribution is the expected value of λ_i and (8) is subject to: $\bigcup_{i\in E^n} D_i^{(n)} \subseteq S, D_i^{(n)} \bigcap D_j^{(n)} = \phi, i \neq j \forall i, j \in E^n$.

$E^n = \{i: Q_i[n]>0\}$, is the user set in which each user's queue is not empty at time nT_s after being served.

4 Subcarrier Allocation Algorithm

Although the optimization objective can be achieved by using (8), there is the case that when $r_i[n]T_s \geq Q_i[n]$, user i 's queue is empty after being served by scheduler, This is called *EaS* case in this paper. In this situation, the scheduler may waste

subcarriers shared by users and the optimization of allocating subcarriers can be obtained by:

$$\max_{D_i^{(n)}, i \in \overline{E}^n} \left(\left| U_i^{'}(W_i[n]) \right| Q_i[n] \right) \Big/ \hat{\lambda}_i T_s \tag{9}$$

where $\overline{E}^n$ is the user set where each user's queue is empty at time nT^{S} after being served. Using (9), some subcarriers provided for the users with the *EaS* case are reassigned to those users without such case to maximize allocating efficiency of each step. To implement it, it must be determined how often the *EaS* event occurs. Assume that for user i, the number of *EaS* events is q_i during $(0, nT_s)$, $i \in \{1,2,\ldots M\}$, the average cycle length of *EaS* event of user i is $L_i = nT_s/q_i$ and we can obtain when $i = 1,2\ldots.M$, L_i is bounded by: $\max\{1/(nT_s/q_i)\} \le 1/L_i \le \sum 1/(nT_s/q_i)$. Moreover, according to [12], we also can obtain the long-term average cycle length for user i, L_i by: $L_i = W_i + 1/\lambda_i$. Therefore, L_i is subject to: $max\{1/(W_i+1/\lambda_i)\} \le 1/L_i \le \sum\{1/(W_i+1/\lambda_i)\}$ ($i = 1,2,\ldots.M$). It is shown that the smaller L_i is the lower W_i is. This means that the downlink is less congested. Let $\hat{m}[k,n]$ be subcarrier k to be user i at time n, and execute the algorithm at start of an *EaS* event, the algorithm is as follows:

Step 1: Obtain the cycle length of *EaS* event, L_i..
Step 2: Compute each user's *BER*, and *SNR*.
Step 3: Allocate subcarriers to user i according to:

$$\hat{m}[k,n] \leftarrow \arg \max_{D_i^{(n)}, i \in \overline{E}^n} \left(\left| U_i^{'}(W_i[n]) \right| \right) Q_i[n] \Big/ \hat{\lambda}_i T_s \tag{10}$$

Step 4: Repeat Step 2, and, if not reaching the start of cycle of next *EaS* event, allocate subcarrier to user i, using:

$$\hat{m}[k,n] \leftarrow \arg \max_{D_i^{(n)}, i \in E^n} \sum_{i \in E^n} \left| U_i^{'}(W_i[n]) \right| \; r_i[n] \Big/ \hat{\lambda}_i \tag{11}$$

Otherwise, return to Step 1.

For the computational complexity, according to (10), (11) and $max\{1/(W_i+1/\lambda_i)\} \le 1/L_i \le \sum\{1/(W_i+1/\lambda_i)\}$ ($i = 1, 2,\ldots M$), it can be verified that the algorithm has linear complexity both in the number of users and the number of subcarriers, hence, it is very simple and efficient.

Based on this algorithm, the utility function, $U(W) = -W^r / r$ ($r \ge 1$) is introduced to provide fairness among users by adjusting r while maintaining the efficiency.

5 Simulation

It the simulation, the number of users M is 20 and each user's channel suffers multipath Rayleigh fading with *bad-urban* delay profile. Let the 20 users have different distance from the BS, thus their average achievable transmission rates are different due to path loss. The 20 users have the same average arrival rate. The

required BER is 10^{-6}. Doppler shift is 15*Hz*, which means each user is slowly moving. The total channel bandwidth is 1.92M*Hz* consisting of 128 subcarriers. For the 20 users, the average achievable transmission rate of best user is 1.26*kbps* (per *kHz*), and that of the worst user is 0.20*kbps* (per *kHz*). An ON-OFF model is used to capture the random traffic stream from 20 users. During a burst ON period, Poisson distribution is utilized to capture the packet arrival rate. An exponential distribution is for the OFF duration [11]. The packet length is constant and a set achievable transmission rate in bits per *Hz*, {1, 2, 4, 6, 8,…} is used instead of being consistent with (2). The performance of the mean delay is expressed as a function with respect to average arrival rate.

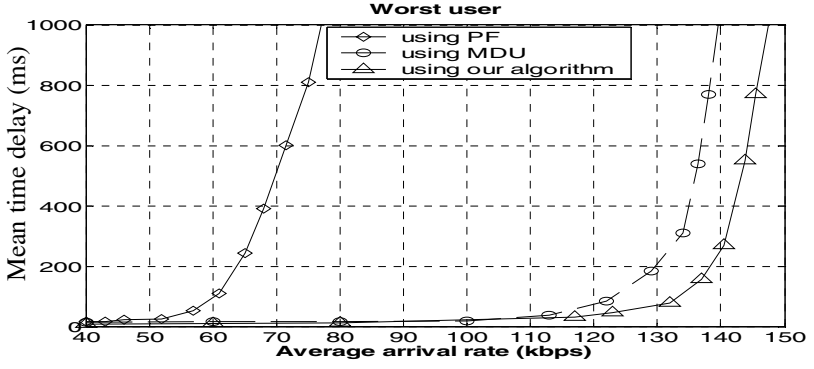


Fig. 2. Delay performance of the worst user with different average arrival rate

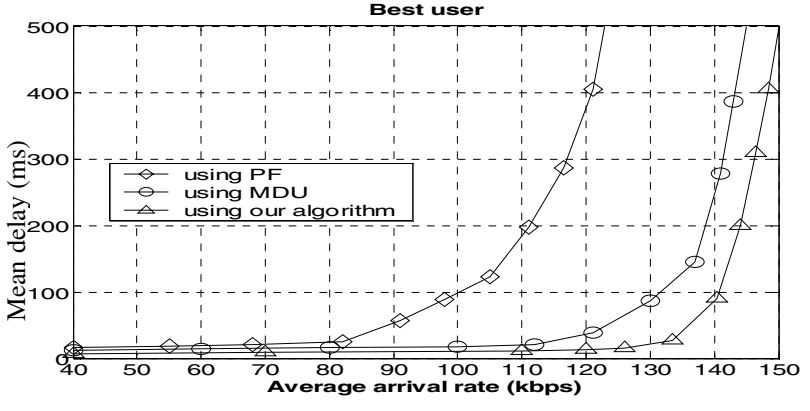


Fig. 3. Delay performance of the best user with different average arrival rate

By the utility function, $U(W) = -W^r / r$ ($r \geq 1$) that follows: $\left| U'(W) \right| = W^{r-1}$, the proposed algorithm are compared with other two algorithms, *PF* that does not consider queue state information and *max MDU* which takes channel conditions and

queue state information into account, but does not consider *EaS* events. Let r =3, the results are shown in Fig.2. and Fig.3.

It is shown in Fig. 2 and Fig. 3 that when *PF* is used, for both the best user and the worst best, if the arrival rate exceeds 80 *kbps,* the system becomes unstable, causing the mean delay to sharply increase. For *MDU* algorithm, the stable region can reach 110*kbps* and the mean delay of the best user is lower than that of the worst user. Utilizing our algorithm, the stable region for the worst user is enlarged to 130*kbps*, and for the best user, it increases to about 132*kbps.*

To estimate the throughput of system, the performance are provided by simulation. In this simulation, we also compare our algorithm with *PF* and *MDU*. The number of users is 20 and the average arrival rate and *SNR* is same for each user. However, different users have different distances from the BS. For all 20 users, the maximum delay is 20*ms* and the length of packet is 8*k* (*bit*). Fig. 4 shows the results that the ratio of throughput is as a function of the average *SNR*.

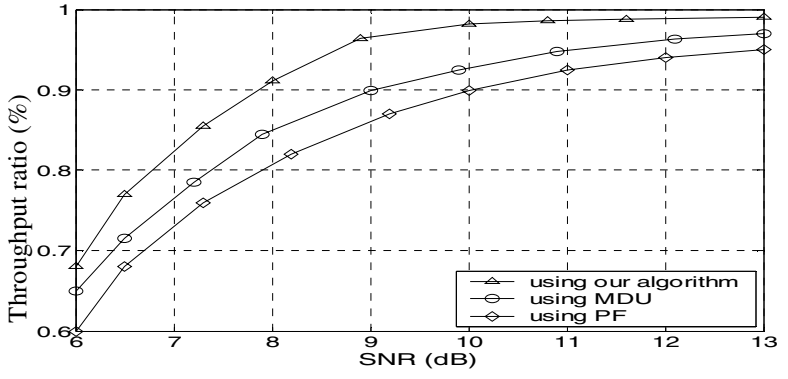


Fig. 4. Throughput ratio of system with respect to the average *SNR*

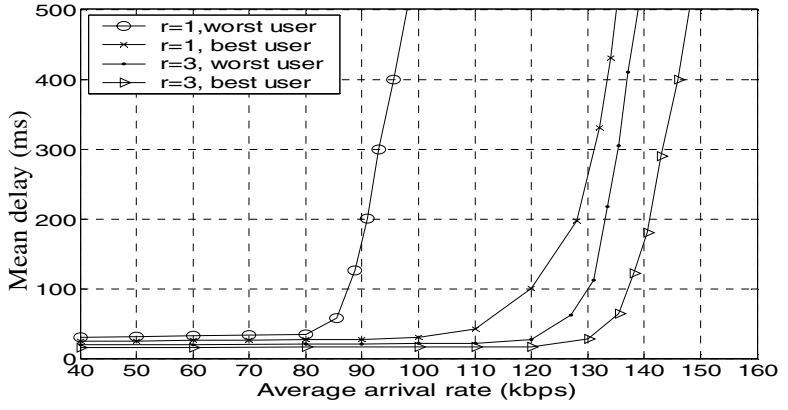


Fig. 5. Mean delay vs. average arrival with different r

Fig. 4 shows that when the average *SNR* is greater than 10dB, the ratio of throughput can arrive at about 97% using our algorithm. However, when *SNR* is11dB, the ratio of throughput correspondingly reaches 94% and 92.5% by using *PF* and *MDU* algorithm respectively. This is because that our algorithm considers the *EaS* events caused by the case that the achievable transmission rate is different among users due to their path difference.

The *utility function,* $U(W) = -W^r / r$ ($r \geq 1$) is introduced that can provide the different degree of delay fairness, which is demonstrated through simulation, shown in Fig. 5.

It is shown in Fig. 5 that regardless of both the best user and the worst user, the stable regions where the value of low mean delay is provided can be enlarged by enhancing *r*. When *r* increases to 3 from 1, for the best user and the worst user, their mean delay is still very low if the average arrival rate is less than 120*kbps*. This provides low delay transmission while guaranteeing the level of fairness.

6 Conclusion

In this paper, concentrating on delay-sensitive applications, we propose a dynamic scheduling scheme for downlink packet transmission in OFDM networks. The current information of buffer length and channel conditions is efficiently utilized to dynamically allocate subcarriers among users. Based on utility function relative to average waiting time, an algorithm is presented that can efficiently allocate subcarriers to users while guaranteeing fairness. Simulation results show that significant extension of stable range and low mean delay can be achieved by using the algorithm. Therefore, the scheme is suitable for delay-sensitive traffic. The efficiency of scheme is verified using mathematical analysis and simulation.

References

1. T. Keller and L. Hanzo, “Adaptive multicarrier modulation: a convenient framework for time frequency processing in wireless communications,” *IEEE Proc, Vol. 88*, pp.611-640, May 2000
2. Y. J. Zhang and K. B. Letaief, “Multiuser subcarrier and bit allocation along with adaptive cell selection for OFDM transmission,” *IEEE Proc. ICC’02. Vol.2*, pp.861-865,2002
3. Y. Cao and V. O. K. Li, “Scheduling algorithms in broadband wireless networks,” *Pro. IEEE, vol.89,no. 1.* pp. 76-87, Jan. 2001
4. J. Cai, X. Shen, and J. W. Mark, “Downlink resource management for packet transmission in OFDM wireless communication systems,” *Proc. IEEE Globecom2003,* San Francisco, CA, Dec. 2003
5. P. Viswanath, D. N. C. Tes, and R. L. Laroia, “Opportunistic beamforming using dumb antemmas,” *IEEE Trans. Inform. Theory, vol. 48, no.6,* pp. 1277-1294, June 2002
6. S. Bost, “User-level performance of channel-aware scheduling algorithms in wireless data network,” *in Pro., IEEE INFOCOM 2003*, Mar 2003, pp. 321-331.
7. G. Song and Y. L, "Utility-based joint physical-MAC layer optimization in OFDM, *IEEE Communications Society,* May 2002, pp. 671-675.

8. G. Song and Y. (G). Li, "Adaptive subcarrier and power allocation in OFDM based on maximizing utility," in Proc., IEEE Veh. Tech. Conf. vol. 2, April 2003, pp. 905-909
9. S. Asmussen, Applied Probability and Queues, 2nd ed. New York: Springern, 2000
10. Zhimei Jiang, Ye Ge, and Ye Li "Max-Utility wireless resource management for best-effort traffic" *IEEE Tran. on wireless communications*, vol.04, No. 1, Jan. 2005
11. X. Qiu and K. Chawla, "On the performance of adaptive modulation in cellular systems," *IEEE Trans. Commun*..,vol. 47, no. 6, pp. 884-895, June 1999.
12. D. Gross and C. M. Harris, Fundamentals of Queuing Theory, 3rd ed New York: Wiley 1998

Comb-Pattern Optimal Pilot in MIMO-OFDM System

Qihong Ge and Huazhong Yang

Dept. of Electronic Engr., Tsinghua Univ., Beijing 100084, China
geqh@wmc.ee.tsinghua.edu.cn

Abstract. Channel estimation is very important for MIMO (Multiple Input Multiple Output) OFDM (Orthogonal Frequency Division Multiplexing) systems, while computation required is relatively large. Block-pattern optimal pilot reduces the computation, but is not suitable for high speed mobile environment. In this paper, comb-pattern optimal pilot is proposed to reduce the error of channel estimation. It can be proved by simulation that performance of system with comb-pattern optimal pilot is much better than that with block-pattern optimal pilot, and the computation required is still small.

Keywords: comb-pattern, pilot, MIMO, OFDM.

1 Introduction

Wide-band wireless communication system faces the challenges of multi-path fading and bandwidth efficiency. OFDM can deal with multi-path fading effectively by changing the wide-band fading channel into several flat sub-channels. On the other hand, MIMO technique can improve the transmission rate without additional bandwidth requirement by introducing multiple transmit antennas and multiple receive antennas. Therefore, a lot of interest has been given to the combination of OFDM and MIMO techniques. [1][2][3][4]

Channel estimation of MIMO-OFDM system is much more complex than that of SISO (Single Input Single Output) OFDM system since the independent channels from transmit antennas to receive antennas share the same bandwidth.[5][6] Block pattern optimal pilot sequence was proposed in [7][8]. It reduces computational complexity by changing the inversion of a common matrix to that of a diagonal matrix. On the other hand, it uses all the sub-carriers in a symbol as pilots, which is only suitable for training period, and is not good for catching up the time-variant property of fading channel.

In this paper, optimal pilot sequence in comb-pattern is proposed. It is verified with simulation that better performance is achieved in high speed mobile environment. In section 2, MIMO-OFDM system is introduced. Optimal pilot sequence in block pattern is discussed in section 3. Section 4 introduces optimal pilot sequence in comb pattern, and section 5 presents simulation results. Finally, conclusion is drawn in section 6.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 84 – 92, 2005.

2 MIMO-OFDM System

The concept of OFDM was proposed by Saltzberg in 1960s, and DFT (Discrete Fourier Transform) was introduced into OFDM in 1971 [9][10].

Consider a data sequence (X_0, X_1, ……, X_{N-1}), where X_k is a complex number expressed as $X_k = a_k + jb_k$. Modulate X_k with orthogonal complex baseband signals, $e^{j\omega_k t}$, and carrier signal, $e^{j\omega_c t}$, we have

$$x(t) = \mathrm{Re}\left[\frac{1}{N}\sum_{k=0}^{N-1} X_k e^{j\omega_k t} \cdot e^{j\omega_c t}\right]. \tag{1}$$

where $\omega_k = k * \omega_0, \omega_0 = 2\pi / T$, T is the symbol duration of the data sequence, and ω_c is the carrier frequency of modulation. The equivalent base-band signal is given by

$$x'(t) = \frac{1}{N}\sum_{k=0}^{N-1} X_k e^{j\omega_k t} . \tag{2}$$

Sampling signal $x'(t)$ at the time $t_n = nT / N$, the process of modulation can be implemented with IFFT (Inverse Fast Fourier Transform) process

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X_k e^{j\frac{2\pi}{N}kn} = IFFT\{X_k\}. \tag{3}$$

Consequently, demodulation is achieved with FFT (Fast Fourier Transform)

$$X_k = FFT\{x(n)\}. \tag{4}$$

Hence, a complete digital implementation could be accomplished to carry out modulation and demodulation of an OFDM system.

To improve capacity of the system, multiple transmit antennas and multiple receive antennas can be adopted. Compared with SISO system with flat fading channel, a MIMO system can improve the capacity by the factor of the minimum number of transmit and receive antennas. [11]

In practical MIMO-OFDM systems, STC (Space Time Code) is introduced to achieve better performance. Fig. 1 presents the structure of a MIMO-OFDM system with STC. Accurate STC decoding needs information of channel, which implies the importance of channel estimation.

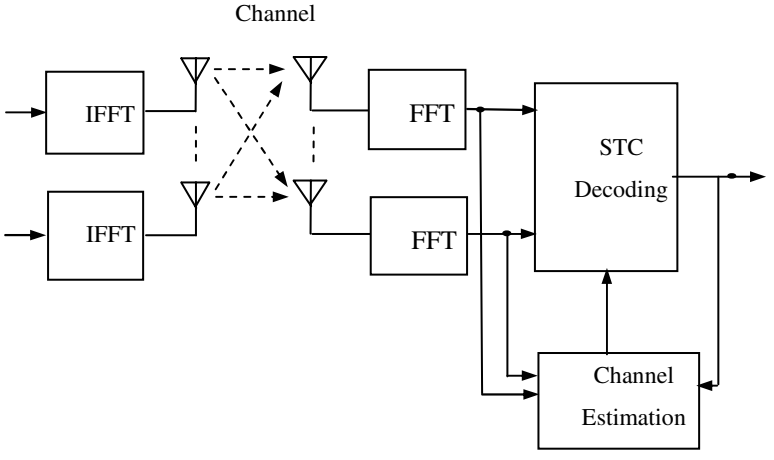


Fig. 1. MIMO-OFDM system with STC

3 Optimal Pilot Sequence in Block Pattern

Consider a MIMO-OFDM system with M transmit antennas, N receive antennas, and K sub-carriers. The received signal of the k-th sub-carrier from the j-th receive antenna in the n-th symbol can be expressed as

$$r_j[n,k] = \sum_i H_{ij}[n,k]t_i[n,k] + \omega_j[n,k]. \tag{5}$$

where $t_i[n,k]$ is the transmitted signal from the *i*-th transmit antenna, $\omega_j[n,k]$ is the noise in channel, and $H_{ij}[n,k]$ is the channel transfer function from the i-th transmit antenna to the j-th receive antenna.

The channel transfer function is independent from different receive antennas, thus the subscript of receive antenna can be ignored, and (5) can be expressed as

$$r[n,k] = \sum_i H_i[n,k]t_i[n,k] + \omega[n,k]. \tag{6}$$

If the impulse response of channel has K_0 non-zero values, the channel transfer function in frequency domain can be derived from

$$H_i[n,k] = \sum_{l=0}^{K_0-1} h_i[n,l] \cdot W_K^{kl}. \tag{7}$$

where $h_i[n,l]$ is the channel impulse response from the *i*-th transmit antenna in the *n*-th OFDM symbol, and $W_K = e^{-j\frac{2\pi}{K}}$.

In a MIMO-OFDM system with two transmit antennas, when all the sub-carriers are used as block-pattern pilots, the channel impulse response is given by [5]

$$\tilde{\mathbf{h}}(n) = \mathbf{Q}^{-1}[n]\mathbf{p}[n]. \tag{8}$$

where $\tilde{\mathbf{h}}(n) = \begin{pmatrix} \tilde{\mathbf{h}}_0(n) \\ \tilde{\mathbf{h}}_1(n) \end{pmatrix}$ is the impulse response from two transmit antennas to receive antenna,

$$\mathbf{Q}[n] = \begin{pmatrix} \mathbf{Q}_{00}[n] & \mathbf{Q}_{01}[n] \\ \mathbf{Q}_{10}[n] & \mathbf{Q}_{11}[n] \end{pmatrix}. \tag{9}$$

can be obtained from Fourier transform of correlation function of the pilots, and

$$\mathbf{p}[n] = \begin{pmatrix} \mathbf{p}_0[n] \\ \mathbf{p}_1[n] \end{pmatrix}. \tag{10}$$

in which

$$\mathbf{p}_j[n,l] = \sum_{k=0}^{K-1} r[n,k] t_j^*[n,k] W_K^{-kl}. \tag{11}$$

But matrix inversion in (8) needs much computation. According to this, optimal pilot sequence in block pattern

$$t_i[n,k] = t_0[n,k] W_K^{-\overline{K}_0 ik}. \tag{12}$$

was proposed, where $\overline{K}_0 = \left\lfloor \frac{K}{M} \right\rfloor$.[7][8] It changes the matrix $\mathbf{Q}[n]$ to a diagonal matrix, which greatly reduces computation of matrix inversion.

But pilot sequence in block pattern uses all the sub-carriers as pilots, which is unnecessary when the number of sub-carriers is relatively large. And at the same time, it is difficult to catch the property of channel between pilot symbols in high-speed mobile environment.

4 Optimal Pilot Sequence in Comb Pattern

As described in the previous section, pilot sequence in block pattern is not good for catching up time-variant multi-path channel. But if keeping pilots in continuous symbols, and using enough pilots to make over-sampling of the channel transfer function, information of the channel transfer function will be kept, and continuous detection will be accomplished.

If there are K_0 non-zero samples in channel impulse response $h(l)$, and K sub-carriers in MIMO-OFDM system, the channel transfer function of the whole bandwidth can be expressed as

$$H_K(n) = \sum_{l=0}^{K-1} h(l) W_K^{nl} = \sum_{l=0}^{K_0-1} h(l) W_K^{nl} . \tag{13}$$

If P ($\frac{P}{M} \geq K_0$) pilots are uniformly distributed in the sub-carriers, pilot channel transfer function will be sampling of $H_K(n)$

$$H_P(p) = \sum_{l=0}^{P-1} h(l) W_P^{pl} = \sum_{l=0}^{K_0-1} h(l) W_P^{pl} , \ p = 0,1,....., P-1 . \tag{14}$$

In a system with M transmit antennas and N receive antennas, the p-th received pilot in the n-th symbol can be expressed as

$$R(n, p) = \sum_{i=0}^{M-1} H_i(n, p) S_i(n, p) + \omega(n, p) . \ p = 0,1,....., P-1 . \tag{15}$$

where $\omega(n, p)$ is white Gaussian noise in channel. Since there is no Inter-Symbol Interference (ISI) among the OFDM symbols due to the guard interval, the parameter n can be omitted, and the received pilot from estimated channel impulse response becomes

$$\tilde{R}(p) = \sum_{i=0}^{M-1} (\sum_{n=0}^{P-1} \tilde{h}_i(n) W_P^{np}) S_i(p) = \sum_{i=0}^{M-1} (\sum_{n=0}^{K_0-1} \tilde{h}_i(n) W_P^{np}) S_i(p) ,$$

$$p = 0,1....., P-1 . \tag{16}$$

where $W_P = e^{-j\frac{2\pi}{P}}$. Then we can get error of received pilot

$$\phi = \sum_{p=0}^{P-1} | R(p) - \tilde{R}(p) |^2 = \sum_{p=0}^{P-1} | R(p) - \sum_{i=0}^{M-1} (\sum_{n=0}^{K_0-1} \tilde{h}_i(n) W_P^{np}) S_i(p) |^2 . \tag{17}$$

where $R(p)$ is accurate value of received pilot. Let $\frac{\partial \phi}{\partial \tilde{h}_i(n)} = 0$, we can have

$$\sum_{p=0}^{P-1} (R(p) - \sum_{i=0}^{M-1} (\sum_{n=0}^{K_0-1} \tilde{h}_i(n) W_P^{pn}) S_i(p)) W_P^{-mp} S_i^*(p) = 0 . \tag{18}$$

where $m = 0,1,....., K_0 - 1$.

Let

$$X_i(n) = \sum_{p=0}^{P-1} R(p) S_i^*(p) W_P^{-np} , \ n = 0,1,.....K_0 - 1 , \tag{19}$$

$$\varphi_{ij}(n)=\sum_{p=0}^{P-1}S_i(p)S_j^*(p)W_P^{-np}\ ,\ n=0,1,.....,K_0-1, \tag{20}$$

Then Eqn. 18 can be rewritten as

$$\sum_{i=0}^{M-1}\ \sum_{n=0}^{K_0-1}\tilde{h}_i(n)\varphi_{ij}(m-n)=X_j(m)\ ,\quad m=0,1.....,K_0-1, \tag{21}$$

Let $\tilde{h}=\begin{pmatrix}\tilde{h}_0\\ \vdots\\ \tilde{h}_{M-1}\end{pmatrix}$, $X=\begin{pmatrix}X_0\\ \vdots\\ X_{M-1}\end{pmatrix}$, $\Psi=\begin{pmatrix}\Psi_{00} & \cdots & \Psi_{0,M-1}\\ \vdots & \ddots & \vdots\\ \Psi_{M-1,0} & \cdots & \Psi_{M-1,M-1}\end{pmatrix}$,

where

$$\tilde{h}_i=[\tilde{h}_i(0),\tilde{h}_i(1),.....\tilde{h}_i(K_0-1)]^T\ ,\quad i=0,1,.....,M-1$$

$$X_j=[X_j(0),X_j(0),.....X_j(K_0-1)]^T\ ,\quad j=0,1,.....,M-1$$

$$\Psi_{i,j}=\begin{pmatrix}\varphi_{i,j}(0) & \varphi_{i,j}(-1) & \cdots & \varphi_{i,j}(1-K_0)\\ \varphi_{i,j}(1) & \varphi_{i,j}(0) & \cdots & \varphi_{i,j}(2-K_0)\\ \vdots & \ddots & \ddots & \vdots\\ \varphi_{i,j}(K_0-1) & \varphi_{i,j}(K_0-2) & \cdots & \varphi_{i,j}(0)\end{pmatrix},$$

$$i=0,1,.....M-1,\ j=0,1,.....,M-1$$

the previous equation can be expressed in matrix form $X=\Psi\bullet\tilde{h}$, or $\tilde{h}=\Psi^{-1}\bullet X$.

From the analysis above, we can get channel impulse response of MIMO-OFDM system with comb pattern pilot, but inversion of matrix Ψ needs much computation.

Introducing some restriction among the pilot from different antennas

$$S_i(p)=S_0(p)\cdot W_P^{-ipQ}. \tag{22}$$

where $Q=\left\lfloor\frac{P}{M}\right\rfloor$ to ensure that pilot sequences from different antennas are different,

then

$$\varphi_{ij}(n)=\sum_{p=0}^{P-1}S_i(p)S_j^*(p)W_P^{-np}=\sum_{p=0}^{P-1}|\ S_0(p)\ |^2\ W_P^{((j-i)Q-n)p}\ . \tag{23}$$

When $|S_0(p)|=1$,

$$\varphi_{ij}(n) = P\delta((j-i)Q-n). \tag{24}$$

The sub-matrixes of Ψ will be analyzed below.

As to the sub-matrixes on the diagonal of Ψ, or $i=j$, only when $n=0$, $\varphi_{ij}(n)\neq 0$. As to the sub-matrixes not on the diagonal of Ψ, or $i\neq j$, when $Q\geq K_0$, $\Psi_{ij}=0$. In the summary, when $Q\geq K_0$, Ψ can be written in diagonal form $\Psi = P\cdot U_{M\cdot K_0}$, where $U_{M\cdot K_0}$ is a $(M\cdot K_0)\times(M\cdot K_0)$ unit matrix. This way, inversion of Ψ becomes inversion of a diagonal matrix, and the estimation of channel impulse response becomes $\tilde{h}=\frac{1}{P}X$.

5 Simulation Results

A QPSK MIMO-OFDM system with 2 transmit antennas and 2 receive antennas is used in the simulation with carrier frequency of 2GHz and bandwidth of 5MHz. The vehicle speed is 20m/s, resulting in the maximum Doppler frequency of 133Hz. The total number of all sub-carriers is 2048. In block pattern, 1/8 of all the OFDM symbols are used as pilot, while in comb pattern, 1/8 of all the sub-carriers in an OFDM symbol are used as pilot. The channel model used in this study is the Rayleigh channel recommended by ETSI (European Telecommunication Standards Institute) for European 3G standard. The channel parameters are shown in Table 1.

Table 1. Parameters of channel

Tap	Relative delay (ns)	Average power (dB)
1	0	0.0
2	310	–1.0
3	710	–9.0
4	1 090	–10.0
5	1 730	–15.0
6	2 510	–20.0

Space-Time Block Code (STBC) is used to transmit payload. Complex vector $\begin{pmatrix} S_n & S_{n+1} \end{pmatrix}$ after symbol mapping is transmitted as $\begin{pmatrix} S_n & S_{n+1} \end{pmatrix}$ and $\begin{pmatrix} -S_{n+1}^* & S_n^* \end{pmatrix}$ from two transmit antennas.

Fig. 2 presents error of channel estimation with different pilot patterns, and Fig. 3. gives out BER (Bit Error Ratio) of the system. It can be concluded from the figures that the performance with comb pattern pilot is much better than that with block pattern pilot.

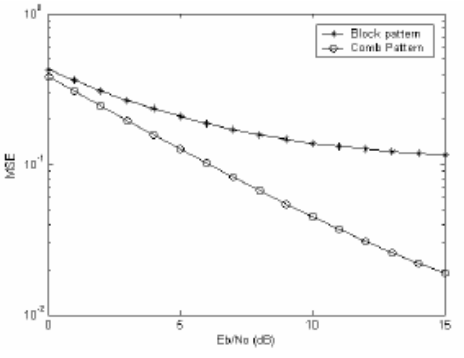


Fig. 2. Error of channel estimation

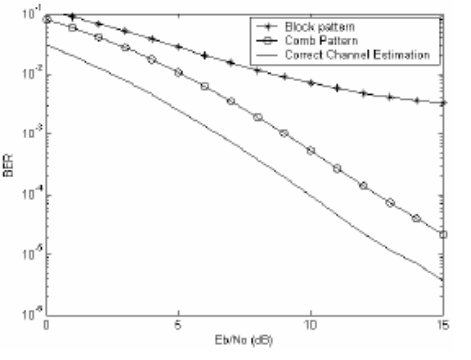


Fig. 3. BER of the system

6 Conclusion

A lot of interest has been given to MIMO-OFDM for achieving high bandwidth efficiency in multi-path fading channel, however channel estimation of it needs much computation. In this paper, comb pattern optimal pilot sequence is proposed to reduce computation of channel estimation. At the same time, it is more suitable for the high-speed mobile environment, comparing with traditional block pattern optimal pilot sequence. Error of channel estimation and BER of the whole system are greatly reduced, implying potentials of the proposed algorithm in the future MIMO-OFDM systems.

References

1. Sampath H., Talware S., Tellado J., et al. A fourth-generation MIMO-OFDM broadband wireless system: disign, performance, and field trial results. IEEE Communications Magazine, 2002. vol. 40, no. 9: 143-149.

2. Stuber G. L., Barry J.R., et al. Broadband MIMO-OFDM Wireless Communications. Proceedings of the IEEE , 2004, vol. 92, no. 2: 271–294.
3. Zelst A., Schenk T.C.W. Implementation of a MIMO OFDM-Based Wireless LAN System. IEEE Trans. Signal Processing, 2004, vol. 52, no. 2: 483 – 494.
4. Li Y.G., Winters J.H., Sollenberger N.R. MIMO-OFDM for wireless communications: signal detection with enhanced channel estimation. IEEE Trans. Commun., 2002, vol. 50, no. 9: 1471 – 1477.
5. Li Y., Seshadre N., Ariyavisitakul S. Channel estimation for OFDM systems with transmitter diversity in mobile wireless channels. IEEE Journal on Selected Areas in Communications, 1999, vol. 17, no. 3: 461-471.
6. Suthaharan S., Nallanathan A., Kannan B. A computationally efficient channel estimation with signal detection for MIMO-OFDM systems. In IEEE editions. Proceedings of IEEE PIMRC 2003, Singapore : National University of Singapore, 2003. vol. 2, 1245-1249.
7. Li Y. Optimum training sequences for OFDM systems with multiple transmit antennas. In IEEE editions Proceedings of IEEE GLOBECOM '00, Red Bank, NJ, USA: Wireless Syst. Res. Dept., AT&T Labs.-Res., 2000. vol. 3, 1478–1482.
8. Li Y. Simplified channel estimation for OFDM systems with multiple transmit antennas. IEEE Trans. Wireless Commun., 2002, vol. 1, no. 1: 67-75.
9. B. R. Saltzberg, “Performance of an efficient parallel data transmission system,” IEEE Trans. Comm., vol. 15, pp. 805-811, Dec. 1967.
10. S. B. Weinstein, and P. M. Ebert, “Data transmission by frequency division multiplexing using the discrete Fourier transform,” IEEE Trans. Comm., Vol.COM-19, pp. 628-634, Oct. 1971.
11. G. J. Foschini, “Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas,” Bell labs Tech. J., pp 41-59, Autumn 1996.

Channel-Adaptive GPS Scheduling for Heterogeneous Multimedia in CDMA Networks

Yongchan Jeong, Jitae Shin, and Hyoung-Kee Choi

School of Information and Communication Engineering,
Sungkyunkwan University, Suwon, Korea, 440-746
{todo76, jtshin, hkchoi}@ece.skku.ac.kr

Abstract. Wireless packet-scheduling is a crucial component for providing quality of service (QoS) in CDMA networks. In this paper, we propose channel-adaptive rate-scheduling based on Generalized Processor Sharing (CA-GPS) to guarantee minimum service rate and to provide proportional fairness among heterogeneous multimedia traffic for QoS differentiation. The CA-GPS scheduler assigns different GPS weights according to traffic priorities, to provide differentiated services under time-varying channel conditions. Soft-uplink capacity analysis is performed and used to improve the utilization of CDMA system resources. The performance analysis and evaluation of proposed CA-GPS is achieved via simulation in terms of achievable throughput, delay, and proportional fairness.

1 Introduction

There are many needs to support heterogeneous multimedia traffic having different quality of service (QoS) requirements in wideband CDMA networks. This paper tackles an efficient scheduling to provide maximum uplink system throughput, minimum average delay and proportional fairness in accordance with traffic priority through dynamic resource allocation under the different traffic QoS requirements and time-varying channel conditions.

An ideal fairness scheduler is Generalized Processor Sharing (GPS) [1] which assigns each traffic a different fixed weight and dynamically allocates bandwidth to all backlogged traffic according to their weights and traffic load. Several GPS-based schedulers have been proposed for wired/wireless networks [2~7]. However these schedulers are implemented using a time-scheduling approach, which represents high complexity due to the extensive computation for each packet's virtual time. The time-scheduling approach is suitable for time-division multiple access (TDMA) or hybrid time division duplex (TDD)/CDMA. The CDMA system is interference-limited and the system capacity depends on the sum of the allocated rate in each block of data. The optimum scheduling scheme is required to incorporate relationships with these traffic rates and received powers.

Previous work relating to GPS-based uplink- scheduling for CDMA environments are mentioned in Refs. [4][8][10]. However the available system capacity in this work is treated as static, with patterns that do not vary over time, resulting in poor total

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 93 – 101, 2005.

system throughput and which cannot adapt efficiently to channel conditions. Also the scheduling schemes do not consider QoS requirements and requested QoS differentiation at the same time. Efficient scheduling in a CDMA system should consider entire CDMA system resources representing as **soft capacity** [8]. Uplink system capacity is especially subject to the variation of signal-to-interference ratio (SIR) and requested rates of users in the cell.

In this work, we will tackle channel-adaptive wireless packet scheduling method, considering the time-varying system capacity in order to maximize the system throughput while providing proportional fairness using different GPS weightings, and also providing QoS differentiation in accordance with traffic priorities. This paper proposes QoS-aware traffic and channel-adaptive scheduling based on GPS to estimate up-link capacity and to provide weighted service rates upon traffic priority. Our scheduling function has the features of (1) higher system throughput by analyzing uplink capacity; (2) QoS differentiation among traffic; and (3) proportional fairness via different GPS weights; and (4) guaranteeing the minimum service rate in time-varying CDMA systems.

This paper is organized as follows: The system model we consider in CDMA cellular networks is briefly described in Section 2. The formulation of a CDMA system capacity analysis is described in Section 3. In Section 4, we propose a channel-adaptive GPS (CA-GPS) scheduling algorithm to achieve objectives of high system throughput, delay, and fairness under time-varying system capacity. Simulation results are shown in Section 5 to demonstrate the performance of CA-GPS schemes, followed by our conclusion in Section 6.

2 System Model

Direct Sequence (DS)-CDMA systems are considered in this work. This paper focuses on an uplink scheduler that resides at each base station (BS). The physical data channels in the uplink are distinguished by pseudo-noise (PN) codes. In this paper we assume the uplink capacity is interference-limited and is not limited by the number of available PN codes due to multiple-access interferences (MAI).

The power control of the user with a low speed is nearly perfect for maintaining the target bit error rate (BER). On the other hand, the power control is difficult for fast-moving users due to fast channel fading. In this paper we assume that all users in the cell move slowly for perfect power control without loss of generality.

The transmitting channel rate of each mobile station (MS) is scheduled on a time-slot basis. The required BER is different according to voice, video, and data traffic. The minimum SIR to meet the minimum required BER, however, is targeted to satisfy CDMA systems. The leaky-bucket regulator is required to shape each traffic source in order to achieve a bounded delay for a user. Fig. 1 is the system queueing model for our proposed CA-GPS scheduling. All active users share time-varying uplink capacity and the scheduler can allocate each user's rate differently. The available system capacity may be different from different time slot.

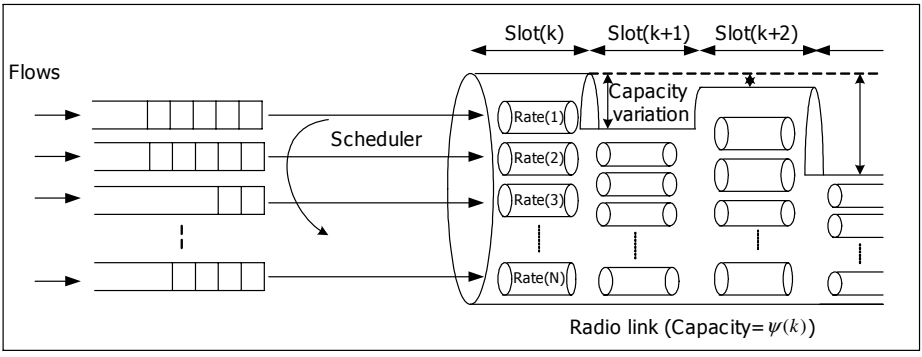


Fig. 1. System model of CA-GPS

3 The Estimation of CDMA System Capacity

Uplink capacity of a CDMA system is dynamically changing according to time and users' position. In this section, we will estimate the uplink capacity in order to maximize the total throughput. Let us consider the uplink of a CDMA system containing B(t) backlogged mobile users at time t. Without loss of generality we assume that each user has only one flow active at a given time. The transmission power of a mobile user i is denoted by p_i. Let G_i be the spreading gain of user i and γ_i be its minimal SIR required to satisfy its QoS requirements. The corresponding QoS constraints are given by: [10]

$$\frac{G_i h_i(t) P_i}{\sum_{j=1, j\neq i}^{B(t)} h_j(t) P_j + W\eta_0} \geq \gamma_i \quad i = 1,2,...,B(t) \tag{1}$$

where W : Bandwidth, $h_i(t)$: Channel gain at time t, η_0 : One-sided power spectral density of additive white Gaussian noise.

The optimal power solution can easily be derived when all QoS constraints are met with equality.

$$\frac{G_i h_i(t) P_i^*}{\sum_{j=1, j\neq i}^{B(t)} h_j(t) P_i^* + W\eta_0} = \gamma_i \quad i = 1,2,...,B(t) \tag{2}$$

Eq. (2) can be modified to:

$$h_i(t) P_i^* = \frac{\gamma_i}{G_i + \gamma_i} \left[\sum_{j=1}^{B(t)} h_j(t) P_j^* + W\eta_0 \right] \tag{3}$$

If we write Eq. (3) for all i's and add those equations, it can be shown that:

$$\sum_{i=1}^{B(t)} h_i(t)P_i^* = \frac{\sum_{i=1}^{B(t)} \frac{\gamma_i}{G_i+\gamma_i}}{\left(1-\sum_{i=1}^{B(t)} \frac{\gamma_i}{G_i+\gamma_i}\right)} W\eta_0 \tag{4}$$

By employing Eq. (4) in Eq. (3), the optimum power solution is determined:

$$P_i^* = \frac{W\eta_0}{h_i(t)\left(1-\sum_{j=1}^{B(t)} \frac{\gamma_j}{G_j+\gamma_j}\right)} \frac{\gamma_i}{G_i+\gamma_i} \tag{5}$$

Since the power value should be positive and limited, the following necessary condition, $\sum_{i=1}^{B(t)} \frac{\gamma_i}{G_i+\gamma_i} < 1$, must be satisfied.

However, when the left-hand side of the necessary condition is close to 1, the optimal power levels may be too high to be sustainable. Moreover, the increase in the total power of users in one cell may adversely affect the surrounding cells and stimulate an increase in intercell interference. Therefore, it is necessary to impose the inequality:

$$\sum_{i=1}^{B(t)} \frac{\gamma_i}{G_i+\gamma_i} < 1-\delta, \quad \delta: \text{Small positive number} \tag{6}$$

From now on, we're going to show the available uplink capacity based on above statements. The received power level at the base station is restricted. Here, the power constraints are defined as:

$$0 \le P_i \le P_i^{\max} \tag{7}$$

where $P_i^{\max}$ is the maximum transmission power limit of user i. Using Eqs. (5) and (7) implies that [10][11]:

$$\sum_{i=1}^{B(t)} g_i < 1-\frac{W\eta_0}{\min_i\left(P_i^{\max} h_i(t)/g_i\right)} \tag{8}$$

where $g_i \left(\triangleq \frac{\gamma_i}{G_i+\gamma_i}\right)$ is power index of user i.

Finally, we can obtain the available maximum uplink capacity at time t, $\psi(t)$, by comparing Eqs. (6) and (8).

$$\psi(t) = 1 - \frac{W\eta_0}{\min_i \left(P_i^{\max} h_i(t) / g_i \right)} \tag{9}$$

Eq. (9) indicates that $\psi(t)$ is mainly affected by channel gain and power index of user *i*.

4 Proposed Channel-Adaptive GPS (CA-GPS) Scheduling

In this section, the CA-GPS scheme is introduced, allocating resources to all users, while considering the soft capacity. Let $r_i(k)$ be the allocated service rate of user *i*, $B(k)$ and $R(k)$ be the set of active users and the total amount of allocated service rates (i.e., $\sum_{i \in B(k)} r_i(k)$) for time slot *k*, respectively. Also, let $\psi(k)$ be the available maximum uplink capacity in time slot k. ϕ_i denotes the weight of user *i*. The set of user *i* is not allocated by any service rate that is represented for compensation as $\alpha(k)$. The CA-GPS scheduler allocates each allocated rate to user *i*, $r_i(k)$, using the following steps as in Fig.2:

In this algorithm, we calculate $R(k)$ using $\psi(k)$ for estimating the available up-link capacity. Users have fixed-service rates in previous work. However we consider a minimum service rate in order to serve more users in the same period. This causes the main difference in terms of total system throughput and efficiency. After that we allocate the extra resources to other active users considering various traffic priorities. This is for providing proportional fairness to all active users.

The case of branch 1 is for bad channel. If the amount of all active users' minimum service rates is more than $R(k)$, we allocate resources to users randomly until $C(k)$ is less than $\min_i(r_i^{\min})$ as shown in the middle area in Fig.2. Since traffic has higher priority and a higher minimum service rate, we have to check the minimum service rates of all users in order to maximize utilization in the remaining resources. The reminder of resource has to be compared with priority order and allocated to the other user having a minimum service rate less than the reminder. If a minimum service rate is not considered as other works, we can't guarantee quality of service when channel state is bad. We give more priority to the total system throughput rather than achieving proportional fairness. The case of branch 2 is for good channel. (we show it in the left side of Fig.2).

Another feature in this algorithm is compensation, i.e., we serve users who were not served on time due to the channel state condition. This also improves total system throughput. Consequently, the proposed algorithm provides proportional fairness and higher throughput compared to other GPS-based schemes.

The above resource allocation procedure is aimed at finding $r_i(k)$ of user *i,* while maximizing the total throughput and satisfying the GPS fairness constraint. This

algorithm also provides the minimum service rate with higher priority and increases the throughput of each user due to the compensation in subsequent user allocation for the users who are in the set $\alpha(k)$ and received insufficient services in previous time interval.

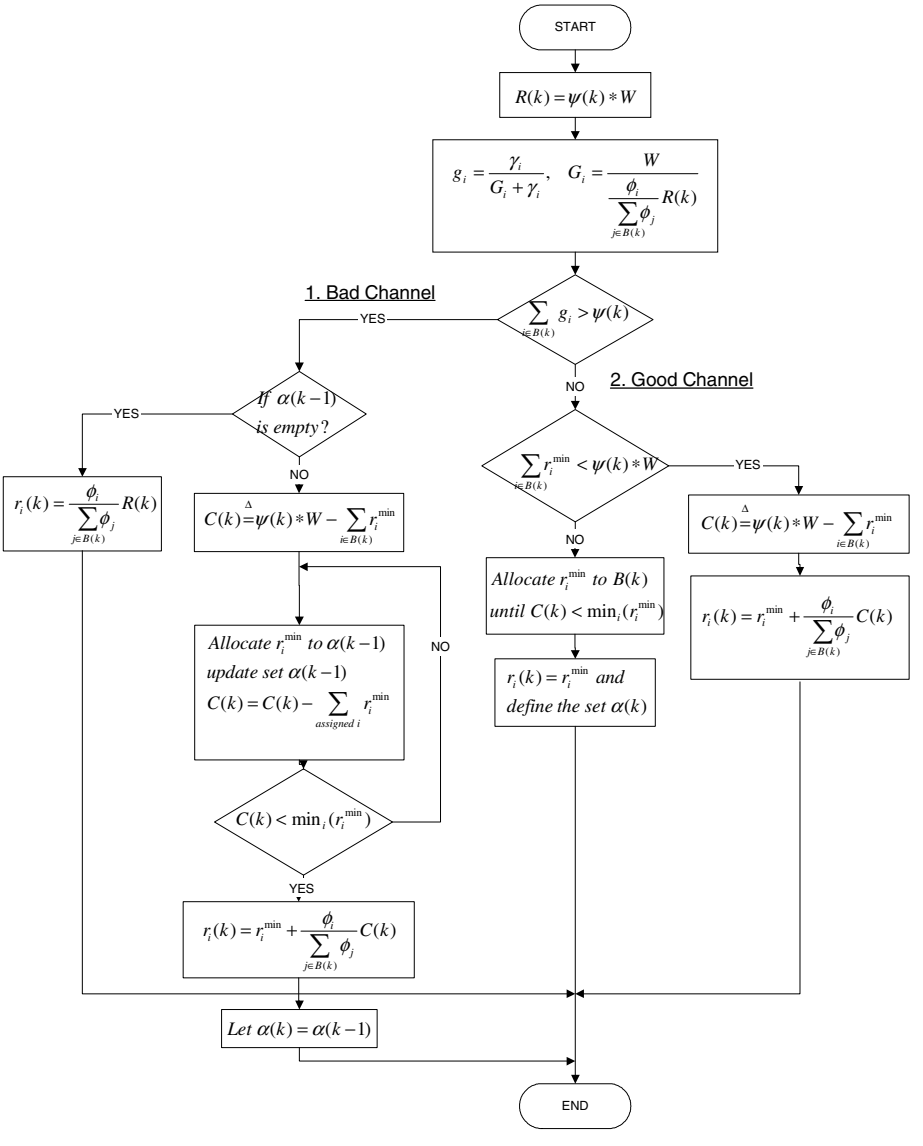


Fig. 2. CA-GPS Algorithm

5 Simulation Results

In this section, simulation results are presented to demonstrate the performance of the proposed CA-GPS scheme in terms of delay, system throughput and proportional fairness. The scheduling period T is 10ms. In simulation, the CA-GPS scheme is compared with the CDGPS [8] under heterogeneous traffic environments.

The total bandwidth is assumed to be a constant W = 5Mb/s. The total available uplink capacity is estimated by solving Eq. (9). Ten flows are considered, and assigned as different weights. All flows are modeled by a Poisson process with average arrival rate λ and packet length L, shaped by a leaky-bucket regulator for providing the bounded delay. The corresponding values of all the parameters used throughout our study are shown in Table 1 [10][12].

Table 1. Simulation Parameter Values

Parameter	Value
Packet size	5kbits
AWGN spectral density (η_0)	10^{-6}
Minimum channel gain $(h_i(t))$	0.25
Maximum transmission power $(P_i^{\max})$	0.5W
Weight(user 0,1,2,3,4,5,6,7,8,9)	(1,1,1,1,1,2,2,2,4,4)
Required SIR (E_b/I_e)	5dB
Scheduling cycle (T)	10ms
Minimum required service rate $(r_i^{\min})$	192,320,640kbps

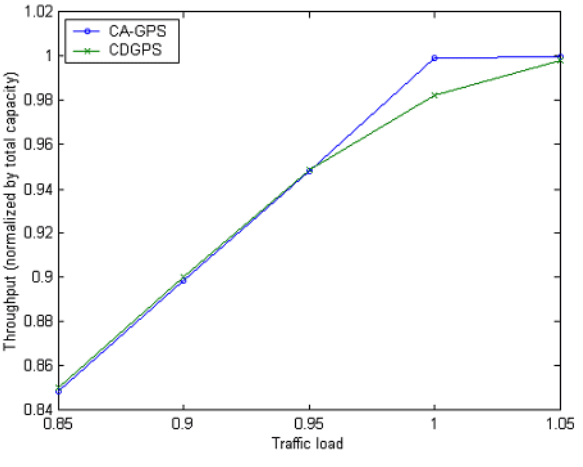


Fig. 3. Throughput comparison: CA-GPS and CDGPS

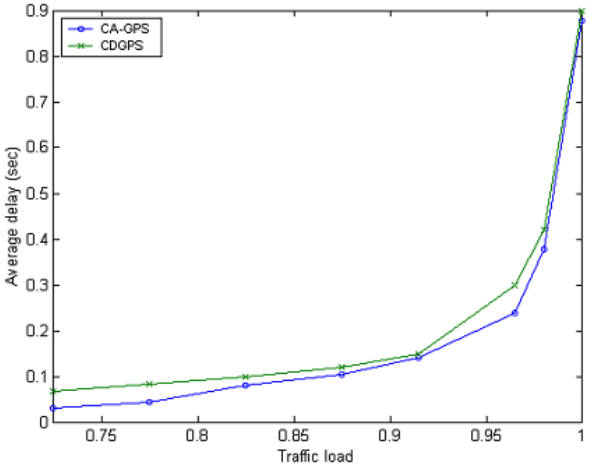


Fig. 4. Average delay

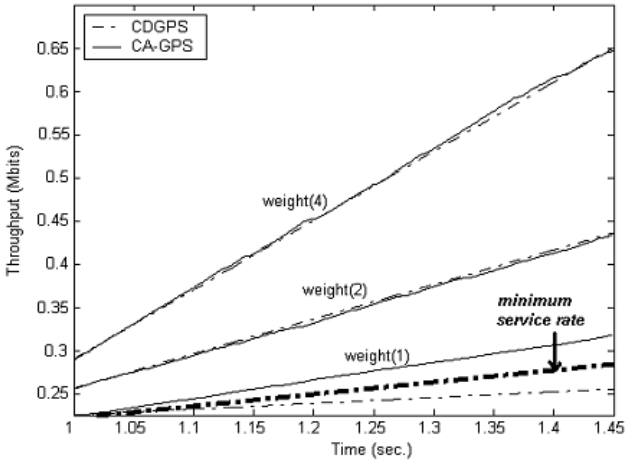


Fig. 5. Proportion fairness

Fig. 3 shows the throughput comparison of being used CA-GPS and CDGPS. The traffic load is the sum of average arrival rates of the ten data flows. The proposed scheme throughput is higher than CDGPS because CA-GPS uses the concept of minimum service rate. It is shown that CA-GPS can improve the uplink throughput.

Fig. 4 shows the average delay with different system loads. In this figure, it can be seen that the average delay performance of CA-GPS with a soft capacity is better than CDGPS with a fixed capacity. The minimum required service rate is the first consideration in CA-GPS. Therefore, more users can be served.

Fig. 5 shows the throughput with different weights. Flow weights have proportions of 1:2:4 and the throughput of flows are close to the proportion in CA-GPS. On the

other hand, CDGPS does not consider the minimum service rate. Consequently, the flow has a lower weight that sometimes cannot be served when the channel condition is bad.

6 Conclusion

In this work, an efficient scheduler is proposed to satisfy the QoS requirements of multimedia traffic in a CDMA uplink system. The time-varying capacity is estimated with the consideration of a user's QoS requirements, channel fading effect, and transmitting power. The proposed channel-adaptive scheduling based on GPS (CA-GPS), adapts the time-varying channel capacity and minimum required service rate in order to improve system utilization, average delay and proportional fairness. The performance of proposed scheduling is compared with CDGPS [8] as "fixed capacity" shown in Fig. 3~5. The proposed scheduling method is closer to fluid-modeled GPS than other GPS-based scheduling for CDMA systems.

References

1. A.K.Parekh and R.G.Gallager,"A generalized processor sharing approach to flow control in integrated services networks: The single-node case,"IEEE/ACM Trans. Networking,vol.1, pp.344-357, June 1993.
2. P. Ramanathan and P. Agrwal, "Adapting packet fair queueing algorithms to wireless networks," in ACM/IEEE MOBICOM'98, Dallas, TX, pp. 1-9.
3. S.Lu and V. Bharghavan, "Fair scheduling in wireless packet networks," IEEE/ACM Trans. Networking, vol. 7, no. 4, pp.473-489, 1999.
4. M. A. Arad and A. Leon-Garcia, "A Generalized Processor Sharing Approach to Time Scheduling in Hybrid CDMA/TDMA," IEEE INFOCOM, Mar. 1998, pp.1164-7.
5. A.Demers and S.Shenker,"Analysis and simulation of a fair queuing algorithm," Internetworking: Res. Exper.,vol.1,no.1,pp. 3-26,1990.
6. D.Stiliadis and A.Varma,"Efficient fair queueing algorithms for packet-switched networks,"IEEE/ACM Trans. Networking,vol.6,pp. 175-185,Apr.1998.
7. Nandagopal,S.Lu,and V.Bharghavan,"A unified architecture for the design and evaluation of wireless fair scheduling algorithms," Wireless Networks,vol.7,pp.231-247,Aug.2002.
8. Liang Xu, X.Shen, Mark, J.W,"Dynamic Fair Scheduling With QoS Constraints in Multimedia Wideband CDMA Cellular Networks"Wireless Communications, IEEE Trans, Vol.3 , Issue.1, Jan.2004 pp.60-73
9. S.Ariyavisitakul,"Signal and interference statistics of a CDMA system with feedback power control-Part Ⅱ,"IEEE Trans.Commun., vol.42,pp.597-605,Feb./Mar./Apr.1994.
10. Chengzhou Li, Papavassiliou, S., "Dynamic fair bandwidth allocation in multiservice CDMA networks" 23rd International Conference on Distributed Computing Systems Workshops, 2003. Proceedings. , 19-22 May 2003.
11. M.A.Arad, A.Leon-Garcia, "Scheduled CDMA: a hybrid multiple access for wireless ATM networks," in Proc. of PIMRC '96, Taipei, Taiwan, 1996.
12. Ashwin Sampath, P.Sarath Kumar, Jack M.Holtzman, "Power control and resource management for a multimedia cdma wireless system", PIMRC'95, vol.1, pp.21-25,1995.
13. J.Zander,"Performance of optimum transmitter power control in cellular radio systems," IEEE Trans. Veh.Technol.,vol.41,pp57-62, Feb.1992

An Adaptive Scheduled Transmission Strategy for Multimedia Services in WCDMA Systems

Eric Hsiao-Kuang Wu[2], Chiang Jui-Hao[1], and Hsin-Pu Chung[2]

[1] Department of Computer Science and Information Engineering, National Taiwan University, No. 1, Sec. 4, Roosevelt Rd., Taipei, Taiwan
littlejohn@inrg.csie.ntu.edu.tw
[2] Department of Computer Science and Information Engineering, National Central University, Chung-Li, Taiwan
hsiao@csie.ncu.edu.tw

Abstract. Since the capacity in WCDMA system is limited by interference, an efficient radio resource mechanism is important to enhance overall performance. Previous researches have shown that uni-access mode is an efficient way to reduce intracell interference caused by other transmitting users, and hence higher performance is attained compared to traditional multi-access mode. However, due to the constraint in the practical system, a single user might not fully utilize the available bandwidth. Consequently, the system capacity as well as throughput will be not optimized. This paper introduces a novel Transport block size based Adaptive Scheduling Scheme (TASS) that utilizes radio resource more efficiently and sufficiently. Different from both uni-access and multi-access mode, the TASS can be seen as a mixed version of them called hybrid-access mode. Via the transmission strategy, less power is required while achieving equivalent throughput and thus higher performance is expected. In order to guarantee the delay time of each user, the scheduling algorithm earliest deadline first (EDF) is applied. As a result, the TASS is suitable for real-time traffics. The experiment results demonstrate that this new transmission strategy carries advantages in system capacity, average delay and overall throughput.

1 Introduction

Wideband code-division multiple-access (WCDMA) is selected as the radio interface technology in UMTS and is designed for packet-based multimedia services in 3G wireless communication network [1]. In WCDMA systems, the radio resource can be allocated to the users by regulating their transmission powers and spreading gains so as to satisfy the diverse QoS requirements of the users and maximize utilization of the available bandwidth. The data transmission rate can be controlled by employing a variable spreading factor (VSF) method [2]. The method for dynamic rate variation is attractive due to its simplicity in implementation and potential for low power consumption at the mobile handsets.

Since mobile users in WCDMA cellular system transmit in the same frequency at the same time, spectrum sharing introduces interference, which degrades the ability of reliable communications. By this, the capacity in WCDMA system is limited by

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 102 – 112, 2005.

interference. The total received interference consists of intra-cell interference, resulting from simultaneous transmissions within the same cell, and inter-cell interference, caused by transmitters in other cells. To mitigate the impact of intercell and intracell interference, transmitting power thus is an important controllable resource for balancing the desired signal and interference powers at the receiver. In addition to power control, it is feasible to manage interference by means of scheduling transmissions. Under this scheduled transmission strategy, some traffics are deferred so that only few users are transmitting within each cell. Consequently, the intracell interference is reduced and the throughput of remaining transmitting users is increased.

Under this concept, two radio-access options are classified as uni-access and multi-access [3]. In multi-access mode, all mobiles access the channel simultaneously as they need and this is the way current CDMA systems operate. In uni-access mode, users are operated in the consecutive time slots one after the other and only one mobile is allowed to access the channel at any time instant, which is like TDMA over CDMA. The results show that the uni-access mode can reduce interference thus the available bandwidth is utilized more efficiently and higher data rate is expected.

This paper is organized as follows. In Section 2, radio resource management and transmission strategies in previous studying are described. Our proposed transmission scheme, a novel transport block size based adaptive scheduling Schemes (TASS) is presented in Section 3. In Section 4, we show the simulation models as well as scenarios and the simulation results compared with transitional WCDMA, TDMA over CDMA are demonstrated. Finally, conclusions and the future works are given in Section 5.

2 Related Works

[4] and [5] address the problem of maximizing throughput in cellular CDMA networks by jointly controlling the data rates and transmit powers of users, subject to some constraints. The constraints are specified in terms of minimum required data rate, the maximum power used by a mobile host, and the minimum achieved SIR value. The optimization criterion is the maximization of the sum of the transmission rates. No scheduling policy is considered.

A different QoS constraint for non-real-time traffics based on an average data rate rather than an instantaneous data rate is proposed in [6]. It requires an amount of data of user should be delivered during a period of time. Therefore, as a higher transmission rate is applied, a shorter transmission time will be obtained. By the concept, the transmissions can be scheduled. It shows that with the scheduling policy only one user in the cell is allowed to transmit at a time, the same amount of data can be delivered with less energy.

In [7], two classes of users consist in the system: real-time and non-real-time users. The optimization criterion is maximization of throughput subject to target SIR value. It is found that higher throughput can be obtained by scheduling transmitting users properly. That is, only one user is allowed to transmit at a time optimizes throughput. This gain does not necessarily require more average transmission power. Because of the delay caused by waiting for service, only non-real-time users are scheduled, and the real-time terminals are always allowed to transmit when they wish. However,

there is no scheduling strategy considered. For example, schedule a user locates near the base station to transmit performs better than schedule a user locates far away the base station.

A hybrid multiple access (HMA) transmission strategy is proposed in [8]. Since uni-access and multi-access transmission modes carry advantages in different respects, the HMA coordinate to access these two modes to enhance system performance. Real-time traffics are chosen to operate in multi-access mode because of the delay problem of uni-access mode. In the contrast, the delay-tolerant traffics operate in uni-access mode for the sake that the BER requirements can be easily achieved. Otherwise, scheduling algorithms are applied including the EDF (early deadline first) for real-time traffics and the static priority scheduling (SPS) for non-real-time traffics to fulfill different demands of these two traffic types.

In [9], the authors address the problem of dynamic resource allocation in a CDMA network that supports real-time and non-real-time services. A jointly transmission power and spreading gain allocation strategy is provided for non-real-time users that manages the multiple access interference efficiently so as to maximize throughput, subject to power constraint imposed by real-time users. The target SIR is not specified by non-real-time users, and the reliability is assured via retransmissions. The resulting resource allocation strategy is implemented as a hybrid CDMA/TDMA strategy. Although the proposed allocation policy maximizes the spectrum efficiency of the system, however, an unfair allocation of the resource among users may occur.

Uni-access mode has been considered as a better transmission scheme since no energy is wasted on resisting interference caused by other transmitting users. However, it is hard to always fully utilize the radio resource by a single user due to the constraint in the practical system that data rate is impossible to increase without limitation. For example, for a user transmitting alone, the only interference is background noise. Suppose the maximal usable power level of this user is 3mW. When it is transmitting with maximal possible data rate (that is, with minimal spreading factor), if only 2mW is required for the user to attain its acceptable transmission quality, the remaining 1mW is wasted. In this case, if we allow another user to transmit simultaneously, higher overall throughput is expected. By this, we obtain a novel transmission scheme which is a modification of uni-access but not equivalent to multi-access mode. It can be seen as a mixed version of them and we call this hybrid-access mode, as depicted in Figure 1(c). Compare it to uni-access and multi-access modes.

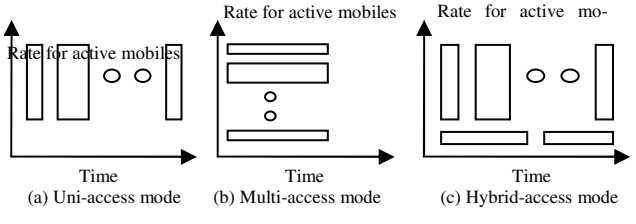


Fig. 1. Different transmission modes

In this paper, we propose a novel Transport block size based Adaptive Scheduling Scheme (TASS) scheme in hybrid-access mode, which utilizes the available bandwidth more efficiently and sufficiently than multi-access mode and uni-access mode, respectively. It can be seen as a dynamic adjustment between these two transmission modes to obtain best performance adapting to different conditions.

3 TASS (Transport Block Sized Based Adaptive Scheduling Scheme)

A. Optimal Power Allocation

A centralized optimal power allocation strategy for multiple rates with diverse QoS requirements has been widely discussed [10] [11]. In this paper, we focus on the uplink WCDMA channel. Each user, i, specifies a target SIR value, denoted as γ_i, which determines the experienced bit error rate (BER) of user i at the receiver. Since uplink transmission power is provided by mobile host's battery, every user also specifies a maximal transmission power limitation P_i^{max}. The path gain of the user is noted as h_i.

The goal of power allocation can be addressed as follows: assign P_i to user i as its transmission power, i = 1, 2 … k, where k is the number of users who are scheduled transmitting, such that the QoS requirements (also expressed as target SIR) and power constraints of different users are satisfied. The problem can be formulated as the following well known inequality:

$$\frac{F_i \cdot P_i \cdot h_i}{\sum_{j \neq i} P_i \cdot h_i + I_{\text{int}\,er} + \eta} \geq \gamma_i \quad \text{Subject to } P_i \leq P_i^{\max}, \text{ and i, j = 1, 2 … k} \tag{1}$$

Here F_i is the spreading factor defined as $F_i = \frac{W}{R_i}$, where R_i is the data rate assigned to user i and W is the total system bandwidth. The η presents the background noise, and the intercell interference is denoted as Iinter. According to (1), the optimal solution can be obtained as:

$$P_i = \frac{g_i \cdot \left(I_{\text{int}\,er} + \eta\right)}{h_i \cdot \left(1 - \Sigma_{j=1}^{k} g_j\right)} \tag{2}$$

$$g_i = \frac{\gamma_i}{\gamma_i + F_i} \tag{3}$$

The term g_i is called as the power index of user i. Equation (2) is in the sense that the SIR requirements of all scheduled users are met with equality. The solution is feasible as long as the conservative constraint is not violated. That is, the aggregate power index has to be always kept below a threshold:

$$\sum_{j=1}^{k} g_j < 1 - \max_i \left(\frac{g_i \cdot \left(I_{\text{int}\,er} + \eta\right)}{P_i^{\max} \cdot h_i} \right) = 1 - \Delta \tag{4}$$

Δ is the reserved power index used to limit the assigned power levels. Equation (4) can be seen as the power capacity in each frame.

B. TASS Transmission Strategy

1) Hybrid-access Transmission Mode

The total power of transmissions is derived from (2):

$$\sum_{i=1}^{k} P_i = \sum_{i=1}^{K} \frac{g_i \cdot (I_{\text{int}er} + \eta)}{h_i \cdot (1 - \sum_{j=1}^{k} g_i)} = \frac{(I_{\text{int}er} + \eta)}{(1 - \sum_{j=1}^{k} g_i)} \cdot \sum_{i=1}^{k} \frac{g_i}{h_i}$$

Regardless of channel condition of each user, it shows that the total transmission power depends on the sum of power index g. From (3), we know that each g is influenced by transmission rate when target SIR is the same. If we can use least g to achieve equivalent overall throughput, the power is saved.

Lemma 1: while achieving equivalent overall throughput over a span of time, the way of allocating resource to fewer users performs better than the way of allocating resource to more users in the aspect of power consumption.

Proof: assume user *A* is transmitting with data rate *X* kb/s. Equivalent throughput can be achieved by sum of *K* other users. Suppose all users specify the same target SIR, and each of *K* users is allocated with data rate Y_i, $i = 1, 2 \ldots K$, $\Sigma Y_i = X$. The sum of power index of *K* users is:

$$g_k = \frac{\gamma}{\gamma + \frac{W}{Y_1}} + \frac{\gamma}{\gamma + \frac{W}{Y_2}} + \ldots \frac{\gamma}{\gamma + \frac{W}{Y_K}} = \frac{Y_1 \cdot \gamma}{Y_1 \cdot \gamma + W} + \frac{Y_1 \cdot \gamma}{Y_2 \cdot \gamma + W} + \ldots + \frac{Y_1 \cdot \gamma}{Y_K \cdot \gamma + W}$$

And the power index of user A is:

$$g_k = \frac{\gamma}{\gamma + \frac{W}{X}} = \frac{X \cdot \gamma}{X \cdot \gamma + W} = \frac{Y_1 \cdot \gamma}{X \cdot \gamma + W} + \frac{Y_2 \cdot \gamma}{X \cdot \gamma + W} + \ldots + \frac{Y_K \cdot \gamma}{X \cdot \gamma + W}$$

Since $X \geq Y_i$, $i = 1, 2 \ldots K$, it appears that $g_A \leq g_K$. Consequently, Lemma 1 is proved.

According to Lemma 1, we know that achieving equivalent throughput by fewer users consumes less power than more users. Since WCDMA is an interference-limited system, lower power is expected to improve performance. By this, we first schedule users to transmit horizontally as uni-access mode rather than vertically as multi-access mode. In order to accommodate most users in horizontal, the data rate assigned to each user is as high as possible such that the amount of time they occupy will be least. If the horizontal capacity is full, we then allocate users vertically.

The TASS allocation scheme is based on this. The transport block is the basic data unit transmitted between the physical layer and the MAC layer. We assume that a transport block is the basic data unit transmitted between the physical layer and the MAC layer. We assume that a transport block can be completely transmitted just in a frame time with the required data rate generated by the user. Suppose that the frame time, denoted by *T*, is fixed and the transmission rate can be selected from the set of rates $\{\Phi_0, \Phi_1 \ldots \Phi_\Pi\}$, where $\Phi_n = n\Phi_1$ ($n = 0, 1 \ldots \Pi$), and Φ_1 is defined as basic rate. For any user *i*, let R^*_i be the minimum required data rate that user *i* specifies, and R_i is the transmission rate assigned to user *i*. If R_i is equal to R^*_i, one frame time, that is *T*, is required to transmit an entire transport block of user *i*. For $R_i = 2R^*_i$, only *T*/2 is needed. If two users, *i*, *j*, are assigned with transmission rate that $R_i = 2R^*_i$, $R_j = 2R^*_j$,

then *T*/2 is needed for both of them. As a result, they can be scheduled to transmit one by one instead of transmitting simultaneously as in conventional CDMA systems do while the delay requirement of them are still satisfied. The time fraction of a frame occupied by user *i* is denoted by τ_i, $0 \le \tau_i \le T$, $i = 0, 1, 2 \ldots k$, and can be expressed as

$$\tau_i = \frac{T \cdot R_i^*}{R_i} \tag{5}$$

For this, if total time fraction for users is less than a frame time T, we can easily schedule users to transmit one by one. Otherwise, some users will be scheduled to transmit simultaneously. That is, if the total time fraction of some "layer" will exceed T when we assign some user to this layer, this user will be assigned to next layer instead of this one. According to such concept, the allocation problem becomes what transmission rate should be assigned to users and during what period of a frame they are allowed to transmit.

Since different layers cause interference to each other, it is expected that the layers should be as less as possible to enhance performance. The allocation problem can be re-addressed as: assign data rate and transmission period to users to accommodate most users in a frame time to transmit with least layers. An exhaustive search for optimal solution is not efficient because of the complexity. We will present a Best Fit Decreasing (BFD) approximation algorithm to solve the problem.

2) BFD Allocation Algorithm

In order to shorten the time fraction of each user, every user uses ΦΠ as its initial transmission rate of the allocation procedure and the time fraction size of each will then be counted by (5) according to the data rate and size of the transport block. The term decreasing means that users are allocated in decreasing order of their time fraction size. When allocating a user, we first find out the layer which after accommodating the user will have the least amount of time left. This is what the term best fit means.

After assigning a user to some layer, the conservative constraint (4) should be checked between users in different layers. If (4) is violated, the user should not be assigned to this layer, and the next fit layer will be tried, and so on until some layer can accommodate the user or all layers are tried and failed. If no layer can accommodate the user while (4) is still satisfied, select the user, suppose it is i, with largest time fraction τi so far and reduce its current transmission rate $R_i = \Phi_n$ to Φ_{n-1}. Note that Ri should be larger than or equal to the minimum required rate R_i^* or this transport block cannot be completely transmitted during T. After that, the BFD allocation algorithm procedure will do again until all transport blocks are allocated, or no user can further decrease its transmission rate, which presents that the system is overload.

The TASS can also be used as admission control. For each transport block, the BFD allocation algorithm is applied with those already admitted ones. The block will be admitted if it can be successfully allocated. Otherwise, it will be blocked.

3) Scheduling

Since delay is the key QoS requirement for real-time traffics, we adopt an earliest deadline first (EDF) algorithm to guarantee that the traffic with most urgent deadline will be serviced first. The transport block with earliest deadline is scheduled to transmit first and the block with next earliest deadline will also be scheduled during the

same time frame as long as it can be successfully allocated in BFD. For those blocks with the same deadline, they are admitted in decreasing order of conservative constraints (4) for the sake of that the larger one can accommodate more users to transmit simultaneously.

C. Implementation Issue

The TASS is centralized and the base station is the logical entity to perform the scheduling procedure. The length of allocation iteration is set to be one frame. The assigning information to each user includes transmit power, transmission rate and transmitting period during the frame. Compare to conventional resource management which contains transmit power and transmission rate, we believe that the signal overhead will not increase too much. In order to follow the time structure specified in 3GPP, slot is the basic time unit of transmission. For example, if a user needs 1/4 frame to transmit a transport block according to the assigned rate, 4 slots (since there are 15 slots per frame) will be given to the user. By this, the transmission format is not necessary to be modified.

Since transmissions in the proposed TASS are separated more detained, synchronization is an important issue or the QoS requirements of each user may be violated. In [12], the USTS addresses the synchronization problem in uplink. This needs additional signal to carry the amount of timing adjustment, it is expected that only a small amount of signaling load will increase at call setup phase and for handover.

4 Simulation Results

A. Single Cell Evaluation

In the following experiment, we evaluate that how TASS scheme enhances performance. Real-time and non-real-time traffics coexist in the system and are uniformly distributed in the serving cell. Each real-time user specifies a minimal required transmission rate 60kbps, while no constraints for non real-time users. The system chip rate is 3.84M chips per bit, and the candidate spreading factor value is 4 to 256 as specified in UMTS. The proposed TASS transmission scheme is compared with conventional CDMA and TDMA-like one by one scheduled transmission strategy.

Since we argue that the TASS scheme is power-saved, we compare the capacities of each strategy first. High capacity means that more users can be tolerant to transmit during a frame time while the quality of each including delay requirement is guaranteed. Thus, higher performance is expected. The effect of different background noise level to capacity for each strategy is depicted in figure 2. We can see that when background noise level is low, TDMA-like scheduling policy performs better than conventional CDMA strategy. As background noise increases, the capacity of TDMA-like scheduling policy reduces quickly while the conventional CDMA strategy only decreases a little and becomes better than the TDMA-like policy. This is because when background noise is relatively low, the interference experienced by users is mainly from other users. Thus defer some traffics will efficiently reduces interference to users, and the TDMA-like scheduling policy performs well. On the contrary, when the background noise is relatively high, the effect of other user becomes out of consideration. As a result, simultaneous transmission may utilize resources more adequately.

However, from figure 2, we find that the maximal capacity of TDMA-like scheduling policy is 15 and hard to get larger. This is by the reason of the spreading factor specified in UMTS is limited to 4. So under the constant system chip rate, a user cannot get higher transmission rate, and the radio resource is then not fully utilized. The proposed TASS overcomes this problem, thus a higher capacity is achieved, and it always performs best no matter under what situation.

Figure 3 shows the comparison of average power consumption of conventional CDMA and TASS scheme in different number of real-time users. The result shows that as user number increases, the consumed power of conventional CDMA increases sharply, while in TASS it is pretty moderate. This is because in CDMA, every user transmits simultaneously. When user number increases, not only interference sources increase, but also the transmit power of each to resist the severer interference, which speeds up the condition to go down. In TASS, the scheduled transmission policy will gentle this situation. Note that the conventional CDMA can accommodate at most 10 users at the same time while the TASS scheme can reach to 17.

In figure 4, we show the maximal achievable throughput for non-real-time users in different number of transmitting real-time traffics while not violate the QoS of them. Assume that the number of non-real-time users is large enough such that the remaining resource after all real-time users are scheduled can be fully utilized.

B. Multicell Simulation

In the following we simulate the conditions may occur in real systems. In the experiment, the multicell environment consists of a center cell surrounded by six cells of the same size. Inter-cell interference is considered, and we evaluate performance only in the center cell.

Mobile users are distributed in the area and the speed of these users is distributed between 0 km/hr and 60 km/hr. We use four traffic models as specified in UMTS, including voice, video and interactive and background applications. Voice is modeled with two states, “TALK SPURT” and “SILENCE”. The probability of both states is 50%. During the “TALK SPURT” state, voice packets are generated as CBR with the bit-rate 60kb/s and delay bound 100ms. A packet is segmented into fixed size transport blocks and each of their size is 600 bits. The portion of voice traffic is 0.6. For video streaming, a mean transmission rate of 240kb/s is assigned. It varies with mean 0.5kbps and standard deviation 10kbps. During the simulation, the bit rate is regenerated every 10ms. The delay bound of video traffic is 250ms. The portion of video traffic is 0.1. The burst nature of data is given to non-real-time traffics. For WWW traffic, the inter-arrival time between packets is 0.4 second, and each time has 12 packets. For e-mail traffic, the inter-arrival time is modified to 0.65 second, and each time has 8 packets. The portion respect to WWW and e-mail traffic is 0.2 and 0.1.

We compare the performance in average delay and throughput between proposed TASS, conventional CDMA and HMA (hybrid multi access) transmission schemes. The HMA proposed in [8] is a strategy that chooses when to operate in uni-access and multi-access in different condition to improve performance.

The mean packet delay includes both queuing delay and the transmission delay, so the delay time counted here is the duration between the time the packet arrives into the queue and the time it is transmitted completely. In figure 6 and figure 7, they show that the TASS performs much better than CDMA and HMA both in voice and video packets average delay.

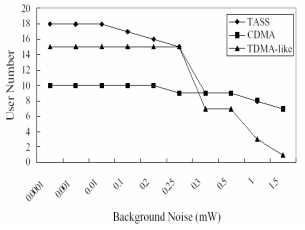


Fig. 2. Capacity in different background noise level

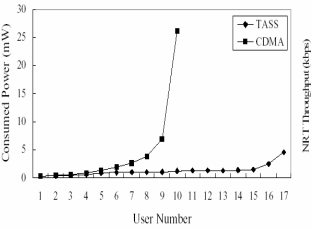


Fig. 3. Power consumption

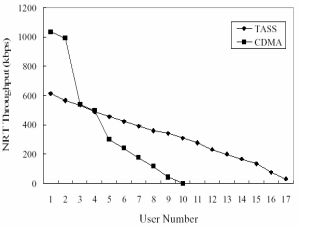


Fig. 4. Maximal achievable throughput of NRT users

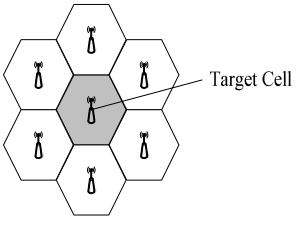


Fig. 5. Multicell environment

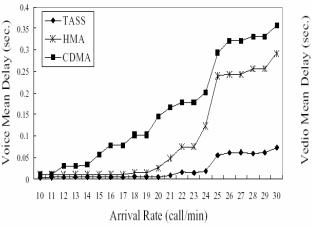


Fig. 6. Average delay for voice users

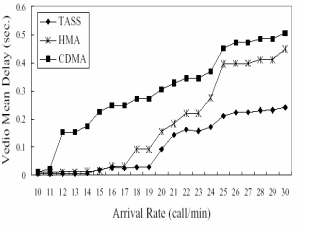


Fig. 7. Average delay for video users

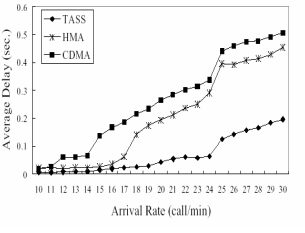


Fig. 8. Average delay for all users

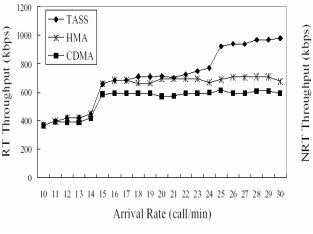


Fig. 9. Average throughput for RT users

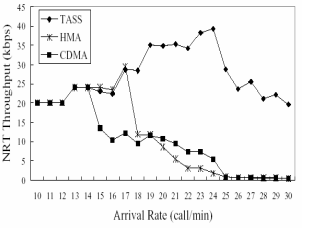


Fig. 10. Average throughput for NRT users

The average delay of all users of three schemes is depicted in figure 8. Totally speaking, the average delay of TASS is the lowest, and the HMA performs better than conventional CDMA. The growing up speed in TASS is also slower than both others.

As system load increases, throughput of real-time users increases as well in TASS. Notice the real-time user throughput of HMA and CDMA is bounded to 690kbps and 590kbps respectively and hardly to increase. This is because the system load limit is achieved in both schemes. Since higher capacity is expected in TASS, the throughput can be up to 1000kpbs. The throughput of non-real-time users in CDMA and HMA is getting decreasing from 15 and 18 respectively due to the system is full loaded and higher priority is for real-time users to transmit, while in TASS, the non-real-time user throughput is start to decrease in 25.

5 Conclusions

In this paper, we propose a novel scheduled transmission scheme for real-time users. The previous uni-access and multi-access transmission strategy performs well under different conditions respectively. The uni-access reduces the interference experienced by the user. However, we argue that the radio resource is not fully utilized. The multi-access mode avoids this problem, but the radio resource is not utilized efficiently because much power is wasted on resisting interference caused by other transmitting users in the same cell. The proposed novel transmission strategy TASS is different from both of them and forms a new transmission scheme called hybrid-access mode. TASS can be seen as a dynamic adjustment between uni-access and multi-access modes to attain best performance adapting to different situations.

The simulation results show that the best capacity is obtained in TASS which means that it can accommodate more users during a frame time while the QoS of each user is not violated compare to conventional CDMA and TDMA-like scheduling policies, and thus it is more suitable for real-time transmissions. The results also show that since less power is required in achieving the same throughput, higher throughput is expected when the same power is consumed.

The effect of multi-cell resource allocation is not demonstrated in the proposed TASS. For example, if more resource is assigned to one cell, the surroundings will experience more intercell interference such that less throughput can be achieved. How to allocate radio resource between different cells to obtain maximal overall throughput will be studied in our future work.

Acknowledgement

This work was supported by Ministry of Economic Affair. under the "Service-oriented Information Management" project (93-EC-17-A-02-S1-029). This work was also supported by National Science Council of Taiwan under the NSC93-2524-S-008-002 Integrated knowledge Management Project.

References

[1] Özgür Gürbüz and Henry Owen, "Dynamic resource scheduling schemes for W-CDMA systems," *IEEE Commun. Mag.*, vol. 38, pp. 80-84, Oct. 2000.

[2] E. Dahlman, B. Gudmundson, M. Nilsson and J. Skold, "UMTS/IMT-2000 based on wideband CDMA," *IEEE Commun. Mag.*, vol. 36, pp. 70-81, Sept. 1998.

[3] Rath Vannithamby and Elvino S. Sousa, "Resource allocation and scheduling schemes for WCDMA downlinks," *Communications, 2001. ICC 2001. IEEE International Conference on*, vol. 5, 2001, pp. 1406-1410.

[4] Ashwin Sampath, P. Sarath Kumar and Jack M. Holtzman, "Power control and resource management for a multimedia CDMA wireless system," *in Proc. PIMRC 1995*, Toronto, Canada, 1995, pp. 21-25.

[5] Deepak Ayagari and Anthony Ephremides, "Optimal admission control in cellular DS-CDMA systems with multimedia traffic," *IEEE Trans. Commun.*, vol. 2, pp. 195-202, Jan. 2003.

[6] Fredrik Berggren, Seong-Lyun Kim, Riku Jantti and Jens Zander, Joint power control and intracell scheduling of DS-CDMA nonreal time data," *IEEE JSAC*, vol. 19, no.10, pp1860-1870, Oct 2001.

[7] Sudhir Ramarkrishna and Jack M. Holtzman, "A scheme for throughput maximization in a dual-class CDMA system," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 830-844, Aug. 1998.

[8] Eric Hsiao-kuang Wu; Hao-Wei Chang; Hsu, K.C., "A QoS-based hybrid multiple access transmission strategy in WCDMA downlink," *IEEE WCNC 2003*, vol. 3, pp. 16-20, Mar. 2003.

[9] Seong-Jun Oh, Danlu Zhang and Kimberly M. Wasserman, "Optimal resource allocation in multiservice CDMA networks," *IEEE Trans. Commun.*, vol. 2, pp. 811-821, Jul. 2003.

[10] Özgür Gürbüz and Henry Owen, "Dynamic resource scheduling for variable QoS Traffic in W-CDMA," *IEEE International Conference on*, vol. 2, pp. 703-707, 1999.

[11] Mohammad Ali Arad and Alberto Leon-Garcia, "A generalized processor sharing approach to time scheduling in hybrid CDMA/TDMA," *IEEE INFOCOM'98*, vol. 3, pp. 1164-1171, 1998.

[12] 3GPP TS 25.854 v5.0.0, "Study report for uplink synchronous transmission scheme (USTS)," 2001.

Semantic Web Enabled VHE for 3rd Generation Telecommunications*

Songtao Lin[1] and Junliang Chen[2]

[1] State Key Laboratory of Networking and Switching, Beijing University of Posts and Telecommunications, 187#, 10 Xi Tu Cheng Rd, Beijing 100876, China
stlin@telestar.bj.cn
[2] School of Computer Science and Technology, Beijing University of Posts and Telecommunications, 10 Xi Tu Cheng Rd, Beijing 100876, China
chjl@bupt.edu.cn

Abstract. Former approaches to VHE resulted in considerable traffic loads and security threats by moving service logics across networks. In order to minimize extra traffic resulted by VHE, here we employ semantic web technology to setup a semantic background, upon which VHE could comprehend the expectations of users and the abilities of services, thus could automatically and precisely select an appropriate local service for a service request of a roaming user. As a first step to reach the semantic web enabled VHE, we model related knowledge into ontologies and present a suggested architectural framework.

1 Introduction

In recent years, 3rd generation mobile telecommunications gradually reached its maturity, while 3rd-party value-added service provider (VASP) plays a more important role in the value chain. Armed by more broadband networks capable of switching packet data and more powerful terminals, future mobile users will be able to enjoy a tremendous amount of value-add services (VASs), just like what they do with their PCs in the current fixed Internet. Different with standardized voice services provided by mobile network operators (MNOs), many VASs only serve a relative small group of users, or even some of them are configured with particular values for some certain parameters in order to meet the requirement of only one single user. Making these personalized services continuously available for roaming users is a big challenge in mobile Internet. In order to fulfill this goal, 3GPP proposed a concept so-called Virtual Home Environment (VHE) [1].

From 3GPP's perspective, the concept of VHE is such that users are consistently presented with the same personalized features, User Interface customization and services in whatever network and whatever terminal (within the capabilities of the

* The research in this paper is supported by the National Natural Science Foundation of China under the contract number 60432010.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 113 – 122, 2005.

terminal and the network), wherever the user may be located [1] [2]. The main idea behind VHE is providing personalized service portability across network boundaries and between terminals, offering users the same look and feel.

VHE offers an opportunity for a new competitor named Mobile Virtual Network Operator (MVNO) to play a role in the mobile Internet arena [3]. MVNOs do not run mobile network infrastructures; neither do they create VASs. A MVNO attracts users by listing out a huge number of services (just like a Internet portal does seen in the fixed Internet today) serving an area maybe wider then a single MNO covers, as well as allowing the user to customize the services they subscribed and enjoy the services regardless of the point of access. There may be a range of flexible and attractive services instead of some certain "killer applications". It is considerably similar to the service model of fixed Internet today [4].

2 Former Approaches to VHE

2.1 Former Research Examples

Although being considered as promising service provision architecture, VHE is merely a concept and has not been fully standardized at this time, hence could be realized using any existing technology. The basic thought of former approaches to VHE is to deploy a VHE middleware between users (terminals) and accessed networks, as well as between services and networks infrastructures. Figure 1 presents the conceptual architecture of VHE reflecting above thought. In this architecture, VHE middleware communicate with underlying networks via Parlay APIs [5] for the purpose of operating over a heterogeneous network environment.

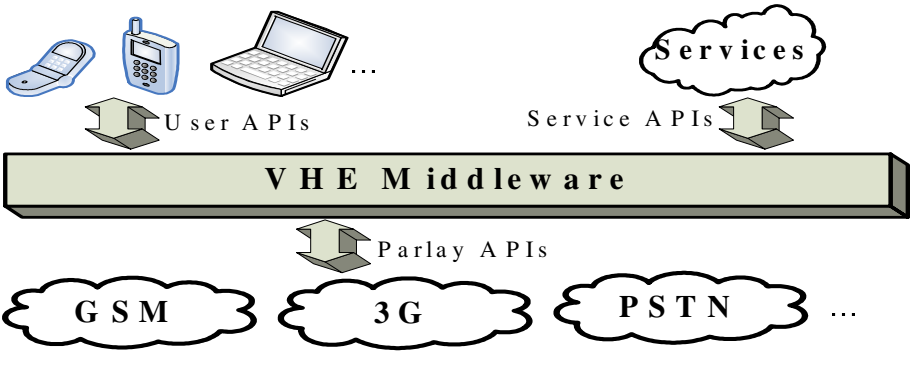


Fig. 1. VHE Conceptual Architecture

In order to consistently provide personalized service portability, VHE middleware should take on the responsibilities of, firstly, storing the home environment of each user, including the services he subscribes and their features; secondly, gathering information involved in the service provision while a roaming user initiates a service, including context of that service request, such as the user's preference about this service and the capability of his currently used terminal; and at last, offering user an

appropriate service with his familiar look and feel. It is really a heavily complicated and dynamic process. Till now, there have been a range of researches trying to realize this process.

Paper [3] illustrates three kinds of control mechanisms apply for VHE services provision, which are home domain control, visited domain control and distributed control. In home domain control scenario, home network provider maintains the user's service profile data, service-related data and service logic, and directly provides service for users roaming in visited network. In visited domain control scenario, while a roaming user requests a service, almost all components of service logic, user's service profile/data and so on are temporarily downloaded to and stored in visited network, which will control required service. Distributed control means that only partial service logic, partial service profile/data are downloaded to visited network, and the required service is controlled by both home and visited network. The authors point out that visited domain control will lead to lowest signaling/traffic load, since all required objects are downloaded/stored within nearby visited network, while distributed control is a trade-off concerning signaling/traffic load and flexibility.

CAMELEON (Communication Agents for Mobility Enhancements in a Logical Environment of Open Networks) [6], an european ACTS (Advanced Communications Technologies and Services) project, proposed an entirely agent-based VHE architecture in which service is split into two distinct parts: user interface implemented by a user interface agent (UIA) and actual (core) service implemented by a service agent (SA). Both UIA and SA are designed as autonomous mobile agents capable to move to users' terminals or visited network nodes, respectively. The use of mobile agents fit the nature of service portability over heterogeneous networks, and assumed by CAMELEON it could significantly reduce network traffic and make service provisioning more efficient, because the service logics' migration from home to visited network could lower signaling load across network boundary.

2.2 Limitations of Former Approaches

Efficiency of service portability is a nature of VHE needs to be considered. VHE is a large-scale system, and needs to control and manage several networks and serves a large number of users. Only reducing imposed extra traffic load as well as costly resources usage as much as possible, could VHE provide users familiar services with same look and feel of price. That is why both approaches described above pay many attentions on efficiency and both of them adopt similar solution of moving service logics to visited operator's domain. But actually, which one of migration of service logics and transfer of service signaling would cause less extra traffic, is hard to be determined beforehand. Therefore, when or in what situation should service logics be moved, or when or in what situation should signaling be transferred, is really an awkward dilemma in practical operation.

Another negative of moving service logics is about security. Particularly, it is not an easy task for visited network nodes to prevent malicious attacks or potential damages hiding in service logics migrate from other foreign networks.

From above points of view, moving service logics between networks maybe is not a very good idea. If possible, providing service portability without migration of service logics would be much better.

3 Semantic Web Enabled VHE

3.1 Introduction of IMSP

A project named "Foundational Research for Intelligent Mobile Service Platform (IMSP)", which is funded by National Natural Science Foundation of China, is currently conducted by the State Key Laboratory of Networking and Switching (Beijing University of Posts and Telecommunications). The objectives of IMSP are: (1) Theory and technologies of providing personalized service with regard to user's habit and hobby and other preferences; (2) Portability of service across networks; (3) Adaptability of service including content delivered to user and method of delivering, with regard to capability of terminal and physical situation of communication channel currently used; (4) Openability and extensibility of platform; (5) Realization of a prototype. The 2nd and 3rd objectives are the standard requirements of VHE, while the 1st objective is not in the coverage of 3GPP's VHE definition but could be looked as valuable features provided to VHE users in order to give them friendly service experiences.

In IMSP, in order to fulfill above objectives, we employ many possible concepts and technologies, such as Parlay, JAIN, OSA (Open Service Access), XML, Web Service, MExE (Mobile Execution Environment), SAT/USAT (SIM/USIM Application Toolkits), and the state-of-art semantic web technology [7]. In the remaining part of this paper, we only focus on the implementing of semantic web in order to fulfill above mentioned VHE-like objectives (1) (2) (3) of IMSP.

3.2 Brief Overview of Semantic Web

The boom of research activities related to semantic web started in the late 1990s. The original intention of semantic web is to make web contents could be comprehended instead of only could be read by machines, thus machines could process those contents as knowledge instead of data.

Ontology [8] and its modeling languages such as Ontology Web Language (OWL) [9] are key enabling foundations for semantic web. Ontology is a formal, explicit specification of a shared conceptualization [10]. It organizes the knowledge of a domain into taxonomies of concepts, each with its attributes, and describes relationships between concepts. Semantic web enables the reasoning about logical relationships hiding in text contents which have been marked up with ontologies; hence machines could understand the semantic in those contents. Ontology modeling languages such as OWL enable creation of ontologies for any domain.

3.3 Semantic Web Based VHE Solution in IMSP

3.3.1 Basic Philosophy

The limitation of former approaches to VHE mentioned above motivates us to try a new approach for VHE in IMSP. The basic philosophy is that, if possible, present roaming users with service provided by local providers, and still keep VHE features.

While a roaming user requests a service he subscribed in his home network, if visited network could provide appropriate service with same or similar look and feel,

it evidently will minimize extra traffic as well as unnecessary occupations of system resources, since there isn't any service logic migration neither any signaling control relationship between user and service provider located in home network, only some necessary data such as description of requested service needs to be transferred across networks. Likewise, the security threat resulted by service logic's migration will also disappear.

In mobile Internet, there are a huge number of all kinds of VASs created by VASPs, and therefore it is quite possible to find a local service meet the requirement of a roaming user. It is just like in today's fixed Internet, users could nearly always find many web-sites providing same or similar contents or services such as news and weather forecasting. It also fits the sense of service provision in real word, for example, we could book an airplane ticket in local agency. Further more, we suggest that VHE users should subscribe VASs from VHE middleware provider (VHEP). In another word, there are subscription relationships between users and VHEP as well as business relationships between VASPs and VHEP, while there is no direct contractual relationship between users and VASPs. So, VHEP could well guide VASs development in order to achieve that in each network there are some counterparts of each kind of service. Another advantage of this business model is that VHEP could maintain a consistent view of the user's preference, which is very important for service personalization.

3.3.2 Related Issues and Requirements

In our vision, VHE is a middleware in charge of filtering local services with regard to roaming user's service personalization, irrespective of terminal and network. In developing of such a VHE middleware, there are some important related issues need to be considered:

(1) In order to provide roaming user a local VAS similar with the one he subscribed in his home network without renegotiating with him, the expectation of his service request (such as booking a tomorrow airplane ticket from Beijing to New York) should be clearly declared. Likewise, the features of each VAS, including its purpose (such as weather forecasting) and behaviors (such as its charging plan) need to be predefined as a VAS profile. The consistency across multiple users and VASPs about service request declarations and VAS profiles should be guaranteed, thus VHE could conduct filtering and matching process and determine an appropriate service for the user. Surely, this consistency is not only in format or syntactic level, but also in human-human semantic level. So that VHE can easily parse and interpret and understand the requirements of users and the abilities and behaviors of VASs, and make a precise and effective matching for these two parties accordingly. Ontology technology, which promises shared understanding of domain knowledge, is considered to be the semantic foundation of IMSP and used to mark up VAS profiles as well as service request declarations.

(2) The context of service is also an important factor of personalized VAS. In the domain of VAS provision, we comprehend context as a term comprising information relative to and maybe influence the service provision in runtime, including user's preferences, network's performances and terminal's capabilities. If a VAS has the ability of intelligently cooperate with context in order to give users more friendly experiences, such context-aware feature should be retained by VHE for roaming

users. It is crucial to obtain a clear and deep understanding of context concept, and model context into a certain ontology. Hence there could be a uniform way for developing context-aware services serving VHE users. A high-level view of suggested context modeling ontology will be presented later in this paper.

(3) VHE middleware should have the ability of managing VAS profiles marked up with some certain ontologies, and automatically selecting an available appropriate service semantically matches the user's expectation. If it results in more than one equivalent service, VHE middleware will select one with regard to some certain policies such as "the cheaper, the better". Particularly, if the user request a service with the feature of context-aware, VHE middleware should select a VAS fits the current context, or select a context-aware VAS and deliver the context marked up with ontologies to that VAS. Moreover, VHE middleware should have the ability of interacting with other VHE middlewares.

3.3.3 Ontologies Design and Usage

Considering the issues described above, we divide related knowledge in the domain of service matching into two groups: the service and the context, and design two ontologies for modeling them respectively.

The service ontology (SO) models the knowledge directly reflects the service characteristics, such as name, provider, purpose, charging plan, QoS, supported geographic area, access method and so on. Figure 2 is a high-level illustration of SO. In this figure, ellipses indicate concepts while arrows indicate properties of concepts.

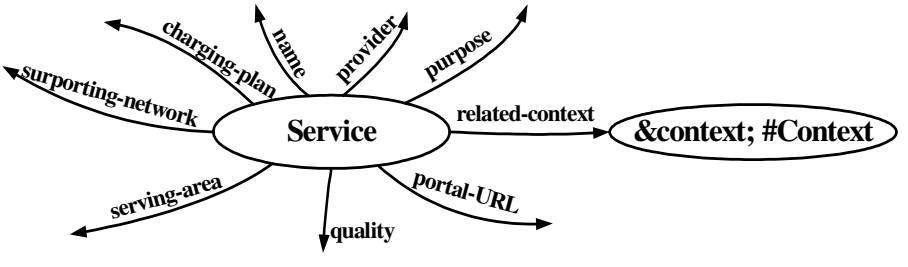


Fig. 2. High-level View of Service Ontology

Each service within VHE domain has a description document so called service profile (SP). All SPs should be marked up with SO and other necessary ontologies. For example, the "purpose" property of an e-commerce service should be marked up with an ontology of e-commerce domain, and the "related-context" property of service should be marked up with context ontology described below. All SPs should be registered in SP storage within VHE middleware when VASPs publish their services.

Likewise, while a user subscribes a service, a description of his personalized service named user service profile (USP) is created with semantic markups and stored in USP storage within VHE middleware. Each user could hold many USPs, because he could subscribe many services as well as many personalized counterparts (each

relate to a USP) of one single service with different sets of features. Actually, while a user initiates a VHE service, his terminal sends the URL of the corresponding USP to VHE middleware.

The context ontology (CO) models the environmental information which maybe will influence matching or adapting of service, including gender of the user, location of the user, available bandwidth, user's generic usage preferences and interface features of terminal, and so on. Figure 3 is a high-level illustration of CO.

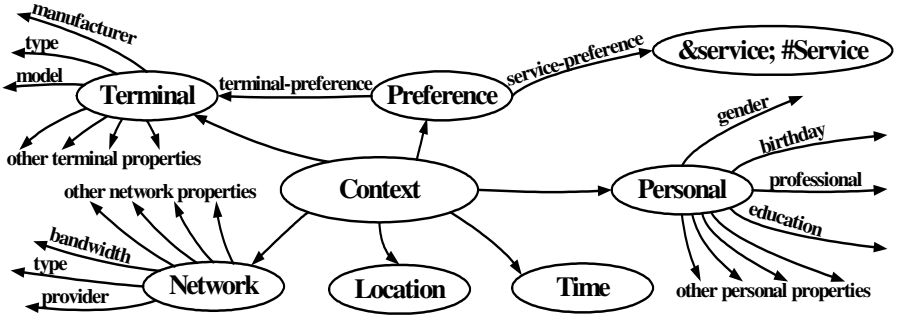


Fig. 3. High-level View of Context Ontology

In our vision, VHE middleware adopts different methods to obtain and manage different kinds of context. At the beginning of a user subscribes to be a VHE user or later, he could register or edit his personal and preference context. This information is marked up with ontologies and formatted as user profile (UP) stored in UP storage within VHE middleware. Other context, such as terminal, network, location and time should be gathered from terminal or network at the time of service request or during service execution. This gathered context could be formatted as a document with semantic markups and sent to VHE middleware. Particularly, for terminal and network context, another alternative method for context gathering is that VHE middleware predefined profiles describing the commonly used terminals and frequently encountered network situations, which are named terminal profiles (TPs) and network profiles (NPs), respectively; and then, VHE middleware will comprehend the terminal or network context of a service by receiving a URL to a certain profile.

3.3.4 Suggested Architectural Framework

Figure 4 illustrate our suggested architectural framework for VHE middleware in client-server paradigm. A client-side context collector is deployed inside the terminal, taking the charge of collecting terminal context and delivering them to the server side. The terminal context delivered could be a formatted document with CO markups, or a URL indicating the predefined TP, just as we described in last section.

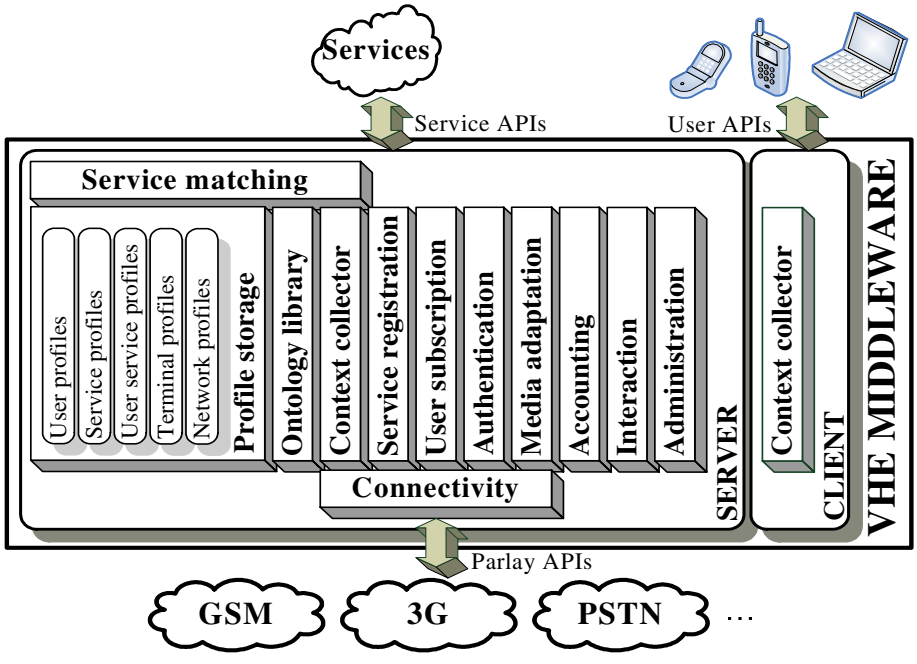


Fig. 4. Architectural Framework for VHE middleware

Modules on the server side are:

(1) Profile storage, stores all profiles within this VHE middleware and temporarily caches profiles migrated from other VHE middlewares.
(2) Ontology library, stores all ontologies including SO and CO. Other ontologies cooperate with SO and CO to mark up profiles are also encouraged to be stored here. Otherwise the failure of node storing necessary ontology will lead to failure of obtaining that ontology, hence lead to failure of service matching.
(3) Context collector, takes charge of collecting context from profile storage or underlying networks or client-side context collector.
(4) Service matching, takes charge of selecting appropriate local service for user, according to the expectation of service request, the abilities of available services and service context. This module mainly cooperates with profile storage, ontology library and context collector.
(5) Service registration, allows VASPs to register their service profile before service publishing.
(6) User subscription, allows users to subscribe services and register or edit their UPs.
(7) Authentication, prevents illegal logins.
(8) Media adaptation, adapts media flow according to the capabilities of user's terminal or performance of network, such as screen size or supported bandwidth.
(9) Connectivity, connects to underlying heterogeneous networks via Parlay APIs.

(10) Accounting, takes charge of service charging and billing.
(11) Interaction, interacts with other VHE middlewares.
(12) Administration, allows system operators to manage the server-side system.

4 Conclusion and Future Work

VHE is proposed by 3GPP as the global ubiquitous service environment for 3rd generation mobile telecommunications and currently under standardization. It promises to provide services for roaming users in their familiar look and feel across multiple providers, various terminals and heterogeneous underlying network, and hence, new business opportunities for network, terminal and service providers.

Several exiting works in the realization of VHE obtain service mobility by service logic migration which leads to extra traffic load between networks and potential security threats. However, we believe that the expected huge number of services available in the future mobile Internet make it possible to select an appropriate service meet the requirement of a roaming user. In another word, migration of service logics is unnecessary in most situations.

This paper elaborates our early-stage approach to VHE in the project of IMSP. Different with former works, our approach establishes a firm semantic foundation for VHE, upon which the users' expectations of requested services and the features of available services as well as the context of services could be specified as semantic knowledge, which could be comprehended and automatically matched by VHE middleware. Ability of selecting local services compatible with home services to serve roaming users minimizes the resulting traffic load across networks and makes services more cost-effective and affordable and safe. As the first step to this target, a couple of ontologies modeling related knowledge of VHE service provision and a suggested architectural framework for realization are presented.

Our future works include investigating the theory and technologies for realization of service composition within VHE middleware. If such objective could be achieved, VHE middleware could provide a roaming user with integrated service composed by several services even in the situation that there is no a single appropriate service ready for the user's request. We will also build a semantic web enabled VHE prototype. Surely, problems of scalability, reliability, security, and QoS-guaranteed mechanism have to be faced.

References

1. 3rd Generation Partnership Project, TR 22.121: Technical Specification Group Services and System Aspects, Service Aspects, Virtual Home Environment (Release 4), V4.1.1 (2002-6).
2. 3rd Generation Partnership Project, TR 23.127: Technical Specification Group Services and System Aspects, Virtual Home Environment (Release 4), V4.3.0 (2002-03).
3. Fawzi Daoud, Seshadri Mohan: Strategies for Provisioning and Operating VHE Services in Multi-Access Networks, IEEE Communication Magazine (2002-1).
4. Lin Songtao, Chen Junliang: An Agent-based Approach to VHE, Proceedings of 2003 International Conference on Communication Technology (2003-4).

5. The Parlay Group: http://www.parlay.org.
6. The European ACTS Research Project, cameleon: http://www.comnets.rwth-aachen.de/~cameleon.
7. Tim Berners-Lee, James Hendler, Ora Lassila: The Semantic Web, Scientific American (2001-3).
8. Mike Uschold, Michael Gruninger: Ontologies: Principles, Methods and Applications, The Knowledge Engineering Review, vol. 11, no. 2 (1996-6).
9. Web Ontology Language (OWL): http://www.w3.org/2004/OWL.
10. Thomas R. Gruber: A Translation Approach to Portable Ontologies, Knowledge Acquisition, vol. 5, no. 2 (1993).

An Adaptive Replication Algorithm in Overlay Networking

Yuancheng Wu, Wenhua Lang, and Mingtian Zhou

School of Computer Science and Engineering, UESTC, Chengdu

Abstract. We present ARK (Adaptive Replication of Keywords) which is a replication algorithm working on DHT overlay networks. We proved in this paper that ARK is near optimal in load balance and is competitive in replica management overhead. Simulations show that ARK has great improvements in load balance and fault tolerance, comparing with existing replication algorithms in CAN, Pastry and Bamboo. ARK is achieved without explicit metadata or the need for a replica directory service, works as an independent building block on top of any overlay system. Nodes unaware of ARK can work well with nodes equipped with ARK. Although our research is part of a content indexing system, the algorithm is suitable for any overlay based data item storage and lookup system.

1 Introduction

Overlay network provides scalability, robustness, flexibility and much more properties that can support large scale distributed system. There are many researches that focus on build distributed systems on top overlay network. The grid community also begins to build grid services on top of p2p network [5], which is one kind of overlay network.

To provide real-world distributed applications on top of overlay network, system load-balance and fault tolerance must be well treated. Replication mechanism is a great choice. The idea of replication is not new, but in the area of overlay network, the research is relatively inadequate. Currently, most DHT-based replication algorithms adapt uniform distributed replicas, which work well in uniform distributed query environment. Unfortunately, real-world queries often change popularity over time, and the popularity follows the distribution of Zipf[3]. Under such situation, nodes that hold popular keywords will have much more load than those holding unpopular keywords. The load for the nodes is extremely unbalanced. This can do harm to the overlay network, where the nodes may be owned by different entities.

Moreover, the placement and subsequent efficient location of replicas in DHT systems remain open problems, especially (1) the requirement to update replicated content, (2) working in the absence of global information, and (3) determination of the locations in a dynamic system without introducing single points of failure.

In this paper, we present Adaptive Replication of Keyword algorithm (ARK). The design goals of ARK include:

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 123 – 132, 2005.

1. The load balance performance of ARK is better than current adapted algorithms.
2. While facing large portion of nodes' failure, the network can still has good availability. And ARK performs better than current algorithms.
3. The replica number is dynamically adaptive to the access pattern and the operations overhead on add/remove replicas must have upper bound even in worst situation.
4. Replica locating scheme must be simple and clean. Nodes without knowing replication algorithm can work together with ARK.
5. On obtaining the previous advantages, the runtime replica number must not increase significantly.

This paper is organized as follows: In section 2, we introduce related works and their limitations. After that, we put forward our work on replication algorithm, with analysis on optimization and competitiveness. Our experiments of replication technology on overlay-based content search will show improvements in respect with load balance and fault resilience. At last, we give conclusion and future works.

2 Related Work

In [11], replication strategies in unstructured P2P network are analyzed. In unstructured P2P network, the main problem is the great amount of query messages the flooding algorithm generates. So the main concern of replication strategy is to reduce the amount of query message. In DHT network, the number of query messages is managed, and we no longer have to cope with it, so we can put our attention in load balancing.

Replication algorithm under unstructured p2p system is discussed in [1], which addresses load balance. In this paper, replicas are added/removed on the read rate of querying node. Replica is created at the source of query, near randomly, so it is difficult to locate and refresh replicas. Moreover, each nodes have to remember all replicas. In our work, we borrowed the idea of control replica number by upper/lower threshold and adapted better method of locating replicas.

CDDR[2] also provide an adaptive replication algorithm, in which number of replicas changes according to read/write pattern of the system. But its algorithm heavily relies on “primary replica” for every data item, which brings central failure point for each data with replicas.

Replication mechanism has already been used in DHT based overlay network. In CAN[5], replication is achieved by using multiple hash functions on the same data item. But to achieve optimized load balance, the number of replica must change according to the access pattern, different hash functions for different replica is not feasible.

In Pastry[6] and Tapestry[7], replication is attained by storing an object on the k Pastry servers whose identifiers are closest to the object key in the namespace. Bamboo[9] also has its replication scheme, which uses fixed number of replicas stored in continuous nodes in identifier-space.

Locality[10] is a great feature being added into overlay networks. In locality-aware networks, nodes that are close in physical location are also close in identifier space, so that to prevent long distance in physical network while close in identifier space,

which brings long and unnecessary overhead of query and data transform. But if we store replicas in identifier-space closed nodes, they will probably be crowded in a same intranet, which may lead to all the replicas in the intranet suddenly unavailable if the router of the intranet fails. So it is not a good idea to store replicas for a same data item in continuous nodes.

3 The ARK Replication Algorithm

In this section, we describe our adaptive replication of keyword (ARK) algorithm. The term "keyword" refers to a data item consists of data related to the keyword. In this paper, keyword is the atom element of replication.

Rule 1: The basic item for replication is the keyword
Although the length of the data for different keywords varies radically, we don't plan to divide keywords into fragments, because that will complicate the algorithm a lot and add much overhead to the operations. The tradeoff must be made.

Rule 2: The location of a replica is determined by the hash result out of the keyword AND the sequence number of the replica
Identifier of a keyword determines which node the keyword will be stored. Without replication, the identifier (id) of a keyword is obtained from (1). In our algorithm, identifiers are calculated with additional ingredient: sequence number. For example, identifier for the second replica for "key1" will be calculated by (2).

$$id = hash(key) \tag{1}$$

$$id = hash(key1,'2') \tag{2}$$

For the first replica of a key, the symbol '1' will not be added in hash calculation, that is, use (1), for back compatible to nodes unware of replication. Because one of basic hash function properties, we can believe that different replicas for a same keyword is distributed evenly among the identifier space, thus replicas are distributed across the network than aggregated to some areas, while the replicas locations are easy obtain.

This replica locating scheme makes it easy to locate any replica immediately, without having to maintain any metadata to manage replica locations. We can see that our design goal 4 is met.

Rule 3: The number of replicas for a keyword is adaptive to the access pattern
We denote the number of replicas for a keyword by r . To leverage r , each replica monitors the access pattern. In this paper, we use the term "time unit" to describe the minimal time about operations. Real system may take one "time unit" as several seconds, one minute, or one hour.

- θ : Recent read rate. The replica records the number of read operations towards it in recent t_0 time units. When θ grows, we consider the replica is more loaded. We use (3) to update θ .

$$\theta(t+1) = \theta(t)\left(1 - \frac{1}{t_0}\right) + \theta_{recent} \tag{3}$$

where θ_{recent} is the read times monitored in recent ONE time unit, and denote by $\theta(t), \theta(t+1)$ the value of θ at time $t, t+1$, respectively.

Some constants are used to regulate the increase/decrease of r.

- τ: Expected read rate for one replica. This value is determined by the system capacity and nodes' capability of handling replicas.
- ρ: Possibility of add/remove replica: This is a damping factor that prevent frequently useless add/remove of replica. $0 < \rho < 1$.
- r_{max}: Maximal number of replicas of a keyword.
- r_{min}: Minimal number of replicas of a keyword.

Given these parameters, we can describe the process of increase (decrease) of r. In a replica, a value called estimated-deviation(δ) is calculated.

$$\delta = \sqrt{r}\left(\theta / t_0 - \tau\right) \tag{4}$$

For $p \in (0,1)$, If δ exceeds τ, when $p < \frac{\delta\rho}{\sqrt{r}\tau}$, a new replica is added. If δ lower than τ, when $p < \left|\frac{\delta\rho}{\sqrt{r}\tau}\right|$, a replica is removed.

Rule 4: The add/remove operations only target the last replica

When an add operation is decided, the node responsible for $id = hash(key1, r+1)$ is notified. The node initializes the state as $\theta = \tau \cdot t_0$ which is the average value of δ, copies the replica to local place. Then all the replicas refresh their state by $r = r + 1$.

The remove operation is alike. The last replica abandons its replica, and all the remaining replicas refresh their state by $r = r - 1$.

Rule 5: The read/write operations

We follow the "read one, write all" rule. When A wants to read, it first obtains the number of replicas of the keyword interested. A consults the first replica to get r, if fails, the second replica is consulted, until r_{max} is reached or one replica answers. With the knowledge of r, A choose one to proceed with read, each replica with equal property. If the chosen replica fails to response, another replica is chosen, until all the possible replicas are all probed, which leads to the keyword to be unavailable.

When A wants to write, it does the same operation in the first step in reading to obtain r. Then A locks all the replicas, writes to them, then unlock.

ARK doesn't treat node failures directly. When one node fails, the lookup process just consults another replica location, instead of trying to fix the problem. We leave this work to the underlying overlay system, waiting for the node to recover or some other node takes over its job. After a write operation, the gap will be filled.

4 Analysis

In this section, we prove that our algorithm is near optimal in load balance, and is competitive in replica operation overhead.

4.1 Optimal Load-Balance Analysis

Assume we have a DHT network of N nodes; each has the capacity of $m_n(n=1,2,..,N)$ to hold replicas. We put K distinct keywords $k_1 k_{2,}..,k_K$ into the network, each has $r_i(i=1,2,..,K)$ replicas, respectively. When a keyword is queried, each of its replicas has the same possibility of $\frac{1}{r_i}$ to be accessed, which is true in ARK (rule 5). The query rate against each keyword $q_i(i=1,2,..,K)$ is not uniformly distributed, but has the distribution $q_i = q(i)$. Thus each replica for keyword k_i will receive queries at rate $\lambda_i = q_i / r_i$ in average. The goal of load balancing is to make λ_i uniform. We use deviation of λ_i to indicate the level of load balance, as shown in (5) and we call D the "load balance factor" (LBF).

$$D=(\sum_{i=1}^{K}(\lambda_i-\bar{\lambda})^2)/K \tag{5}$$

where

$$\bar{\lambda}=(\sum_{i=1}^{K}\lambda_i)/K \tag{6}$$

is the average value for λ_i .

To achieve load balance, the question becomes

$Minimize(D(\lambda_i))$ when $q_i = q(i)$ and $\sum_{i=1}^{K} r_i \le \sum_{n=1}^{N} m_n$.

Theorem 1: The distribution is optimal if the distribution of r_i is proportional to that of q_i .

Proof is omitted due to space limitation.

Theorem 2: ARK is optimal in load balance.

Proof: To prove our algorithm is optimal, we need to show that for different keyword k_1 k_2 , the number of replicas for the keywords is proportional to the request rates against them.

That is, if we define $r_{1/2} = \frac{r_1}{r_2}$ and $q_{1/2} = \frac{q_1}{q_2}$, we need to prove (7).

$$r_{1/2} = q_{1/2} \tag{7}$$

Assume at some time t , $r_{1/2}(t) < q_{1/2}(t)$. Because $q_i = \frac{r_i \theta_i}{t_0}$, where θ_i is "recent read rate" of a replica for k_i , calculated from (3), we get $\theta_1 > \theta_2$. As in rule3, if $\theta_1 > \theta_2 > \tau t_0$ ($\tau t_0 > \theta_1 > \theta_2$), both r_1 and r_2 will decrease (increase), until $\theta_1 > \theta_2 \cong \tau t_0$ ($\tau t_0 \cong \theta_1 > \theta_2$). If $\theta_1 > \theta_2 \cong \tau t_0$, observe (4), r_2 will keep steady and r_1 will continue to increase when $\delta > \tau$, thus $r_{1/2} = r_1 / r_2$ will increase. On the other

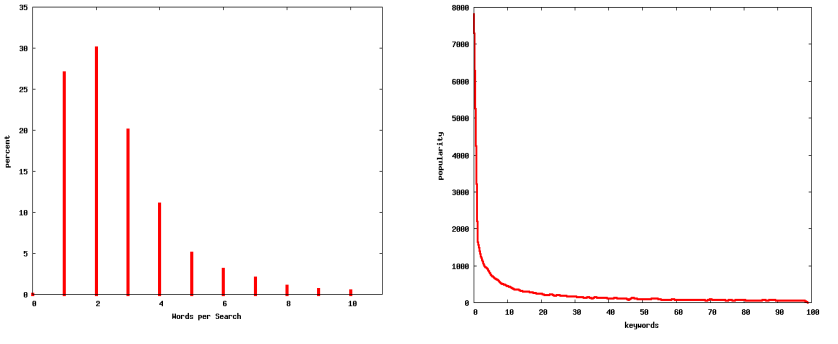
hand, if $\tau t_0 \cong \theta_1 > \theta_2$, r_1 will keep steady and r_2 will decrease, thus $r_{1/2}$ will also increase. So if $r_{1/2}(t) < q_{1/2}(t)$, $r_{1/2}$ will increase. Likely, if at some time t, $r_{1/2}(t) > q_{1/2}(t)$, we can deduce $r_{1/2}$ will decrease.

So we can conclude that the algorithm will adapt towards $r_{1/2} = q_{1/2}$. □

In fact our algorithm is just "near optimal", because there is some constraints that prevent the strict proportional distribution of r. First, every keyword must have at least r_{min} replica(s), even when there is no reads against it for a long time. Second, there must be an upper bound of replica number to put the replication under control, but this also prevents r from growing freely to meet the need of $p < \left| \frac{\delta\rho}{\sqrt{r\tau}} \right|$. Third, to prevent r change too quickly when the query pattern changes, some mechanism is introduced to slow down the add/remove process, which also leads to "near optimal".

4.2 Competitive Analysis

Competitiveness is a widely-accepted way to measure the performance of an on-line algorithm [12]. Formally, a dynamic replication algorithm P is said to be c-competitive if there are two constants c and d, such that for any request sequence ψ, $COST_P(\psi) \le c \cdot COST_A(\psi) + d$, where A is the optimal algorithm that is designed at a priori knowledge of the input sequence. The competitiveness property bounds the worst case cost to be within a constant factor of optimal algorithm A.



a. Number of Keywords per Query *b. Distribution of keywords*

Fig. 1. Query properties in experiment

Note that the optimal offline algorithm A is different with the optimal load-balanced algorithm discussed in the previous section. We are talking about the cost of read/write operations for the replication algorithm in this section.

We assume the least operations required to read or write a replica to be 1, and denote by c_d the cost of transmitting the data of a keyword from one node to another,

and c_c the cost of transmitting a control-message from one node to another, such as messages consulting for number of replicas.

Theorem 3: ARK is $(c_c + c_d) \cdot r_{max}$ -competitive.

The proof is omitted due to space limitation.

With the analysis of competitiveness of ARK, we can see that ARK meets our design goal 3. Note that the competitive ratio has less meaning than the competitive property itself. In fact, algorithms with higher competitive ratio usually work better in practice [12].

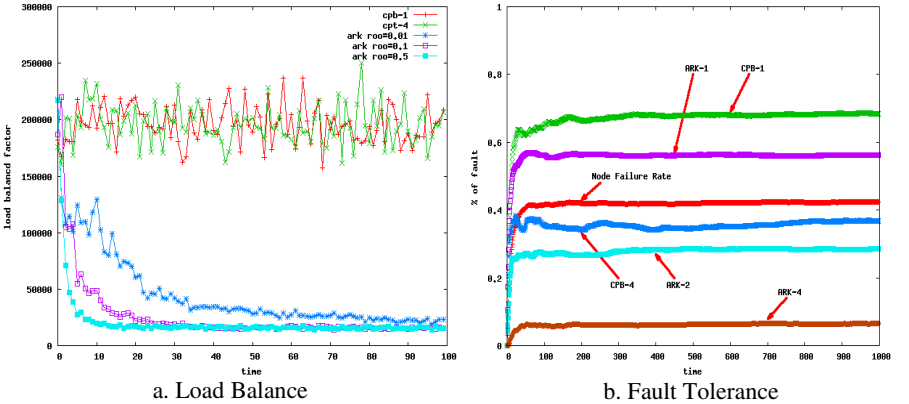
5 Experiments

In this section, we compare ARK with available replication algorithms on top of DHT overlay network. As was mentioned, CAN, Pastry, and Bamboo utilize roughly same replication algorithm, which we call in this paper as CPB. We compare the performance of load balance and communication overhead.

5.1 Load Balance

We simulate an overlay network consists of 1,000 nodes, with 10,000 keywords scatter among these nodes. Each keyword is replicated in two ways: CPB used in CAN, Pastry and Bamboo, and ARK.

The queries against these keywords are issued as follows, each query has one or more keywords, the number is under the distribution shown in Fig1a, which is the typical search pattern discovered by [3]. Researches show that queries of keywords have different popularity. The popularity follows a Zipf distribution [3]. The most popular keywords is searched for a large portion among all the queries. Fig1b is an example of keyword popularity distribution generated by our simulator.



a. Load Balance b. Fault Tolerance

Fig. 2. Performance Comparing

The simulation parameters in this section are $N = 1000, K = 10000, r_{max} = 200$. For each time unit, the system generates 1,000 queries, as described above. After every time unit, the system polls the network parameters. The result for load balance factor defined in (5) is shown in Fig 2.

The load balance result is shown in Fig2a, where 'cpb-1' 'cpb-4' means CPB algorithm with fixed replica number of 1 and 4, respectively. 'roo=' means in ARK algorithm, we take different values of ρ.

We can observe that fix-numbered replication algorithm CPB(used in CAN, Pastry, Bamboo) present a almost steady LBF(load balance factor), while ARK has a much lower LBF. The improvement is about 60 times, which indicates ARK meets our design goal 1. We can also see that ARK first act badly, its LBF is much higher than that of CPB, that is because ARK initially allocates one replica for each keyword, while in CPB each keyword have 4. Soon the system adapts the input queries. In less than 20 time units, ARK reaches a steady state. We can also see from the graph that the different values of ρ result in different speed of aggregation, bigger ρ leads to faster aggregation. But big ρ also has drawbacks. When ρ increases, the add/remove operation overhead increases a lot. For example, when $\rho = 0.5$, the operations needed in first 100 time units are 1802, while when $\rho = 0.01$, the number is 112.

5.2 Fault Tolerance

We use the percent of success queries (PSQ) to measure the fault tolerance performance. For a query consists of 1 to 10 keywords, only when all the involved keywords are successfully read, can be a success query.

The fault model of any node is as follows. Every node works an expected period of T time units before it fails. If one node fails, the nodes that are close to it in the identifier space will also get failed in possibility of p_{fail}, which has the distribution $f(p_{fail}) = \frac{|i - i_0|}{l}, |i - i_0| < l$, where i is the identifier of nearby node, i_0 is the identifier having been failed, l is the range that a failed node will affect. After failure, the node will recover after a while. The time length is t_e, with the distribution $f(t_e) = \frac{2}{\sqrt{2\pi} t_{e0}} e^{\frac{(t_e - t_{e0})^2}{2t_{e0}^2}}, 0 < t_e < 2t_{e0}$, where t_e is the average time of recovery.

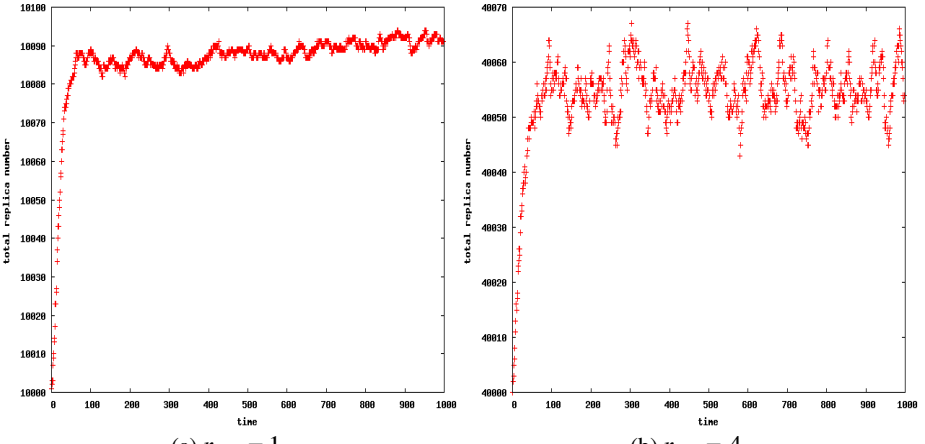
In the simulation, we use $T = 50, t_{e0} = 10, l = 5$, and the results are shown in Fig2b, which illustrates the network failure node rate, CPB and ARK unsuccessful query rate. The number after "CPB-" denotes the number of replicas used in CPB algorithm, and the number after "ARK-" denotes different r_{min} values taken in ARK algorithm.

In our simulation, the overlay network nodes' failure rate is higher than 40%. When using CPB algorithm, when using replica number 1 and 4, the PSQ is 35% and 68%,

respectively. With ARK, when $r_{min} = 1,2,4$, the PSQ is 42%, 70%, 94%, respectively, which is much higher than CPB. This means that ARK meets our design goal 2.

5.3 Runtime Replica Number

Fig. 3 shows the total number of replicas needed, when, $r_{min} = 1,4$ respectively. Comparing that CPB occupies fixed number of replicas, typically, uses 40,000 replicas, ARK in same level uses about 40060, occupying 0.15% more storage. Overall, ARK requires less than additional $0.01K$ replicas, where K is the number of keywords, to trade for much more better performance of load balance and fault tolerance.



(a) $r_{min} = 1$ (b) $r_{min} = 4$

Fig. 3. Total Replica Number for ARK

6 Conclusion and Future Works

In this paper, we present ARK (Adaptive Replication of Keywords). We proved in this paper that ARK is near optimal in load balance and is competitive in replica management.

Comparing with the replication algorithms adapted in the famous DHT systems CAN, Pastry and Bamboo, simulations show that ARK has much better performance in load balance and in fault tolerance. Specially, comparing CPB-4 and ARK with $r_{min} = 4$, the load balance factor of ARK is less than 16,000, while CPB is about 200,000. When the network fail rate is as high as 40%, 94% queries can be successfully served, while CPB-4 only does 60%. As the price for these improvements, the replicas occupied by ARK are $0.01 \cdot K$ more than the CPB counterpart. The algorithm is applicable to any content storage scheme on top DHT based overlay networks.

We are working on build ARK on top of publicly available Bamboo-DHT system [8], which is described by [9]. Bamboo uses the Pastry geometry. Although Bamboo has its own replication algorithm, we changed the source slightly to disable it so as to use ARK.

There are much works left to adapt this theory to the real system. Concurrency issues must be taken into account when all the nodes run ARK at the same time. For example, locks must be put when updating replication status, and in add/remove actions.

We also plan to study optimistic replication strategy, which avoid the atom property of write to all replicas, thus improve the overall system performance. We are considering put this into our future replication algorithm.

References

1. Vijay Gopalakrishnan, Bujor Silaghi, Bobby Bhattacharjee, and Pete Keleher, Adaptive Replication in Peer-to-Peer Systems, In The 24th International Conference on Distributed Computing Systems, March 2004.
2. Yixiu Huang, Ouri Wolfson, A Competitive Dynamic Data Replication Algorithm, International Conference on Data Engineering, 1993.
3. Patrick Reynolds and Amin Vahdat, Efficient Peer-to-Peer Keyword Searching, Proceedings of International Middleware Conference, 2003.
4. Ian Foster and Adriana Iamnitchi, On Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing, IPTPS'03.
5. S.Ratnasamy, P.Francis, M.Handley, R.Karp, and S.Shenker, "A scalable content addressable network," in Proceedings of ACM SIGCOMM'2001, August 2001.
6. A.Rowstron and P.Druschel, "Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems," in Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001), Heidelberg, Germany, November 2001.
7. B.Y.Zhao, J.Kubiatowicz, and A.Joseph. "Tapestry: An infrastructure for fault-tolerant wide-area location and routing," Technical Report UCB/CSD-01-1141, University of California at Berkeley, Computer Science Department, 2001.
8. http://bamboo-dht.org
9. Sean Rhea, Dennis Geels, Timothy Roscoe, and John Kubiatowicz. Handling Churn in a DHT. in Proceedings of the USENIX Annual Technical Conference, June 2004
10. S. Eugene Ng and Hui Zhang. Predicting Internet Network Distance with Coordinates-Based Approaches. In IEEE INFOCOM'02, 2002.
11. Edith Cohen, Scott Shenker, Replication Strategies in Unstructured Peer-to-Peer Networks, SIGCOMM 2002
12. Yixiu Huang, Ouri Wolfson, A Competitive Dynamic Data Replication Algorithm, Proceedings of the Ninth International Conference on Data Engineering, 1993

Performance Modeling of Mobile Peer-to-Peer Systems

Lu Yan

Turku Centre for Computer Science (TUCS),
FIN-20520 Turku, Finland
lu.yan@ieee.org

Abstract. This paper is a performance study of peer-to-peer systems over mobile ad hoc networks. We present a performance model towards an in-depth understanding of mobile peer-to-peer systems. Our results provide potential useful guidelines for mobile operators, value-added service providers and application developers to design and dimension mobile peer-to-peer systems.

1 Introduction

Peer-to-Peer (P2P) computing is a networking and distributed computing paradigm which allows the sharing of computing resources and services by direct, symmetric interaction between computers. With the advance in mobile wireless communication technology and the increasing number of mobile users, peer-to-peer computing, in both academic research and industrial development, has recently begun to extend its scope to address problems relevant to mobile devices and wireless networks.

Mobile Ad hoc Networks (MANET) and P2P systems share a lot of key characteristics: self-organization and decentralization, and both need to solve the same fundamental problem: connectivity. It seems natural and attractive to deploy P2P systems over MANET due to this common nature, but the special characteristics of mobile environments and the diversity in wireless networks bring new challenges for research in P2P computing.

Though both P2P and MANET have recently becoming popular research areas due to the widely deployment of P2P applications over Internet and rapid progress of wireless communication, few research has been done for the convergence of the two overlay network technologies. In fact, the scenario of P2P systems over MANET seems feasible and promising, and possible applications for this scenario include car-to-car communication in a field-range MANET, an e-campus system for mobile e-learning applications in a campus-range MANET on top of IEEE 802.11, and a small applet running on mobile phones or PDAs enabling mobile subscribers exchange music, ring tones and video clips via Bluetooth, etc.

This paper is a performance study of peer-to-peer systems over mobile ad hoc networks. In the following section we will review previous work on P2P and MANET. In section 3, we present a performance model towards an in-depth understanding of mobile peer-to-peer systems. In section 4, we apply our analytical model to practical network design problems and analyze some important QoS issues. Finally, section 5 concludes the paper.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 133 – 142, 2005.

2 Background and State-of-the-Art

Since both P2P and MANET are becoming popular only in recent years, the research on P2P systems over MANET is still in its early stage. The first documented system is Proem [1], which is a P2P platform for developing mobile P2P applications, but it seems to be a rough one and only IEEE 802.11b in ad hoc mode is supported. 7DS [2] is another primitive attempt to enable P2P resource sharing and information dissemination in mobile environments, but it is rather a P2P architecture proposal than a practical application. In a recent paper [3], Passive Distributed Indexing was proposed for such kind of systems to improve the search efficiency of P2P systems over MANET, and in ORION [4], a Broadcast over Broadcast routing protocol was proposed. The above works were focused on either P2P architecture or routing schema design, but how efficient is the approach and what is the performance experienced by users are still in need of further investigation.

Previous work on performance study of P2P over MANET was mostly based on simulative approach and no concrete analytical mode was introduced. Performance issues of this kind of systems were first discussed in [5], but it simply shows the experiment results and no further analysis was presented. There is a survey of such kind of systems in [6] but no further conclusions were derived, and a sophisticated experiment and discussion on P2P communication in MANET can be found in [7]. Recently, B. Bakos etc. with Nokia Research analyzed a Gnutella-style protocol query engine on mobile networks with different topologies in [8], and T. Hossfeld etc. with Siemens Labs conducted a simulative performance evaluation of mobile P2P file-sharing in [9]. However, all above works fall into practical experience report category and no performance models are proposed.

We believe that to understand the performance issues of such kind of systems, rigorous analytical models are needed, which capture the relation between key system parameters and performance metrics. In the remaining sections we present our effort towards an in-depth understanding of mobile peer-to-peer systems, especially from users' point of view, e.g. a download performance model towards an in-depth understanding of mobile peer-to-peer systems. Our results provide potential useful guidelines to design and dimension mobile peer-to-peer systems.

3 Modeling Download Performance

The download performance modeling is a relatively new issue compared to the search performance modeling, which was already extensively studied in some P2P and MANET research [10, 11, 12]. In this section, we would like to present our efforts towards a performance model of downloading in such kind of systems, and thus answer the question: "what is the performance experience when many users try to retrieve data with parallel downloading scheme?"

3.1 Preliminary Assumptions

Though early research on modeling had mainly focused on routing performance and searching efficiency, recently, there were some works on modeling the download performance. The Markov chain approach was brought forth in [13] for a queue system

model and some measurement studies were mentioned in [14]; more recently, Stochastic fluid models are studied in [15, 16, 17], which provide a more intuitive and deterministic approach. Our work uses the same approach as [17, 18]; but taking the idea into mobile environments, more realistic scenarios and physical constraints should be introduced, and old notions should have new interpretations.

Since the introduction of Tornado Code [19, 20] has been a popular technique on recently parallel downloading systems, here we assume: (1) the parallel download process in our model is Tornado-like, which reduces the requirement for coordination and signalling. Due to the limited bandwidth of existing wireless networks (probably accompanied with expensive data transmission charge, e.g. cellular network), (2) it is reasonable to allow *pure downloader* (i.e. *leech*) exist in the system. Therefore, as illustrated in Figure 1, there are three types of peers in our model: (a) normal peer (i.e. *contributor*), which owns part of the file (i.e. ordinary downloader), but still allows others to download from itself. This type is the most common one and it actually constitutes the majority in our system. (b) pure downloader (i.e. *leech*), which just downloads but never uploads. The realistic implication of this type may be physically constrained mobile devices (e.g. cellular phones with limited bandwidth or associated with too expensive data transmission charge). (c) pure uploader (i.e. *seed*), which already have all pieces of the file but still stays in the system to allow others to download from itself. The realistic implication of that type may be content publishers (e.g. mobile operator's service point).

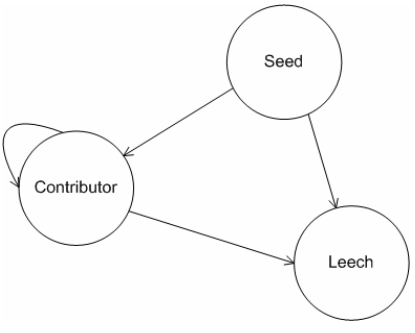


Fig. 1. Three Types of Peers

Although there is heterogeneity in realistic infrastructure [21], such as bandwidth, latency, availability, etc., here we make a trade-off between the simplicity of the model and its ability to capture all facets, and assume (3) all peers in our model have equal capacity (i.e. all peers have the same upload and download bandwidth). With the above assumptions and the parameters in Table 1, we can derive that at time t, there are $\beta\, x(t)$ leeches and $(1-\beta)\, x(t)$ contributors in our system.

3.2 The Model

The queue-like model of one peer in our system is illustrated in Figure 2. As noted here, during the download and upload process, it is also possible that peers will get

offline or abort the process, and in order to make the model simple, here we use abort rate ρ and leave rate κ to model these interrupted processes.

Table 1. Parameters Used in the Model

Parameter	Meaning
$x(t)$	Number of downloaders (i.e. contributors and leeches) at time t
β	Selfish rate (i.e. leech portion)
$y(t)$	Number of seeds at time t
λ	Arrival rate of new download request (Possion process)
μ	Upload bandwidth of each peer
τ	Download bandwidth of each peer
ρ	Abort rate of downloaders
κ	Leave rate of seeds

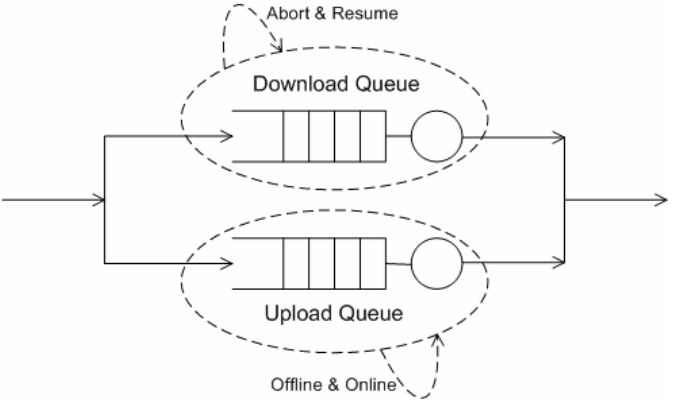


Fig. 2. Queue-Like Model of One Peer

In a P2P download and upload scheme, it is natural to expect more on the download side (i.e. this implies $\tau \geq \mu$); so taken the download bandwidth constraint into account, the total upload bandwidth should be $min(\mu((1-\beta)\,x(t) + y(t)), \tau\,x(t))$, and the arrival and departure rate of download request will be λ and $min(\mu((1-\beta)\,x(t) + y(t)), \tau\,x(t)) + \rho\,x(t)$ respectively, and the arrival and departure rate of upload request will be $min(\mu((1-\beta)\,x(t) + y(t)), \tau\,x(t))$ and $\kappa\,y(t)$. Thus the fluid model is derived as

$$\frac{d}{dt}x(t) = \lambda - \min\left[\mu\left[(1-\beta)x(t) + y(t), \tau x(t)\right]\right] - \rho x(t)$$

$$\frac{d}{dt}y(t) = \min\left[\mu\left[(1-\beta)x(t) + y(t), \tau x(t)\right]\right] - \kappa y(t)$$

In a steady state, the number of downloaders and seeds should be independent of time (i.e. $d(x(t))/dt) = d(y(t))/dt = 0$); and then if we define

$$\frac{1}{\iota} = \frac{1}{1-\beta}\cdot\left(\frac{1}{\mu} - \frac{1}{\kappa}\right)$$

where ι can be interpreted as *effective* upload bandwidth compared to *nominal* upload bandwidth μ (i.e. after considering the impact of leeches), the equations can be solved as

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} \dfrac{\lambda}{\tau\left(1+\dfrac{\rho}{\tau}\right)} \\ \dfrac{\lambda}{\kappa\left(1+\dfrac{\rho}{\tau}\right)} \end{pmatrix} \quad \text{when} \quad \frac{1}{\tau} \geq \frac{1}{\iota}$$

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} \dfrac{\lambda}{\iota\left(1+\dfrac{\rho}{\iota}\right)} \\ \dfrac{\lambda}{\kappa\left(1+\dfrac{\rho}{\iota}\right)} \end{pmatrix} \quad \text{when} \quad \frac{1}{\tau} < \frac{1}{\iota}$$

where the limited download bandwidth and limited upload bandwidth is the constraint respectively. Furthermore, if we define

$$\frac{1}{\phi} = \max\left(\frac{1}{\tau}, \frac{1}{\iota}\right)$$

where φ can be interpreted as *bottleneck* bandwidth intuitively, we obtain an elegant solution

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} \dfrac{\lambda}{\phi\left(1+\dfrac{\rho}{\phi}\right)} \\ \dfrac{\lambda}{\kappa\left(1+\dfrac{\rho}{\phi}\right)} \end{pmatrix}$$

Finally, we derive the average download time for a peer with Little's Law [22]

$$\text{Time} = \frac{1}{\phi + \rho} \qquad \text{where} \qquad \frac{1}{\phi} = \max\left(\frac{1}{\tau}, \frac{1}{\iota}\right)$$

4 Performance Analysis with the Model

In the model presented in the previous section, it is clear that different settings of β, μ, τ, ρ and κ will lead to different performance; so in this section we will use our analysis model to provide some insights in the network.

4.1 Selfish Peers

For a fixed set of network parameters, we first study the impact of β on the network performance. The realistic interpretation of β is interesting, which is somehow related to peer strategy and incentive mechanism (i.e. *selfish* peers or *leeches*).

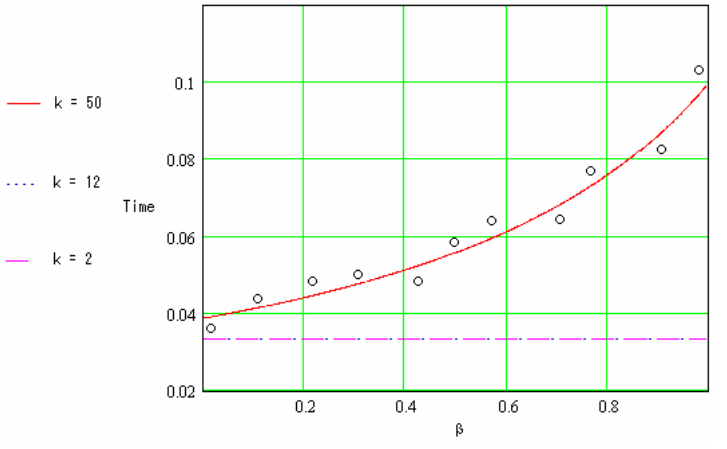


Fig. 3. Impact of β on Network Performance

The network parameters we have chosen are: μ = 12kbps, τ = 20kbps, ρ = 10kbps, κ_0 = 50kbps, κ_1 = 12kbps, κ_2 = 2kbps. In this scenario, we consider the effect of *selfish* peers. Intuitively, the existing *leeches* will degrade the system performance because they just download from others and never upload. The red curve in Figure 3 for κ_0 = 50kbps justifies our intuition; besides the observation that *Time* is a non-decreasing function of β, we can also find the upper bound and lower bound of *Time* if we consider two extreme cases: β = 1 (i.e. all downloaders are *selfish* and no one uploads to others) and β = 0 (i.e. there is no *leeches* in the system). At this point, we are all happy with our intuition; but if we change the value of κ into κ_1 = 12kbps and κ_2 = 2kbps, something strange happens. As shown in Figure 3 as two overlapped horizontal lines, the network performance is constant, independent of β. We briefly comment on this situation: recall the *bottleneck* bandwidth definition in the previous section, it actually means the downloading bandwidth is the bottleneck since $\mu \geq \kappa$; in such a situation, the *leeches* make no harm to the system since the whole system performance is constrained by the limited download speed (i.e. selfishness is *not* always harmful). From this phenomenon, we argue that it is reasonable to introduce *leeches* into our model as in our preliminary assumptions, and actually there are lots of *leeches* existed in realistic systems. In other words, *what is rational is real and what is real is rational*.[1]

[1] Taken from Hegel's famous dictum *Das Wirkliche sei vernuenftig und das Vernuenfitige wirklich*.

4.2 Download Bandwidth's Role

In the previous subsection, we have seen the download bandwidth's impact on the system performance. Intuitively, increasing the download bandwidth will lead to a shorter downloading time, as often observed in our daily experiences; but is this common sense always true? Now we study the impact of τ on the system performance (i.e. download bandwidth's role).

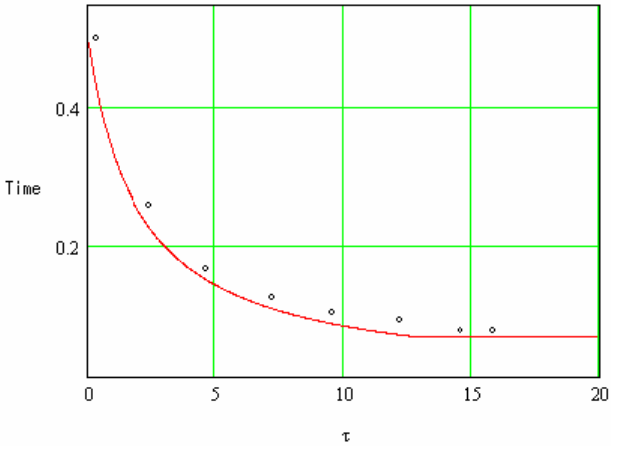


Fig. 4. Impact of τ on Network Performance

The network parameters we have chosen are: $\beta = 0.2$, $\mu = 12$kbps, $\rho = 2$kbps, $\kappa = 50$kbps. Shown as the red curve in Figure 4, *Time* is a non-increasing function of τ. Besides, we can also derive the upper bound and lower bound of *Time* if we set $\tau = 0$ (i.e. the download channel is actually blocked) and $\tau = \infty$ (i.e. the download bandwidth is *much* higher than upload bandwidth) respectively.

The left half part of the curve justifies our intuition perfectly, but the right half seems to yaw from the common sense. The key to the phenomenon is still *bottleneck* bandwidth: initially, when τ increases, *Time* decreases accordingly because download bandwidth is the bottleneck now; however, once τ becomes big enough, increasing τ will not decrease *Time* any more, because the download bandwidth is no longer the bottleneck of the system performance. In fact, if we consider the impact of μ on network performance (i.e. upload bandwidth's role), we will get a similar curve. From these phenomena, we argue that there are not always performance gains with increased download bandwidth, and the key to network performance gains is to keep a good balance of download bandwidth and upload bandwidth, and actually to increase bottleneck bandwidth. In other words, *every coin has two sides.*[2]

4.3 Importance of Seeds

The *seeds* are a special kind of peers, which upload but don't download. Compared to *leeches*, seeds can be deemed as *selfless* peers. Intuitively, it is very important to have

[2] Ancient proverb.

seeds in the system; and in this subsection, we study the impact of κ on the system performance (i.e. seeds' contribution).

The network parameters we have chosen are: $\beta = 0.2$, $\mu = 2$kbps, $\rho = 1$kbps, $\tau_0 = 1$kbps, $\tau_1 = 2$kbps, $\tau_2 = 6$kbps, $\tau_3 = 20$kbps. With the curves shown in Figure 5, we are now not surprised to see the divisions of these curves and their singular points, because we already know their roots in the *bottleneck* bandwidth concept. Here we just briefly comment on the situation $\tau_2 = 6$kbps because this speed seems to coincide with the practical speed of our daily cellular networks (e.g. GPRS): the ideal scenario is $\kappa = 0$ (i.e. all seeds are persistent in the network), where the lower bound of *Time* resides. As κ increases, initially, the slight loss of seeds doesn't degrade the system performance since the system is download bandwidth constrained; however, once κ is big enough, the system turns into upload bandwidth constrained, and the system performance degrades sharply with the loss of seeds; this also explains the singular point in the curve.

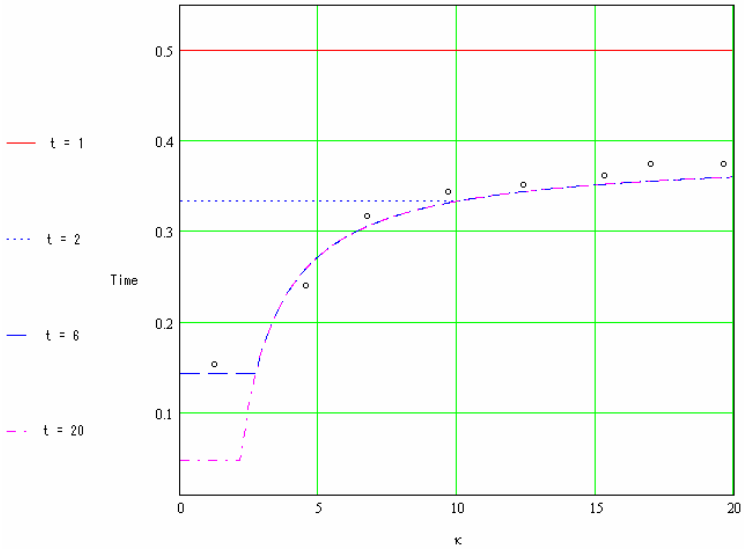


Fig. 5. Impact of κ on Network Performance

The realistic interpretation of *seeds* is service points or completed downloaders (but not all completed downloaders become seeds due to the existence of *leeches*), and the realistic meaning of the phenomenon is: it would be an effective way for mobile operators to improve QoS in such kind of systems via providing more service points.

5 Concluding Remarks

In this paper, we studied the performance issues of peer-to-peer systems over mobile ad hoc networks. After characterizing the variability of the system by taking some preliminary assumptions, we then present a performance model towards an in-depth

understanding of mobile peer-to-peer systems. We also briefly discussed three analytical examples on apply this model to capture the behavior of the system in steady states.

In order to make the paper concise, we didn't use the model to analyze the system in inequilibrious states, though it is not hard to simulate these cases with the given fluid model. Our results provide potential useful guidelines for mobile operators, value-added service providers and application developers to design and dimension mobile peer-to-peer systems, and as a foundation for our long term goals [23, 24].

References

[1] G. Kortuem, J. Schneider, D. Preuitt, T. G. C. Thompson, S. Fickas, Z. Segall. When Peer-to-Peer comes Face-to-Face: Collaborative Peer-to-Peer Computing in Mobile Ad hoc Networks. In *Proc. 1st International Conference on Peer-to-Peer Computing (P2P 2001)*, Linkoping, Sweden, August 2001.

[2] M. Papadopouli and H. Schulzrinne. A Performance Analysis of 7DS a Peer-to-Peer Data Dissemination and Prefetching Tool for Mobile Users. In *Advances in wired and wireless communications, IEEE Sarnoff Symposium Digest*, 2001, Ewing, NJ.

[3] C. Lindemann and O. Waldhorst. A Distributed Search Service for Peer-to-Peer File Sharing in Mobile Applications. In *Proc. 2nd IEEE Conf. on Peer-to-Peer Computing (P2P 2002)*, 2002.

[4] A. Klemm, Ch. Lindemann, and O. Waldhorst. A Special-Purpose Peer-to-Peer File Sharing System for Mobile Ad Hoc Networks. In *Proc. IEEE Vehicular Technology Conf.*, Orlando, FL, October 2003.

[5] S. K. Goel, M. Singh, D. Xu. Efficient Peer-to-Peer Data Dissemination in Mobile Ad-Hoc Networks. In *Proc. International Conference on Parallel Processing (ICPPW '02)*, IEEE Computer Society, 2002.

[6] G. Ding, B. Bhargava. Peer-to-peer File-sharing over Mobile Ad hoc Networks. In *Proc. 2nd IEEE Conf. on Pervasive Computing and Communications Workshops*. Orlando, Florida, 2004.

[7] H.Y. Hsieh and R. Sivakumar. On Using Peer-to-Peer Communication in Cellular Wireless Data Networks. In *IEEE Transaction on Mobile Computing*, vol. 3, no. 1, January-March 2004.

[8] B. Bakos, G. Csucs, L. Farkas, J. K. Nurminen. Peer-to-peer protocol evaluation in topologies resembling wireless networks. An Experiment with Gnutella Query Engine. In *Proc. International Conference on Networks*, Sydney, Oct., 2003.

[9] T. Hossfeld, K. Tutschku, F. U. Andersen, H. Meer, J. Oberender. Simulative Performance Evaluation of a Mobile Peer-to-Peer File-Sharing System. *Research Report 345*, University of Wurzburg, Nov. 2004.

[10] R. Schollmeier and I. Gruber. Routing in Peer-to-Peer and Mobile Ad Hoc Networks. A Comparison. In *Proc. International Workshop on Peer-to-Peer Computing*, Pisa, Italy, 2002.

[11] P. Reynolds and A. Vahdat. Efficient Peer-to-Peer Keyword Searching. In *Proc. Middleware*, 2003.

[12] S. Corson and J. Macker. Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. *RFC 2501*, Jan. 1999.

[13] Z. Ge, D. Figueiredo, S. Jaiswal, J. F. Kurose, D. Towsley. Modeling peer-to-peer file sharing systems. In *Proc. IEEE Infocom*, 2003.

[14] K. P. Gummadi, R. J. Dunn, S. Saroiu, S. D. Gribble, H. M. Levy and J. Zahorjan. Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. In *Proc. 19th ACM symposium on Operating systems principles*, 2003.

[15] F. Clevenot and P. Nain. A Simple Fluid Model for the Analysis of the Squirrel Peer-to-Peer Caching System. In *Proc. IEEE Infocom*, 2004.

[16] X. Yang and G. Veciana. Service Capacity of Peer to Peer Networks. In *Proc. IEEE Infocom*, 2004.

[17] D. Qiu and R. Srikant. Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks. In *Proc. ACM SIGCOMM*, 2004.

[18] F. L. Piccolo, G. Neglia. The Effect of Heterogeneous Link Capacities in BitTorrent-Like File Sharing Systems. In *Proc. Intl. Workshop on Hot Topics in Peer-to-Peer Systems (HOT-P2P'04)*, Oct, 2004.

[19] J. Byers, M. Luby, M. Mitzenmacher. Accessing Multiple Mirror Sites in Parallel: Using Tornado Codes to Speed Up Downloads. In *Proc. IEEE Infocom*, 1999.

[20] J. W. Byers, J. Considine, M. Mitzenmacher and S. Rost. Informed Content Delivery Across Adaptive Overlay Networks. In *Proc. ACM SIGCOMM*, 2002.

[21] S. Saroiu, P.K. Gummadi, S.D Gribble. A Measurement Study of Peer-to-Peer File Sharing Systems. In *Proc. Multimedia Computing and Networking (MMCN'02)*, 2002.

[22] P. J. Denning and J. P. Buzen. The Operational Analysis of Queueing Network Models. In *ACM Computer Survey*, 1978.

[23] L. Yan, K. Sere, X. Zhou, and J. Pang. Towards an Integrated Architecture for Peer-to-Peer and Ad Hoc Overlay Network Applications. In *Proc. 10th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS 2004)*, May 2004.

[24] L. Yan. MIN: Middleware for Network-Centric Ubiquitous Systems. In *IEEE Pervasive Computing*, Vol. 3, No. 3, 2004.

A Random Walk Based Anonymous Peer-to-Peer Protocol Design

Jinsong Han[1], Yunhao Liu[1], Li Lu[2], Lei Hu[2], and Abhishek Patil[3]

[1] Dept. of Computer Science, Hong Kong University of Science & Technology, Clear Water Bay, Kowloon, Hong Kong
{jasonhan, liu}@cs.ust.hk
[2] The State Key Laboratory of Information Security, Chinese Academy of Sciences, Beijing, China
luli@mails.gscas.ac.cn, hu@is.ac.cn
[3] Dept. of Computer Science and Engineering, Michigan State University, East Lansing, MI, 48823, USA
patilabh@msu.edu

Abstract. Anonymity has been one of the most challenging issues in Ad Hoc environment such as P2P systems. In this paper, we propose an anonymous protocol called **R**andom **W**alk based **A**nonymous **P**rotocol (RWAP), in decentralized P2P systems. We evaluate RWAP by comprehensive trace driven simulations. Results show that RWAP significantly reduces traffic cost and encryption overhead compared with existing approaches.

1 Introduction

Thanks to its high scalability and easy implementation, Peer-to-Peer (P2P)[3, 6, 7] becomes a killer application in distributed environments. P2P applications have been rapidly developed from the very beginning of this millenary. Researchers are strongly encouraged to dig into this up-and-coming approach further. In short, P2P architecture has predominant features including scalability, redundancy, flexibility, autonomy, and anonymity compared with traditional Client-Server models [9, 10, 13, 14, 16]. Although, anonymity [1], which is a concern of user's privacy, has been given a lot of attention, it has not been fully addressed. The primary P2P systems merely provide incomplete anonymity designs. With tremendous increase in users, current P2P systems face urgent needs for both privacy and security.

1.1 Anonymity Categories

Previous anonymity studies fall into three categories: resistant-censorship (or publishing anonymity); initiator or responder anonymity; and mutual anonymity (giving both the initiator and responder anonymity). This paper focuses on the third category.

Usually initiator or responder anonymity is only a one-way model, in which a system can provide an initiator an anonymous transmission from a sender to receivers, responder anonymity vice versa. Aside from them, mutual anonymity is a more

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 143 – 152, 2005.

completed situation for the privacy requirement from users. Strictly defined, mutual anonymity includes three aspects: an anonymous initiator, an anonymous responder, and the anonymous communication between these two units.

1.2 P2P Anonymity

Although P2P file-sharing paradigms have many advantages over traditional client-server approaches, its open and free join-leave policy leads to a complete lack of protection for system participators, which exposes them to attacks from malicious peers. A sharply increasing amount of P2P users exaggerates the probability of suffering this threat. As a basic design purpose, user's privacy is an important issue over the P2P systems. However, most P2P prototypes are vulnerable under malicious attacks. We argue that P2P's weak anonymity feature cannot guarantee safety for their good users without suffering attacks from collaborating ones based on the following observations. First, some malicious peers can acquire information easily by monitoring packet flows, distinguishing packet types, (e.g., the QueryHits message [4] sent from responder), and analyzing the TTL value of these queries. Consequently, initiators and responders are completely exposed to their neighbors and P2P systems fail to provide anonymity in each peer's local environment. Second, in an untrustworthy public network, when the files or messages are transferred in a plain text, their contents also help the attackers on the path to collaborate and guess the identities of the communication parties. Therefore, current P2P systems cannot provide anonymity guarantees. In this paper, we propose a mutual anonymity protocol in decentralized P2P systems, called **R**andom **W**alk based **A**nonymous **P**rotocol (RWAP). RWAP allows users achieve a mutual anonymity with low traffic and cryptographic overhead compared with previous mutual anonymity protocols.

The rest of this paper is organized as follows. In Section 2, we introduce the background knowledge of random walk. In Section 3, we propose the RWAP approach. In Section 4, we discuss anonymity degree of RWAP. We evaluate RWAP by comprehensive trace driven simulations in Section 5. We present related works in Section 6 and conclude this work in the last section.

2 Random Walk

Before we present the RWAP design, we give a short introduction on random walk, which recently acts as one of the basic algorithms in P2P protocols to deal with content location and topology maintenance issues. As an optional search model of the content location, it has been discussed in many previous works [12, 15, 17].

In most of current unstructured P2P systems, such as Gnutella and KaZaA, peers use flooding-based search to locate desired contents. Blind flooding causes large amount of unnecessary traffic, particularly in densely connected graphs. To keep the system scalable, researchers made considerable efforts in order to reduce traffic cost caused by search operations. Random Walk is such a substitute, in which a query message is forwarded to one or several randomly chosen neighbors at each hop until it

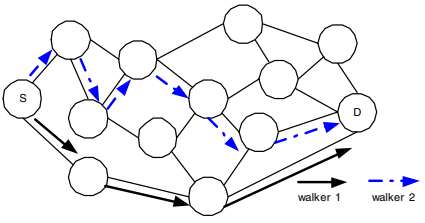


Fig. 1. Basic Random Walk

reaches a volunteer resource provider within a limited hop number. An example is shown in Fig. 1. The message here is called a walker. The original version of random walk has only one walker. Compared with flooding search, one walker technique significantly improves search efficiency. However, users might suffer a long wait time before it receives a response.

3 RWAP

In this section, we present our **R**andom **W**alk based **A**nonymity **P**rotocol. RWAP is designed for decentralized file sharing system like Gnutella–KaZaA. Without modifying the basic architecture, we revise Gnutella 0.6 [4] protocol by adding a local storage space to each servant. Packets can be temporarily stored in each reached peer.

As we mentioned in Section 2, a maximum hop number is set to a peer-count field of each walker. When the message is randomly walking, this value is decremented after each hop. The message continues walking until this field equal to zero. As a result, the length of random walk path is constrained.

3.1 System Architecture and Initialization

When a peer joins our system, it first contacts the bootstrap server to obtain a peering node list. With the help of this list, it can construct several neighboring relationships with chosen peers.

Before presenting our protocol, we define some notations used in this paper. We assign F as the requesting file and f as its query id; we use S denote the initiator, and R denote the query responder. P_i denotes a middle peer in the P2P system, and sq denotes a sequence number to link query message and key message. We use $A \rightarrow B$: M to represent A sending a message M to B. We define $E_K(\mathrm{M})$ as encrypting the message M with a symmetric key K, and $D_K(C)$ as decrypting a cipher C with the symmetric key K.

3.2 Anonymous Query

For an initiator S, before searching some query, it first creates symmetric session Key K and a pair of RSA keys K_{S+} (public key of S) and K_{S-} (private key of S). Here we choose 128bit-AES [2] as the symmetric encryption algorithm. S encrypts f with K: $C = E_K(f)$. For distinguishing same messages coming from an identical source, S

brands this cipher message with a sequence number *SN*. To construct an anonymous return path, *S* also generates a reversed onion path *RP*:

$$RP = \{X, \{Y, \{...\{S, \{SN\}K_{S+}\}, ...\}K_{Y+}\}K_{X+}\}$$

Before starting query, *S* encapsulates two message packets: One is the query message which contains *C*, and the other is the key message includes *K*, K_{S+} and *RP*. We let *C* and *K* to represent them respectively.

When *C* and *K* are ready, *S* randomly chooses one neighbor to forward cipher *C*. After a short waiting time (more like the SIFS of CSMA/CA in IEEE 802.11), *S* chooses another neighbor to send *K*. This key message will also be embedded with same sequence number *SN* as the cipher message.

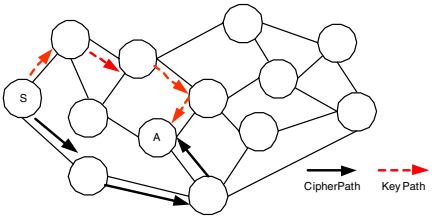


Fig. 2. RWAP anonymous query

Each peer receiving a cipher message saves a copy in its local storage and randomly forwards it to one of its neighboring peers. For those who receive the key message, they check in their stored cipher records to find those ones with the same *SN* as in this message. If there is any record, for example, a *C*' matching the key message, this peer try to recover *M* from *C*' with the *K*: *M*' = $D_K(C')$. If M' is a meaningful query, this peer has successfully recovered an original query message. Otherwise, it stops the key message randomly walking. Figure 2 presents an example of such a procedure.

3.3 Finding Responders and Delivering Files Back

A voluntary middle peer, who recovers a query message *M*, selects itself as a query agent for an unknown peer. It then starts a normal flooding search. To help the potential responder sending files back, it attaches the reversed path *RP* and K_{S+} in each query message.

We suppose that this query message reaches a certain volunteering responder *R* that occupies such a file. Before it sends the file to *S*, it should do some following preparations:

R generates a session *K*', and encrypts file *F*: $C_F = E_{K'}(F)$. Then *R* encrypts session key *K*' to $C_{K'} = E_{K_{S+}}(K')$. After this digital envelope being sealed, *R* sends $C_F + C_K + RP'$ to the first relay peer *X*.

This combined chunk data will eventually flow back to *S* along the *RP* path. Thus, *S* can first recover the session Key *K*' using its private key and then obtain the desired file by decrypting C_F. We illustrate this procedure in Fig. 3.

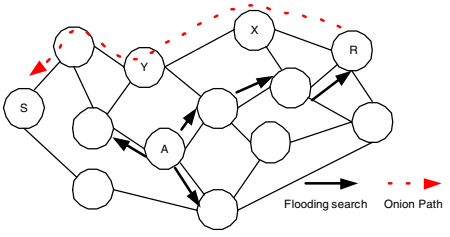


Fig. 3. Searching and driving files back

In a practical environment, to decrease the delay, the initiator can increase the number of cipher walkers. That is, the initiator peer sends out multiple cipher walkers, and each walker takes its own random walk.

3.4 Mimic Traffic

In our basic algorithm presented in the previous section, peers are vulnerable to the attacks from local eavesdropper. Those malicious peers can observe all communications sent from and to an individual user. Without any protection, when the initiating peer issues a query, it is immediately exposed to the local eavesdropper. Since all the input and output traffics are under their monitoring, local eavesdroppers can easily distinguish an initiator or a responder by detecting the non-corresponding input or output.

To prevent against the local malicious peers listening, each peer periodically sends out noise cipher or key messages as mimic traffics. To confuse attackers, we use the average interval time between two consecutive normal queries as the interval time of mimic traffic queries of each peer. In our experiments, the basic interval between two consecutive queries is about 10 seconds. In RWAP design, the bootstrap server can provide this default parameter to each joining peer. And every peer can dynamically adjust this setting according its query load. At one time, an idle peer should send out just one noise packet. If this peer wants to obtain a high anonymity, it can increase the frequency of sending noise packets. Since each message uses a hop-count to control its coverage range, the traffic overhead could be limited salable. Meanwhile, such mimic traffic would cause some extra decryption operations in P2P systems. Since the recovering action simply uses the AES, which is a fast symmetric encryption algorithm, the total additional decryption overhead is also very little.

4 Anonymity Degree

In this section, we first analyze anonymity degree of our protocol.

Initiator Anonymity: We define the anonymity degree of any initiator as the possibility of make a correct guess to identify this sender by a receiver or other observers. It is obvious that a receiver, which can only monitor its own links, cannot recognize an original initiator, since the initiator information is not included in the packet. And we call such a receiver a passive guesser. With the help of other collaborating

adversary, a receiver is able to obtain a list of the possible peers which could have sent the packet. We call such a peer correspondingly an active guesser.

Theoretically the degree of sender anonymity is *1/(n-1)* from the passive peer's point of view, where *n* is the number of P2P peering nodes in the system. For those active receivers, they can make some collusion with other adversary.

Theorem 1. The probability that collaborating adversaries correctly guess an initiator is less than $2(a+1)/n$, where n is the number of total peers and a is the number of attackers in the system.

Proof: Supposing the A_k and A_C are the attackers in the key and cipher message random walk paths respectively. Let H_k, $(k \geq 1)$, denote the event that the first adversary on the path occupies the *k*th position on the path, where the initiator itself occupies the 0th position. We denote *I* an event that initiator is the direct predecessor of an attacker. We also define $H_{m+} = H_m \vee H_{m+1} \vee H_{m+2} \vee \ldots$. Obviously $H_1 = I$. Therefore we will calculate $\Pr[I \mid H_{1+}]$, a probability that attackers guess correctly the initiator identity.

For adversaries A_k: $\Pr[H_i] = (\frac{n-a}{n})^{i-1} \frac{a}{n}$.

$$\Pr[H_{2+}] = \frac{a}{n} \sum_{k=1}^{\infty} (\frac{n-a}{n})^k = \frac{a}{n} (\frac{\frac{n-a}{n}}{1-\frac{n-a}{n}}) = \frac{n-a}{n}$$

$$\Pr[H_{1+}] = \frac{a}{n} \sum_{k=0}^{\infty} (\frac{n-a}{n})^k = \frac{a}{n} (\frac{1}{1-\frac{n-a}{n}}) = 1$$

We also get $\Pr[H_1] = \frac{a}{n}$, $\Pr[I \mid H_1] = 1$, $\Pr[I \mid H_{2+}] = \frac{1}{n-a}$. The last one is coming from the observation that if the first adversary just occupies the second or posterior position, it can only randomly guess the initiator identity with a probability of $\frac{1}{n-a}$. Now, we can get the $\Pr[I]$ as

$$\Pr[I] = \Pr[H_1]\Pr[I \mid H_1] + \Pr[H_{2+}]\Pr[I \mid H_{2+}] = \frac{a+1}{n},$$

then we get

$$\Pr[I \mid H_{1+}] = \frac{\Pr[I \wedge H_{1+}]}{\Pr[H_{1+}]} \leq \frac{\Pr[I]}{\Pr[H_{1+}]} = \frac{a+1}{n}.$$

Analogously, the probability that A_c guess the initiator correctly is $\frac{a+1}{n}$. So the probability that the adversaries (A_k and A_c) guess the initiator correctly is at most $\frac{2(a+1)}{n}$.

Responder Anonymity: With the onion path, the expected number of path reformations required for c attackers to determine the initiator out of n participants is $O((n/c)^l)$, where l is the length of the path between the initiator and responder [19].

5 Performance Evaluation

We use DSS Clip2 [5] trace to simulate P2P topologies. The results are consistent with different traces and here we show two of them: May 17, 2001 and May 31, 2001, denoted as Trace 1 and Trace 2, respectively.

In our simulation, we implement random walk used in a decentralized P2P network by conducting Depth First travels algorithm from a specific node. A query operation is simulated by randomly choosing a peer as the sender, and a keyword according to Zipf [8] distribution. In each run, 2,000 search operations are simulated.

We also run the crypto software kits on a desktop PC with PIV 3.2GHz CPU, 512MBytes memory, 40G hard disk, and 10/100M Ethernet card. We found that the average 1024-bit RSA decryption rates are 210 per second. The encryption rates are about 5600 per second. In our simulation, we choose the 1024-bit RSA as the crypto process in the onion path and use these average values as the reference value of the 1024-bit RSA performance. Meanwhile, our statistics show that the encryption speed for 128 bit AES is averagely 63.230 Mbps. Therefore, we choose this value as AES speed.

We also simulate the dynamic changes in P2P systems by assigning a lifetime to every peer. The average of this value is 10 minutes. The lifetime decreases by one after each passing second. When a peer's lifetime reaches zero, it leaves in the next second. After a certain number of peers leave the network, we then randomly pick the same number of peers from the physical network to join the P2P overlay.

In our experiment, we define the unit of traffic as a link between two neighboring peers. The traffic cost added by RWAP is mainly caused by two kinds of actions: mimic traffic and random walk of massage. The latter has a really trivial additional overhead. The upper bound of traffic caused by an anonymous query is $L_K + L_C$, where L_C and L_K are the length of cipher message and key massage walk path respectively. Meanwhile we show the mimic traffic in Figures 4 and 5. In our experiment, we increase the average interval for each peer, and observe the diminished change of average mimic traffic. We find that the optimal average interval of sending noise packet is about 20 second. Under this situation, the total additional mimic traffic is from 16000 to 16550, and is just about half of a single original flooding search.

Correspondingly, the relevant AES encryptions and decryptions overhead are also trifling. We show the changes in different average query time intervals in Fig. 6 and 7. Since the file is transferring to initiator along an onion path, the average RSA encryptions are employed ($L_O + 1$) times per query as well as RSA decryptions, where L_O is the onion path length.

Although RWAP make a scalable traffic increase to P2P systems, we also observed the hit rate of key message. We define a hit happening when a cipher message is successfully recovered when a peer receives a key message. Figures 8 and 9 show that if the path length, which equals the hop-count, is greater than 3, the average hit rate reaches approximately 60%. Moreover, if we extend the path length to 7, the average hit rate is much closed to 100%.

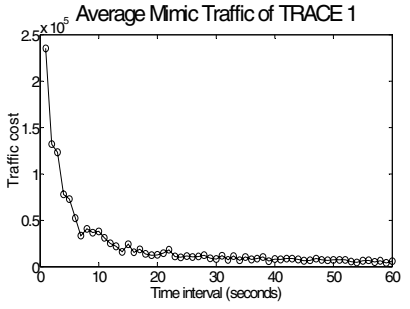


Fig. 4. Mimic Traffic Cost of TRACE 1

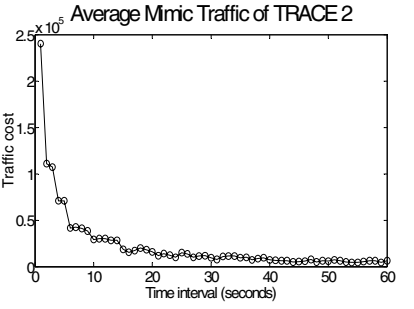


Fig. 5. Mimic Traffic Cost of TRACE 2

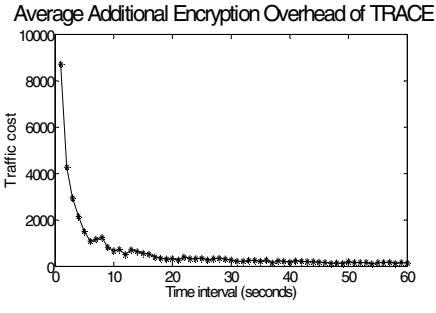


Fig. 6. Mimic Encryption of TRACE 1

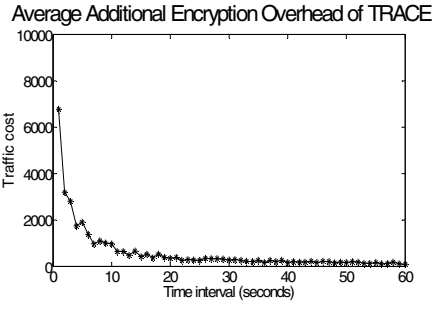


Fig. 7. Mimic Encryption of TRACE 2

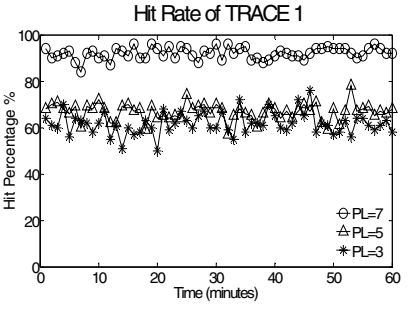


Fig. 8. Query Hit Rate of TRACE 1

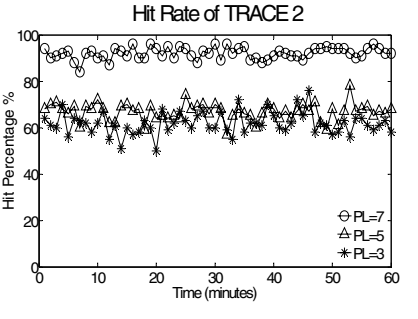


Fig. 9. Query Hit Rate of TRACE 2

6 Related Work

Anonymous transmission includes initiator, responder, and mutual anonymity as we mentioned above, as well as anonymous data transferring. In MorphMix [19] and Tor [11], initiator predetermines an anonymous path to hide original identification information, called as path-based approaches. Analogously, Crowds [18] and Hordes [21] complete the path selection through middle peers.

Some studies, such as Anonymous Peer-to-Peer File Sharing (APFS) [20], and Shortcut-responding Protocol [22], have been proposed to provide mutual anonymity in P2P systems.

APFS is more like a hybrid structured anonymous system. Some coordinator nodes act as a superior peer and maintain a list of all the peering nodes. Some peers in these lists volunteer to issue queries for others. All communications of this framework are based on the onion path to guarantee the anonymity and hence no centralized authority exists in this system.

In Shortcut-responding Protocol [22], the initiator establishes an reply block, which includes a onion-based reversed path, before sending each query. Each peer that receives the query determines whether or not devotes itself as a query agent peer in a probability of *pv*. If a peer acts as the query agent for the initiator, it floods this query into P2P systems. Upon requests, a responder builds another onion path to anonymously send the file to the query agent peer. The query agent peer delivers the file along the reversed path to the initiator and drops duplicate copies. Because of reducing the length of the return path, this approach achieves a shorter response time than other anonymous protocols.

7 Conclusion

In this paper, we propose a mutual anonymity protocol **R**andom **W**alker based **A**nonymous **P**rotocol (RWAP) in decentralized P2P systems. By employing random walk concept, RWAP allow peers issue queries and deliver requested files anonymously. RWAP achieves mutual anonymity in P2P systems with a satisfied degree of anonymity and low traffic overhead. We also evaluate RWAP by trace-driven simulations.

In future work, we will improve RWAP model in decentralized P2P systems by employing Bias *k*-random walkers schemes to acquire higher efficiency and accuracy. We will also implement RWAP prototype over a real decentralized P2P system.

References

1. Anonymity.- http://freehaven.net/anonbib/topic.html
2. FIPS-197: Advanced Encryption Standard, National Institute of Standards and Technology (NIST).- http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
3. Gnutella.- http://gnutella.wego.com/
4. Gnutella Protocol Development.- http://rfc-gnutella.sourceforge.net/index.html
5. The Gnutella Protocol Spedification v0.4.- http ://www.clip2.com/GnutellaProtocol04.pdf
6. KaZaA.- http://www.kazaa.com
7. Napster.- http://www.napster.com
8. L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker.- Web Caching and Zipf-like Distributions: Evidence and Implications. In Proceedings of IEEE INFOCOM, 1999
9. Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker.- Making Gnutella-like P2P Systems Scalable. In Proceedings of ACM SIGCOMM, 2003
10. E. Cohen and S. Shenker.- Replication Strategies in Unstructured Peer-to-peer Networks. In Proceedings of ACM SIGCOMM, 2002

11. R. Dingledine, N. Mathewson, and P. Syverson.- Tor: The Second-Generation Onion Router. In Proceedings of the 13th USENIX Security Symposium, 2004
12. C. Gkantsidis, M. Mihail, and A. Saberi.- Random Walks in Peer-to-Peer Networks. In Proceedings of IEEE INFOCOM, 2004
13. K. P. Gummadi, R. J. Dunn, S. Saroiu, S. D. Gribble, H. M. Levy, and J. Zahorjan.- Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload. In Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP), 2003
14. W. Jia, D. Xuan, W. Tu, L. Lin, and W. Zhao.- Distributed Admission Control for Anycast Flows. IEEE Transactions on Parallel and Distributed Systems (TPDS), 2004
15. C. Law and K.-Y. Siu.- Distributed Construction of Random Expander Networks. In Proceedings of IEEE INFOCOM, 2003
16. Y. Liu, X. Liu, L. Xiao, L. M. Ni, and X. Zhang.- Location-Aware Topology Matching in Unstructured P2P Systems. In Proceedings of IEEE INFOCOM, 2004
17. Q. Lv, P. Cao, Edith, Cohen, K. Li, and S. Shenker.- Search and Replication in Unstructured Peer-to-Peer Networks. In Proceedings of International Conference on Supercomputing (ICS'02) ACM, 2002
18. M. K. Reiter and A. D. Rubin.- Crowds: Anonymity for Web Transactions. ACM Transactions on Information and System Security, 1998
19. Rennhard and B. Plattner.- Introducing MorphMix: peer-to-peer based anonymous Internet usage with collusion detection. In Proceedings of ACM workshop on Privacy in the Electronic Society, 2002
20. V. Scarlata, B. N. Levine, and C. Shields.- Responder Anonymity and Anonymous Peer-to-Peer File Sharing. In Proceedings of the 9th International Conference of Network Protocol (ICNP), 2001
21. C. Shields and B. N. Levine.- A Protocol for Anonymous Communication over the Internet. In Proceedings of 7th ACM Conference on Computer and Communication Security (ACM CCS), 2000
22. L. Xiao, Z. Xu, and X. Zhang.- Low-cost and Reliable Mutual Anonymity Protocols in Peer-to-Peer Networks. IEEE Transactions on Parallel and Distributed Systems, 2003

A System for Power-Aware Agent-Based Intrusion Detection (SPAID) in Wireless Ad Hoc Networks

T. Srinivasan[1], Jayesh Seshadri[2], J.B. Siddharth Jonathan[3], and Arvind Chandrasekhar[4]

Department of Computer Science and Engineering,
Sri Venkateswara College of Engineering,
Sriperumbudur, India 602105
[1] tsrini@svce.ac.in
[2] jayeshs2000@yahoo.co.in
[3] jonathansiddharth@yahoo.co.in
[4] arvindcac@gmail.com

Abstract. In this paper, we propose a distributed hierarchical intrusion detection system for ad hoc wireless networks, based on a power level metric for potential ad hoc hosts, which is used to determine the duration for which a particular node can support a network monitoring node. We propose an iterative power-aware power-optimal solution to identifying nodes for distributed agent-based intrusion detection. The advantages that our approach entails are several, not least of which is the inherent flexibility SPAID provides. We consider minimally mobile networks in this paper, and considerations apt for mobile ad hoc networks and issues related to dynamism are earmarked for future research. Comprehensive simulations were carried out to analyze and clearly delineate the variations in performance with changing density of wireless networks, and the effect of parametric variations such as hop-radius.

1 Introduction

An intrusion is defined as "any set of actions that attempt to compromise the integrity, confidentiality, or availability of a resource" [1]. Several algorithms have been published in recent years to deal with intrusion detection, which incorporated the essence of the *wireless* nature of wireless ad hoc networks. Intrusions in wireless networks amount to *interception, interruption, or fabrication* of data transmitted across nodes. Intrusion into a wireless network is possible if an intruder node attempts to access unauthorized data. *Ad hoc networks* are particularly prone to such dangers, considering the dynamic and geographically distributed nature of the nodes. Ad hoc networks can hence be classified on the basis of their dynamism as minimally mobile or highly mobile. In this paper, we primarily focus on minimally mobile networks, where the power levels of the nodes are absolutely critical in determining the kinds of processes they can run in a sustainable fashion. We briefly discuss PLANE, a metric we suggest for comparing power levels across nodes for running agent-based network monitoring processes.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 153 – 162, 2005.

Agent-based systems are inherently reconfigurable, since the agents can easily be migrated to other hosts, and are by themselves lightweight, and thus suit the power sensitive nature of networks such as wireless sensor networks. A complete analysis of possible network threats to general ad-hoc networks is found in [2]. We adopt the hierarchical model proposed in [5] and, in SPAID, extend it to include power awareness of individual nodes.

2 Existing Approaches

The intrusion problem can typically be tackled by adding additional intrusion detection layers on top of the protocol, or through alterations to the wireless protocol itself. For the former style of enforcing security, two types of intrusion detection systems (IDS) are typically used, as a reminiscence of wired intrusion detection techniques [5].

2.1 Network Based Systems

Network-based systems (NIDS) can be passive or active systems, listening in on network traffic. By capturing and examining individual packets flowing through a network, NIDS can analyze across all layers of the network protocol and are able to look at the payload within a packet, to see which particular host application is being accessed and with what options, and raise alerts when an attacker tries to exploit a bug in such code, by detecting known attack signatures. NIDS often require dedicated hosts or special equipment, and thus can be prone to network attacks. Further considerations are discussed in [6,7].

2.2 Host Based Systems

Host-based intrusion detection systems [8,9] monitor each individual host by running on each host. They are able to detect actions such as repeated failed access attempts or changes to critical system files, and normally operate by accessing log files or monitoring real-time system usage [5]. To ensure effective operation, host IDS clients have to be installed on every host on the network, tailored to the specific host configuration. Host-based systems require dedicated processes to run for network monitoring, and, as their name suggests, are not bandwidth dependent. The disadvantage of such comprehensive host-based systems is that they can considerably slow down the hosts that have IDS clients installed.

To circumvent these problems agent-based lightweight models were proposed for wireless networks, which are more bandwidth efficient, and provide a heuristic approach to intrusion detection. Our approach combines the approach in [5] of providing a hop-based hierarchical agent-based model with our own approach for power-awareness in selection of nodes.

2.3 “Secure” Protocols for Wireless Ad Hoc Networks

Protocol-based security measures provide for encryption mechanisms and other extensions such as one-way hash chains used in [4], to deal with routing update

attacks. Some approaches also suggest symmetric cryptographic methods to alter the MAC sub-layer to improve security [10]. Research on secure versions of existing protocols such as link state protocols [13] and variations of distance vector protocols [4] has been performed. It is clear that such "secure" additions correspond to an increase in the rigidity of wireless ad hoc networks, which in essence curtails their usefulness. Certain protocols such as that in [4] seem to improve on the base protocol but comparisons with other approaches is still in its infancy and the increased cryptographic overhead in some applications cannot be justified. Examples of protocol-based security extensions and measures can be found in [3], [4], [11], [12] and [13].

3 Preliminary Considerations for SPAID

The agent-based model proposed in [9] approaches the IDS problem with a technique that handles intrusions with an agent running on each system. Also, the suggestion of statistical methods for classifying network data seems to have been proven to be inappropriate, in light of the Support Vector Machine (SVM) based model suggested in [14]. Further, the model in [9] is not suitable for a power-aware IDS, since such a system warrants energy consumption in systems irrespective of their current battery levels, i.e. it suggests an IDS without considering the feasibility of the assumption that network monitoring and analysis is justified in nodes with minimal power, such as robust wireless sensor networks (WSN).

3.1 Modular IDS Architecture

The IDS we propose is built on a mobile agent framework. It is a non-monolithic system and employs several sensor agents that perform certain functions, such as:

- *Network monitoring*: Only certain nodes will have sensor agents for network packet monitoring, since we are interested in preserving the total computational power and battery power of mobile hosts.
- *Host monitoring:* Every node on the mobile ad hoc network will be monitored internally by a host-monitoring agent. This includes monitoring system-level and application-level activities.
- *Decision-making:* Every node will decide the intrusion threat level on a host-level basis. Certain nodes will collect intrusion information and make collective decisions about network level intrusions.
- *Action:* Every node will have an action module that is responsible for resolving intrusion situation on a host (such as locking out a node, killing a process, etc).

A hierarchy of agents has been devised in order to achieve the above goals. We will adapt the hierarchy for our purposes. There are three major agent classes as used in [5], categorized as monitoring, decision-making and action agents. Some are present on all mobile hosts, while others are distributed to only a select group of nodes, as discussed further. The monitoring agent class consists of packet, user, and system monitoring agents. The following diagram shows the hierarchy of agent classes.

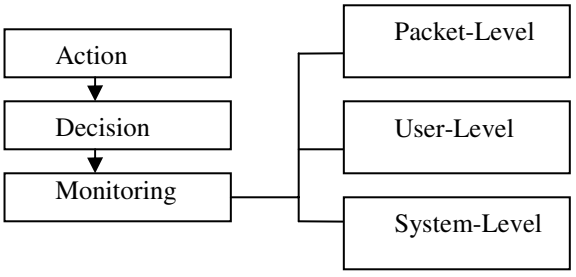


Fig. 1. Typical agent hierarchy, depicting the multi-level decision making process for intrusion detection

3.2 Agent Distribution

As mentioned above, not all the nodes on a wireless ad-hoc network will host all types of IDS agents. To save resources, some of the functionality must be distributed efficiently to a (small) number of nodes. The modular architecture we use employs the architecture in [5]. The decision making module incorporates the energy metric Power Loss/Availability for Network-monitoring Estimate (PLANE), a node-specific measure of the mean power loss per node for running the network monitoring agent. PLANE is can directly be related to the wireless protocol used, mean number of wireless links for the specific node, average node maintenance energy consumption, and the battery power remaining. PLANE ultimately estimates the duration the node can last on the same power without replenishment. To calculate the power consumption metrics such as those in [15] are often used. The reception costs are multiplied by the number of links for the node to yield an average reception cost, to which we add the average cost of sending a message. Thus, these costs are dependent on the density of the network and the routing/link exchange protocols used.

3.3 Calculating PLANE

The calculation of PLANE involves calculating the duration for which the node can continue to support a network monitor along with its normal operations. We therefore calculate PLANE by calculating the time for which node can last as the network monitoring node as shown below in Equation 1.

$$PLANE = \frac{BPR}{TEC_{nm}} \tag{1}$$

In Equation 1, BPR is the total battery power remaining at the instant of node selection, i.e. SPAID in Section 4, and TEC_{nm} is the total energy consumption with network monitoring node processes running. In the absence of measurement of exact networking monitoring energy consumption, we assume PLANE as PLANE'. The value PLANE' is typically available directly from most distributed wireless networks, such as sensor networks, and hence finds a presence in the above calculation.

$$PLANE' = \frac{BPR}{TEC} \tag{2}$$

TEC is the total energy consumption before the node is selected for network monitoring. PLANE can be tailored to suit the needs of the type of network monitoring required and the nature of the actual node on which it runs. We shall not deal in further detail with PLANE in this paper, but rather focus our attention on the iterative algorithm for network monitoring node selection. TEC values are represented by different wireless nodes running in different ad hoc modes which consume between 741 mW and 843 mW [15].

4 The SPAID Algorithm

In SPAID, we deal with multi-hop network monitoring clustered node selection. This type of a node selection has its inherent advantages in allowing complete coverage of all nodes and links in a network, but with an added factor of redundancy in the collection of intrusion detection data. Additionally, by varying the hop-radius of the algorithm and the PLANE/Topology constraints, redundancy in overlap of monitored nodes can be achieved, which allows us to prune the set of nodes selected for network monitoring. Considering that we are dealing with minimally mobile wireless ad hoc networks, topological changes shall not be considered in PLANE evaluation, and are deemed to be constant during the process of node selection.

4.1 SPAID Node Selection Algorithm for Network Monitoring Nodes

The SPAID algorithm uses the agent hierarchy presented in Fig.1, with a significantly adapted node selection mechanism to incorporate power-awareness, and is best detailed by the following six steps.

Step 1: Set PLANE Constraint/ Topology Constraint. Set a constraint on the PLANE value of nodes which are allowed to compete for becoming a network monitoring node. These depend upon the duration for which the topology is expected to be unchanged, and IDS active duration. Further, certain nodes which have very small number of adjacent nodes may be discarded by setting the Topology Constraint.

Step 2: PLANE Calculation and PLANE Ordered List (POL). Arrange the different nodes in increasing values of PLANE as calculated previously, for all nodes which satisfy the PLANE Constraint. This implies that nodes that can last longer as a network monitoring node take higher precedence in consideration for selection.

Step 3: Hop Radius. Set the hop radius to one initially, and increment for each insufficient node selection with the current hop radius.

Step 4: Expand Working Set of Nodes. Consider node selection incrementally, initially from the first node, (node with highest PLANE), to finally the set of all nodes in the network, incrementing the set of nodes under consideration by one node each time. We call this set the working set (WS) of nodes. The WS is expanded only if the addition leads to an increase in number of represented nodes.

Step 5: Voting. We use the voting system for Node Selection, as used in [5], except that we limit the candidates to just the nodes which are part of WS. Under this voting system, each node votes for that node within the hop radius which it feels is the best-

connected node in the network. The connectivity indices used in [5] do not have to be calculated in our approach.

Step 6: Check acceptability of nodes. If all links/nodes are not represented by the set of nodes covered by the voting scheme, then we expand the WS and repeat the process from Step 4. If WS equals the POL, then increment the hop radius, and repeat from Step 3. It is suggested that the increment in hop radius be considered a final resort, as it effectively increases the amount of processing per monitoring node.

4.2 An Example

Let us consider a network (with node density D=3) given below in Fig. 2, listed with the PLANE values for different nodes. The node density is a count of the number of nodes on average adjacent to other nodes at the instant of running SPAID.

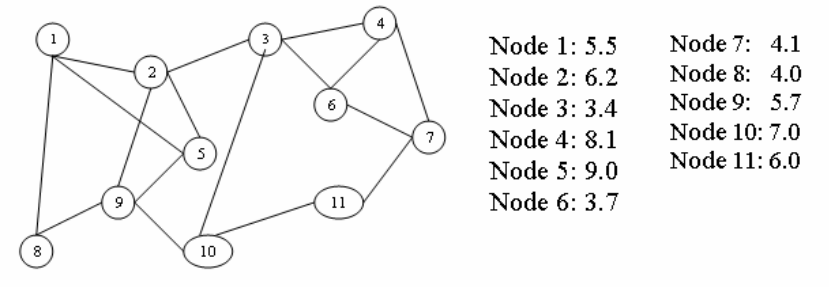


Fig. 2. An example network with a D = 3. The PLANE values (in relative time) for the different nodes are shown

The POL is therefore given by {5,4,10,2,11,9,1,7,8,6,3} where each number represents the node number. We initially set the hop radius to 1, in case an allocation is not possible, SPAID continues with higher hops. We depict the Working Set as WS {<node list>}, and iteratively augment the list with nodes from the PLANE Ordered List. For this example, we begin with WS {5}, i.e. we take the first node, Node 5, which has the highest PLANE.

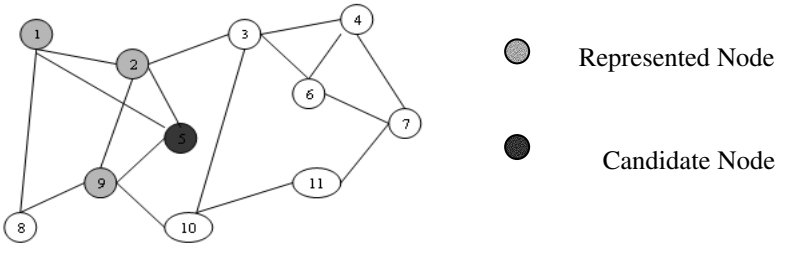


Fig. 3. Example 1 with WS {5}

Next, considering that all nodes have not been covered, we choose the next node, in this case node 4, and so on. Fig. 4 and Fig. 5 portray the next two steps.

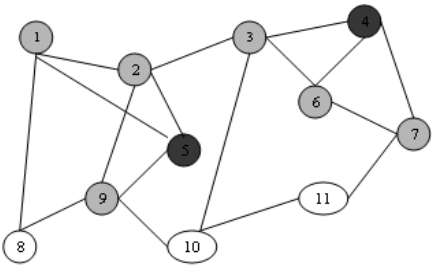


Fig. 4. Example 1 with WS{5,4}

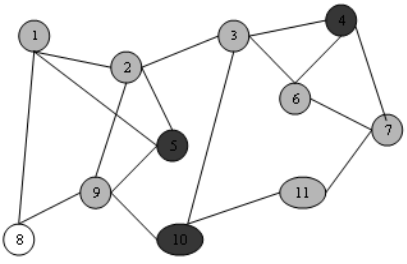


Fig. 5. Example 1with WS{5,4,10}

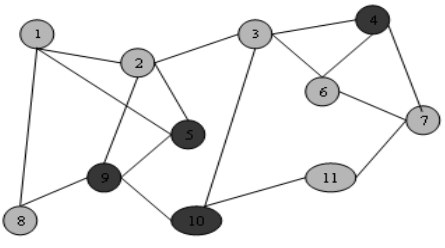


Fig. 6. Example1. Final node selection {5,4,10,9}. WS{5,4,10,2} and WS{5,4,10,2,11} were skipped since no new nodes are represented

The current WS {5,4,10} has the next choice in accordance with SPAID, as node 2, followed by node 11. The addition of these nodes, however, provides no additional information from any node that cannot be obtained from the current WS. Thus we skip potential working sets {5,4,10,2} and {5,4,10,11}.

All nodes are represented and hence the solution set WS is {5,4,10,9} as shown in Fig 6. It is clear from the above example that the percentage of packet monitors varies inversely with the node density, but the node selection varies in slightly different fashion in low density and high density networks as illustrated in Section 5.

4.3 Rerunning SPAID

Dynamism in SPAID is a very important concept, considering that power levels drop considerably if a node persistently runs as a network monitoring node. The SPAID algorithm needs to be run when a change in the power level of the current WS indicates that another node has a better chance of lasting longer as a network monitoring node.

In Example 1, after current WS {5,4,10,9} has run for about 200 seconds (assuming idle power consumption as uniform), the power level in node 9 would have dropped below that in node 1. In this case, the SPAID algorithm needs to be run again, to ensure a power-optimal solution to the multiple-sensor network monitoring problem is maintained as the power levels change.

5 Performance Comparisons

Comparisons between single-hop and multiple-hop radius for allocating network monitoring nodes provides a neat measure of the tradeoff involved vis-à-vis the number of nodes needed. As the node density (D) increases drastically, the percentage of nodes allocated for network monitoring increases gradually and then stabilizes. The performance of SPAID can be appraised using the percentage of nodes selected as

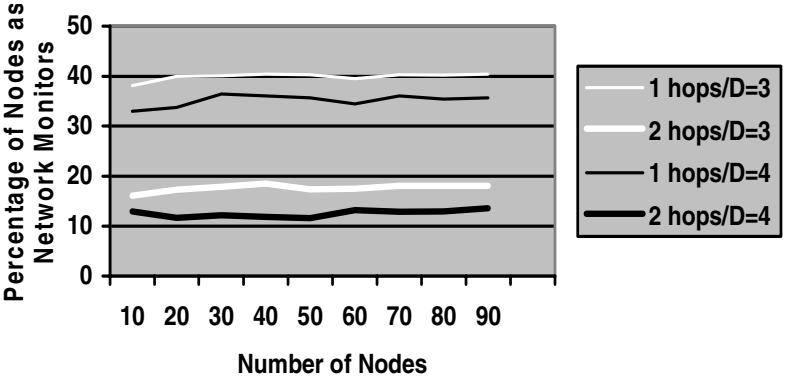


Fig. 7. Performance in Sparse Wireless Networks with low average number of adjacent nodes (D) per node using SPAID. The near constant percentage of nodes used is evident

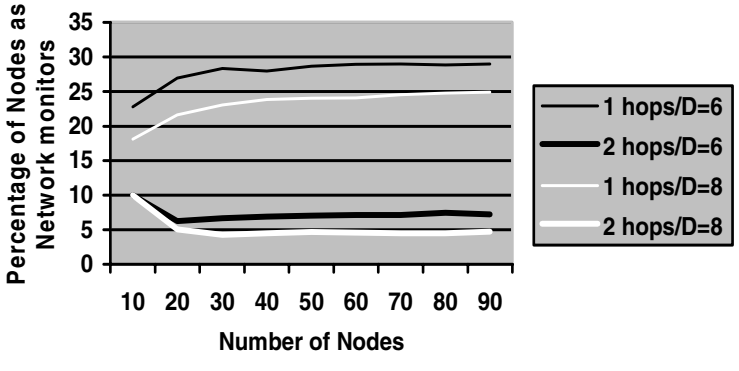


Fig. 8. Performance in Dense Wireless Networks with high average number of adjacent nodes (D) per node using SPAID. The gradual change in percentage of nodes towards stable levels is to be noted

network monitors as a metric. The density of the network clearly plays a major role, since the more the number of adjacent nodes per node, the fewer the network monitors needed to verify their authenticity. For high density (D greater than or equal to 8) wireless ad hoc networks, such as wireless sensor networks, we find that the density of network monitors stabilizes to near constant levels, and mimics the values

presented in [5], which represent the performance of a non-power-aware iterative node selection algorithm. As evident from the succeeding graphs, increasing node density and adjacency reduces the percentage of nodes to be selected as network monitoring nodes. Thus, intrusion detection systems adapt quite efficiently to SPAID when using high-density ad-hoc networks with a large number of nodes.

A practical limit of 2 hops is necessary, as this limits the amount of network monitoring traffic to be transferred through intermediate nodes, since the amount of traffic varies as a quadratic of the hop-limit.

6 Conclusion

In this paper, we have suggested an iterative algorithm SPAID, that culminates from our consideration of individual node power-levels using PLANE. Our algorithm, SPAID, provides a capable means of power-aware node selection. Selection of network monitoring nodes plays a key role in determining the effectiveness of coverage of any intrusion detection technique which runs on each node, and through this paper we propose an a scheme that combines power-awareness with agent-based node selection. We are currently working on an adaptive version of SPAID, specifically for multi-tiered hierarchical IDS architectures, where different node selection procedures can be employed for rapid learning and efficient detection.

References

1. Heady, R., Luger, G., Maccabe, A., and Servilla, M.: The architecture of a network level intrusion detection system, Technical report, Computer Science Department, University of New Mexico, 1990.
2. Zhou, L., and Haas, Z.J.: Securing Ad Hoc Networks, IEEE Networks Special Issue on Network Security. November,1999.
3. Zapata, M.G.: Secure Ad hoc On-Demand Distance Vector (SAODV) Routing, IETF MANET Mailing List, Message-ID 3BC17B40.BBF52E09, 2002.
4. Hu, Y.C., Johnson, D.B. and Perrig, A.: SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks Fourth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), 2002.
5. Kachirski, O. and Guha, R.: Efficient Intrusion Detection using Multiple Sensors in Wireless Ad Hoc Networks", 36th Annual Hawaii International Conference on System Sciences (HICSS'03) - Track 2 , January 06 - 09, 2003.
6. Dasgupta, D. and Brian, H.; "Mobile Security Agents for Network Traffic Analysis", Proceedings of DARPA Information Survivability Conference & Exposition II,2001.
7. Tao, J., Ji-ren, L., and Yang, Q.: "The Research on Dynamic Self-Adaptive Network Security Model Based on Mobile Agent", Proceedings of 36th International Conference on Technology of Object-Oriented Languages and Systems, 2000.
8. Bernardes, M.C., and Moreira, E.S.: "Implementation of an Intrusion Detection System based on Mobile Agents", Proceedings of International Symposium on Software Engineering for Parallel and Distributed Systems, pp. 158-164, 2000.
9. Zhang, Y., and Lee, W.: "Intrusion Detection in Wireless Ad-Hoc Networks", Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, MobiCom, pp. 275-283, 2000.

10. Perrig, A., Hu, Y.C., and Johnson, D.B.: Wormhole Protection in Wireless Ad Hoc Networks. Technical Report TR01-384, Department of Computer Science, Rice University, 2001.
11. Awerbuch, B., Holmer, D., Nita-Rotaru, C., and Rubens, H.: An On-Demand Routing Protocol Resilent to Byzantine Failures In ACM Workshop on Wireless Security (WiSe), 2002.
12. Bhargava, S., and Agrawal, D.P.: Security enhancements in aodv protocol for wireless ad hoc networks. Vehicular Technology Conference, 2001.
13. Papadimitratos, P., and Haas, Z.J.: Secure Link State Routing for Mobile AdHoc Networks, IEEE Workshop on Security and Assurance in Ad hoc Networks, 2003.
14. Deng H., Zeng, Q.A., and Agrawal, D.P.: SVM-based Intrusion Detection System for Wireless Ad Hoc Networks Proceedings of the IEEE Vehicular Technology Conference (VTC'03), Orlando, 2003.
15. Feeney L.M., and Nilsson, M.: Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment, Proceedings of IEEE INFOCOM, 2001.

BSMON: Bandwidth-Satisfied Multicast in Overlay Network for Large-Scale Live Media Applications

Yuhui Zhao[1,2], Yuyan An[2], Jiemin Liu[1], Cuirong Wang[1], and Yuan Gao[1]

[1] Northeastern University at QinHuangDao, 066000, QinHuangDao, China
yuhuizhao@mail.neuq.edu.cn
[2] QinHuangDao Foreign Language Professional College,
066004, QinHuangDao, China

Abstract. ISP needs multicast service to save the bandwidth and costs when more and more large-scale live media applications are emerging in the Internet. There are no proper methods because the localization of IP multicast and Application Level Multicasts. Here we present a suggestion called Layered Overlay Multicast Network (LOMN) for the ISP's live media services. LOMN set up the core multicast tree for live streams by strategically deploying the service brokers (SvBs) and the algorithm of Bandwidth-satisfied multicast in overlay network (BSMON), which is to set up an efficient QoS-satisfied overlay connections between SvBs, balance the overlay traffic and the load on the SvBs. we address critical issues such as scalability, low delays, little delay variations and minimal the cost of multicast tree.

1 Introduction

More and more large-scale live media applications using group communication technology are emerging in the Internet, such as video conferencing, video on-demand, network games, distributed interactive simulation, etc. Because Multicast is better than multiple unicast for group communication for Internet Service Providers (ISPs), there have been tremendous efforts to provide multicast service, ranging from IP multicast to recently proposed application-layer multicasts(ALM), which are in development. IP multicast is still far from being widely deployed on the Internet [1] and its most critical ones include: the lack of a scalable interdomain routing protocol, the state scalability issue with a large number of groups, the lack of support in access control, the requirement of global deployment of multicast-capable IP routers and the lack of appropriate pricing models, as make ISPs reluctant to deploy and provide multicast service. ALM's multicast-related features are implemented at end hosts, data packets are transmitted between end hosts via unicast, and are replicated at end hosts. They are not require infrastructure support from intermediate nodes (such as routers), and thus can be easily deployed. But its lacks are less QoS for live media and generally low scalable to support large multicast groups due to its relatively low

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 163–172, 2005.

bandwidth efficiency and heavy control overhead caused by tree maintenance at end hosts. In addition, it is hard for it to have an effective service model to make profit for an ISP.

To facilitate the support for existing and new overlay applications for ISP, we propose a framework called Layered Overlay Multicast Network (LOMN) that not only can be shared by a variety of applications, but also can provide scalable, efficient, and practical multicast support for a variety of group communication applications. Services brokers (SvBs) composed of the cores of LOMN, and Bandwidth-satisfied multicast in overlay network for large-scale live media applications (BSMON) is proposed for LOMN. The main function of BSMON is to search bandwidth-satisfied overlay multicast paths forming overlay networks for upper layer QoS-sensitive overlay applications, and balance overlay traffic load on SvBs and overlay links. With the increase in QoS-sensitive applications in the Internet, it is necessary to support QoS-aware routing service for the whole Internet. With the placement of SvBs by ISPs, BSMON can also be used to provide end-to-end QoS-aware multicasting services without significant changes in the Internet infrastructure.

2 Relative Works

Recently the efforts on overlay networks have been very active. In the proposals, Some is for specific application such as Host multicast [2], content distribution networks [3], peer-to-peer file sharing [4]; and the others aiming at developing generic overlay service networks for a variety of applications. For example, Yoid [5] is a generic overlay architecture, which is designed to support a variety of overlay applications that are as diverse as net news, streaming broadcasts, and bulk email distribution. Another similar effort is the Planet-lab [6] experiment whose goal is to build a global testbed for developing and accessing new network services. X-Bone [7] operates at the IP layer and based on IP tunnel technique. OverQoS [8] is an architecture proposed to provide Internet QoS using overlay networks. Our effort is complementary to most of the above approaches. We share common goals as OverQoS, SON, and Yoid. However, this paper is mainly focused on to set up QoS-satisfied overlay connections between SvBs, balance the overlay traffic and the load on the SvBs. None of the other work has addressed this aspect adequately. In addition, we address critical issues such as scalability, low delays, little delay variations and minimal the cost of multicast tree.

3 Layered Overlay Multicast Network

LOMN consists of service brokers which are strategically deployed by ISP who dimensions its overlay network according to end user requests and sells its multicast services to group coordinators via a service contract. Most of the Internet domains can have one or more SvBs that depends on the needs of live media communication. These SvBs provide a unified platform to serve several overlay

applications at the same time. For supporting large-scale live media communication to enhance scalability, LOMN have a hierarchical architecture. The groups of contiguous SvBs are grouped to form clusters, which in turn are grouped together to form super-clusters. And an end host could form a cluster with other end hosts close by. In the cluster, overlay multicast is used for efficient data delivery between the limited number of end users. The user clusters connect to the LOMN by the access SvBs on the edge of LOMN.

3.1 Service Brokers

SvBs are specialized nodes that can be placed in the Internet to provide generic overlay service support to overlay multicast applications. These SvBs are interconnected over the transfer layer to form the LOMN. They cooperate with each other across the Internet to provide overlay services support. SvBs can be placed either at the edge of a domain or in the core and subscribe high bandwidth connections to the Internet backbone. Usually they are designed to different levels according to the relationship of the ASs. The SvBs of one domain know the addresses of the SvBs of the neighboring domains using dynamic hash table (DHT). This knowledge can be incorporated during deployment or through exchange of messages with the neighboring domains. The SvBs are also responsible for encapsulation and decapsulation of the outgoing and incoming packets of the overlay network, respectively.

3.2 Formulation of the Hierarchical Topology

If the physical network topology is modelled as an undirected graph $G = (V, E)$, where V and E denotes the sets of network nodes (or routers) and physical links respectively, the overlay dimensioning problem can be formulated as follows: given a set of groups $\{M_i\}$ with group member distribution and bandwidth requirement, and a physical network topology $G=(V,E)$, find a virtual topology G' on top of G so that G' can accommodate all the groups, while keep the cost of G' minimum under the bandwidth waste threshold. Here, we assume the multicast group set M_i is obtained from the service contract (between the users and the ISP) or from long-term measurement in the steady state of the network and group dynamics. The G' is the LOMN network.

In LOMN, Using D_x denotes a AS x, the set $\{D_x, x \in N\}$ composed of the graph G, and from the overlay point, it also composed of the graph of G', $G' = (V', E')$, V', $E'$are the set of the SvBs and the edges of the overlay edges. V'_i denotes the node SvB i, E'_{ij} denotes the overlay edge between the node SvB i and the node SvB j, $|V'|$ denotes the numbers of the SvBs, $|E'|$ denotes the numbers of the connections between the node SvB i and the node SvB j.

Basic Link Regular: If $V'_i, V'_j \in D_x$, then existing E' between them and $|E'| = 1$.

InterDomain Link Regular: If $V'_i \in D_x$ and $V_j \in D_y$, then existing E', which link D_x and D_y, called $interE'$, and $|E'|$=1; If more than one physical links between the D_x and D_y, then existing corresponding $interE'$, and $|E'| \geq 1$.

Node Hierarchy Regular: If V_i' has an $interE'$, it is called top SvB, denotes as $topV'$; if V_i' is at the edge of D and directly links to the terminal agent, it is called an access SvB, denotes $accV'$; others are called core SvB, denotes as $corV'$.

Overlay Hierarchy Regular: if a subset of G' only composed of the node like $topV'$ and the edge like $interE'$, then it is call top multicast overlay sub-graph of G', denotes as $topG'$; if a subset of G' only composed of the node like $corV'$, then it is call core multicast overlay sub-graph of G', denotes as $corG'$; else, it is called access multicast overlay sub-graph of G', denotes as $accG'$.

From the regulars defining above, the hierarchy overlay multicast network is set up. These also mean that the overlay link connecting two or more SvBs in the overlay topology may physically pass through multiple ASes that do not have any SvBs. And the SvBs within the same domain can form full mesh connecting each other, but the LOMN interconnecting multiple domains is not full mesh topology, this can reduce the connections messages and control the size of the DHT.

3.3 Performance Measurements of Overlay Links

An overlay link is usually composed of multiple physical links. And the non-overlay traffic would also be using the same physical links. The SvB cannot control or manage the IP-layer resources. To obtain the performance of an overlay link, many efficient measurement methods have been proposed in the literature, such as, Ping and Sting [9]. Here we can use direct measurements and actively send traffic between two SvBs and see how much of traffic can get through before the path gets saturated and starts losing packets.

3.4 Multicasting in LOMN

The resource allocation function mainly deals with allocating the network resources response to the QoS requirements.In the process of the multicast, LOMN is advocated as the service backbone domain. Outside LOMN, end users subscribe to services by transparently connecting to the access SvB advertised by the ISP. Each access SvB organizes some end users into a "cluster", where an application layer multicast tree is formed for data delivery among the cluster members. When a new customer wants to subscribe overlay services from the LOMN, it first contacts an service web portal, that is access SvB. the access SvB receives a join request for a multicast group, it first looks up a multicast tree for the group from his DHT, sets up one or more connections to the core SvBs in the multicast tree, then to the top SvBs, all of these are the multicast distribution trees building for data delivery. To reduce the management overhead of a large number of trees and improve the multicast state scalability, an aggregated multicast approach[10] is used, in which multiple groups are forced to share one delivery tree. Data packets are encapsulated at access SvB, transmitted on aggregated trees, and decapsulated at outgoing access SvBs.

4 Bandwidth-Satisfied Multicasts in Overlay Network

Based on the features of LOMN, a multicast routing framework, BSMON, is supposed. To provide a QoS-satisfied overlay multicast path from the source SvB to the destination SvBs in the overlay topology, it is necessary to identify a subset of the overlay topology that provides the connectivity, and satisfies the required QoS between the source SvB and the destination SvBs. Critical resources includes not only the overlay link capacity, but also the SvBs capacity. The connectivity depends on the bandwidth availability of overlay links and the capacity of the SvBs.

4.1 The Main Ideas of BSMON

BSMON uses the following approaches to provide bandwidth-satisfied overlay multicasting services in the dynamic network environment.

1) While selecting the overlay links, BSMON tries to balance the traffic among the overlay links and SvBs in addition to satisfying the QoS requirement. This approach ensures that the overlay traffic will be resilient to the background non-overlay traffic. At the same time, additional overlay traffic will have less impact on the existing traffic when the overlay path quality is degraded.

2) BSMON realizes the hierarchical architecture of LOMN, provides a scalable topology for distributing the overlay link and SvB state information. Each SvB can then have the aggregated overall mesh. When an overlay routing request arrives, the SvB can utilize the aggregated topology to find an approximate path. It then contacts some of the SvBs on the path to get detailed and up-to-date information about the path performance. Even though this approach incurs some control message overhead, the up-to-date path performance information helps in improving the possibility for providing QoS service satisfaction.

3) During the overlay data routing process, the non-overlay traffic may increase suddenly and thereby could affect the normal overlay data traffic. To cope with this situation, BSMON uses an adaptive routing approach. When an SvB realizes that the additional overlay link capacity of an overlay link is less than the overlay traffic it is currently servicing, it begins to search several backup overlay paths connecting itself to this neighboring SvB. These backup paths will make sure that the overlay traffic can bypass this overlay link if the quality degrades. If the SvB cannot find enough backup paths, it will notify some of the previous hop SvBs to search for backup paths to bypass the degraded overlay link.

To achieve these ideas, let the users feel satisfied to the bandwidth, which the ISP offers to them, the delay and delay-variation is the key factors in the multicast process.

4.2 The Bandwidth-Satisfied Overlay Multicasting

By the Performance Measurements among the SvBs, an overlay weighted undirected graph G' comes into being. Here the function of the link delay is defined as $Delay$: $E' \longrightarrow$R+, $Delay(l)$ denotes the delay of the packets passing the overlay link l, $l \in E'$. And the function of the performance of a node is defined as

Perfo: $Cpu \longrightarrow 100\%$, *Perfo(SvB)* denotes the OccupyTime of the CPU of the SvB, $SvB \in V'$.

Let $M \subset V'$ be a set of nodes involved in a group communication. V'_M is called multicast group member, $V'_M \in M$. Packets originating from a source node, V'_s, have to be delivered to a set of receiver nodes $M - \{V'_s\}$. A multicast tree $\mathrm{T}(V'_M, E'_M)$ is a subgraph of G' that centers as c_i, $\forall c_i \in coreG'$, and spans all the nodes in M. The path from a source node V'_s to a receiver node V'_d in the tree T, is denoted by $P_T(V'_s, V'_d)$, where $V'_s, V'_d \in M$.

According the approaches mentioned in 3.1, the multicast tree should meet the following conditions:

$$Perfo(SvB_i) \leq \lambda,\ \forall\ SvB_i \in M\ \&\ SvB_i \in D_x \tag{1}$$

$$\min_{SvB_i \in M}(Perfo(SvB_1), Perfo(SvB_2), \ldots, Perfo(SvB_j)) \tag{2}$$

$$\min_{l_j \in P_t}(\sum_{j=1} Perfo(SvB_{l_j,i}), \ldots, \sum_{j=m} Perfo(SvB_{l_j,i})),\ \forall i, j, m \in N \tag{3}$$

$$\sum_{l \in P_T(V'_s, V'_d)} delay(l) \leq \triangle,\ \forall V'_s, V'_d \in M \tag{4}$$

$$|\sum_{l \in P_T(u,v)} delay(l) - \sum_{l \in P_T(x,y)} delay(l)| \leq \delta,\ \forall u, v, x, y \in M \tag{5}$$

where λ is the maximum occupy time of the SvB's CPU, Δ is the maximum of delay, δ is the maximum of the delay variation between the any different l.

The algorithm basic idea is described as follows.

Step 1: if some terminal of AS_x belongs to a multicast group, then the algorithm can select one or more SvBs in AS_x as the multicast intermediate node, and selecting SvB need to satisfy Expressions 1, which means only to select the free SvB as a relay node.

Step 2: the selected SvBs composed of the multicast group.

Step 3: compute the center nodes. Assume any node SvB_i in the multicast group as center, and create a multicast tree l_j using Dijkstra Algorithm, then compute the delay, if it satisfies the expression 4, sequentially compute the delay variation, if it satisfies the expression 5, the node is sent to the set of the center nodes.

Step 4: in turn compute the cost order of the relating multicast tree l_j using expression 3.

Step 5: in the set of the center nodes, in turn compute the order of the candidate center node using expression 2.

Step 6: select the freest SvB as the center, and the relating tree as the current multicast tree, the others as candidates.

From the steps, we will achieve the following objectives: (1) Balancing overlay traffic among the overlay links; (2) Balancing the overlay traffic overhead among the SvBs; (3) Finding and providing QoS-satisfied paths connecting the source SvBs and destination SvBs.

5 Performance Evaluations

5.1 Simulation Setup

We have implemented a session-level event-driven simulator to evaluate the performance of BSMON. The simulations are based on the Georgia Technology Internetwork Topology Model (GT-ITM) [11], which is used to generate the network topology. These topologies have 50 transit domain routers and 500-2000 stub domain routers. Each end host is attached to a stub router uniformly at random. To test the scalability of different schemes, we focus on large group sizes and vary the number of members in each group from 200 to 1000. When simulating the algorithms, the SvBs form a Three-level hierarchical topology at the application layer. Each of the clusters in the hierarchy has an average of ten members (subclusters or SvBs).

The dynamics of the overlay Multicasting is modeled as follows. The overlay multicasting request arrives at a random accSvB node according to a Poisson distribution with rate λ. The destination domain is randomly chosen. The holding time of the overlay session is exponentially distributed with a mean of 2 min. Similar to [12], the offered load of the overlay routing request is defined as $\rho = (\lambda * h/u * (\sum L_i))$, where h is the mean of overlay path hops (number of SvBs in the path), and $\sum L_i$ is the sum of the overlay link capacities in the corresponding overlay topology. During the process of simulation, we vary the value of u to test BSMON' s performance under different offered loads. The physical links' bandwidths during the simulation are randomly selected between 40 and 280 units with delay 2 ms, while the SvBs' capacities are uniformly set as 900 units. The non-overlay traffic occupies around 50% of each physical link's capacity. The non-overlay traffic varies its volume $\pm 20\%$ every 500 ms. The SvBs exchange their state information every 1000 ms. We assume that the error of available bandwidth measurement result is within $\pm 10\%$. For each overlay routing request, we use an overlay routing protocol to set up an overlay path connecting the source SvB and the destination SvB with a bandwidth requirement range of 1-6 units. The computation capacity requirement is varied between 6-10 units.

In the Internet, most of the interdomain traffic is concentrated across a smaller subset of ASes. To simulate the network situations, we repeat our simulations on the following network scenarios: 80% of the overlay routing requests' source and destination pairs are from 30% SvBs, while others are uniformly distributed among all the other SvBs, which is reflective of the real Internet environment.

5.2 Simulation Results and Discussions

Bandwidth-Satisfaction Rate(BSR): Because of the unbalanced distribution of Internet traffic, in many situations, the shortest path-based routing protocol cannot provide a QoS-satisfied path connecting the source and destination domains. To quantify this factor, BSR is defined as

$$BSR = \frac{Number\ of\ Bandwidth\ satisfied\ overlay\ paths}{Number\ of\ overlay\ request\ paths}$$

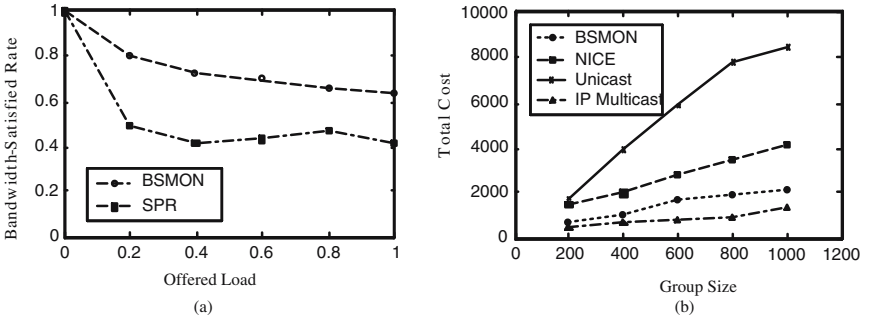


Fig. 1. (a) BSR comparison (b)Group Size and Tree cost

The results obtained for BSMON are compared with that of the shortest-path routing(SPR) algorithm, which refers to the shortest path in the overlay network, not in the IP layer. Fig.1(a) shows the QSR of BSMON compared with SPR. From the figure, we can observe that BSMON can greatly improve the QSR. In addition to finding QoS-satisfying overlay paths, BSMON also helps in finding paths that are not affected significantly by the non-overlay traffic.

Multicast Tree Cost: Multicast tree cost measures by the number of links in a multicast distribution tree. It quantifies the efficiency of multicast routing schemes. Application level multicast trees and unicast paths may traverse an underlying link more than once, and thus they usually have a higher cost than IP multicast trees. In Fig.1(b), we plot the average tree cost of BSMON, NICE, ,unicast and IP multicast as group size increases from 200 to 1000. As a reference, we also include the total link cost for unicast. Compared with the cost of unicast paths, NICE trees reduce the cost by 35%, BSMON trees reduce the cost by approximately 70%, and IP multicast trees save the cost by 68-80%. Clearly, the performance of BSMON is comparable to IP multicast. In addition, BSMON outperforms NICE in all cases, and their difference magnifies as group size is increased.

Average Link Stress: Link Stress is defined as the number of identical data packets delivered over each link. IP multicast trees has the least link stress since only a single copy of a data packet is sent over each link. Fig.2(a) shows the average link stress as the group size varies. IP multicast maintains a unit stress since no duplicate packets are transmitted on the same link. BSMON trees exhibit average link stress between 1.16 and 1.56, whereas the average link stress of NICE trees is always higher than 2.00. For BSMON and NICE, the link stress does not vary greatly with group size. However, unicast is not as scalable as BSMON and NICE, since its link stress keeps increasing when group size grows.

Average Path Length: Path Length is the number of links on the path from the source to a member. Unicast and shortest-path multicast schemes are usually optimized on this metric and thus have smallest path lengths. In simulation experiments, end hosts join the multicast group during an interval of 200 seconds. The results for average path length are shown in Fig.2(b). As expected,

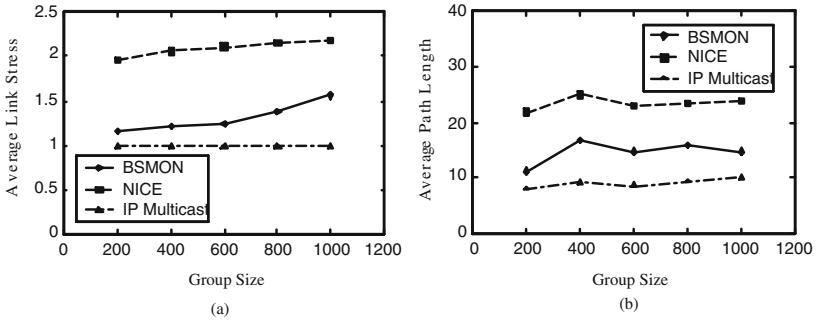


Fig. 2. (a) Group Size and average link stress (b)Group Size and average Length

IP multicast has the shortest end-to-end paths. Additionally, the path lengths of BSMON trees are much shorter than those of NICE trees on average. For instance, at group size of 1000, the average path lengths of BSMON and NICE trees are approximately 15 and 21, respectively.

6 Conclusions and Future Works

In this paper, LOMN locates to support emerging large-scale live media applications and tries to build an overlay multicast network in the Internet for ISP. LOMN assumes that most Internet AS has one or more service brokers, which form a layered overlay network and cooperate with each other to facilitate the deployment of overlay service. The key part of LOMN is BSMON, which is designed as a generic overlay multicast protocol. The goal of BSMON is to find Bandwidth-satisfied paths and select adaptively route the overlay traffic in spite of the unpredictable overlay link performance. BSMON can set up an efficient bandwidth-satisfied overlay connections between SvBs, balance the overlay traffic and the load on the SvBs. we address critical issues such as scalability, low delays, little delay variations and minimal the cost of multicast tree. The simulation results show that the BSMON algorithms can effectively find and provide QoS-assured overlay services and balance the overlay traffic burden among the SvBs, as well as the overlay links.

Our work is just beginning, only limited testing and simulation has done, the design needs to be validated in using. It is necessary to do more optimize for some multi-objective problem during the process of looking for the best multicast trees in the future.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under grant No.60273078 and the Doctor Fund of Hebei under grant No.05547010D-3.

References

1. C. Diot, B. Levine, J. Lyles, H. Kassem, and D. Balensiefen. Deployment issues for the IP multicast service and architecture. IEEE Network, Jan. 2000.
2. B. Zhang, S. Jamin, and L. Zhang, "Host multicast: A framework for delivering multicast to end users," presented at the INFOCOM'02, New York, June 2002.
3. B. Krishnamurthy, C.Wills, and Y. Zhang, "On the use and performance of content distribution networks," presented at the ACM SIGCOMM Internet Measurement Workshop, San Francisco, CA, Nov. 2001.
4. I. Stoica, R. Morris, D. Karger, M. F. "Chord: A scalable peer-to-peer lookup service for internet applications," in Proc. ACM SIGCOMM, Aug. 2001, pp. 149-160.
5. P. Francis. Yoid: Extending the Internet Multicast Architecture [Online]. Available: http://www.aciri.org/yoid/docs/index.htm
6. Z. Duan, Z. Zhang, and Y. T. Hou, "Bandwidth provisioning for service overlay networks," presented at the SPIE ITCOM Scalability and Traffic Control in IP Networks (II)'02, Boston, MA, July 2002.
7. XBone [Online]. Available:http://www.isi.edu/xbone
8. L. Subramanian, I. Stoica, H. Balakrishnan, and R. H. Katz, "Over QoS: Offering Internet QoS using overlays," presented at the HotNetIWorkshop, Princeton, NJ, Oct. 2002.
9. S. Savage, "Sting: A TCP-based nework performance measurement tools," in Proc. 2nd USENIX Symp. Internet Technologies and Systems, Oct. 1999, pp. 71-79.
10. A. Fei, J.-H. Cui, M. Gerla, and M. Faloutsos. Aggregated Multicast: an approach to reduce multicast state. Proceedings of Sixth Global Internet Symposium(GI2001), Nov. 2001.
11. GT-ITM: Modeling Topology of Large Internetworks [Online]. Available: http://www.cc.gatech.edu/projects/gtitm/
12. A. Shaikh, J. Rexford, and K. Shin, "Evaluating the overheads of sourcedirected quality-of-service routing," in Proc. 6th IEEE ICNP, Oct. 1998, pp. 42-51.

A Routing and Wavelength Assignment Algorithms Based on the State Level of Links

Xiaogang Qi, Sanyang Liu, and Junfeng Qiao

Department of Mathematics Science,
Xidian University, Xi'an 710071, P.R.China

Abstract. For the problem of routing and wavelength assignment in Wavelength Division Multiplexing(WDM)optical transport network, an algorithm based on the state of links is proposed, which is named Trade-off_LSDRAW, and which can select a path with the higher state level between a pair of nodes in a network. Finally, by a example network, we show that the algorithm has the stronger capability of selecting a better path between a pair of nodes, and can achieve the load balancing and reduce the congestion probability in WDM optical transport networks.

1 Introduction

For the extremely great capacity and high speed, the technology of optical transmission system is superior to the other transmission system. Now with the increasing development of WDM technology and the update of the equipments of Optical Cross-Connect (OXC) and Optical Add-Drop Multiplexing (OADM), all-optical networks employing the concept of routing and wavelength assignment (RWA) has made great improvements in the flexibility of selecting a route, and all kinds of RWA algorithms have emerged for different traffic requirements and different optimal objectives. Some kinds of metric parameters are used as the standard to select a path in the algorithms [1-4], in which the current state of the communication link of the network is so little considered that congestions will take place in some cases and the performance of the network will deteriorate. Therefore the RWA algorithms with the ability of load balancing in different network state is required, and which can effectively solve the problem of the network congestion and the corresponding decline of the network performance due to load imbalance in WDM transport networks. Finally, a dynamic RWA algorithm based on the state level of links is proposed.

2 Problem Description and Definition of Objective Function

Assume that the WDM transport network is denoted by a directional graph $D = (V, A, S)$, where V is the set of nodes and A is the set of directional edges or arcs in D , and S represents the current state of all the arcs in A that

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 173–177, 2005.

Level_link_state	Set_wavelength_avai	Bandwidth(λ_k), $\lambda_k \in$ Set_wavelength_Avai

Fig. 1. The example WDM network

correspond to the communication links. For any a_{ij} in $A = (v_i, v_j)$ $(i \neq j)$, there exists s_{ij} denotes the current state of link a_{ij}.

As a rule, dynamic routing is composed of two steps, the collection of the state information of links in the network and the route selection based on the collected information. The state information of a link includes the available channel of the link, the available bandwidth corresponding to each channel, the link delay, the reliability and et al. The link state information is updated with the standard threshold-based triggered approach.

In the WDM transport network D described here, there is a routing table for any node v to other nodes in D, and the state information of links is exchanged among different routing tables to update the total state information of the network. When the state level of a link is changed, the nodes connected with it will firstly detect and then sent out the change to the other nodes. Here the state information of links involves the state level, the available wavelength of the link and the available bandwidth corresponding to each available wavelength. The state information of a link is expressed as the following Figure 1.

Here $Leve\_link\_state$ denotes the state level of a link,$Set\_wavelength\_avai$ denotes the set of available wavelengths, and $Bandwidth(\lambda_k)$is the bandwidth on the available wavelength λ_k. The state level of a link can be represented as followings:

$$\begin{aligned} Leve\_link\_state(a_{ij}) &= Level(Set\_wavelength\_avai(a_{ij}), Bandwidth(\lambda_k)) \\ &= f[Cap\_avai\_rate(a_{ij})] \end{aligned} \tag{1}$$

where $\lambda_k \in Set\_wavelength\_avai(a_{ij})$, and the state of link a_{ij} can be classified by $Cap\_avai\_rate(a_{ij})$ and the different evaluation set, and $Cap\_avai\_rate(a_{ij})$ denotes the relative available capacity of the link a_{ij}. Suppose that the evaluation set is

$$T = (t_1, \cdots, t_i, t_{i+1}, \cdots, t_{n-1}). \tag{2}$$

satisfying

$$0 < t_1 < \cdots < t_i < t_{i+1} < \cdots < t_{n-1} < 1. \tag{3}$$

and $Cap\_avai\_rate(a_{ij})$ can be divided into corresponding level from 1 to n by T, therefore the state level of a link takes values from 1 to n, in which the highest level corresponds to with the maximal value n and the lowest level 1 corresponds to with the minimal value. In formula (1),

$$Cap\_avai\_rate(a_{ij}) = \frac{Cap\_avai(a_{ij})}{Max_{a_{ij} \in A} Cap\_all(a_{ij})} \tag{4}$$

$$Cap\_avai(a_{ij}) = \sum_{\lambda_k \in Set\_wavelength\_avai(a_{ij})} Bandwidth(a_{ij}) \tag{5}$$

$$Cap\_all(a_{ij}) = \sum_{\lambda_k \in Set\_wavelength\_all(a_{ij})} Bandwidth(a_{ij}) \tag{6}$$

In formula (3)and(4), $Cap_avai(a_{ij})$denotes the absolute available capacity of the link a_{ij}, $Set\_wavelength\_avai(a_{ij})$ denotes the set of available wavelength on the link a_{ij},$Cap\_all(a_{ij})$ denotes the total communication capacity of the link a_{ij}, $Set\_wavelength\_all(a_{ij})$ denotes the set of all the wavelength on the link a_{ij}.

The state level of path P can be calculated by

$$Leve\_link\_state(P) = Min_{a_{ij} \in P}\{Leve\_link\_state(a_{ij})\}. \tag{7}$$

In formula (1,5,6), we suppose the value of $Bandwidth(a_{ij})$ on different link is different in general. Based on this, two different wavelength assignment methods named PWFPreceding Wavelength First, PWFand LBWFLeast Bandwidth Waste First, LBWFare proposed.

2.1 PWF Wavelength Assignment

Assume that wavelength λ_f is occupied on the link a_{ki} prior to node v_i and the set of the available wavelengths on the following link $a_{ij}$is$Set_wavelength_avai(a_{ij})$, and $\lambda_f \in Set\_wavelength\_avai(a_{ij})$, we will set λ_f as the working wavelength on link a_{ij}, otherwise we will set $\lambda_c \in Set\_wavelength\_avai(a_{ij})$ as the working wavelength on a_{ij} link randomly.

2.2 LBWF Wavelength Assignment

In LBWF, we suppose that the communication requirement and the corresponding bandwidth of each wavelength because the available bandwidth might be different, and the bandwidth request on the link a_{ki} is F , if $Bandwidth(\lambda_c)$ of λ_c on link a_{ij} satisfies

$$Bandwidth(\lambda_c) = Min_{\lambda_k \in Set\_wavelength\_avai(a_{ij})}\{Bandwidth(\lambda_c) \geq F\}. \tag{8}$$

we will firstly set the λ_c as the working wavelength on link a_{ij} .

Notes: For both of the wavelength assignment methods, the first aims at minimizing the times of the wavelength conversion, and the second is to minimize the waste of the available bandwidth. So both of them may be selected according to the different optimal objectives.

3 Routing and Wavelength Assignment Algorithm -Tradeoff_LSDRWA

In Tradeoff_LSDRWA, according to the state level of links and distance, the cost function of the link is firstly calculated for each link according to the equation

$$Cost(a_{ij}) = \begin{cases} b_l * [n - Level(a_{ij})] + b_d * Dis(a_{ij}) & \text{if } n \geq 1, \\ \infty & \text{if } n = 0. \end{cases} \tag{9}$$

where b_l and b_d is the weighed coefficient of state level and the distance of the link respectively, and $b_l \geq 0$,$b_d \geq 0$, n is the number of the state level of links, $Dis(a_{ij})$ is the distance of the arc a_{ij} , and $Level(a_{ij})$ is the state level of the current arc a_{ij} .

The cost function of wavelength conversion is formulated as the following:

$$Convert(v_i) = \begin{cases} M & \text{if there is a wavelength conversion,} \\ 0 & \text{if there is no wavelength conversion.} \end{cases} \tag{10}$$

The Tradeoff_LSDRWA algorithm is described as following:

Step 1. Let $D = (V, A, S)$ be the WDM network, s and d be the source node and the destination node respectively, s,$d \in V$, U={s} , $Cost(s) = 0$and for any other node $v \in V$, $Cost(s) = \infty$, $i = 0$;

Step2. we assume that a_{ti} is the arc from $v_t$tov_i , the working wavelength is $\lambda_{w_{ti}}$ on the a_{ti}, $v_i \in U$, $v_j \in V/U$, the cost of can calculated as the following

$$Cost(v_j) = Cost(v_i) + Cost(a_{ij}) + Convert(v_i) \tag{11}$$

Step3. Let u be the node whose cost equal to $Min_{v_j \in V/U} Cost(v_j)$, and if $u = d$, go to Step4;otherwise let $i = i + 1$,$U = U \bigcup \{u\}$ go to step2;

step4. End.

4 Example

In Figure 2, a path from the node $d$to node $c$is established according to the Tradeoff_LSDRWA algorithm, which is superior to that according to the existing algorithms for the problem in the WDM optical networks.

The different path may be established based on the different algorithm, and they are described as the following respectively.

1) The path established according to the shortest path algorithm[5] is $d \rightarrow c$, whose level is 6 and the distance is 45, and the working wavelength on $d \rightarrow c$ is set as randomly.

2)If we only consider State Level of Links, the path $d \rightarrow a \rightarrow b \rightarrow c$ may be established, whose level and distance are 8 and 80 respectively, and the wavelength on the path are $\lambda_1, \lambda_3, \lambda_2$.

3)For the Tradeoff_LSDRWA, we suppose that $n = 10$,$b_l = 0.9$,$b_d = 0.1$, the path $d \rightarrow b \rightarrow c$ may be established, whose level is 7 and distance is 60, and the wavelength on it are λ_4 and λ_4.

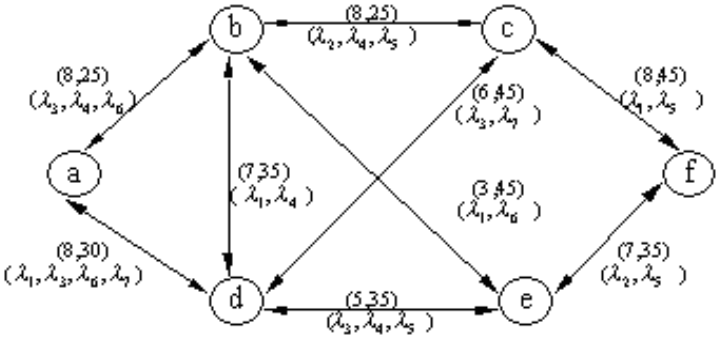


Fig. 2. The example WDM network

5 Conclusions

Based on the state level of links, the RWA problem of the WDM networks is studied and a dynamic RWA algorithm called Tradeoff_LSDRWA is proposed for the problem. On the one hand, we assume that the state level of links changes dynamically, so the algorithm proposed is also dynamic, on the other hand, the algorithm is relatively steady because of the stability of the state level of links because only when the enough great change of state level of links occurs, the state level of links is updated. The example network shows that the selected route by the Tradeoff_LSDRWA is of the higher state level and the shorter distance, and the Tradeoff_LSDRWA can achieve load balancing in WDM optical transport networks.

References

1. Thomas EStern,Krishna Bala(1998). Multiwavelength optical networks:A layered approach. Addison Wesley,Inc 1999
2. IChlamtacAFarago and TZhang(1996). Lightpath (Wavelength) routing in large WDM networks. *IEEE Journal of Selected Areas in Communications*, 14(5),909-913
3. Deying Li ,et al(2000). Minimizing number of wavelength in multicasting routing trees in WDM networks. *Networks*, 35(4),260-265
4. Aradhana Narula-Tam and Eytan Modiano(2000). Dynamic load balancing in WDM packet networks with and without wavelength constraints. *IEEE Journal of Selected Areas in Communications*, 18(10), 1972-1979
5. Dijkstra E W(1959). A note on two problems in connection with graphs. *Numer-Math*, 1, 269-271

Cooperative Determination on Cache Replacement Candidates for Transcoding Proxy Caching*

Keqiu Li[1], Hong Shen[1,2], and Francis Y.L. Chin[3]

[1] Graduate School of Information Science,
Japan Advanced Institute of Science and Technology,
1-1, Asahidai, Nomi, Ishikawa 923-1292, Japan
[2] Department of Computer Science and Technology,
University of Science and Technology of China,
Hefei, Anhui 230026, China
[3] Department of Computer Science and Information Systems,
University of Hong Kong, Pokfulam Road, Hong Kong

Abstract. Transcoding proxy caching is an important technology for improving the services over Internet, especially in the environment of mobile computing systems. In this paper, we address cooperative determination on cache replacement candidates for transcoding proxies. An original model which determines cache replacement candidates on all candidate nodes in a coordinated fashion with the objective of minimizing the total cost loss is proposed. We formulate this problem as an optimization problem and present a low-cost optimal solution for deciding cache replacement candidates.

1 Introduction

Web caching is an important technology for improving the services over Internet. Since the majority of web objects are static, caching them at various network components (e.g., client browser, proxy server) provides a natural way of decreasing network traffic. Moreover, web caching can also reduce users' access latency and alleviate server load.

A key factor that affects the performance of web caching is the cache replacement policy, which is a decision for evicting an object currently in the cache to make room for a new object. A number of cache replacement policies, which attempt to optimize various performance metrics, such as hit ratio, byte hit ratio, delay saving ratio, etc., have been proposed in the literature. However, all these polices are local replacement models that determine cache replacement candidates from the view of only a single node. Furthermore, they become inefficient in transcoding proxies due to the new emerging factors in the transcoding

* This work was partially supported by Japan Society for the Promotion of Science (JSPS) under its General Research Scheme **B** Grant No. 14380139). Corresponding author H. Shen (shen@jaist.ac.jp).

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 178–187, 2005.

proxy (e.g., the additional delay caused by transcoding, different sizes and reference rates for different versions of a multimedia object) and the aggregate effect of caching multiple versions of the same multimedia object. Although the authors have elaborated these issues in [1], they considered the cache replacement problem at only a single node. Cooperative caching, in which caches cooperate in serving each other's requests and making storage decisions, is a powerful paradigm to improve cache effectiveness [3,6]. There are two orthogonal issues to cooperative caching: object location (i.e., finding nearby copies of objects) and object management (i.e., coordinating the caches while making storage decisions). The object location problem has been widely studied [2,4,8]. Efficient coordinated object management algorithms are crucial to the performance of a cooperative caching system, which can be divided into two type of algorithms: placement and replacement algorithms. There are a number of research on finding efficient solutions for cooperative object placement [5,7,9]. However, there is little work done on finding efficient solutions for cooperative object replacement. Due to the interrelationship among different versions of the same multimedia object, cooperative caching in transcoding proxies becomes more important and complicated. We claim that this is very significant for the performance of a cooperative caching system since when a updated version is to be cached, an efficient replacement policy should decide cache replacement candidates by considering the cooperation of all the nodes on the path from the server to the client. Another important point is that the replacement decision on each node should be beneficial, i.e., the profit gained by caching the new object should be no less than the profit lost by removing some objects from the cache to make room for the new object. As the transcoding proxy is attracting an increasing amount of attention in the environment of mobile computing, it is noted that new efficient cache replacement policies are required for these transcoding proxies. In this paper, we address cooperative determination on cache replacement candidates for transcoding proxies. We first propose an original model which determines cache replacement candidates among all candidate nodes in a coordinated fashion with the objective of minimizing the total cost loss. Moreover, we formulate this problem of an optimization problem and present a low-cost optimal solution for deciding cache replacement candidates.

The rest of this paper is organized as follows: Section 2 introduces some preliminaries. We formulate the problem and present an optimal solution for this problem in Section 3. Finally, we conclude this paper in Section 4.

2 Preliminaries

We first introduce multimedia object transcoding in Section 2.1, and then notations and definitions in Section 2.2.

2.1 Multimedia Object Transcoding

Transcoding is used to transform a multimedia object from one form to another, frequently trading off object fidelity for size, i.e., the process of converting a

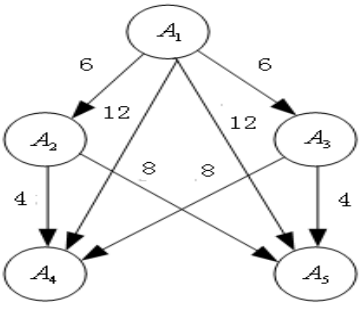


Fig. 1. An Example of A Weighted Transcoding Graph

media file or object from one format to another. Transcoding is often used to convert video formats (i.e., Beta to VHS, VHS to QuickTime, QuickTime to MPEG). But it is also used to fit HTML files and graphics files to the unique constraints of mobile devices and other Web-enabled products. These devices usually have smaller screen sizes, lower memory, and slower bandwidth rates. In this scenario, transcoding is performed by a transcoding proxy server or device, which receives the requested document or file and uses a specified annotation to adapt it to the client.

The relationship among different versions of a multimedia object can be expressed by a weighted transcoding graph. An example of such a graph is shown in Figure 1, where the original version A_1 can be transcoded to each of the less detailed versions A_2, A_3, A_4, and A_5. It should be noted that not every A_i can be transcoded to A_j since it is possible that A_i does not contain enough content information for the transcoding from A_i to A_j. In our example, transcoding can not be executed between A_4 and A_5 due to insufficient content information. The transcoding cost of a multimedia object from A_i to A_j is denoted by $w(i,j)$. The number beside each edge in Figure 1 is the transcoding cost from one version to another. For example, $w(1,2) = 6$, and $w(3,4) = 4$. $\phi(i)$ is the set of all the versions that can be transcoded from A_i, including A_i. For example, $\phi(1) = \{1,2,3,4,5\}$, $\phi(2) = \{2,4,5\}$, and $\phi(4) = \{4\}$. In this paper, we use G to denote a weighted transcoding graph.

2.2 Notations and Definitions

We model the network as a graph $G = (V, E)$ in this paper, where $V = \{v_0, v_1, \cdots, v_n\}$ is the set of nodes or vertices, and E is the set of edges or links. We assume that every node is associated with a cache with the same size B and there are m multimedia objects, i.e., $O_1, O_2, \cdots, O_l$, maintained by server v_0. For each multimedia object O_j, we assume that it has m_j versions: $O_{j,1}, O_{j,2}, \cdots, O_{j,m_j}$ and all versions have the same size. Thus, each node can hold at most B objects. We denote the set of objects cached at node v_i by $Y^i = \{A_1^i, A_2^i, \cdots, A_m^i\}$, where $A_j^i \subseteq \{O_{j,k_1}, O_{j,k2}, \cdots, O_{j,k_j}\}$ is the set of different versions of object O_j cached at node v_i. Obviously, $Y = \{Y^1, Y^2, \cdots, Y^n\}$

is the set of all objects cached. For each version of object O_j, we associate each link $(u, v) \in E$ a nonnegative cost $L_{j,k}(u, v)$, which is defined as the cost of sending a request for version $O_{j,k}$ and the relevant response over the link (u, v). In particular, $L_{j,k}(u, u) = 0$. If a request goes through multiple network links, the cost is the sum of the cost on all these links. The cost in our analysis is calculated from a general point of view. It can be different performance measures such as delay, bandwidth requirement, and access latency, or a combination of these measures. Let $r_{i,j,k}$ denote the request for $O_{j,k}$ at node v_i and $f_{i,j,k}$ be the frequency of $r_{i,j,k}$.

For notational tidiness, we omit argument j in all parameters and functions throughout the following analysis since our analysis is based on a specific object. For example, O_k denotes version k of object j, A^i is the set of different versions of object j cached at node v_i, $L_k(u, v)$ denotes the cost of sending a request for version O_k and the relevant response over the link (u, v), $r_{i,k}$ denotes the request for O_k at node v_i, and $f_{i,k}$ denotes the frequency of $r_{i,k}$. We also make the following assumptions.

- *Assumption* 1*:* $L_k(v_{i_1}, v_{i_2}) = (i_1 - i_2)L$ for all $1 \leq k \leq m$ as there are $i_1 - i_2$ links on the path between node v_{i_1} and node v_{i_2}, and the cost on each link for each version of O_j is L.
- *Assumption* 2*:* The transcoding graph is a linear array and the transcoding cost between any two adjacent versions is constant, i.e., $t(O_{k_1}, O_{k_2}) = \sum_{k=k_1}^{k_2-1} t(O_k, O_{k+1}) = (k_2 - k_1)^+ T$, where $x^+ = x$ if $x \geq 0$ else $x^+ = \infty$.
- *Assumption* 3*:* There exists some positive integer δ such that $(\delta - 1)T \leq L$, and $\delta T > L$. If there does not exist such a δ, i.e., $L \gg T$ or $T \gg L$. Obviously, these are two trivial cases.

3 Cooperative Cache Replacement for Transcoding Proxies

3.1 Problem Formulation

Before formulating the problem, we give some explanation on how the requests are served. As shown in Figure 2, a request goes along a routing path from the client (node v_n) to the server (node v_0). Note that any request $r_{i,k}$ could find the service from $S(r_{i,k})$, where $S(r_{i,k})$ denotes the serving object for $r_{i,k}$. Assume that $S(r_{i,k}) = O_{k_1} \in A^{i_1}$ with $k_1 \leq k$ and $i_1 \leq i$, then there may be the following ways of serving $r_{i,k}$ by $O_{k_1} \in A^{i_1}$.

- O_{k_1} is first sent from node v_{i_1} to node v_i and then transcoded to O_k at node v_i.
- O_{k_1} is first transcoded to O_k at node v_{i_1} and then O_k is sent from node v_{i_1} to node v_i.
- O_{k_1} is first sent from node v_{i_1} to node v_{i_2}, transcoded to O_k at node v_{i_2}, and then then O_k is sent from node v_{i_2} to node v_i.

– O_{k_1} is first sent from node v_{i_1} to node v_{i_2} and transcoded to O_{k_2} at node v_{i_2}, and then O_{k_2} is sent from node v_{i_2} to node v_{i_3} and transcoded to O_{k_3} at node v_{i_3}, then O_{k_3} is sent from node v_{i_3} to node v_i and transcoded to O_k at node v_i.

– ⋮

Fig. 2. System Model for Multimedia Object Caching

All these cases would cost the same under our cost model even though in practice. However, when a new or updated version of a multimedia object to be cache, denoted by O_{i_0}, is passing through each node between nodes $v_{i'}$ and v_i, it should be decided where O_{i_0} should be cached and which version should removed from the relevant cache to make room for it depending on how $r_{i,k}$ is served. Given X (i.e., the set of cached objects) and $O_{k'} \in A^{i'}$ $(i' \leq i)$. Let $d(r_{i,k}, O_{k'})$ denote the cost of serving $r_{i,k}$ by $O_{k'}$ at node $v_{i'}$. Then $d(r_{i,k}, O_{k'})$ is defined as follows:

$$d(r_{i,k}, O_{k'}) = (i - i')L + (k - k')^+ T \tag{1}$$

where $(x-y)^+ = \begin{cases} x-y & \text{if } x-y \geq 0 \\ 0 & \text{if } x-y < 0 \end{cases}$

Now we begin to formulate the problem addressed in this paper, i.e., determining where a new or updated version O_{i_0} should be cached among nodes $\{v_1, v_2, \cdots, v_n\}$ and which version of object j should be removed at that node to make room for O_{i_0} such that the total cost loss is minimized. Suppose that $P \subseteq V$ is the set of nodes at each of which $X_{i,k_i} \in A^i$ should be removed to make room for O_{i_0}, then this problem can be formally defined as follows:

$$L(P^*) = \min_{P \subseteq V} \{L(P)\} = \sum_{v_i \in P} (l(X_{i,k_i}) - g_i(O_{i_0})) \tag{2}$$

where $L(P)$ is the total relative cost loss, $l(X_{i,k_i})$ is the cost loss of removing X_{i,k_i} from node v_i, and $g_i(O_{i_0})$ is the cost saving of caching O_{i_0} at node v_i.

3.2 Dynamic Programming-Based Solution

Before presenting the solution, we evaluate the two items, i.e., $l(X_{i,k_i})$ and $g_i(O_{i_0})$, shown in Equation (2) in detail .

First, we begin with presenting a solution for finding the best way of serving $r_{i,k}$, i.e., finding $S(r_{i,k})$. Based on Equation (1), the cost of serving $r_{i,k}$, denoted by $c(r_{i,k})$, is defined as follows:

$$c(r_{i,k}) = \min \left\{ \min_{O_{k'} \in A^{i'},\ 1 \le i' \le i} d(r_{i,k}, O_{k'}), iL \right\} \tag{3}$$

Therefore, the object for serving $r_{i,k}$, denoted by $S(r_{i,k})$, is determined as follows:

$$S(r_{i,k}) = \begin{cases} O_{k'} \in A^{i'} & \text{if } c(r_{i,k}) \ge d(r_{i,k}, O_{k'}) \\ v_0 & \text{if } c(r_{i,k}) = iL \end{cases} \tag{4}$$

The following property will help us simplify the problem of finding the best way of serving $r_{i,k}$.

Theorem 1. *If both O_{k_1} and O_{k_2} are cached at node $v_{i'}$, then we have $d(r_{i,k}, O_{k_1}) < d(r_{i,k}, O_{k_2})$ for $k > k_1 > k_2$.*

Proof. Based on the definition of $d(r_{i,k}, O_k)$, we have $d(r_{i,k}, O_{k_1}) = (i - i')L + (k - k_1)^+ T$ and $d(r_{i,k}, O_{k_2}) = (i - i')L + (k - k_2)^+ T$. Since $(k - k_1)^+ < (k - k_2)^+$, we have $d(r_{i,k}, O_{k_1}) < d(r_{i,k}, O_{k_2})$. Hence, the theorem is proven.

From Theorem 1, we can see that for request $r_{i,k}$, we can consider only the least detailed version that can be transcoded to version k. Thus, Equation (3) can be simplified as follows:

$$c(r_{i,k}) = \min \left\{ \min_{1 \le i' \le i} d(r_{i,k}, O_{k^*}), iL \right\} \tag{5}$$

where O_{k^*} is the least detailed version of object j cached at node $v_{i'}$ that can be transcoded to version k.

It is easy to see that the time complexity for computing $S(r_{i,k})$ is $O(\log n)$, where n is the number of nodes in the network. So the total complexity for computing all $S(r_{i,k})$ ($1 \le i \le n$ and $1 \le k \le m$) is $O(mn \log n)$ since there are n nodes and object j has m different versions.

For each object $x \in X$, the set of requests served by x is expressed as $R(x) = \{r_{i,k} | S(r_{i,k}) = x\}$ and the total cost for the requests served by x is $C(x) = \sum_{r_{i,k} \in R(x)} f_{i,k} d(r_{i,k}, x)$. In this paper, we use R_s to denote the set of requests served by the server.

Regarding to $R(x)$, we have the following property.

Property 1. If $r_{i,k} \in R(x)$, then $r_{i',k'} \in R(x') \ \forall\ i' \le i$ and $k' \le k$.

Proof. Suppose that $x \in A^{i_1} = O_{k_1}$, $x' \in A^{i_2} = O_{k_2}$ and there exists $i' \le i$ and $k' \le k$ such that $r_{i',k'} \in R(O_{i_2})$. Since $S(r_{i',k'}) = x'$, we have $d(r_{i',k'}, x') \le d(r_{i',k'}, x)$. Therefore we have $(i' - i_2)L + (k' - k_2)T \le (i' - i_1)L + (k' - k_1)T$, i.e., $(i_2 - i_1)L + (k_2 - k_1)T \le 0$. Therefore we have $d(r_{i,k}, x) = (i - i_1)L + (k - k_1)T = (i - i_2)L + (k - k_2)T + (i_2 - i_1)L + (k_2 - k_1)T = d(r_{i,k}, x') + (i_2 - i_1)L + (k_2 - k_1)T \le d(r_{i,k}, x')$. So we have $S(r_{i,k}) = x'$, which contradicts $r_{i,k} \in R(x)$. Hence, the property is proven.

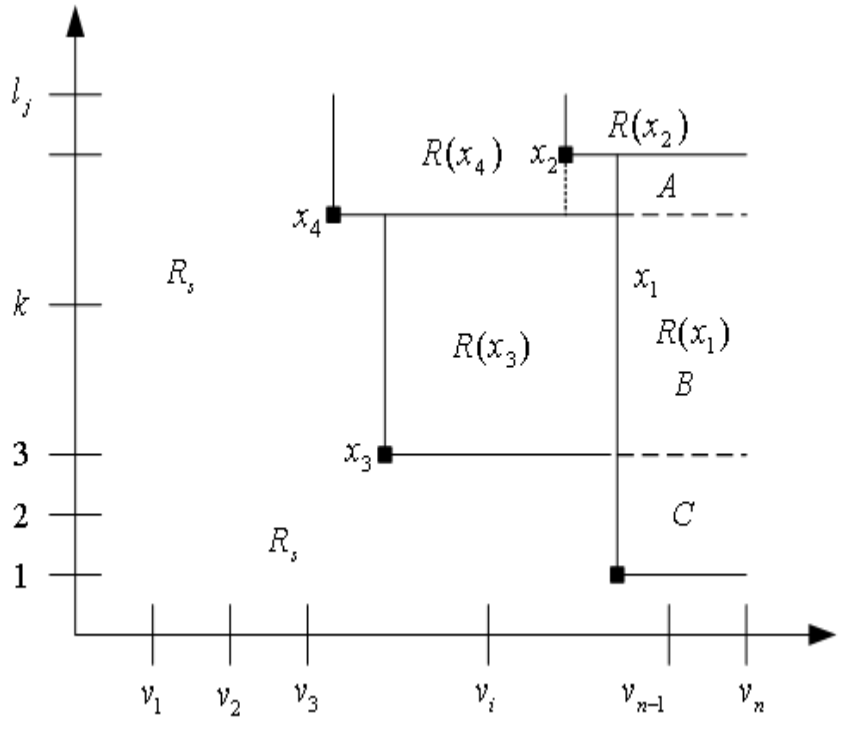


Fig. 3. Example for Calculating $l(x)$

From Property 1, we can see that $R(x)$ should be a region that can be divided into several rectangular regions. This can be seen from Figure 3. For example, $R(x_4)$ can be divided into two regions by the vertical broken line from x_2.

Regarding to calculating $l(X_{i,k_i})$, we first give the following theorem.

Theorem 2. *Suppose that only X_{i,k_i} is cached at node v_i, then we have* $l(X_{i,k_i}) = \sum_{r_{i,k} \in B_0} f_{i,k}[i \cdot L - d(r_{i,k}, X_{i,k_i})] + \sum_{i=1}^{n} \sum_{r_{i,k} \in B_i} f_{i,k}[d(r_{i,k}, X^i_{k_i}) - d(r_{i,k}, X_{i,k_i})]$, *where* $B_0 = \{(\alpha, \beta) | \alpha = i_0, \beta \in R_0 \bigcap R(X_{i,k_i})\} \bigcap R(X_{i,k_i})$ *and* $B_i = \{(\alpha, \beta) | \alpha = i_0, \beta \in R(X^i_{k_i}) \bigcap R(X_{i,k_i})\} \bigcap R(X_{i,k_i})$.

Proof. It is obvious that $B_i \bigcap B_j = \phi$ for $i \neq j$. This guarantees that each request's access cost is only calculated one time. Now we prove the correctness of the calculation of $l(X_{i,k_i})$, i.e., the requests in B_i should be served by $X^i_{k_i}$. Suppose that there exists a request $r_{i',k'} \in b_i$ which is not served by $X^i_{k_i}$. Based on Property 1, we have all the requests in the region $B'_i = \{(\alpha, \beta) | i \leq \alpha \leq i_0, k_i \leq \beta \leq k_0$ will be not served by $X^i_{k_i}$. It is easy to see that $R(X^i_{k_i}) \bigcap B'_i \neq \phi$, i.e., there exist some requests in region $R(X^i_{k_i})$ that are not served by $X^i_{k_i}$. This

obviously contradicts the fact that all the requests in region $R(X^i_{k_i})$ are served by $X^i_{k_i}$. Hence, the theorem is proven.

For example, in Figure 3, if x_1 is removed, $R(x_1)$ can be divided into three regions (i.e., A, B, and C), which will be served by x_4, x_3, and the server, respectively. Thus, we have $l(x_1) = \sum_{r_{i,k} \in A} f_{i,k}[d(r_{i,k}, x_4) - d(r_{i,k}, x_1)] + \sum_{r_{i,k} \in B} f_{i,k}[d(r_{i,k}, x_3) - d(r_{i,k}, x_1)] + \sum_{r_{i,k} \in B} f_{i,k}[i \cdot L - d(r_{i,k}, x_1)]$.

In practice, the general case is that several versions of the same multimedia object are cached at node v_i at the same time (see Figure 4). In this case, calculating $l(x)$ should also consider the mutual effect of the least more detailed cached version on the removed version since the requests served by the removed version could be satisfied by this detailed version. For example, when calculating $l(x_2)$, $R(x_2)$ might be divided into four parts A, B, C, and D which will be served by x_4, x_5, x_3, and x_1, respectively.

Taking into consideration the caching dependence along the path, calculating $l(X_{i,k_i})$ becomes more complex and it is so obvious to obtain an optimal solution.

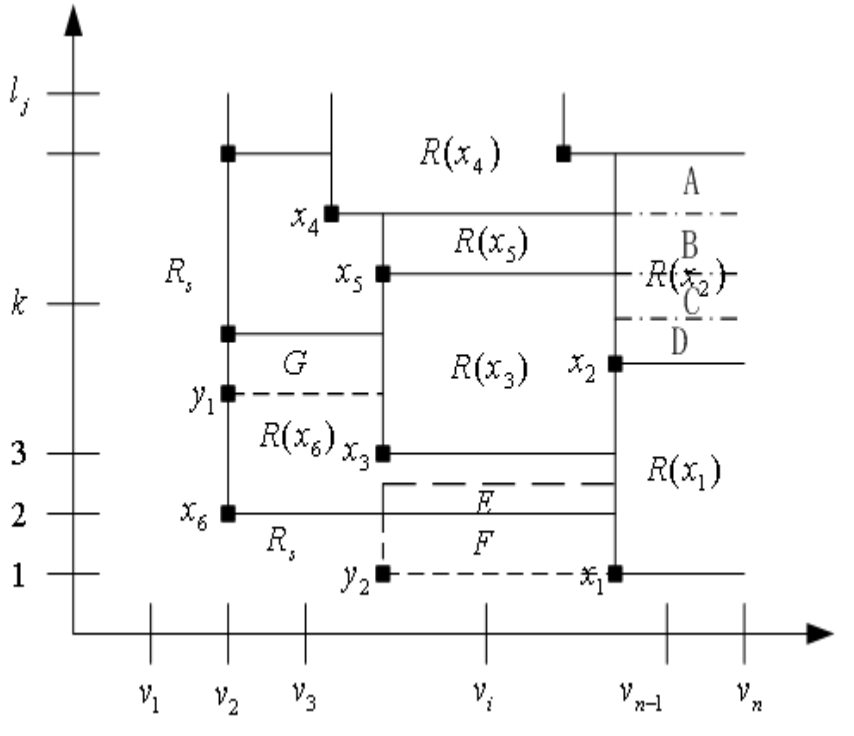


Fig. 4. Example for Calculating $l(x)$

Similarly, we can calculate the cost saving of caching O_{i_0} at node v_i. For example in Figure 4, if $i_0 = y_1$, then $R(x_6$ can be divided in to two parts: E and F; if $i_0 = y_2$, then $R(x_6$ can also be divided in to two parts. So we have $g(y_1) = \sum_{r_{i,k} \in G} f_{i,k}[d(r_{i,k}, y_1) - d(r_{i,k}, x_6)]$ and $g(y_2) = \sum_{r_{i,k} \in E} f_{i,k}[d(r_{i,k}, x_6) - d(r_{i,k}, y_1)] + \sum_{r_{i,k} \in F} f_{i,k}[i \cdot L - d(r_{i,k}, y_1)]$.

Now we begin to present an optimal solution for the problem as defined in Equation 2. In the following, we call the problem a k-optimization problem if we determine cache replacement candidates from nodes $\{v_1, v_2, \cdots, v_k\}$. Thus, the original problem (Equation (2)) is an n-optimization problem. Theorem 3 shows an important property that the optimal solution for the whole problem must contain optimal solutions for some subproblems.

Theorem 3. *Suppose that* $X = \{X_{i_1,k_{i_1}}, X_{i_2,k_{i_2}}, \cdots, X_{i_\alpha,k_{i_\alpha}}\}$ *is an optimal solution for the* α*-optimization problem and* $X' = \left\{X_{i'_1,k_{i'_1}}, X_{i'_2,k_{i'_2}}, \cdots, X_{i'_\beta,k_{i'_\beta}}\right\}$ *is an optimal solution for the* $k_{i_\alpha} - 1$*-optimization problem. Then* $X^* = \left\{X_{i'_1,k_{i'_1}}, X_{i'_2,k_{i'_2}}, \cdots, X_{i'_\beta,k_{i'_\beta}}, X_{i_\alpha,k_{i_\alpha}}\right\}$ *is also an optimal solution for the* α*-optimization problem.*

Proof. By definition, we first have $L(X^*) = l(X_{i'_1,k_{i'_1}}) + l(X_{i'_2,k_{i'_2}}) + \cdots + l(X_{i'_\beta,k_{i'_\beta}}) + l(X_{i_\alpha,k_{i_\alpha}}) = L(X') + l(X_{i_\alpha,k_{i_\alpha}}) \geq l(X_{i_1,k_{i_1}}) + l(X_{i_2,k_{i_2}}) + \cdots + l(X_{i_\beta,k_{i_\beta}}) + l(X_{i_\alpha,k_{i_\alpha}}) = L(X)$. On the other hand, since X is an optimal solution for the α-optimization problem, we have $L(X) \geq L(X^*)$. Therefore,we have $L(X) = L(X^*)$. Hence, the theorem is proven.

Based on Theorem 3, an optimal solution for the n-optimization can be obtained by checking all possible removed candidates from node v_1 to node v_n in order. Therefore, it is east to get that the time complexity of this solution is $O(n^2 + mn \log n)$ based on our previous result that the complexity for computing all $S(r_{i,k})$ is $O(mn \log n)$, where n is the number of nodes in the network and m is the number of versions of object j.

4 Conclusion

The transcoding proxy is attracting more and more attention since it plays an important role in the functionality of web caching. In this paper, we presented a coordinated cache replacement model in transcoding proxies where multimedia object placement and replacement policies are managed in a coordinated way. Our model is formulated as an optimization problem and the optimal solution is obtained using a low-cost dynamic programming-based solution.

References

1. C. Chang and M. Chen. *On Exploring Aggregate Effect for Efficient Cache Replacement in Transcoding Proxies*. IEEE Trans. on Parallel and Distributed Systems, Vol. 14, No. 6, pp. 611-624, June 2003.
2. A. Chankhunthod, P. Danzig, C. Neerdaels, M. Schwartz, and K. Worrell. *A Hierarchical Internet Object Cache*. Proc. of the USENIX Technical Conference, pp. 22-26, 1996.
3. M. D. Dahlin, R. Y. Wang, T. E. Anderson, and D. A. Patterson. *Cooperative Caching: Using Remote Client Memory to Improve File System Performance*. Proc. of First Symp. Operating Systems Design and Implementations, pp. 267-280, 1994.
4. L. Fan, P. Cao, and J. Almeida. *Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol*. Proc. of ACM SIGCOMM Conference, pp. 254-265, 1998.
5. K. Li, H. Shen, F. Chin, and S. Zheng. *Optimal Methods for Coordinated En-Route Web Caching for Tree Networks*. ACM Trans. on Internet Technology (TOIT), Vol. 5, No. 2, May 2005.
6. M. R. Korupolu and M. Dahlin. *Coordinated Placement and Replacement for Large-Scale Distributed Caches*. IEEE Trans. on Knowledge and Data Engineering, Vol. 14, No. 6, pp. 1317-1329, 2002.
7. X. Tang and S. T. Chanson. *Coordinated En-Route Web Caching*. IEEE Trans. on Computers, Vol. 51, No. 6, pp. 595-607, June 2002.
8. X. Tewari, M. Dahlin, H. M. Vin, and J. S. Kay. *Design Considerations for Distributed Caching on the Internet*. Proc. of the 19th Int'l Conference Distributed Computing Systems (ICDCS), pp. 273-284, 1999.
9. J. Xu, B. Li, and D. L. Li. *Placement Problems for Transparent Data Replication Proxy Services*. IEEE Journal on Selected Areas in Communications, Vol. 20, No. 7, pp. 1383-1398, 2002.

High Performance Embedded Route Lookup Coprocessor for Network Processors[1]

Kai Zheng, Zhen Liu, and Bin Liu

Department of Computer Science, Tsinghua University, Beijing, P.R.China, 100084
Phone: +86-10-62773441, Fax: +86-10-62773616
{Zk01, Liuzhen02}@mails.tsinghua.edu.cn
Liub@tsinghua.edu.cn

Abstract. Embedded Route Lookup Coprocessors (RLCs) are attractive for their potential in building high-performance Network Processors. But compared with conventional lookup schemes, it always imposes more severe restrictions on table size and power consumption, which poses challenge in the state of art. In this paper, we propose a novel lookup mechanism, Compounded CAM with Optimized Bitmap Compression (CCAM-OBC), which employs different lookup methods for prefixes of different length ranges, so as to combine the benefits of CAMs and bitmap compressed tries. With this scheme, table size, power consumption and update complexity are all well optimized while very high lookup throughput is achieved, which makes it a perfect solution to embedded RLC. For a real-life 130K-prefix route table, the implemented prototype performs more than 100 Million Packets Per Second (MPPS) with only 24KB TCAM, 48KB BCAM and 251KB SRAM. Furthermore, each update needs only 2 memory accesses averagely.

1 Introduction

IP address lookup is one of the key issues in designing high performance routers and Network Processors (NPs). The challenge arises from that: a) the length of IP prefix is variable; one IP address may match multiple prefixes in the forwarding table, and the longest matching prefix (LMP) should be chosen; b) advances in fiber-optic technology is pushing the line rate of core routers to 40Gbps or even higher, which implies that a single line card would support the packet processing rate of more than 100Mpps (packets per second). Also note that, with the rapid progress of the Very Large Scale Integrated (VLSI) technology, accessing rate of the memories can be further improved and the pin resource can be significantly saved if the memories are embedded into the host NPs. Therefore it would be both feasible and attractive to design high performance

[1] This work is supported by NSFC (No. 60173009 and 60373007), China 863 High-tech Plan (No. 2002AA103011-1 and 2003AA115110), China/Ireland Science and Technology Collaboration Research Fund (CI-2003-02) and the Specialized Research Fund for the Doctoral Program of Higher Education of China (No. 20040003048).

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 188 – 197, 2005.

embedded Route Lookup Coprocessors (RLCs) for NPs. However, this makes the RLC design even more challenging, because it demands not only high lookup operation throughput, but also small memory requirements and low power consumption.

Currently, packet forwarding based on LMP address lookup is well understood with both trie-based algorithms and TCAM-based schemes. Classic trie-based algorithm is a kind of time-consuming algorithm. Gupta et al. employs indirect lookup schemes for LMP search. The DIR-21-3-8 algorithm [1] furnishes LMP lookup at hardware accessing speed; but the memory requirement is relatively large that more than 9MB DRAM is required. This makes it impracticable to be embedded into the NPs. The idea of Degermark's SFT [2] and Huang's BC-16-16 [3] are very similar, both of which use a called Bitmap Compression (BC) technique to improve storage efficiency. The idea is to use bit vectors to represent part of the prefix trie. With certain amount of pre-calculation, fast LMP lookup can be achieved while the memory requirement is fairly small. However, these algorithms have a fatal flaw of not supporting incremental updates.

Ternary Content Addressable Memory (TCAM) is a promising device to build a high-speed LMP lookup engine, because it returns the matching result within a single memory access cycle. However, the high cost to density ratio and low power efficiency of the TCAM make it very hard for embedded RLC.

In this paper, we propose a fast IP lookup scheme called CCAM-OBC, which combines the benefits of both CAMs and BC-tries. High performance is achieved while the memory requirement of CCAM-OBC is small enough for building embedded RLC, and the proposed optimization of the BC algorithm effectively solves the "Hard-to-Update" problem.

2 Definitions

For the purpose of better understanding the data structure used in our design, a full binary trie is introduced to represent the whole prefix space, with one node for each possible prefix. The prefix of a route table entry defines a path in the trie ending in

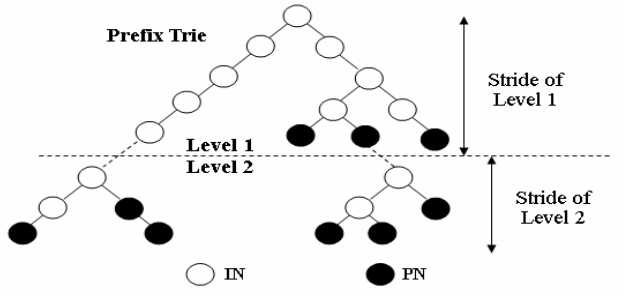


Fig. 1. Prefix Trie and the corresponding definitions

some node, which is called Prefix Nodes (PN) in our scheme; if a node itself is not a PN but its descendants include PNs, then we call it an Internal Node (IN) (See Fig. 1). Node that is either a PN or an IN is called Genuine Node (GN), which in fact carries route information. We let GN(i) denotes the set of all of the GNs on depth i of the prefix trie, while |GN(i)| denote the number of nodes that GN(i) contains.

For each IP address lookup operation, it may contain 1 to 2 phases, which are called the Level 1 (L1) lookup and the Level 2 (L2) lookup respectively (Also see Fig. 1). Stride is defined as the number of consecutive bits in the destination IP address used in L1 or L2.

3 Constructing the Data Structure

3.1 Some Useful Observations

We have analyzed a large amount of real-life route tables collected from four famous route service projects [4-7], and get some useful characteristics of the prefix distribution. In order to illustrate these characteristics more clearly, here we pick four typical ones from these route tables, as is described in Table 1. These route tables are both spatially and temporally widely distributed, which ensures that the characteristics discussed here are not specific to a particular router or time interval.

Table 1. Four real-life route tables

Name of Data Base	Date	Number of Prefixes	Number of Next Hop
Mae-West [4]	2001-03	33,960	45
SD_NAP [5]	2001-06	3,935	2
Route View [6]	2003-10	123,384	4
RRC06 [7]	2003-11	131,372	35

3.1.1 Characteristic I

As shown in Fig. 2 (note the logarithmic scale on the y-axis), despite the elapse of time, the prefix length distributions keep a relatively stable form: The historical 24-bit Class C Prefix still dominates the number of entries (about 50% alone); the ratio of prefixes longer than 24-bit is very tiny (less than 1% in each of the four cases); and over 90% of the prefixes are between 18-bit and 24-bit.

3.1.2 Characteristic II

As shown in Fig. 3, the Genuine Ratio (GR) of depth i, defined as the ratio of |GN(i)| to 2^i (the total number of nodes on Depth i), is relatively low when i is larger than 16, and GR drops sharply as i grows larger. This characteristic is very helpful for saving memory requirement (the reason will be revealed shortly in Section 3.3).

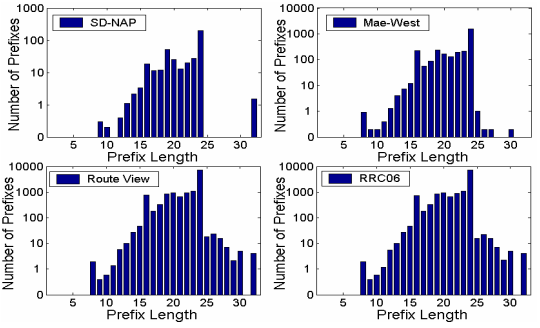


Fig. 2. Prefix length distribution of 4 typical routers

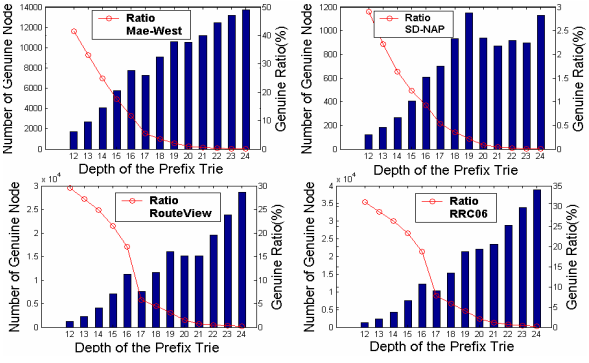


Fig. 3. Numbers and ratios of GNs

3.2 Principles of the Architecture Construction – Prefixes Set Partitioning

The significance of these observations is that they help us make clever decisions in forwarding table architecture design. One of the main ideas is that we may choose different lookup schemes for prefixes with different length ranges. In this paper, we divide the route prefixes into three parts according to their lengths and apply different lookup schemes to them respectively: 1) According to Characteristic I, few prefixes (e.g. <1%) are longer than 24-bit, and note that it is just this minority that has the highest processing and storing complexity. So we prefer to use TCAM for the associated search. 2) For the majority of the prefixes whose lengths are between 18-bit and 24-bit (over 90%), using TCAM is too expensive and power consuming for an embedded RLC solution. So a trie-based indirect lookup scheme seems more appropriate. 3) According to the thought in indirect lookup scheme with stride of L1 segment table being s, a prefix of length $l<s$ should be expanded to $2^{(s-l)}$ s-bit prefixes. This expansion not only wastes memory but also increases the update complexity. Since the prefixes shorter than 18-bit is very few, we choose TCAM for them again.

3.3 Adopting BCAM

Compared to TCAM, Binary CAM (BCAM) also returns the matching result within a clock cycle, except that BCAM cannot handle variable-length lookup. However, the cost and power consumption of BCAM are much less than those of TCAM.

As is mentioned in the Definition Section, we apply a 2-level indirect lookup mechanism to the prefixes between 18-bit and 24-bit. And the lookups on the first level are actually fixed-length search, thus it is advisable to introduce BCAM into our lookup scheme. In contrast, according to Characteristic II, using SRAM to construct a segment table at L1 will waste large amount of memory space because of the low ratio of the GNs. For instance, the BC-16-16 scheme uses a full hash segment table on L1 (stride=16), so there should be fixedly 2^{16}=64K 24-bit SRAM entries (totally 192Kbyte) in any cases. As far as the SD_NAP (3935 prefixes) route table is concerned, there are only 692 GNs on depth 16, which means only $692/2^{16}$ <1% of the memories store genuine information, and the other 99% are wasted in storing duplicated information.

3.4 The Optimized Bitmap Compression (OBC) Algorithm

The Compression Technique helps reduce storage requirement. However the native BC-Trie algorithm does not allow incremental updates, because in most cases one prefix may relate to lots of other prefixes. As is mentioned above, we perform indirectly lookup for prefixes between 18-bit and 24-bit. We chose 18 as the stride of L1 (where BCAM is used), therefore the height of a L2 sub-trie (where BC-Trie is used) is at most 24-18=6 and the maximum number of prefixes correlate with each other is at most 2^6=64. This implies that a data structure encapsulating at most 64 prefixes would be modified when we update a route prefix. As will be revealed shortly in the latter section, with the assists of a set of dedicated designed hardware logics, both lookup and update performance of the OBC algorithm are high enough for embedded RLCs.

For detailed idea of the OBC algorithm, imaging a cut through a certain L2 sub-trie, there would be totally $2^{(24-18)}$ =64 nodes on the cut (see Fig. 4). The PNs on the L2 sub-trie divide the cut into several prefix intervals (e.g. there are seven prefix intervals in the example). Note that each node within the same interval shares the same routing information (i.e. NHI). The idea of the OBC algorithm is to define a 64-bit Bitmap Vector (BV) for each L2 sub-trie, representing the 64 nodes on the cut. BV is used to indicate the borders of the prefix intervals: If a node on the cut has a different NHI (prefix interval) than its neighbor just before it, the corresponding bit is set, otherwise, zero. Since the nodes within an interval share the same NHI, we just store one NHI for each interval instead of one for each node. Then the pruning set of the NHIs belonging to a L2 sub-trie forms a Compressed NHI (CNHI) vector.

Now finding the NHI of a specific node on the cut turns out to be finding the CNHI in the corresponding CNHI vector. The number of '1's in the Bitmap Vector before the corresponding bit of a specific node on the cut indicates the location of the NHI in the CNHI vector.

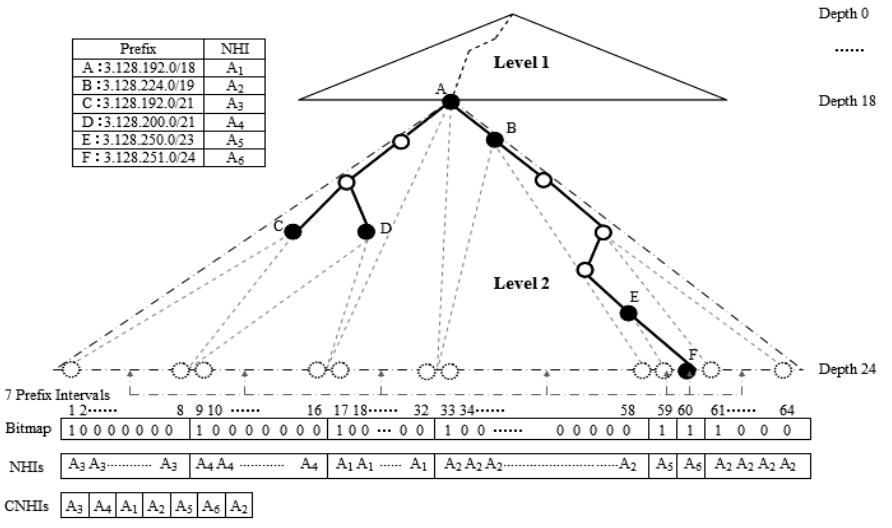


Fig. 4. A demonstration of the OBC algorithm

4 Hardware Implementation

The complete implementation scheme of the embedded RLC is shown in Fig. 5.

4.1 Level 1 Table Structure (Compound CAM)

As mentioned in the previous sections, we adopt different lookup schemes for prefixes of different lengths range. Both TCAM and BCAM are used in our scheme, and we call the combination of TCAM and BCAM used in the L1 lookup, the Compound CAM.

- For the set of prefixes whose lengths are longer than 24bits, we adopt a 32-bit-wide TCAM. The prefixes and masks are stored in the TCAM, while the corresponding NHIs are stored in the associated SRAM. For the set of prefixes whose lengths are shorter than 18bits, the memory organization is similar, except that the width of the TCAM is 17-bit.
- For the set of the prefixes whose lengths are between 18 and 24 bits, we use a BCAM to perform the L1 fixed-length lookup. Each GN on Depth18 of the prefix trie corresponds to a BCAM entry, which stores the entry to a L2 (OBC) data structure. Figure 6 shows the format of the associative SRAM of BCAM.

4.2 Level 2 Table Structure (The OBC Module)

The L2 offset table comprises of 4 key components (also see Fig. 5).

- Bitmap Array: The Bitmap Array is a SRAM, each word of which stores a 64-bit bitmap vector.

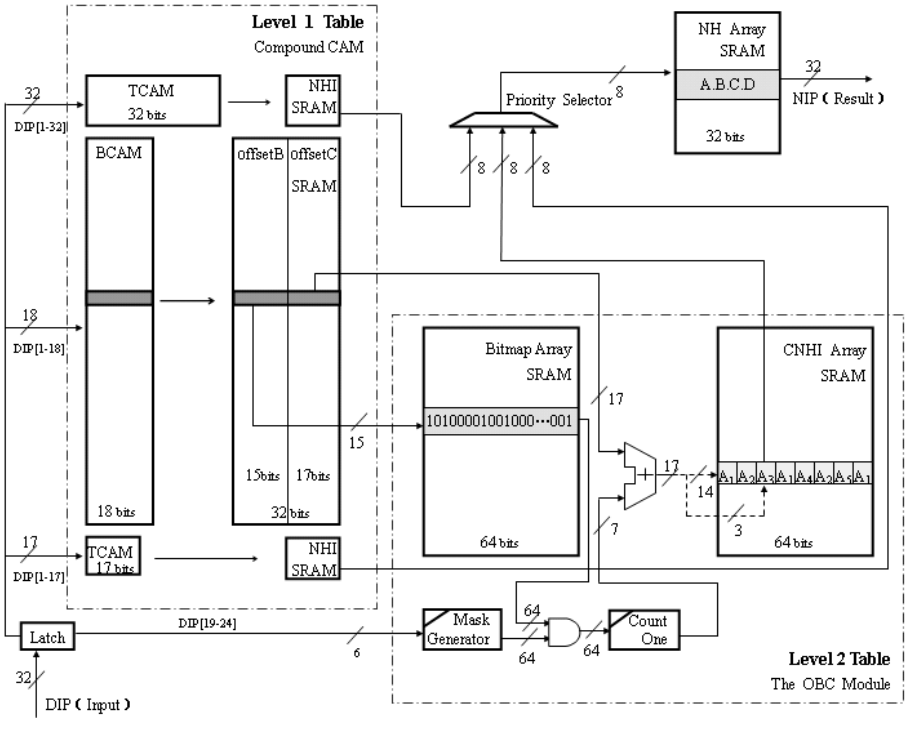


Fig. 5. Schematics of the complete implementation architecture

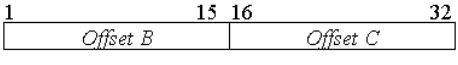


Fig. 6. Format of the SRAM associated with the BCAM

- CNHI Array: It is also a SRAM. Since each CNHI is 8-bit wide (an octet), then a single word in the CNHI Array can store 64/8=8 CNHIs. As shown in Fig. 6, we use a 17-bit (the 17th to 33rd bits of the L1 entry) pointer to locate a specific CNHI: The first 14 bits indicate the address of the 64-bit SRAM word, while the rest 3 bits indicate the position of the octet in the SRAM word.
- Mask Generator: According to the OBC algorithm, to perform a L2 OBC lookup, we need to count the number of 1's before a certain bit in a specific 64-bit Bitmap vector. The function of Mask Generator is to denote the boundary-bit in the 64-bit Bitmap Vector indicating that only the 1's before this boundary-bit should be counted. It is actually a simple decoder with a truth-value table as depicted in Table 2.
- Count-One Logic: Its task is to count the number of 1s in a given 64-bit word. It is actually a parallel adder, the function of which is to add up all of the bits within the given word.

Table 2. The truth value table of the Mask Generator

Input（DIP[19-24]）	Output（Mask[1-64]）
000000	00000……00000
000001	10000……00000
……	……
111111	11111……11111

4.3 Process of the CCAM-OBC Route Lookup Scheme

The lookup operation for the prefixes within the three length ranges (1-17bits, 18-24bits, and 25-32bits) are performed in parallel. The lookups for the prefixes within the first and the third length ranges are handled by TCAM, which is similar to that of conventional TCAM scheme. The search operation for prefixes within the second length range (18-24bits prefixes) employs indirect lookup: Firstly, the most significant 18bits of the IP address are extracted and sent to BCAM for an exact matching, then the pointer to the BV and the pointer to the CNHI vector are returned. Secondly, the 64-bit BV is obtained from the Bitmap Array via the corresponding pointer; meanwhile, the 19-24 bits of the IP address are extracted and sent to Mask Generator to form a 64-bit mask. And then the *AND* results of the BV and the Mask is sent to the Count-One logic. Then the result of Count-One logic indicates the position in which the corresponding CNHI lies within the CNHI vector. Finally, according to the three returning results (corresponding to the three length ranges), the longest and in-void one is selected as the finally results.

5 Experimental Results

5.1 Memory Requirement Evaluation

A comparison of memory requirements of different lookup schemes is shown in Table 3. Note that in the case of the smallest table, BC-16-16 has its fixed 192KB SRAM requirement (for the L1 segment table), which is too wasteful. On the other hand, in the largest route table case, conventional TCAM consumes more than 500KB, which is too expensive and high power consumption to be embedded.

Table 3. Memory requirement comparison between CCAM-OBC and other lookup schemes (Assuming that 1KB TCAM=2KB BCAM)

Schemes	SD NAP		Mae-West		Route View		RRC06	
	TCAM	RAM	TCAM	RAM	TCAM	RAM	TCAM	RAM
CCAM-OBC	2.2	14.4	19.1	140.5	44.5	181.5	47.7	250.8
TCAM	15.4	15.4	133	133	482	482	511	511
DIR-21-3-8	None	>9k	None	>9k	None	>9k	None	>9k
BC-16-16	None	213	None	391	None	574	None	617

5.2 Lookup Throughputs and Latency Evaluation

We use an ALTERA APEX II FPGA (EP2A15) to implement the prototyped CCAM-OBC mechanism. The synthesis result shows that the critical path delay is 8.47ns, indicating a working frequency of 118MHz. With a fully pipelined mechanism, a lookup throughput of up to 118Mpps can be achieved, supporting wire-speed forwarding of up to OC768 (40Gbps). According to the prototyped implementation, the longest pipeline is with seven stages (for the BCAM-OBC part). Hence the processing latency of a lookup operation is 59.29ns.

5.3 Update Performance

The update process of CCAM-OBC varies with the prefix length. Updating prefixes longer than 24bits or shorter than 18bits is the same with conventional TCAM scheme, only the number of entries is tiny and the length range is restricted. So, incremental update can be easily implemented using the algorithm presented in [8].

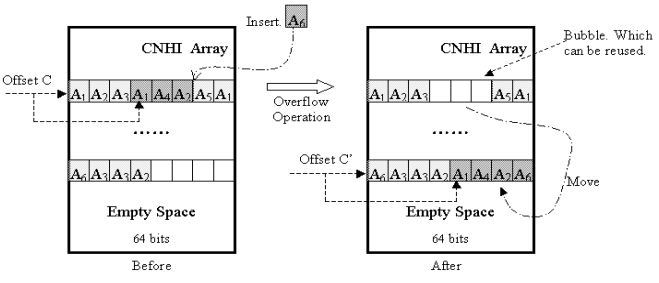


Fig. 7. Overflow caused by the insertion within the consecutive data structure

Updating a prefix between 18 and 24 bits is a little more complicated. If it is a simple next hop IP address alteration, only the corresponding CNHI will be modified. For a prefix deletion or insertion not causing the CNHI vector overflow, Bitmap and CNHI vector will be adjusted and in the worst case, 9 memory accesses are needed, one for Bitmap and eight for CNHI vector. In the case of a CNHI vector overflow (see Fig.7), new CNHI Array space should be allocated and the corresponding BCAM entry will be modified, too. However, the actual update operation will not cost so much. For instance, in the case of RRC06 route table, over 80% CNHI vectors contain less than 8 CNHIs, and the average number is only 5.8, which means in most cases, $\lceil 8bits \times 5.8/64bits \rceil = 1$ memory access is enough for a CNHI vector update.

6 Conclusions

In this paper, by analyzing quantities data of real-world route table data and adopting the strong points of both CAM-based lookup scheme and the Trie-based algorithms, we propose originally a novel IP lookup scheme, CCAM-OBC, for the embedded RLC

designs. In this scheme BCAM is, for the first time, introduced to perform the LMP IP address lookup. We optimize the original *Bitmap Compression* algorithm and solve the Hard-to-Update problem. The three goals of high lookup throughput, small memory requirement and low update complexity are achieved simultaneously, which makes CCAM-OBC especially suitable for embedded RLC.

References

1. Gupta, P., Lin, S., and McKeown, N.: Routing lookups in hardware at memory access speed, Proc. of IEEE INFOCOM'98 (1998) 1240-1247.
2. Degermark, M., Brodnik, A., Carlsson, S., and Pink, S.: Small Forwarding Tables for Fast Routing Lookups, Proc. of ACM/SIGCOMM'97 (1997) 3-14.
3. Huang, N.F., Zhao, S.M., Pan, J.Y., and Su, C.A.: A Fast Routing Lookup Scheme for Gigabit Switching Routers, Proc. of INFOCOM'99 (1999) Vol. 3, 21-25.
4. Database of the Mae-West router from the IPMA Project (A joint effort of the University of Michigan and Merit Network), http://www.merit.edu/ipma.
5. Database of the SD_NAP route server (sd-nap-dmz.pch.net), from, http://archive.pch.net/archive/
6. Database of the route-views.wide.routeviews.org route server from the Route-view Project (University of Oregon), http://www.routeviews.org.
7. Database of the RRC06 router from the RRCC Project (Routing Registry Consistency Check Project), http://www.ripe.net/rrcc/.
8. Shah, D., and Gupta, P.: Fast Updating Algorithms for TCAMs, IEEE Micro Magazine (2001), Vol. 21(1), 36-47.

An Efficient Distributed Dynamic Multicast Routing with Delay and Delay Variation Constraints*

Kun Zhang, Hong Zhang, and Jian Xu

Computer Department,
Nanjing University of Science & Technology,
210094 Nanjing, P.R.China
zhangkunw@263.net, njust@126.com

Abstract. The delay and delay variation-bounded multicast tree (*DVBMT*) problem is known to be NP-complete. In this paper, we propose an efficient distributed dynamic multicast routing algorithm to produce routing trees with delay and delay variation constraints. The proposed algorithm is fully distributed, and supports the dynamic reorganizing of the multicast tree in response to changes for the destination. Simulations demonstrate that our algorithm is better in terms of tree delay and routing success ratio as compared with other existing algorithms, and performs excellently in delay variation performance under lower time complexity, which ensures it to support the requirements of real-time multimedia communications more effectively.

1 Introduction

In real-time multicast applications, messages must be transmitted from the source node to their destinations within a certain amount of time which requires the communication to be done within a pre-specified end-to-end delay bound. The stringent delay constraint imposed on multimedia traffic to ensure that audio and video data are delivered smoothly to the audience. Besides, the multicast tree must also guarantee a bound on the variation among the delays along the individual source-destination paths, which can probably avoid causing inconsistency or unfairness problem among users. Such a bound provides synchronization among the various receivers and ensures that no receiver is "left behind" and that none is "far ahead" during the lifetime of the session.

Our research subject is concerned with multicast routing with delay and delay variation constraints. The issue first defined and discussed in Ref. [1] is that of minimizing multicast delay variation under multicast end-to-end delay constraint. The authors referred to this problem as *the delay and delay variation-bounded multicast*

* Supported by: (1)the National High Technology Research and Development Plan of China (No.2001AA113161); (2)the National Natural Science Foundation of China (No.60273035); (3)the Research and Development Foundation of Nanjing University of Science & Technology (No.96126).

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 198 – 207, 2005.

tree (*DVBMT*) problem, and proved it to be NP-complete. Table 1 gives a summary of some existing algorithms for multicast routing with delay and delay variation constraints.

Table 1. Some existing heuristics for *DVBMT* problem

Algorithm	Author	Complexity	Strength/weakness
DVMA	Rouskas[1]	$O(klmn^4)$	First heuristic for *DVBMT*, smart performance in terms of the delay variation, but high time complexity.
DDVCA	Sheu[2]	$O(mn^2)$	Based on CBT and minimum delay path, with a lower complexity and a satisfactory performance.
SP-DVMA	Yu[3]	$O(mkn^3)$	An improved version of DVMA, but increasing the scope of optional paths and causing the argument of the delay variation.

Notes: m is the number of destination nodes, n is the number of network nodes, k and l are the parameters of k-th shortest path algorithm.

However, there are two major difficulties in deployment and application of the existing algorithms to real-time communication networks. Firstly, most of the existing algorithms are centralized in nature. A centralized algorithm requires a central node to be responsible for computing the entire routing tree, and this central node must have the full knowledge about the global network. It suffers from some drawbacks in large networks, such as poor fault tolerance, heavy computing load at the central node, high communication cost in keeping network information up-to-date, and inaccuracy of routing information. The other difficulty is that, only a few existing algorithms are concerned with the dynamic change of multicast memberships. As we know, in many multicast applications, multicast participants are free to leave or join a multicast session dynamically. Therefore, it is important to ensure that any change of multicast memberships will not affect the traffic on the current connection, and the routing tree remains minimally disruptive to the multicast session.

To the best of our knowledge, little work has been done on finding delay and delay variation-bounded multicast routing tree in a distributed manner so far. In this paper, we propose a distributed multicast routing algorithm for obtaining multicast trees with delay and delay variation constraints, aimed at overcoming the above two difficulties.

The rest of the paper is organized as follows. Section 2 gives a formal definition of *DVBMT* problem. The proposed algorithm is described in Section 3. Section 4 presents an approach to dynamically reorganizing the tree in response to changes in multicast group. Simulation results are presented in Section 5. Section 6 concludes the paper.

2 The Definition of the *DVBMT* Problem

We represent a communication network by an undirected graph $G(V,E)$, where V denotes the set of nodes, and E, the set of edges, corresponds to the set of communication links connecting the nodes. Any link has a delay $d(e)$: $E \rightarrow R^+$ associated with it, where $d(\bullet)$ represents the delay that the packet experiences on link including queuing, transmission, and propagation delay. Let $s \in V$ be a source node and $M \subseteq V-\{s\}$ be the set of destination nodes, called the multicast group. A multicast tree $T(T \subseteq G)$ is a tree rooted at s and spanning the nodes in M. Let $p(u,v)$ denote the path from u to v. Then, multicast packets from u to v experience a total delay of $\sum_{e \in p(u,v)} d(e)$.

For the sake of convenience, we use Δ_T and δ_T to represent the multicast end-to-end delay and the multicast delay variation in a multicast tree T. Based on these definitions, we can formally present the *DVBMT* problem as follows:

Definition 1. *Delay and delay variation-bounded multicast tree* (*DVBMT*) *problem. Given a network* $G(V,E)$, *a source node* s, *destination node set* M, *a link delay function* $d(\bullet)$, *a positive delay bound* Δ *and a positive delay variation bound* δ, *the objective of the DVBMT problem is to construct a multicast tree* $T(V_T, E_T)$ *which spans s and M such that the delay and delay variation constraints are satisfied*, *i.e.*,

$$\Delta_T = \max_{m \in M} \left(\sum_{e \in p(s,m)} d(e) \right) \leq \Delta \tag{1}$$

$$\delta_T = \max_{u,v \in M} \left\{ \left| \sum_{e \in p(s,u)} d(e) - \sum_{e \in p(s,v)} d(e) \right| \right\} \leq \delta \tag{2}$$

3 Our Proposed Algorithm

In this section, we firstly give the assumptions and basic ideas of our algorithm. Then describe the proposed algorithm in detail. Finally, the analysis of correctness and complexity for our algorithm is discussed.

3.1 Assumptions and Basic Ideas

The basic idea of our proposed algorithm derives from following theorem.

Definition 2. *Adding a path* $p(u,v)$ *into a tree* T *refers to all nodes and links on the path are included into the tree, denoted by* $T + p(u,v)$.

Theorem 1. *Given a network* $G(V,E)$, *a source node s, destination set* M. Δ *and* δ *are the delay bound and the delay variation bound of multicast session respectively. Suppose* T' *is a subtree, and* $\Delta_{T'} \leq \Delta$, $\delta_{T'} \leq \delta$. $Sub(M)$ *is the destinations covered in* T' *so far, and* $Sub(M) \subset M$. *We use maxd and mind to represent the maximal delay and minimal delay of the path among the paths from s to each destination of* $Sub(M)$ *in* T', *respectively.* $\forall m \in M$, $m \notin Sub(M)$, *if* $p(s,m)$ *satisfies*,

$$\max\{0, maxd - \delta\} \le d(p(s,m)) \le \min\{mind + \delta, \Delta\} \tag{3}$$

$$T = T' + p(s,m) \tag{4}$$

then $\Delta_T \le \Delta$, $\delta_T \le \delta$.

Proof. See[4].

Theorem 1 shows that during of constructing a multicast tree which meets delay and delay variation constraints, if the delay of a path from *s* to next uncovered destination satisfies (3), and then the tree after adding this path is still a feasible tree.

The basic idea of our proposed algorithm works as follows. We firstly compute the *k* least delay paths from *s* to each destination node $m \in M$ by using the distributed *k*-Bellman-Ford (*k*BF) algorithm [5] as a *candidate-paths-set*. Then, a destination node is randomly selected, and the least delay path from *s* to this destination in *candidate-paths-set* is added into an initial empty tree *T*. At each step, we select the first path in *candidate-paths-set*, which starts from *s* to a nontree destination and satisfies (3), and add it to *T*. This operation repeats until all nodes in *M* are included in the tree.

We assume that each node has the information about the *k* shortest paths (in terms of delay) and the delay of each path to every destination node. The information is stored in the local routing table denoted by *Route* at each node. This can be achieved by running the distributed *k*BF algorithm on delay metric. In *Route*, each node $v \in V$ consists of $k \times |M|$ entries, one entry for the *k*th shortest path of every destination node. An entry *Route*[*i*][*m*] has two fields *Route*[*i*][*m*].*n* and *Route*[*i*][*m*].*d*, representing the next neighbor node on the *i*th least delay path from *v* to *m* and the delay of this path, respectively.

A simplified data structure for a control message is a 3-tuple $< type, kth, dest >$, where *type* is the type of the message, *kth* denotes the *k*th least delay paths in *candidate-paths-set*, and *dest* is the current selected destination. Five main types of messages are used in our algorithm, which are

open – opening a multicast connection, and getting *candidate-paths-set*;
start – starting the construction of the multicast tree;
add – adding a candidate path from the source to a destination node into the tree;
notify – notifying the source that a destination has been added to the tree;
finish – finishing the construction of the multicast tree.

3.2 Algorithm Details

Every node in the system executes the same routing algorithm. It is initially in an idle state waiting for connection setup requests.

When a node receives a request (*open* message) for opening a multicast connection, with parameters such as destination set *M*, a delay bound Δ and a delay variation bound δ , it computes the *k* least delay paths from *s* to each destination node $m \in M$ by using the distributed *k*BF algorithm. Let P_m be the set of *k* least delay paths for the destination *m*, i.e.,

$$P_m = \{p_1(s,m), p_2(s,m), ..., p_k(s,m)\}$$
$$\text{with } d(p_1(s,m)) \le d(p_2(s,m)) \le ... \le d(p_k(s,m)) \tag{5}$$

When source s receives an *start* request, an empty tree T is first initialized. Then a destination node m is randomly selected. Let *maxd* and *mind* be the maximal delay and minimal delay of the path among the paths from s to each destination covered in T so far, respectively. *maxd* and *mind* are initialized as $d(p_1(s,m))$(i.e. *Route*[1][m].*d*). A connection *add* message $< add, 1, m >$ is sent to the neighbor v via which the selected destination m can be reached by $p_1(s,m)$. The edge (s, v) is added into T.

When the *add* message arrives at an intermediate node, say u, on the way to the designated destination m, it passes this *add* message $< add, kth, m >$ to its next neighbor (u'), leading to m (u'=*Route*[*kth*][m].*n*). Then the edge (u, u') is added to T.

When the *add* message reaches the designated destination, an *notify* message is sent to s to show a destination has been added to the tree. Upon the receipt of this *notify* message, a nontree destination, say m', is selected. Then, the first candidate path whose delay is distributed between $[\max\{0, maxd - \delta\}, \min\{mind + \delta, \Delta\}]$ is picked out from *candidate-paths-set*. Suppose this candidate path is the ith least delay path, then the message $< add, i, m' >$ is sent to the neighbor v' via which m' can be reached by the selected path. After updating the values of *maxd* and *mind*, the edge (s, v') is added into T.

The above operation continues as the multicast connection is extended to destinations one after another, until all destinations in M are included in T. When the *add* request reaches the last destination in M, it sends a *finish* message to s. The construction of multicast tree satisfying both delay and delay variation constraints is completed. The pseudo code of whole algorithm is given in Fig. 1.

3.3 Discuss of the Algorithm

Theorem 2. (*Correctness of proposed algorithm*). *A delay and delay variation-bounded multicast routing tree will be always found if one exists.*

Proof. Suppose there exists a delay and delay variation-bounded tree for a source s, a set of destination M. During the construction of multicast tree, algorithm firstly adds the least delay path from s to one of selected destination into the tree, which obviously does not violate the two bounds. At each step, we always select a path satisfying (3) from s to a nontree destination in *candidate-paths-set*, and add it to the tree. According to Theorem 1, the tree, after this path is added, is still feasible. As a result, our algorithm can always find a delay and delay variation-bounded multicast tree if one exists.

Theorem 3. *In the worst case, the message complexity of our proposed algorithm is* $O(mn)$, *and the time complexity is* $O(k^2m^2n\log k)$, *where m is group size, n is network size, and k is the number of paths generated by using kBF.*

Proof. In our algorithm, the *add* message for each destination will be sent at most n times. Since there are m destinations, there will be at most $O(mn)$ number of *add*

```
Variables:
/* local = local node */
/* msg = control message */
/* Route = the local routing table */
/* T = the multicast tree */
/* maxd(mind) = the maximal(minimal) delay of the path among the paths to each
                destination covered in T */
1.  main( )
2.      wait for until receiving a message;
3.      switch (msg.type)
4.          case open: getting candidate-paths-set;
5.          case start: start( );
6.          case add: add( );
7.          case notify: notify( );
8.          case finish: finish( );
9.  end main;
10. start( )
11.     s = local; T = ∅ ;
12.     randomly choose a destination m ∈ M ;
13.     msg = < add, 1, m>;
14.     maxd = mind = Route[1][m].d;
15.     n = Route[1][m].n;
16.     add edge (local, n) to T; send (n, msg);
17. end start;
18. add( )
19.     dest = msg.dest; kth = msg.kth;
20.     if local ≠ dest then //pass add msg to the next neighbor
21.         msg = <add, kth, dest>;
22.         n = Route[kth][dest].n;
23.         add edge (local, n) to T; send (n, msg);
24.     else if all destinations are included in T then
25.         msg = <finish, kth, dest>; send (s, msg);
26.     else //notify source node to add next destination
27.     msg = <notify, kth, dest>; send (s, msg);
28. end add;
29. notify( )
30.     choose a destination m' ∈ M and m' ∉ T ; kth = 1;
31.     while kth ≤ k do
32.         if max{0, maxd − δ} ≤ Route[kth][m'].d ≤ min{mind + δ, Δ}  then
33.             break;
34.         kth++; msg = < add , kth, m' > ;
35.         update maxd and mind;
36.         n = Route[kth][ m' ].n;
37.         add edge (local, n) to T; send (n, msg);
38. end notify;
39. finish( )
40.     finish the construction of multicast tree T ;
41. end finish;
```

Fig. 1. The pseudo code for the proposed algorithm

messages. Other messages will not be set more than m times. Therefore, the worst message complexity of our algorithm is $O(mn)$.

In terms of time complexity, generating k least delay paths for m destinations by using kBF costs $O(k^2m^2n\log k)$ [5], and the rest of our algorithm costs $O(m)$. So, the worst time complexity of our algorithm is $O(k^2m^2n\log k)$.

4 Dynamic Reconstruction of the Tree for Membership Changes

For certain multicast applications, multicast participants may join or leave the multicast group dynamically during the lifetime of the multicast connection. It is important to ensure that any change of multicast memberships will minimize both the cost incurred during the transition period and the disruption caused to the receivers, and the routing tree after the change will always satisfy the constraints (1) and (2) for the current destination set.

In our method, when a destination node $m \in M$ decides to leave the multicast group, if m is not a leaf node, then no action needs to be taken. The new tree can be the same as the current tree T, with the only difference being that node m will stop forwarding the multicast packets to its local user and perform only switching operations. If, however, m is a leaf node, then a *leave request* is sent upward (to the source direction) along the tree, node by node, until it reaches the source node or another destination. At each node this request passes through, the connection is released. As the result, the new tree is essentially the same as T except in parts of the path from the source to m.

When a node $v \notin M$ wants to join an existing multicast group, it sends a *join request* to the source. We distinguish following three cases:

If $v \notin V_T$, we get k least delay paths from source to v. Then select the first path satisfying (3) and add it to T, which is similar to the main steps of our algorithm. If this fails to discover such a path, then deny the participation of node v in the multicast session and discard its *join request*.

If $v \in V_T$, and the path from source to v is such that the delay variation constraint (2) is satisfied for the new multicast group $M \bigcup \{v\}$. T is then a feasible tree for the new group, and can be used without any change other than having node v now forward multicast packets to its user, in addition to forwarding them to the downstream nodes.

If $v \in V_T$, but the path from source to v is such that constraint (2) is not satisfied for the new group $M \bigcup \{v\}$. It shows that v must be an intermediate node in the path from source to other destination or destinations. As a result, we will delete the paths which contain v and the destination(s). Then add v and the destination(s) to the routing tree, one by one, until all of them are included in the tree.

5 Simulation

In the following simulations, we will compare the performance of our algorithm with other four delay and delay variation-bounded routing algorithms. Five algorithms, namely a distributed version of Bellman-Ford Shortest Path Algorithm (SPT) [6],

DVMA [1], DDVCA [2], SP-DVMA [3], and the one we proposed (Zhang's) have been implemented in a QoS routing simulator (QRSIM) designed by us and written in C++. All simulations are run on a Pentium IV 2.8 GHz, 512 MB RAM, DELL PC.

Generating network topology is based on the random link generator (based on Waxman's generator [7] with some modifications) developed by Salama [8], which yields networks with an average node degree of 4. The positions of the nodes are fixed in a rectangle of size $4000km \times 4000km$. The Euclidean metric is then used to determine the distance between each pair of nodes. Edges are introduced between pairs of nodes u, v with a probability that depends on the distance between them. The edge probability is given by $P(u,v) = \beta \exp(-l(u,v)/\alpha L)$, where $l(u, v)$ is the distance from node u to v, L is the maximum distance between two nodes. α and β are parameters, and are set to 0.15 and 2.2 respectively. Larger values of β result in graphs with higher edge densities, while small values of α increase the density of short edges relative to longer ones.

The link delay function $d(e)$ is defined as the propagation delay of the link, and queuing and transmission delays are negligible. The propagation speed through the links is taken to be two thirds the speed of light. At each simulation point, we run the simulation 500 times and the result is the mean value of the results produced by these 500 runs. Each time, the source node and the destination nodes are randomly picked up from the network graph. Note that δ is kept constantly at 0 in DVMA algorithm (it forces DVMA to return the smallest delay variation that it can find).

Table 2. Delay and delay variation for different network size. m=10, Δ =35ms, δ =25ms

Alg. \ n	60	70	80	90	100
SP-DVMA	31.14 / 19.07	32.09 / 18.28	32.35 / 19.28	33.01 / 18.23	33.84 / 18.19
DDVCA	29.78 / 20.39	30.30 / 20.60	30.49 / 20.02	30.76 / 20.07	30.74 / 20.27
DVMA	27.08 / 17.38	27.26 / 17.40	27.32 / 17.56	27.41 / 17.08	28.17 / 17.33
SPT	25.42 / 18.43	25.93 / 18.61	26.11 / 18.27	26.70 / 18.76	26.83 / 18.40
Zhang	24.74 / 17.53	24.78 / 17.54	25.93 / 17.93	26.00 / 17.62	26.53 / 17.81

Table 3. Delay and delay variation for different group size. n=100, Δ =35ms, δ =25ms

Alg. \ m	4	5	6	7	8
SP-DVMA	31.55 / 13.40	33.27 / 14.66	33.76 / 15.86	34.14 / 16.77	34.42 / 17.31
DDVCA	29.68 / 12.21	30.04 / 15.10	30.00 / 16.09	30.21 / 17.04	30.79 / 18.32
DVMA	25.01 / 11.74	25.84 / 12.93	26.18 / 13.80	26.51 / 15.41	27.42 / 16.26
SPT	23.94 / 12.37	24.99 / 13.82	25.33 / 14.82	25.74 / 15.75	26.34 / 17.00
Zhang	24.12 / 12.09	25.13 / 13.91	25.12 / 14.78	25.45 / 15.44	25.42 / 16.40

Tables 2 and 3 show the delay and delay variation of various heuristics for different number of network nodes (from 60 to 100 in steps of 10, group size = 10) and different number of group members (from 4 to 8 in steps of 1, network size = 100), respectively. Parameters that are kept constant are Δ =35ms and δ =25ms. The numbers in Tables (e.g. 31.14 / 19.07) represent the delay and delay variation for that

algorithm, respectively, and the units is millisecond (ms). It can be seen from Tables 2 and 3 that our algorithm has the best delay performance among all algorithms. SPT algorithm gives slightly higher delay than our algorithm. As the number of network nodes and group members increase, the maximum end-to-end delay of all algorithms increases, but below the 35ms delay bound.

As for the delay variation, we can see that DVMA algorithm has the optimum delay variation performance as expected. Our algorithm gives slightly higher delay variation than DVMA, but has lower delay variation than other three algorithms. Table 3 shows the delay variation performance of all algorithms increases as the group size increases. This is expected since, the larger the size of the multicast group, the larger number of the destination nodes physically closer or farther to the source, which results in the increase of the delay variation between destination nodes.

Finally, we compare the routing request success ratio (SR) for three algorithms (DDVCA, SPT and Zhang's). SR is defined as the ratio of the number of multicast routing requests accepted and the total number of requests generated. Table 4 shows the SR of routing requests for these algorithms in above simulation environment. We observe from Table 4 that our algorithm achieves higher SR than other two algorithms for all scenarios we tested. It is obvious that as the group size increases, the SR of all algorithms decreases. This is because the delay variations between destinations increase as the group size increases, and then possibility of satisfying δ will decrease.

Table 4. A comparison on the success ratio of routing request

n	DDVCA	SPT	Zhang
60	56.4%	88.8%	93.0%
70	58.0%	88.2%	94.0%
80	54.0%	87.0%	93.0%
90	59.0%	87.8%	93.8%
100	59.6%	86.8%	90.8%

m	DDVCA	SPT	Zhang
4	86.2%	97.2%	98.8%
5	79.6%	96.2%	97.0%
6	72.4%	95.8%	96.6%
7	69.8%	93.0%	95.8%
8	67.0%	92.6%	95.0%

6 Conclusion

In this paper, we discuss the problem of constructing multicast routing trees satisfying the end-to-end delay bound and delay variation bound, which is called *DVBMT* problem and has been proved to be NP-complete. We have presented an efficient distributed dynamic multicast routing algorithm for obtaining such trees. We firstly compute candidate least paths in terms of delay from source to each destination. Then starting from an empty tree, we iteratively add a candidate path satisfying specific condition to the selected destination into the tree. This operation repeats until all destinations are included in the tree. The proposed algorithm has the following advantages.

(1) *Fully distributed*. Each node operates based on its local routing information and coordination with other nodes is done via network message passing.

(2) *Dynamic changes of multicast memberships*. We also give a method to dynamically reorganize the multicast tree in response to changes for the destinations, and guarantee the minimal disruption to the multicast session.

(3) *High performance with low complexity*. A large amount of simulation has been done to show that our algorithm performs excellently in delay, delay variation, and routing success ratio with a lower time complexity, which ensures it to support the requirements of real-time multimedia communications more effectively.

References

1. Rouskas, G.N., Baldine, I.: Multicast Routing with End-to-End Delay and Delay Variation Constraints. IEEE Journal on Selected Areas in Communications, Vol. 15, No. 3 (1997) 346-356
2. Sheu, P.R., Chen, S.T.: A Fast and Efficient Heuristic Algorithm for the Delay- and Delay Variation-bounded Multicast Tree Problem. Computer Communications, Vol. 25, No.8 (2002) 825-833
3. Yu, Y.P., Qiu, P.L.: A Heuristic of Multicast Routing with Delay and Delay Variation Constraints. Journal of China Institute of Communications, Vol. 24, No. 2 (2003) 132-137
4. Zhang, K., Wang, H., Liu, F.Y.: Distributed Multicast Routing for Delay and Delay Variation-bounded Steiner Tree using Simulated Annealing. Computer Communications, Vol. 28, to be published, (2005)
5. Jia, Z.F., Varaiya, P.: Heuristic Methods for Delay Constrained Least Cost Routing Using *k*-Shortest-Paths. Proc. of IEEE INFOCOM'01 (2001)
6. Bellman, R.E.: Dynamic Programming. NJ: Princeton University (1997)
7. Waxman, B.M.: Routing of Multipoint Connections. IEEE Journal on Selected Areas in Communications, Vol. 6, No. 9 (1988) 1617-1622
8. Salama, H.F.: Multicast Routing for Real-time Communication on High-speed Networks. PhD Dissertation, North Carolina State University, Department of Electrical and Computer Engineering (1996)

Data Caching in Selfish MANETs

Jian Zhai, Qing Li, and Xiang Li

Department of Computer Engineering and Information Technology,
City University of Hong Kong,
83 Tat Chee Avenue, Kowloon, HKSAR, China
jian.zhai@student.cityu.edu.hk
itqli@cityu.edu.hk
xiang.li@student.cityu.edu.hk

Abstract. There are a lot of cooperative cache algorithms in Mobile Ad hoc Network (MANET) environment today. However, little attention was paid to the case that the mobile clients within a MANET are non-cooperative. These non-cooperative behaviors include selfish, faulty and malicious ones. In this paper, we focus on the selfish behavior and study it thoroughly. The essential of our cache algorithm within a selfish MANET is that service provider can be paid for its caching service. We adopt game theory in this paper and theoretically prove that the proposed cache algorithm within a selfish MANET can run into an equilibrium status after several steps. Some useful conclusions are drawn, and experiment results are given to show the validity and efficiency of our work.

1 Introduction

Caching data in its neighbor node is a feasible approach to accelerate the data access delay under the MANET (*Mobile Ad hoc Network*) environment. However, the problem becomes tough when the clients are some types of light-weight mobile devices such as mobile phones and PDAs due to their limited storage space, which prevents the clients from holding a large cache. We call such mobile devices *weak nodes*. In contrast, there are also strong mobile devices such as notebooks and tablet PCs that have large main memory, high-speed wireless connection, and ample power supply. We call them *strong nodes*. The strong nodes can provide data caching service to the weak ones if they belong to the same MANET.

Currently, there are a lot of data cache algorithms for the MANET environment. These algorithms make an assumption that all nodes within a MANET are willing to provide cache services, which is called *cooperative caching*. However, in most real situations, a lot of clients within a MANET are weak nodes. Thus, the clients have to consider the serious hardware limitations and may be unwilling to provide cache service for free. This phenomenon has been noticed by some researchers, and some compulsive rules are introduced to ensure the cooperation. Nevertheless, users can crack the compulsive program and escape themselves from the obligation. Therefore we do not think compulsive method is a good approach to solve the "selfish" problem of cooperative data cache within MANET. Instead, cache service providers may be

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 208 – 217, 2005.

willing to provide the service if they can be rewarded/paid by cache users. Hence, data cache service can work within a MANET even mobile clients are still selfish. In the above scenario, data cache in a MANET becomes a "chargeable" service instead of a private behavior of a single node. The transactions among all the nodes within a MANET can be treated as business dealings in the market. Every strong node that is willing to provide the data cache service acts as a separate *seller*. The seller places its auction on the market and tries to earn the most out of it by attracting as many *customers* as possible to buy its service and making the net income from each customer as high as possible. At the same time, customers are free to buy such services from one or more sellers in the interest of the best performance ratio. However, better service means larger storage space, stronger emission power for wireless communication, higher power cost, and so on, which are all against the increase of the net income. So, the price and the quality of service pose a total contradiction to the seller. Similar situation occurs on the customers who bid for caching services of high performance ratio from one or more sellers, where they must compete against each other to get the desired services cost-effectively.

In a selfish MANET, the main problem we are concerned with is whether the system will enter an equilibrium status or not. The theory of Nash Equilibrium (NE) [2] from microeconomics is applicable to predict whether there exists equilibrium, and if so, what the outcome will be. In this paper, we aim at modeling the market behaviors in the MANET environment. From the theory of microeconomics, it is well known that pricing is a tool in the market to induce social behavior (also called Pareto Efficiency, PE for short) in microeconomics [3]. A set of well-defined protocols can encourage the sellers to provide caching services to the MANET market in a competitive price, which may lead them to set the price properly and thus induce PE in MANET.

The rest of the paper is organized as follows. We give a brief review of related work in section 2. The problem is formally described in section 3. Section 4 provides the market protocols. Section 5 discusses the strategies of seller nodes and consumer nodes, respectively, and discusses the relationship between NE and PE. Experiment results are given in section 6. Section 7 summarizes our work and sheds light on the future research issues.

2 Related Work

Under the MANET environment, earlier research efforts (e.g. [4] and [5]) were always based on an assumption that each node of the MANET is able and willing to provide caching service. However, in many real world MANET systems, the nodes are self-interested. Loo addressed the issue of selfish behavior of the nodes within a MANET in [6]. Miranda *et al.* described the selfish behavior in a MANET environment caused by the limitation of resources such as battery power [7]. They also found ways to prevent selfishness in an open MANET environment. Srinivasan *et al.* examined the selfish users in MANET where energy is treated as a valuable resource [8].

There are two ways to prevent the nodes of a MANET from selfish behaviors: to establish a compulsory mechanism and to introduce incentives. The former includes the work from Hatzis [9] and Chatzigiannakis [10] where compulsory protocols are provided. The later includes the work from Chen *et al.* [11] and Ma *et al.* [12], where incentive is used to promote contribution such as package forwarding and information sharing from the nodes.

Besides the research from computer science, some economists have also contributed their works on examining the behavior in a non-cooperative environment. Of particular interests to us is the work of Nash Equilibrium (NE) by John Nash (1950) who identified a fundamental solution of non-cooperative games, which he called an "equilibrium point". Our work is proposed by adopting this concept in selfish MANET.

3 Model Formalism

A MANET can be modeled as a triple (V, E, M), where $V=\{v_1,\ldots,n_n\}$ represents the n nodes inside the network, $E=V\times V$ represents the set of directed edges, each of which has the form of (v_i, v_j), $i\neq j$, that connects two nodes, and the function $M : E \rightarrow \Re$ for each edge (v_i, v_j) stands for the price that node v_j is willing to pay for the service from node v_i. A directed edge (v_i, v_j) exists if and only if node v_i provides caching service to node v_j. For each v_i, there are four variables describing its behavior in the (non-cooperative) *game*:

b_i is the total outgoing bandwidth;
r_i is the minimum tolerable transfer bit rate of the requested data file;
s_i is the total main memory space reserved by v_i for caching service;
w_i is the wireless signal emission power of v_i.

A node v_i can be either a consumer node or a seller node or both. A seller node must be a strong node but there is no precondition for a node to be a consumer node. If v_i is a consumer node only, then $s_i=0$ and $w_i=0$. If v_i is a seller node only, then $r_i=0$. In the model, a node is allowed to serve both as a consumer and seller. For a consumer node v_i, r_i should be less than its maximum bandwidth for downloading; otherwise, caching the data file on other seller nodes is meaningless.

It is assumed that, in order to download the data file in a tolerable time, the data should be prefetched for T seconds in advance. T depends on the stability of the network. Meanwhile, the same T value can be uniformly applicable to the whole MANET for simplicity. So, the required cache size is $\eta_i=T\cdot r_i$ for a consumer node v_i.

4 Protocols

In this section we define a set of protocols that each node in the "market" should obey. These protocols are executed compulsorily by asking every node who wants to join the MANET to install a plug-in.

The first protocol is the communication protocol. Fig. 1 briefly shows the typical conversation between a seller and a consumer.

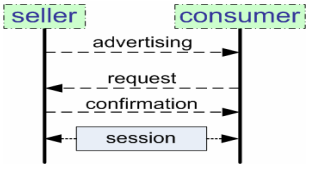


Fig. 1. Conversation between a seller and consumer

In Fig. 1, the seller broadcasts a message within the MANET periodically to let other nodes know that it provides chargeable caching service.

After getting the broadcast messages, consumers then "know" which node(s) can provide such services. When a consumer wants to use the service, it sends a request to the most preferred seller (i.e., the one who has the best performance ratio). Assuming the target seller's signal can still reach the consumer with power w_i (we say the target seller is in the service range of the consumer), the seller replies to the consumer with a confirmation signal and starts a conversation session with the consumer.

The advertisement is exemplified by Fig. 2, where the square represents for a consumer node and the five circles represent seller nodes.

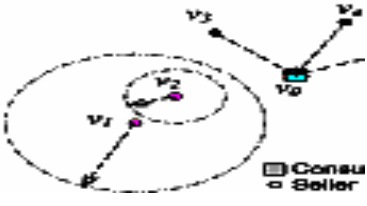


Fig. 2. Advertising in MANET

In Fig. 2, each v_i (i=1,2,…,5) is willing to provide caching service to the consumer v_0 within a MANET. Recall that seller nodes must be strong nodes, so v_i (i=1,2,…,5) can broadcast the advertisement message with signal emission power w_i. In the case shown in Fig. 2, signals from v_1 and v_2 cannot reach to v_0, and v_0 only knows that v_3, v_4, and v_5 provide caching service hence chooses a most preferred one from them. So, the stronger the signal emission power of a seller is, the more potential consumers it has. However, larger w_i may imply a greater cost. So, a seller node needs to make a balance between the advertising cost and the expected net income.

The second protocol is the payment protocol. There have already been a lot of research works that address the payment models, such as Nulglets [15] and Sprite [16]. In Nuglets, there is a central bank that manages the virtual money of all nodes. The consumer nodes store money in the central bank, while the seller nodes get their payment from it. The central-bank is supposed to be fair and truthful. In Sprite, each node must be installed with a plug-in. The plug-in truthfully records the owner's virtual money. Trading is done directly among the nodes without the help of a central bank.

In our model, either type of the payment protocols is applicable since it does not affect the equilibrium of the MANET market.

The third protocol is the unit cost protocol. The unit cost vector Θ_i (to be thoroughly discussed in section 5) should remain constant for the same session. That is, each time when a seller and a consumer make an agreement about the caching service, Θ_i should not be changed during the process of downloading a data file.

5 Strategies

Each seller node v_i providing service to a set of consumer nodes $\Gamma(v_i)=\{v_j|1\leq j\leq n$ and (v_i,v_j) exists$\}$ must meet the following two conditions:

$$\sum_{k=1}^{n} r_k \cdot x_k \leq b_i \tag{1}$$

$$\sum_{k=1}^{n} \eta_k \cdot x_k \leq s_i \tag{2}$$

where $x_k=1$ if $v_k \in \Gamma(v_i)$ and $x_k=0$ if $v_k \notin \Gamma(v_i)$.

Let c_k be the net income v_i earned from v_k, the aim of v_i is to maximize $\sum_{k=1}^{n} c_k \cdot x_k$ subject to the above two constraints.

The problem above is dynamic and has an exponential computational complexity [13]. Even when the MANET consists of less than several hundreds of nodes, and the number of possible directed edges started from v_i is much smaller than the total number of nodes, it is still a heavy computation burden for a mobile device. However, because $\eta=T\cdot r_i$, we can rewrite inequation (2) as $\sum_{k=1}^{n} r_k \cdot x_k \leq \frac{s_i}{T}$. So, if $b_i < s_i/T$, the set $\{x_k|1\leq k\leq n\}$ satisfying inequation (1) should definitely satisfy inequation (2), and vice versa. Then, the problem can be transformed into a knapsack problem. In a market with NE, the difference among different c_k does not vary too much. So, a greedy algorithm can be devised to find a near optimal solution.

To calculate the net income c_k of v_i from v_k, v_i must synthetically consider r_k, η_k, w_i, and the price p_k that node v_k pays to v_i for the service. c_k can be calculated using the following formula:

$$c_k = p_k - r_k \cdot y_1 - \eta_k \cdot y_2 - w_i \cdot y_3 \tag{3}$$

Note that v_i must provide the unit cost vector $\Theta_i=(y_1, y_2, y_3)^T$ (y_k is a non-negative real number for k=1,2,3) before it joins the MANET. Θ_i can be adjusted according to the market requirement and the condition of v_i. For example, if v_i realizes that too many consumers are requesting caching service from it, the owner of v_i may increase y_k (k=1,2,3) in the next advertisement broadcast. Let $\Xi_i=(r_i, \eta_i, w_i)$, then the net income can be expressed as $c_k=p_k - \Xi_i \cdot \Theta_i$. The seller's strategy is described with pseudo-code in Fig. 3.

```
Seller_Strategy()
1.  Broadcast advertisement with power w;
2.  Create child process Cost_Change();
3.  Repeat
    If no request
      Sleep time_intv;
      Continue;
    Else
      For all requests
        Find near optimal consumer set;
        Confirm their request;
        Create process to
        start conversation.

Cost_Change()
1.  Repeat
    If Θ_i changes
      Broadcast advertisement with
      new Θ_i and w;
    Else if w changes
      Broadcast advertisement with new w;
    Else
      Sleep mon_intv;
```

Fig. 3. Seller's Strategy

The consumer's strategy is relatively simple because a consumer only needs to rank the available caching service providers, and chooses the one(s) with the best performance ratio. The pseudo-code of the consumer's strategy is given in Fig. 4:

```
Given a set of available seller nodes V^s
Consumer_Strategy(V^s)
1.  For each v_i ∈ V^s
    Send request to v_i
    If v_i has enough space and bandwidth
      ρ_i = w_i/p_i
    Else
      ρ_i = 0
2.  Ranking ρ = { ρ_i where v_i ∈ V^s } in desc order
3.  Let n = ‖ ρ ‖
4.  Select top-k (k ≤ n) ρ_i from ρ as ρ' = {ρ_1', ..., ρ_k'}
5.  Request v_1', ..., v_k' for caching service
```

Fig. 4. Consumer's Strategy

Based on the modeling of the behaviors between nodes, the *game* within a MANET can be defined as a triple $(V, \{A_i\}, \{\Phi_i(\Theta_i, \Lambda_i)\})$, where V is the set of nodes that participate in the market of the MANET, $\{A_i\}$ is the set of actions for node v_i, and

$\Phi_i : \prod_i A_i \to \Re$ is the payoff or utility function for node v_i given the set of actions of all the nodes. Node v_i's action is denoted by g_i. Thus, g_i is a feasible action if inequations (1) and (2) are met. The payoff or utility function of node v_i is the sum of net income it gets from the market. Let $\Lambda_i = \sum_{k \in \Gamma(i)} p_k$, Φ_i can be calculated using formula (4):

$$\Phi_i(\Theta_i, \Lambda_i) = \sum_{k \in \Gamma(i)} c_k \tag{4}$$

In this *game*, each node wants to maximize its own payoff. By applying NE theory in our model, the following two lemmas can be obtained (whose proofs are given in [20]).

Lemma 1: A pure strategy of Nash Equilibrium always exists in the data cache game.

Lemma 2: Each node v_i correctly computes and declares its true unit cost Θi.

In the field of microeconomics, the concept of Pareto Efficiency stands for the "social optimal", which means the maximum value of *F*. An allocation is *Pareto Efficient* (PE) if there is no other allocation in which some other individuals are better off and no individual is worse off. We can get the following lemma, whose proof is again given in [20].

Lemma 3: The NE status of the MANET market can be induced to PE under our protocols.

Another issue discussed in NE is how long a system can reach NE if it exists. Under our market protocols, the MANET market can reach NE quickly (in only one or two rounds of conversation), no matter what the initial value of Θ_i is. The above conclusion is tested and verified by our experiment studies as detailed in the next section.

6 Experiment

In this section we demonstrate the efficiency of our caching scheme in a 2-tier emulation network environment. The first tier is the ad hoc network layer and the second tier is the node behavior based on the MANET.

For the first tier, there are two natural methods to evaluate the application performance in an ad hoc network environment. The first one is to construct a real ad hoc network test-bed with desired scenarios upon which the applications or protocols are tested. The second method is using network simulator, which offers the ability to repeat and control the network conditions according to the tester's requirement. However, both the test-bed and simulator methods have pros and cons: the former is very realistic but expensive and non-repeatable; the later requires re-implementing the network protocols/applications and the modeling is a nontrivial procedure. In our experimental study, a network emulator proposed by Ke *et al.* [18] is adopted, which represents a trade off between the real test-bed and pure simulation approaches. In Ke's system, a real-time scheduler based on a reordering algorithm is presented with

improved accuracy and scalability of emulation. In our experiment, we set the number of nodes to be 120; the rate of background traffic ranges from 0 packets/second to 120 packets/second (the packet size is 512 bytes). In the system, we let γ be an adjustable argument which is as defined in formula (5).

$$\gamma = \frac{N_{strong}}{N_{weak}} \tag{5}$$

We assume that every *strong node* wants to be a seller within the MANET market, and every *weak node* is a consumer node.

As for the second tier, Janssen *et al.* proposed an agent-based simulation for evaluating intermediation roles in e-commerce [19]; its library for simulating the auction and biding behavior in a market is used in our experiment to simulate the node behaviors.

For a distributed caching system, *Bit Hit Ratio (BHR)* and *Average Service Delay (ASD)* are two important indicators. *BHR* is defined by the ratio of total bytes from cached objects over the total bytes of objects requested by all the clients. When a user request arrives and the requested segments are not in the cache, it has a delayed start. The *BHR* is defined by formula (6).

$$BHR = \sum_{i=1}^{N} P_i \lambda_i / \sum_{i=1}^{N} \lambda_i \tag{6}$$

In formula (6), λ_i is the frequency of the data request, which ranges from 0 to 5 in our experiment. P_i is the possibility that node i's request is cached by other node(s).

The *ASD* is as defined in formula (7).

$$ASD = \frac{1}{\sum_{i=1}^{N} \lambda_i} \sum_{i=1}^{N} \frac{\lambda_i \cdot BHR}{b_i} \tag{7}$$

Fig. 5 and Fig. 6 show, respectively, the variation of *BHR* and *ASD* along with the time line for the cases of γ = 0.5, 1, and 2.

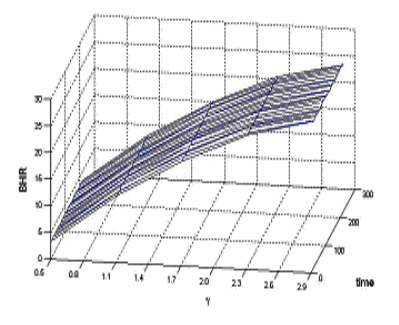


Fig. 5. Variation of BHR with different γ along timeline

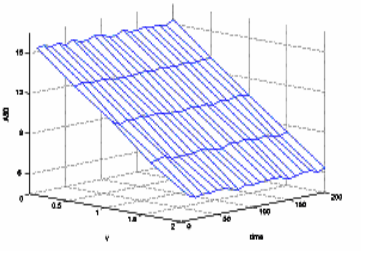


Fig. 6. Variation of *ASD* with different γ along timeline

From Fig. 5 and Fig. 6, we can see that BHR and ASD are nearly constant along with time. We also see that, when the number of the strong nodes is half of that of the weak nodes (i.e., γ=0.5), the BHR is around 5%, in comparison with the BHR around 25% when γ=2. Inversely, when γ=2 (that is, the number of the strong nodes is twice of that of the weak nodes), the *ASD* is around 6s, in comparison with the *ASD* around 12s when γ=0.5.

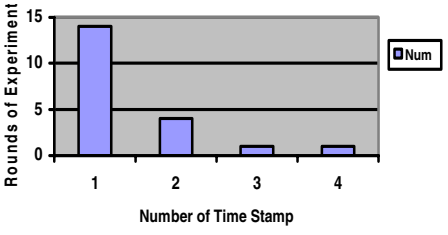


Fig. 7. Number of time stamps required for the market to fall into equilibrium

We also notice that, in the emulation experiment, no matter what the initial value of Θi is, the system can quickly come into equilibrium. By repeating the experiment 20 rounds, Fig. 7 shows the number of time stamps that the market falls into equilibrium. From Fig. 7, we can see that the market can fall into equilibrium within one or two time stamps in most (18 out of 20) rounds of the experiment. As depicted in Fig. 7, our (plug-in) mechanism has an acceptable computational complexity, which can be applied to the MANET environment realistically.

7 Conclusions

We have presented a data caching mechanism and a set of protocols to encourage more powerful mobile devices (strong nodes) to provide data cache service to other weak nodes (e.g., mobile phones) in the MANET environment. Our assumption is that each node in the MANET is greedy and self-interested, always trying to maximize its own income. We have found that the "market" finally reaches an equilibrium (or called Nash equilibrium in microeconomics). However, it is known that the maximum income of individual nodes may not lead to the Pareto Efficiency or social optimum (i.e., the maximization of the outcome of the game). Through game theory, we have proved (cf. [20]) that our scheme and protocol design can actually achieve social optimum.

References

[1] J. Jones. Game Theory: *Mathematical Models of Conflict*. Albion/Horwood Publishing House, 2000.
[2] J. Nash. Non-Cooperative Games. *Annals of Mathematics 54, 2* (Sep. 1951), 286-295.

[3] Saraydar, N. Mandayam, and D. Goodman. Efficient Power Control via Pricing in Wireless Data Networks. *IEEE Transaction on Communications, 50, 2* (Feb. 2002), 291-303.
[4] Bestavros, and S. Jin. OSMOSIS: Scalable Delivery of Real-Time Streaming Media in Ad-Hoc Overlay Networks. *In Proc. of the IEEE Int'l Conf. on Distributed Computing Systems (ICDCS '03)* (Rhode Island, USA, May 2003). IEEE Press.
[5] W. H. O. Lau, M. Kumar, and S Venkatesh. Mobile Ad Hoc Networks: A cooperative cache architecture in support of caching multimedia objects in MANETs. *In Proc.5th ACM int'l workshop on Wireless mobile multimedia* (Atlanta, USA, Sept 2002). ACM Press.
[6] F. Y. Loo. Ad Hoc Network: Prospects and Challenges. Technical Report D1, AML, University of Tokyo, Japan, 2004.
[7] H. Miranda, and L. Rodrigues. Preventing selfishness in open mobile ad hoc networks. *In Proceedings of the 7th CaberNet Radicals Workshop* (Bertinoro, Italy, October, 13-16 2002). IEEE Press.
[8] V. Srinivasan, P. Nuggehalli, C. F. Chiasserini, and R R. Rao. Energy Efficiency of Ad Hoc Wireless Networks with Selfish Users. *In Proc. of European Wireless conference (EW '02)* (Florence, Italy, February 26-28, 2002). IEEE Press.
[9] K. P. Hatzis, G. P. Pentaris, P. G. Spirakis, V. T. Tampakas and R. B. Tan. Fundamental Control Algorithms in Mobile Networks. *In Proc. 11th Annual Symposium on Parallel Algorithms and Architectures (SPAA '99)* (Saint-Malo, France, June, 1999). ACM Press.
[10] Chatzigiannakis and S. Nikoletseas. An Adaptive Compulsory Protocol for Basic Communication in Highly Changing Ad-hoc Mobile Networks. *In Proc. 2nd Int'l Workshop on Parallel and Distributed Computing Issues in Wireless networks and Mobile Computing (PDCIWNMC '2002)* (Marriott Marina, Florida, USA, April 15-19, 2002). IEEE Press.
[11] K. Chen, and K. Nahrstedt. iPass: an Incentive Compatible Auction Scheme to Enable Packet Forwarding Service in MANET. *In Proceedings of the 24th IEEE International Conference on Distributed Computing Systems (ICDCS '04)* (Tokyo, Japan, March 23-26, 2004). IEEE Press.
[12] T. B. Ma, C. M. Lee, C. S. Lui, and K. Y. Yau. Incentive P2P Networks: A Protocol to Encourage Information Sharing and Contribution. *ACM SIGMETRICS Performance Evaluation Review, 31, 2, Special issue on the fifth workshop on MAthematical performance Modeling and Analysis (MAMA '03)* (Sep. 2003), 23-25.
[13] Montet, and D. Serra. *Game Theory and Economics.* Palgrave Macmillan, NY, 2003.
[14] K. Chandra, D. S. Hirschberg, and C. K. Wong. Approximate Algorithms for Some Generalized Knapsack Problems. *Theoretical Computer Science, 3*, 3 (Dec. 1976): 293-304.
[15] L. Buttyan and J. Hubaux. Nuglets: a virtual currency to stimulate cooperation in self-organized ad hoc networks. Technical Report EPFL, DSC, 2001.
[16] S. Zhong, Y. R. Yang, J. Chen. Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-hoc Networks. In *Proceedings of the INFOCOM conference (INFOCOM '03)* (San Francisco, USA, March 30-April 3, 2003). IEEE Press.
[17] http://www.ietf.org/rfc/rfc1321.txt.
[18] Q. Ke, D. Maltz, and D. B. Johnson. Emulation of Multi-Hop Wireless Ad Hoc Networks. *In Proceedings of the 7th International Workshop on Mobile Multimedia Communications (MoMuC '00)* (Tokyo, Japan, Oct 23-26. 2000). IEEE Press.
[19] M. Janssen, A. Verbraeck, and H. G. Sol. Agent-based Simulation for Evaluating Intermediation Roles in Electronic Commerce. *In Proceedings of the 1st Agent-Based Simulation (ABS '00)* (Passau, Germany, May 2-3, 2000).
[20] http://www.hkws.org/appendix.doc

Optimal Scheduling for Link Assignment in Traffic-Sensitive STDMA Wireless Ad-Hoc Networks

Hengchang Liu and Baohua Zhao

Department of Computer Science, Univ. of Science and Technology of China,
Hefei, Anhui, P.R. China, 230027
hcliu@mail.ustc.edu.cn

Abstract. In this paper, we consider the resource optimization problem to maximize the network throughput by efficiently using the network capacity, where multi-hop functionality and spatial TDMA access scheme are used. The objective is to find the minimum frame length with given traffic distributions and corresponding routing information. Because of the complex structure of the underlying mathematical problem, previous work and analysis become intractable for networks of realistic sizes.We address the problem through mathematical programming approach, develop the linear integer formulation optimizing the network throughput, and then show the similarity between the original problem and the graph edge coloring problem through the conflict graph concept. We propose a column generation solution approach and make several enhancements in order to fasten its convergence. Numerical results demonstrate that the theoretical limit of the throughput can be efficiently computed for networks of realistic sizes.

1 Introduction

The emergence of wireless ad-hoc networking is due to the increasing interactions between communication and computing, which is changing information access from "anytime, anywhere" into "all the time, everywhere"[4]. The defining characteristic of ad-hoc networks is their loose and self-organized structure, as opposed to the centralized structure in cellular networks. This flexibility has made ad-hoc networks a very attractive approach for many applications, such as peer-to-peer community wireless networks [7, 13] and mobile sensor networks [14]. In ad-hoc networks, a direct communication link is set up between two distinct radio units, if the signal-to-noise ratio (SNR) is sufficiently high. Taken energy consumption into account, not all pairs of units may establish direct links, and traffics between them have to be relayed through some other units. This is the so-called multi-hop functionality in ad-hoc networks.

There are many challenges in the rational design of ad-hoc networks [9]. One particular issue is the problem of allocating physical and data link layer resources, to minimize a cost function while fulfilling certain network layer communication demands, such as the traffic load and transmission rate. In an access control method using TDMA, the transmission capacity is divided into time slots, and each direct link is assigned a dedicated slot. A promising approach for increasing its efficiency is the spatial TDMA (STDMA) scheme, which takes into account the fact that radio units

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 218 – 228, 2005.

are usually spread out geographically, and hence units with a sufficient spatial separation can use the same timeslot for transmission.

In current wireless ad-hoc network, resource utilization lies heavily on the performance of the algorithm used for generating the transmission schedule. In this paper, we consider the link scheduling problems in wireless ad-hoc networks from a mathematical point of view. Given an arbitrary network traffic distribution, we address the problem of finding a schedule that maximizes the network throughput. The contribution of our work is two-fold. Firstly, we note that this is the first paper to investigate this certain problem deeply and show that it can be efficiently solved by set covering programming formulation coupled with a column generation approach. Secondly, we present several practical enhancements to fasten the convergence of the column generation process and these improvements are heuristic enough to be applied in other corresponding problems.

Network throughput is one of the most important issues when evaluating the performance of a wireless ad-hoc network. The methods to get high throughput can be divided into heuristic methods and optimization methods. Heuristic approach is a way to construct a schedule making sensible but not necessarily optimal decisions along the way [10, 11, 12]. As an example, Somarriba et al. propose one scheduling way in which they put the links that are compatible into each timeslot until all links have been allocated. And the links with higher relative traffic load get higher priority to be put in the timeslots first, or else they will become the bottleneck of the communication network. This method does not require much computation compared to other methods and is easy to be implemented, but not optimal. In addition, the authors did not get the approximate ratio of the algorithm, which implies that more theoretical support is needed.

Optimization approach, the problem of optimal scheduling of transmissions in multi-hop wireless networks has a long history (see, for example, [17, 18]). Recently, several approaches have been proposed for further improving network-wide performance [2, 3, 6, 8]. However, the optimization process in [2, 6] relies on explicit enumeration of all transmission groups, which results in an optimization problem whose number of variables is exponential with the number of direct links. Consequently, results are only given for small networks.

Explicit enumeration of all transmission links is avoided when using a column generation technique for solving the network optimization problems [1, 5]. In [1], the authors consider the optimal scheduling for maximum uniform throughput under STDMA scheme. Transmission groups are generated by solving an integer programming sub-problem at each iteration. Although results are reported for larger networks than those in [2, 6], the “homing in, tailing off” property [5] of this approach makes it really a long time to get the optimal solution. Also the objective of optimizing uniform throughput is somehow impractical for realistic networks in which traffic load is different for different direct links. Our work can be viewed as a traffic-sensitive extension for [1] and in addition we develop methods to fasten the convergence of column generation process, which can be applied to larger networks than other current approaches.

2 Network Model and Assumptions

We will use the network model and assumptions used in [1, 16] when defining the optimization problems for maximizing the network throughput. An ad-hoc network can be characterized by a directed graph $D = (N, L)$, where the node set N represents the radio units, and the link set L represents the communication links. A directed link (i, j) is established if the SNR is no less than a given threshold, that is, if

$$SNR(i, j) = \frac{P_i}{L_b(i, j) \times N_r^3} \quad g_0 \tag{1}$$

Where P_i is the transmission power of i, $L_b(i, j)$ is the path-loss between i and j, N_r is the effect of the thermal noise, and g_0 is the threshold. We assume that every node transmits at its maximum power when sending data packets to another node.

Several assumptions are commonly made in our work. The communication model we consider is interference–limited, i.e., any node can communicate with at most one other node at a moment. Meanwhile, a node can not transmit and receive packets at the same time. Finally, the so called SIR–criterion states that a link is free of error only if its signal–to–interference ratio (SIR) is above a given threshold, which can be formulated as

$$SIR(i, j) = \frac{P_i}{L_b(i, j)(N_r + \sum_{k \in S, k \neq i} \frac{P_k}{L_b(k.j)})} \geq \gamma_1 \tag{2}$$

where S is the set of simultaneously transmitting nodes. Obviously not all pairs of nodes can establish direct links; therefore traffic between two units may have to be relayed through some other units. We assume fixed routing in the network flow model and that traffics, represented by multiple source-destination pairs, are given. Only single-path transmission is considered since for data flow across multiple pre-specified paths for each source-destination pair, we can regard different paths between the same pair as different routing at packet level. It is clear that the sum of all rates between the same pair is the end-to-end rate of this pair.

As to the STDMA, we assume that time is slotted and divided into frames. Each frame consists of identical time slots, which are of equal length and can be used to transmit one fixed-length packet per slot. The length of the schedule determines the size of a data frame, and the schedule repeats itself from one frame to the next. In our approach, bandwidth requirement is not considered and this assumption is reasonable in many practical conditions, such as the ultra-wideband (UWB) systems [15].

3 Problem Statement

When network traffic distribution is taken into account, that is, the amount of packets for each source-destination pair in a frame is given, then the frame length of the STDMA schedule determines the efficiency of the spatial reuse of the timeslots and further represents the network throughput, which is defined as the largest traffic load

in unit time. We next define the optimization problem for minimum-length scheduling for optimal link assignments.

Given the node and link sets N and L, the transmitting power of each node P_i, the path-loss between every pair of node $L_b(i, j)$, the noise effect N_r, the two threshold g_0, and the amount of packets for each source-destination pair to be transmitted in a frame, our objective is to find the minimum-length traffic-sensitive scheduling for link assignments in order to maximize the network throughput, such that every link that is involved in the transmission of a certain packet receives at least one timeslot, and such that the following interference constraints are satisfied:

- Two links that share a common node must be assigned different timeslots.
- A timeslot can be assigned to a link only if the SIR-constraint (2) for the link is satisfied.

4 Computational Complexity

In this section, we will show that, from the computational complexity point of view, the problem defined in the previous section, denoted by MLSP (Minimum-Length Scheduling Problem) for short, is NP-hard.

Proposition 1. Problem MLSP is NP-hard.

Proof. We show that a special case of MLSP is NP-hard. As is shown in [1], the problem for finding link assignment schedules to maximize the uniform network throughput is NP-hard because it can be transformed into the graph edge coloring problem through the "conflict graph" concept [20]. Consider a special case of MLSP, in which each link is active exactly once in a frame. It can be easily realized that an optimal solution to MLSP is also optimal to the problem in [1], and vice versa. So we have proved that MLSP is NP-hard.

A link-slot integer programming formulation is described in [1] and is not suitable to use because the numbers of variables and constraints grow rapidly with respect to the network size. As to our traffic-sensitive problem, the computational bottleneck can not be overcome and this formulation becomes more intractable. Next we reformulate the original problem using alternative set covering formulations, which has a very simple constraint structure.

The set covering formulation is based on the concept of transmission group, which represents a group of links that can be in simultaneous transmission. Denote ζ_L by the set of all transmission groups of links. We introduce the following integer variables.

- x_l stands for the number of timeslots assigned to transmission group l;
- $s_{ij}^l = 1$ if group l contains link (i, j) and $s_{ij}^l = 0$ otherwise

We suppose there are K source-destination pairs in the network, each consists of two distinct nodes. The k-th pair has n_k links through the fixed multi-hop routing

process and r_k packets to transmit in a frame. Thus from a global point of view, there are $\sum_{k=1}^{K} r_k$ packets and corresponding $\sum_{k=1}^{K} n_k r_k$ links to be scheduled and the traffic load t_{ij} for each link (i, j), that is, the number of packets that pass this link in each frame, can be computed easily from th given routing information.

Then the problem MLSP can be formulated using the following set covering formulation, denoted by SCF for short.

$$SCF: \min \sum_{l \in \zeta_L} x_l, s.t. \tag{3}$$

$$\sum_{l \in \zeta_L} s_{ij}^l x_l \geq t_{ij}, \forall (i, j) \in L \tag{4}$$

$$x_l \geq 0, inetger, \forall l \in \zeta_L \tag{5}$$

In SCF, the objective function (3) is to minimize the total number of assigned timeslots. Constraints (4) ensure that each link can be assigned enough timeslots to support the transmission and constraints (5) indicate that x_l is an integer variable.

The complexity of the set covering formulation lies mainly in the cardinality of the set ζ_L. For networks of realistic size, there are huge numbers of transmission groups. However, the key strength of this formulation is the very simple structure that can be exploited by using a column generation approach. Column generation is especially efficient for problems that typically contain a large number of columns, although very few of them are used in the optimal solution. We note that it has been proposed in [19] to solve the graph coloring problem, which has an equivalent structure to the MLSP problem as we have described in the previous sections. Next we will show the detailed process to solve the original problem using the column generation approach.

5 Column Generation Approach

Column generation approach is considered as a successful story in solving large-scale integer programming problem [21]. Strictly speaking, it is a decomposition technique for solving a structured linear programming (LP), such as the LP-relaxation of the integer programming. In this decomposition scheme, the original LP is decomposed into two parts: a master problem and a subproblem. The master problem contains a subset of columns and the subproblem is solved to identify whether the master problem needs a new column or not. If the master problem has to be enlarged, one or several columns are added to the master problem, which is then re-optimized, and the procedure repeats until it is large enough to fins an optimal solution of the original LP.

5.1 MLSP Solution Method

To apply column generation to MLSP, we consider the LP-relaxation of SCF.

$$LP-SCF: \min \sum_{l \in \zeta_L} y_l, s.t. \tag{6}$$

$$\sum_{l \in \zeta_L} s_{ij}^l y_l \geq \mu_{ij}, \forall (i,j) \in L \tag{7}$$

$$0 \leq y_l \leq 1, \forall l \in \zeta_L \tag{8}$$

It is easily realized that the optimum of LP-SCF is invariant to the predefined value $|T|$. Variable $y_l = x_l / |T|$ has its own meaning: it is the proportion of timeslot that is assigned to group l. Obviously the optimal solution of LP-SCF provides a lower bound to that of SCF.

To get the column generation master problem, we consider a subset of ζ_L, denoted by ζ_L'. To ensure feasibility of the master problem, we let ζ_L' be the set of $\sum_{k=1}^{K} n_k r_k$ links to be scheduled in a frame derived by TDMA scheme, i.e., each group in ζ_L' contains a single link. This yields the following master problem:

$$MASTER: \min \sum_{l \in \zeta_L'} y_l, s.t. \tag{9}$$

$$\sum_{l \in \zeta_L'} s_{ij}^l y_l \geq \mu_{ij}, \forall (i,j) \in L \tag{10}$$

$$0 \leq y_l \leq 1, \forall l \in \zeta_L' \tag{11}$$

When the master problem is solved, we need to determine whether ζ_L' is sufficiently large to find an optimal solution or not. This is equivalent to examining whether there exists any element $l \in \zeta_L$, for which the corresponding variable y_l has a positive reduced cost. Using the LP-dual theory described in [21], the reduced cost of variable y_l is:

$$RC_l = 1 - \sum_{(i,j) \in L} \beta_{ij} s_{ij}^l \tag{12}$$

where $\beta_{ij}, \forall (i,j) \in L$ are the optimal dual variables to constraints (10). So the subproblem should be solved iff the minimum of (12) is negative. We transform it to the following subproblem:

$$\min_{l \in \zeta_L} RC_l = 1 - \max_{(i,j) \in L} \beta_{ij} s_{ij}^l \tag{13}$$

We use the following the following variables and reformulate the subproblem from a computational point of view.

s_{ij} =1 if link (i,j) in included in the group and s_{ij} =0 otherwise;

v_i =1 if node i is transmitting and v_i =0 otherwise.

Thus the subproblem can be formulated as below:

$$SUBPROB.\max \sum_{(i,j)\in L} \beta_{ij} s_{ij}, s.t. \tag{14}$$

$$\sum_{j:(i,j)\in L} s_{ij} + \sum_{j:(j,i)\in L} s_{ij} \le 1, \forall i \in N \tag{15}$$

$$s_{ij} \le v_i, \forall (i,j) \in L \tag{16}$$

$$\frac{P_i / N_r}{L_b(i,j)} s_{ij} + \gamma_1 (1 + M_{ij})(1 - s_{ij}) \ge \gamma_1 (1 + \sum\nolimits_{k\in N, k\ne i,j} \frac{P_k / N_r}{L_b(k.j)} v_k), \forall (i,j) \in L \tag{17}$$

If the solution to the subproblem results in a non-positive reduced cost, the optimal LP-value to the master problem is found. Otherwise, the master problem is re-optimized with a new column added to ζ_L', and the procedure continues until we get the upper bound to the integer optimum of the IP problem.

5.2 Performance Enhancement

The solution process of the column generation approach when solving large-scale network instance, always meets two main difficulties. One is the computing effort at one iteration, that is, solving the subproblems which are integer programming problems, may require excessive computing time; and the other is its poor convergence property, that is, the solution process exhibits a long "tail" to reach the optimal, which is called the "tailing off" effect [21]. Next we propose two enhancements for accelerating the convergence of this method, each to overcome one of the difficulties shown above.

The first enhancement is to set a threshold value for termination control when solving the subproblem, instead of solving it to optimality. In practical, we stop running the subproblems after a predefined time. If the optimal solution found so far means a reduced cost that is no more than the threshold, then the corresponding column is chosen into the master problem. Otherwise, the threshold value is reset and the solution process is resumed. In addition, we impose an upper bound of the threshold (e.g. -0.001). Meanwhile, we note that this improvement will not compromise the solution optimality in our implementation and the upper bound of the threshold value ensures the finite number of iterations when solving the LP-relaxation.

The second enhancement concerns the "maximum feasible group". Intuitively speaking, a transmission group(column) is maximum feasible, if the addition of any new link will make the group infeasible. By ensuring columns added to the master problem are maximum feasible, we can minimize the number of iterations needed before reaching optimality. We need only an additional step after solving the subproblem to obtain a maximal feasible group. The solution can be made maximal feasible by considering the revised subproblem, in which the objective function is to maximize the total number of links and the additive constraints to ensure links that contained in the solution must also be contained in the maximal feasible group.

5.3 Integer Solution

Having introduced how the column generation method solves the LP-SCF, we still need ideas on how to actually obtain integer solutions. The most common tool to get integer solutions is the branch-and-price technique used in [19] for embedding column generation into a branch-and-bound framework. Alternatively, integer solutions can be found using heuristics. In our implementation, the branch-and-bound tree is used and we found that the LP-SCF provides very good approximations to the integer optimal solutions.

6 Numerical Results

In this section, we show our numerical results of the traffic-sensitive scheduling and the column generation method. The former is to show how the network traffics affect the scheduling results and the latter is to demonstrate the efficiency of the column generation method. The test networks used in our numerical experiments are as follows: A 4×4 grid network for the traffic-sensitive scheduling and several networks with different nodes for the column generation method. For each of the test networks, the following computations have been carried out. We used our column generation approach to solve the LP-relaxation of the set covering formulation, and to obtain a feasible schedule using the columns that have been generated. The column generation method is implemented using AMPL [16] and CPLEX (Version 7.0) and the computer we used to conduct our experiments has a 667 MHz CPU and 1GB RAM.

6.1 Traffic-Sensitive Scheduling

As mentioned above, the goal of this simple experiment is to observe the affect of the network traffic on the scheduling results. The grid network is shown in Figure 1. The traffic is unicast and there are 4 paths (1-2-3-4, 5-6-7-8, 9-10-11-12, 13-14-15-16) in the network. The interference radius of each node is the same, and is set to be equal to the distance between the adjoint nodes.

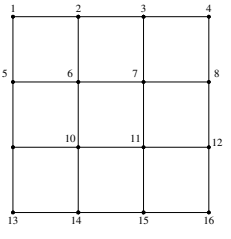


Fig. 1. The 4×4 grid network

Firstly, we set the amount of packets that pass through each path to (1, 1, 1, 1), which implies that in each frame the traffic load for each path is one packet. The result shows that there are at least 5 timeslots in a frame. Then we change the traffic to (1, 1, 1, 2), and now the result is 7. We declare that that with various network traffics, the optimal link scheduling is different and thus the network throughput can change

dynamically. Obviously the optimal schedule may not be unique, though they have the same number of timeslots in each frame.

6.2 Results of the Column Generation Method

We have used three test networks of different sizes, the numbers of the nodes in which are 20, 40 and 60. The characteristics of these networks and computational results of link assignment are summarized in Table 1. The second and third columns in the table show the network dimension in the numbers of the nodes and (directed) links. The following two columns display the maximum and minimum traffic load of a link. Using CPLEX, we solve both the master problem and the subproblem and the last two columns represent the solution time in seconds and the number of iterations, which is equal to the number of columns generated in the solution process. Since the excessive computational effort is required to solve the subproblems, we have used the enhancements we have explained in the above section.

Table 1. Results of the column generation approach

Network	$\|N\|$	$\|L\|$	$\min_{(i,j)\in L} t_{ij}$	$\max_{(i,j)\in L} t_{ij}$	Iterations	Time (second)
N1	20	147	1	45	165	236
N2	40	202	1	267	188	812
N3	60	385	1	511	365	7546

To speed up the column generation process, the subproblem is not solved to optimality and a threshold value on the reduced cost is used for termination control. In particular, we halt the solution process for the subproblem after a time limit of 15 seconds. If the best solution found so far yields a reduced cost that is not less than the threshold, we then consider this column as the one added into the master problem. Otherwise, we half the threshold and the process is run again. The initial threshold value is 0.005 and we impose a minimum value of 0.001. We observe that the LP-bound obtained from the column generation method is very close to the integer optimum. We also note that spatial reuse is achieved for all the networks and thus demonstrate the efficiency of the spatial TDMA scheme and also column generation approach.

7 Conclusions and Future Work

Resource allocation to maximize the total network throughput is a crucial issue in the design of STDMA wireless ad-hoc network. In this paper, we have studied the scheduling problem of allocating timeslots to the network units taking traffic load into account. We prove that the original problem is NP-hard, and when using a set-covering formulation, a column generation technique can be applied to compute the minimum

frame length efficiently even for networks of realistic sizes. Our numerical experiments demonstrate the efficiency of this approach not only because it yields very tight bounds to the optimal scheduling results, but for its computational efficiency compared to other previous approaches.

There are several directions for further research. One particular problem is the cross-layer optimal scheduling problem, that is, joint routing and link assignment. We note that, using our framework of methodology, it is possible to find the optimal scheduling efficiently. Another very interesting topic is to take QOS constraint for traffic load into account; for example, we can consider the situation under which each packet has its own delay bound for multi-hop routing, which is expressed in timeslots.

References

[1] P. Bjorklund, P. Varbrand and D. Yuan, "Resource optimization of spatial TDMA in ad hoc radio networks," In Proceedings of the 2003 Infocom, San Francisco, CA, April1-3 2003.

[2] S. Toumpis, A. J. Goldsmith, "Capacity Regions for Wireless Ad hoc Networks", International Symposium on Communication Theory and Applications, April 2001.

[3] T. Elbatt and A. Ephremides. "Joint Scheduling and Power Control for Wireless Ad-hoc Networks", Proc IEEE Infocom, 2002.

[4] Juha Leino. Applications of Game Theory in Ad Hoc Networks. MASTER'S thesis in Department of Engineering Physics and Mathematics of Helsinki University of Technology.

[5] M. Johansson and L. Xiao. "Cross-layer optimization of wireless networks using nonlinear column generation", In Proceedings of Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks, 2004.

[6] R.L.Cruz and A.Santhnam, "Optimal Routing, Link Scheduling, and Power Control in Multi-hop Wireless networks", Proceedings of the IEEE INFOCOM, San Francisco, CA, April1-3 2003.

[7] Bay area wireless users group. http://www.bawug.org/.

[8] P. Varbrand, D. Yuan and P. Bjorklund. "Resource optimization of spatial TDMA in ad hoc radio networks: A column generation approach". In Proceedings of the 2003 Infocom, San Francisco, CA, April 2003.

[9] A.J.Goldsmith, and S.B.Wicker. "Design challenges for energy-constrained ad-hoc woreless networks". IEEE Transactions on Wireless Communications, pp.8-27, August, 2002.

[10] O.Somarriba and T.C.Giles, "Transmission Power Control for Spatial TDMA in Wireless Radio Networks", Proceedings of the 4th IEEE Conference on Mobile and Wireless Communications Networks, Stockholm, Sweden, September 2002.

[11] O.Somarriba, "Multi-hop Packet Radio Systems in Rough Terrain", Licentiate Thesis, Radio Communications Systems, Department of Signals, Sensors and Systems Royal Institute of Technology, 10044 Stockholm, Sweden, October 1995.

[12] J.Gronkvist, "Assignment Strategies for Spatial Reuse TDMA", Licentiate Thesis, Radio Communications Systems, Department of Signals, Sensors and Systems Royal Institute of Technology, 10044 Stockholm, Sweden, October 2002.

[13] Seattle wireless. http://www.seattlewireless.net/.

[14] A.Howard, M.J.Mataric, and G.S.Sukhatme, "Mobile Sensor Network deployment using potential fields: Adistributed scalable solution to the area coverage problem", in Proc. Int. Conf. Distributed Autonomous Robotic Systems, Fukuoka, Japan, June 2002, pp. 299-308.

[15] http://www.ntia.doc.gov/osmhome/reports/uwb/uwb.pdf
[16] R.Fourer, D.M.Gay, W.Kernighan, AMPL – A Modeling Language for Mathematical Programming, Boyd & Fraser, Danvers, MA, 1993.
[17] B.Hajck and J.Wieselthier, "Link Scheduling in polynomial time". IEEE Transactions on Information Theory, 34(5): 910-917, September 1988.
[18] D.J.Baker, J. Wieselthier, and A. Ephremides. "A distributed algorithm for scheduling the activation of links in a self-organizing mobile radio network". In Proceedings of the IEEE International Conference on Communications, pages 2.F.6.1- 2.F.6.5, Philadelphia, PA, June 1982.
[19] A.Mehrotra, and M.A.Trick, A column generation approach for graph coloring, INFORMS Journal on Compuiting 8 (1996): 344-354.
[20] Kamal Jain, Jitendra Padhye, Venkat Padmanabhan, and Lili Qiu, "Impact of Interference on Multihop Wireless Ntwork Performance", MobiHoc 2003, Maryland, USA, June 1-3, 2003.
[21] Marco E.Lubbecke, Selected Topics in Column Generation. Marco E. Lubbecke. Department of Mathematical Optimization. Braunschweig University of Technology. Technical report 008-2004.

Modeling and Performance Evaluation of Handover Service in Wireless Networks*

Wenfeng Du[1], Lidong Lin[2], Weijia Jia[1,2], and Guojun Wang[1]

[1] College of Information Science and Engineering,
Central South University, Changsha 410083, China
Duwenfeng@yeah.net, Itjia@cityu.edu.hk
Csgjwang@mail.csu.edu.cn
http://sise.csu.edu.cn/cise/index.asp
[2] Department of Computer Engineering and Information Technology,
City University of Hong Kong, Hong Kong, SAR China
Lin.lidong@student.cityu.edu.hk

Abstract. With the development of wireless network, more and more applications require the QoS for message transmission. In order to enhance the QoS service in the dynamic changing wireless networks, this paper proposes two channel allocation schemes to improve the resource utilization of the base station, named FSC and FRC. FSC assigns the available shared channel to the handover call or the reserved channel when the shared channels are fully occupied. FRC, however, assigns the free reserved channel or the shared channel when the reserved channels are fully occupied. We use two-dimension Markov model to analysis the new call blocking probability and handover call dropping probability. Extensive numeric results show that the two schemes have strongly influence on the network resource utilization. In order to improve the performance of base station, the tradeoff between the number of services channel and the QoS of base station must be considered.

1 Introduction

Recently, with the quick development of wireless networks, more and more people begin to access the Internet by using wireless equipments [1]. It is supposed to provide services to mobile user anytime, anywhere in an uninterrupted and seamless way, and a lot of services which were provided by the wired network have been supported by the wireless equipments now. QoS guarantee for end-to-end service has been the fundamental issues in wireless cellular networks.

One of the key elements to provide QoS guarantee is an effective bandwidth allocation policy, which not only fully utilizes the scarce wireless bandwidth, but also guarantees call termination and call blocking probabilities. When a call

* This work is supported by the National Grand Fundamental Research 973 Program of China under Grant No.*2003CB317003*. Research Grant Council RGC Hong Kong, SAR China (projects nos.*CityU 1039/02E*,*CityU 1055/01E*) and Strategy Grant of City University of Hong Kong under nos *7001709* and *7001587*.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 229–238, 2005.

reaches a base station it may be accepted or rejected. When the call finishes within the coverage area of current base station, it will release the channel, or it will be handed over to another cell which the mobile equipment may move in.

There are some schemes to handle the handover call in a priority way by reserved channel [7][8], but there are less discussion about the reserved channel allocation scheme. Due to the different performance of the base station with different channels allocation scheme, how to make good use of the reserved channel becomes an important issue. In this paper, we will analyze the performance of base stations from the viewpoint of the reserved channels allocation and the number of reserved channels, and give the analysis on the call termination probabilities, call blocking probabilities and the percentage of bandwidth utilization.

The rest of the paper is organized as follows. Section 2 introduces the general channel allocation process and proposes of two channel allocation schemes. A two-dimension Markov process is used to analyze the new call blocking probability and the handover call dropping probability for the channel allocation schemes. Two performance metrics were provided in Section 3 to evaluate the schemes. In Section 4, some performance results are presented for base station with different channel allocation schemes and the relationship between channel allocation and the number of reserved channels is also analyzed. Section 5 concludes the paper and discusses the future research.

2 Channel Allocation Process and Model

The radio coverage of each base station is called a cell, such as Personal Communication System (PCS) [2][3]. Within each cell, there are mainly two classes of call traffic: new call and handover call. A new call is the one which initiates in the current cell, while a handover call is the one which initiates in another cell, and is transferred (handed over) to the current cell [4].

When a call enters the current cell, the unused channel will be assigned to it. If no channel is available, which depends on the channel allocation scheme, the call may be blocked in the system. If a call is assigned to a channel, it will not release the channel until the call is finished or handed over to another cell. From the viewpoint of a user, it is better to block a new call than dropping an ongoing call [6]. Since all handover calls are very sensitive to interruption and have more stringent QoS requirement, such as voice communication, the forced termination of an ongoing call should be considered to be more serious than blocking a new call attempt. Therefore, it is a good method to queue a new call and give way to handover call. Due to the scarce resource, with more traffic, the residual capacity of the cell's size is getting smaller and smaller, which may increase the call handover frequency [5]. Therefore, it is critical to analyze the tradeoff between the QoS and number of mobile devices served.

Many previous proposed approaches treat the handover calls with priority [7][8], thus the handover calls are generally given a higher priority than a new call in the proposed schemes through reserved channels [9][10]. The channels are categorized into SC (Shared channel) and RC (Reserved channel). The SCs can

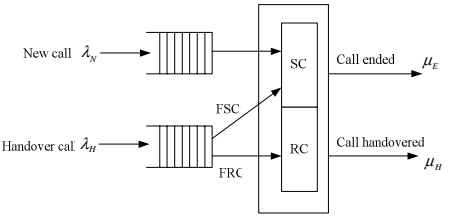


Fig. 1. The channel allocation model in wireless networks

be shared by new calls and handover calls, but RCs can be only allocated to the handover calls.

There are different channel holding percentage and call blocking probability for different channel allocation scheme. When a handover call enters the coverage of current cell, it can attain the available channel in RCs or in SCs. There are two ways to offer channel to the handover call: The first way is the cell tries to allocate a SC to the handover call first. If all SCs are busy, the cell stops providing services to the coming new calls and allocate RCs to the handover call; The other way is the cell chooses the available RCs first to the handover call. When there is no RC available, the handover call will share the SCs with the new calls.

The different number of RCs will make the base station runs in different performance. We propose two strategies for allocating channels to the new and handover calls and present a probability model for analysis the performance evaluation. The working process is shown in Fig. 1.

Without loss of generality, we make the following assumptions and notations. There are totally N channels in each cell, including m reserved channels and $N-m$ shared channels. Handover and new calls arrive at the cell follows Poisson processes with rates λ_H and λ_N respectively. The call end in each channel follows an exponential process with rate μ_E and the call that will be handed over to the neighbor cells in each channel follows an exponential process with rate μ_H. Two channel allocation strategies, FSC (Fistt SC) and FRC (First RC), were considered based on the arrival of handover call.

2.1 FSC (First SC) Allocation Scheme

The handover calls are first allocated with SC then allocated with RC. If there are unused SCs, the handover call shares SCs with the new call. If all SCs are fully occupied, the coming new call is blocked and the handover calls will be assigned to the RCs. Two cases are analyzed as follows:

1. SC is not fully occupied at time t: Assume that SCs have already been allocated to k calls, where $0 \leq k \leq N-m$. The number of arrival calls during the time interval $(t, t+\Delta t)$ is $(\lambda_N + \lambda_H)\Delta t + o(\Delta t)$ and the released channels are $k(\mu_E + \mu_H)\Delta t + o(\Delta t)$,where we generically denote $o(\Delta t)$ as a function of Δt such that

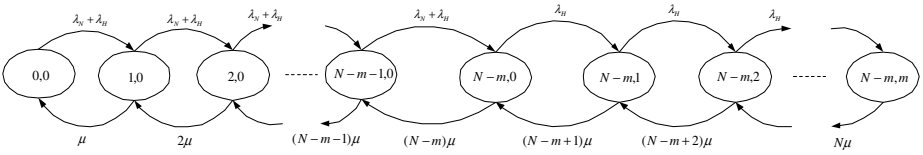


Fig. 2. Transition diagram of FSC

$$\lim_{\Delta t->0} \frac{o(\Delta t)}{\Delta t} = 0$$

Let $p_{ij}(t)$ be the probability of the number of assigned SCs from i to j, it yields

$p_{k,k+1}(\Delta t) = (\lambda_N + \lambda_H)\Delta t + o(\Delta t), k < N - m$
$p_{k,k-1}(\Delta t) = k(\mu_E + \mu_H)\Delta t + o(\Delta t)$
$p_{k,j}(\Delta t) = o(\Delta t), |k - j| \geq 2$
The call arriving rate $\lambda_{k,1}$ and the leaving rate μ_k are as follows.
$\lambda_{k,1}$=$(\lambda_N + \lambda_H)$, $k = 0, 1, \ldots, N - m$ and $\mu_k = k(\mu_E + \mu_H)$

2. SC is fully occupied at time t, i.e., $k \geq N - m$. The arriving rate $\lambda_{k,2}$ and the leaving rate μ_k are expressed as follows:

$\lambda_{k,2} = \lambda_H$, $k = N - m, \ldots, N$ and $\mu_k = k(\mu_E + \mu_H)$

Denote $s(t)$ as the number of occupied SCs at time t , $r(t)$ as the number of occupied RCs at time t. Consider in the steady state, a bi-dimension process $\{s(t), r(t)\}$ is a discrete-time Markov chain. According to above transition state equations, the transition diagram of FSC is shown in Fig. 2.

2.2 FRC (First RC) Allocation Scheme

The handover calls are first allocated with RC then allocated with SC. When the RCs are fully occupied, the handover call shares SCs with the new call. Assume the number of allocated RCs is j. Similar to the above discussion, two cases should be considered:

1. When $j < m$ at time t, then the call arriving rate $\lambda_{j,2}$ and the leaving rate μ_j on queue of RC are shown as follows:

$\lambda_{j,2} = \lambda_H$, $j = 0, 1, \ldots, m - 1$ and $\mu_j = j(\mu_E + \mu_H)$

Consider at time t, the number of calls in SC is k, $k \in [0, N - m]$. Then the call arriving rate λ_k and the leaving rate μ_j on queue of SC are as follows:

$\lambda_k = \lambda_N$, $k = 0, 1, \ldots, N - m$ and $\mu_k = k(\mu_E + \mu_N)$

2. When $j > m$, then RC is saturated, consider the number of calls in the SC is k, $k \in [0, N - m]$ at time t. Then the call arriving rate $\lambda_{k,1}$ and the leaving rate μ_k as follows.

$\lambda_{k,1} = \lambda_N + \lambda_H$, $k = 0, 1, \ldots, N - m$ and $\mu_k = (k + m)(\mu_E + \mu_H)$

According to the above transition state equations, we can derive a two-dimensional transition diagram of FRC as shown in Fig. 3.

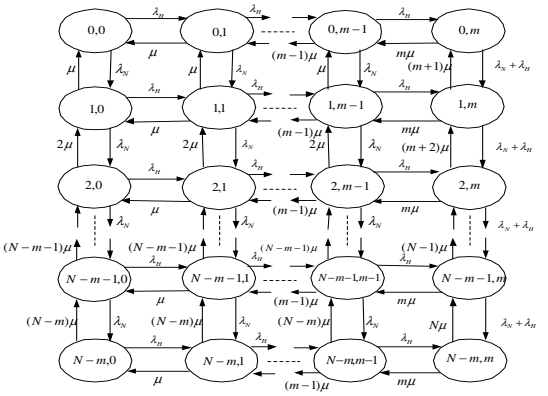


Fig. 3. Transition diagram of FRC

3 Performance Metrics

Some previous research works analyzed the handover scheme using some simple weight functions [11]. Their goal of designing handover scheme is to reduce the Grade of Service (GoS) and to improve the channel utilization [12]. In this section, we discuss two main performance metrics which are used to evaluate our schemes: GoS and channel utilization. In order to describe the impact of terminating an ongoing call on the wireless network's QoS, a punish factor γ was introduced to GoS.

$$GoS = PB + \gamma \times PD$$

Where PB is new call blocking probability and PD is handover call dropping probability. Following the rationale discussed in [12], we set $\gamma \in [5, 20]$. The analysis of the performance metric in the two allocation schemes are illustrated as following:

1. FSC: According to the state-transition diagram in Fig. 2, the stationary distribution is deduced as follows:

$$p_k = \frac{\prod_{i=0}^{k-1} \lambda_{i,1}}{\prod_{i=1}^{k} \mu_i} p_0 = \frac{1}{k!} \cdot \left(\frac{\lambda_N + \lambda_H}{\mu_E + \mu_H} \right)^k \cdot p_0, \quad k \le N - m$$

$$p_k = \frac{\prod_{i=0}^{N-m-1} \lambda_{i,1}}{\prod_{i=1}^{N-m} \mu_i} \cdot \frac{\prod_{i=N-m}^{k-1} \lambda_{i,2}}{\prod_{i=N-m+1}^{k} \mu_i} \cdot p_0 = \frac{(\lambda_N + \lambda_H)^{N-m} \cdot (\lambda_H)^{k-(N-m)}}{k! \cdot (\mu_E + \mu_H)^k} \cdot p_0, \quad k > N - m$$

We can derive p_0 of FSC as

$$p_0 = \left[1 + \sum_{k=1}^{N-m} \frac{\prod_{i=0}^{k-1} \lambda_{i,1}}{\prod_{i=1}^{k} \mu_i} + \frac{\prod_{i=0}^{N-m-1} \lambda_{i,1}}{\prod_{i=1}^{N-m} \mu_i} \cdot \sum_{k=N-m+1}^{N} \frac{\prod_{i=N-m}^{k-1} \lambda_{i,2}}{\prod_{i=N-m+1}^{k} \mu_i} \right]^{-1}$$

The channel busy percentage α of FSC is described as follow.

$$\alpha = \frac{L}{N} = \frac{\sum_{k=1}^{N} k p_k}{N}$$

The blocking probability PB that a new call arrival will find all $N-m$ shared channels busy and will therefore be lost is

$$PB = p_{N-m} = \frac{1}{(N-m)!} \left[\frac{\lambda_N + \lambda_H}{u_E + u_H} \right]^{N-m} \cdot p_0$$

The dropping probability PD that a handover call arrival will find all $N-m$ shared channels and m reserved channels busy and will therefore be lost is

$$PD = p_N = \frac{(\lambda_N + \lambda_H)^{N-m} \cdot (\lambda_H)^m}{N! \cdot (\mu_E + \mu_H)^N} \cdot p_0$$

2. FRC: According to state-transition diagram in Fig.3, when $j \leq m$, we derive the stationary distribution of RC as follows.

$$p_k = \frac{\prod_{i=0}^{k-1} \lambda_{i,2}}{\prod_{i=1}^{k} \mu_i} p_{0,RC} = \frac{1}{k!} \cdot \left(\frac{\lambda_H}{\mu_E + \mu_H} \right)^k \cdot p_{0,RC}, \quad k \leq m$$

$$p_{0,RC} = \left[1 + \sum_{k=1}^{m} \frac{\prod_{i=0}^{k-1} \lambda_{i,2}}{\prod_{i=1}^{k} \mu_i} \right]^{-1} = \left[1 + \sum_{k=1}^{m} \frac{1}{k!} \cdot \left(\frac{\lambda_H}{\mu_E + \mu_H} \right)^k \right]^{-1}$$

On the other hand, the stationary distribution of SC is deduced as follows.

$$p_j = \frac{\prod_{i=0}^{j-1} \lambda_i}{\prod_{i=1}^{j} \mu_i} p_{0,SC} = \frac{1}{j!} \cdot \left(\frac{\lambda_N}{\mu_E + \mu_H} \right)^j \cdot p_{0,SC}, \quad j \leq N - m$$

$$p_{0,SC} = \left[1 + \sum_{j=1}^{N-m} \frac{\prod_{i=0}^{j-1} \lambda_i}{\prod_{i=1}^{j} \mu_i} \right]^{-1} = \left[1 + \sum_{j=1}^{N-m} \frac{1}{j!} \cdot \left(\frac{\lambda_N}{\mu_E + \mu_H} \right)^j \right]^{-1}$$

Similarly, we achieve the channel busy percentage α as follow.

$$\alpha = \frac{L}{N} = \frac{\sum_{k=1}^{m} k \cdot p_k + \sum_{j=1}^{N-m} j \cdot p_j}{N}$$

The blocking probability PB and dropping probability PD are described as follows.

$$PB = \frac{1}{(N-m)!} \left[\frac{\lambda_N}{u_E + u_H} \right]^{N-m} \cdot \left[1 + \sum_{j=1}^{N-m} \frac{1}{j!} \cdot \left(\frac{\lambda_N}{\mu_E + \mu_H} \right)^j \right]^{-1}$$

$$PD = \frac{1}{m!} \left[\frac{\lambda_H}{u_E + u_H} \right]^{m} \cdot \left[1 + \sum_{k=1}^{m} \frac{1}{k!} \cdot \left(\frac{\lambda_H}{\mu_E + \mu_H} \right)^k \right]^{-1}$$

When $j > m$, all channels of RCs are busy. The stationary distribution of SC is denoted as

$$p_k = \frac{\prod_{i=0}^{m-1} \lambda_{i,2}}{\prod_{i=1}^{m} \mu_i} \cdot \frac{\prod_{i=0}^{k-1} \lambda_{i,1}}{\prod_{i=1}^{k} \mu_i} \cdot p_0 = \frac{(\lambda_H)^m \cdot (\lambda_N + \lambda_H)^k}{k! \cdot m! \cdot (\mu_E + \mu_H)^{k+m}} \cdot p_0$$

We can derive p_0 of FRC as

$$p_0 = \left[1 + \sum_{k=1}^{m} \frac{\prod_{i=0}^{k-1} \lambda_{i,2}}{\prod_{i=1}^{k} \mu_i} + \frac{\prod_{i=0}^{m-1} \lambda_{i,2}}{\prod_{i=1}^{m} \mu_i} \cdot \sum_{k=1}^{N-m} \frac{\prod_{i=0}^{k-1} \lambda_{i,1}}{\prod_{i=1}^{k} \mu_i} \right]^{-1}$$

The channel busy percentage α thus can be derived,

$$\alpha = \frac{L}{N} = \frac{\sum_{k=1}^{N-m} k \cdot p_k + m}{N}$$

The blocking probability PB and the dropping probability PD is

$$PB = PD = \frac{(\lambda_H)^m \cdot (\lambda_N + \lambda_H)^{N-m}}{(N-m)! \cdot m! \cdot (\mu_E + \mu_H)^N} \cdot p_0$$

4 Numeral Results and Discussion

Through consideration of different arriving rates for the new calls and handover calls as well as the number of RCs in each base station, we have observed the different performance data. Initially, the parameters are set as follows. There are 10 channels in each cell; New call and handover call arriving are in Poisson process with rate $4/sec$ and $3/sec$ respectively; the residence time of call is an exponential process with rate $1.5/sec$; The value of γ in GoS is 10.

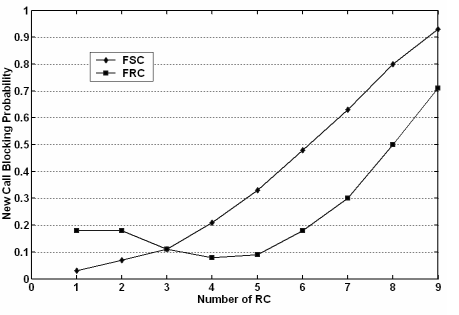


Fig. 4. New call blocking probability with the number of RC

Fig. 4 shows the new call blocking probability of base station with different number of RCs. It can be seen that the new call blocking probability increases with the increment of the number of RC. When the number of RC is lower than 3, the FRC has a higher new call blocking probability than FSC. But when the number of RC is over 4, the new call blocking probability of FSC exceeds that of FRC. Since the FRC scheme allocate RC to the handover call firstly, only all channels in the RC are fully occupied, the handover call will share the channel of SC with a new call, whereas by FSC scheme, the handover call will firstly share the channel of SC with new call. Therefore FSC has higher new call blocking probability than that of FRC.

Figure 5 shows the handover call dropping probability of different number of RCs. According to this figure, we can see that the handover call dropping probability decreases with the increase of RC. It also shows that the number of RC is a critical factor to determine the handover call dropping probability. The more is the number of RC, the smaller the handover call dropping probability is.

Figure 6 shows the change of GoS with the number of RCs. According to this figure, we have the following observations. When the number of RC $<$ 5, GoS of FRC $>$ GoS of FSC. When the number of RC $>$ 5, GoS of FSC $>$ GoS of FRC. The figures also show that the different channel allocation schemes strongly influence the GoS value.

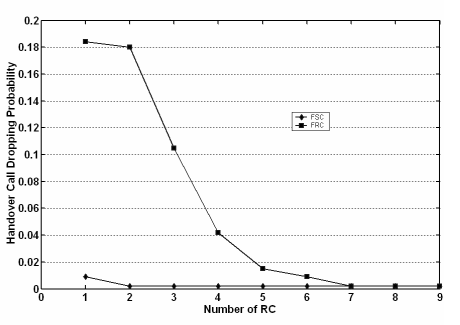


Fig. 5. Handover call dropping probability with the number of RC

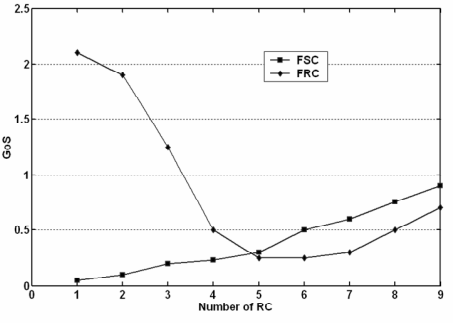


Fig. 6. GoS with the number of RC

5 Conclusions

We have proposed two channel allocation schemes to improve the utilization of base station: FSC (First SC) and FRC (First RC). We also use two-dimension Markov model to analyze the new call blocking probability and handover call dropping probability. The performance metric of *GoS* is proposed to evaluate the two schemes. Extensive numeric results show that the two schemes strongly affect the network utilization. In order to improve the utilization of base station, it is advised that the tradeoff between the number of services channel and the QoS of base station should be considered. Furthermore, channel allocation scheme is critical for improving the network's performance and resource utilizations to achieve low call dropping or blocking probability.

This probability model has discussed in the circuit switch network in which each call will hold the channel until the call is ended or handed over to another cell. But in packet switching networks, the model is not valid and we are currently investigating the model fitting to such networks.

References

1. C. Lindemann and A. Thmmler. Performance analysis of the General Packet Radio Service. IEEE International Conference on Distribute Computing Systems, pp.673-680, 2001.
2. Y. Fang, I. Chlamtac, and Y. B. Lin, Call performance for a PCS net-work, IEEE J. Select. Areas Commu, vol. 15, pp. 1568-1581, Oct.1997.
3. Y. B. Lin, S. Mohan, and A. Noerpel, Queueing priority channel assign-ment strategies for PCS hand-off and initial access, IEEE Trans. Veh.Technol., vol. 43, pp. 704-712, Aug. 1994.
4. I. C. Panoutsopoulos and S. Kotsopoulos, Handover and new call admission policy optimization for G3G systems. ACM Wireless Networks, vol. 8, pp. 381-389, July.2002.
5. Y. B. Lin, A. Noerpel and D. J. Harasty. The sub-rating channel assignment strategy for PCS hand-offs. IEEE Trans on Veh Technol, vol.45, no. 1, pp. 122-130, Feb.1996.
6. S. Tekinay. Handover & Channel Assignment in Mobile Cellular Networks. IEEE Commu Mag, Novem.1991.
7. M.-H. Chiu and Mostafa A. Bassioni, Predictive Schemes for Handoff Prioritization in Cellular Networks Based on Mobile Positioning, IEEE on Selected Areas in Comm, vol. 18, no. 3, pp.510-522, March. 2000.
8. L. Ortigoza-Guerrero and A. H. Aghvami, A Prioritized Handoff Dynamic Channel Allocation Strategy for PCS, IEEE Trans on Veh Technol, vol. 48, no. 4, pp.1203-1215, July. 1999.
9. Hang Chen, Qing-An Zeng, and Dharma, A Novel Analytical Modeling for Optimal Channel Partitioning in the Next Generation Integrated Wireless and Mobile Networks, Proceedings of the 5th ACM international workshop on Modeling analysis and simulation of wireless and mobile systems, pp.120-127, Sept. 2002.
10. W. Li, X. Chao, Modeling and Performance Evaluation of a Cellular Mobile Network, IEEE/ACM Trans on Networking, vol.2, no. 1, pp. 131-145, Feb. 2004.
11. R. Ramjee, D. Towsley, R. Nagarajan, On optimal call admission control in cellular networks, Wireless Networks, no 3, pp. 29-41, 1997.
12. J.Moreira, Nelson, E.A. de, Comparison of Handoff Resource Allocation Strategies through the State-Dependent Rejection Scheme, 17th International Teletraffic Congress, pp.323-334, 2001.

The Optimum Parameter Design for WCDMA Intra-frequency Handover Initiation

Donghoi Kim[1] and Joinin Kim[2]

[1] Mobile Telecommunication Research Laboratory,
Electronics and Telecommunications Research Institute,
161 Gajong-dong, Yuseong-gu, Daejeon 305-350, Korea
donghk@etri.re.kr
[2] Korea Advance Institute of Science and Technology (KAIST),
371-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Korea
jongin97.kim@samsung.com

Abstract. WCDMA handover algorithms employ signal averaging, hysteresis and the Time-to-Trigger mechanism to optimize the trade off between number of unnecessary handover, reported events (system load) and handover delay time. We investigate optimal parameters for the WCDMA intra-frequency handover algorithm and the impact of each parameter on the system performance. Number of reporting events triggered for handover and handover delay are key performance measures in this paper. The study shows various tradeoffs between the parameters related to averaging, hysteresis and Time-to-Trigger. We have also discovered that the Layer3 filter and Time-to-Trigger mechanism may cause negative effects on each other in some cases and there are optimum values, when used simultaneously.

1 Introduction

Design of handover initiation can be made to use several measurements such as the received signal level from the communicating and neighboring base stations, the path loss to the base stations and bit error rate. In general, hysteresis and signal averaging is employed to enhance the performance of handover(i.e. probability of unnecessary handover at the expense of handover delay). Previous studies on handover initiation have revered that there are trade offs between handover delay and number of unnecessary handover.

Handover initiation criteria analyzed in literature are mainly based on the length of averaging window, the threshold level and the hysteresis margin. In addition, WCDMA introduce the Time-to-Trigger mechanism to reduce unnecessary signaling and ping pong effects. Also, averaging window is used to smooth out random signal fluctuations and to make handover decisions to be based on underlying trends and not instantaneous changes.

Soft handover is essential for intra-frequency in WCDMA. The active set is defined as the set of base stations to which the mobile users are simultaneously connected. Soft handover involves active set update procedure which include

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 239–248, 2005.

signaling of appropriate event triggered by the mobile based on the measurement of the measurement quantity (i.e. Ec/Io, path loss, etc). Frequent reporting may cause unnecessary handover and signaling overload. On the other, if the reporting is too seldom, it may increase the handover delay.

WCDMA (3GPP) recommendation does not specify the measurement and averaging interval be fixed or variable. Actual physical layer measurement depends on the implementation of the mobile unit. However, WCDMA specifies the network controlled features to enhance the the performance, which include the hysteresis, Time-to-Trigger and Layer3 filtering. A network controlled Layer3 filtering (exponentia1 smoothing) provides same options as to hysteresis and Time-to-Trigger to some extent, but give some extra benefits which makes it possible to control the rate of reporting, i.e. system loads. Therefore, it is our interest to investigate the impacts of each of the network controlled elements, including Layer3 filter, hysteresis margin and Time-to-Trigger, in order to analyze handover performance after applying a minimal physical layer measurement. Our goal is to optimize the parameters for these handover mechanisms considering various tradeoff relations. By using an appropriate combination of filter, hysteresis and Time-to-Trigger, it is possible to fine tune the real time decisions to be optimal in time and amplitude. Therefore, we can optimize parameters related to handover decision.

2 System Description

2.1 Measurements and Signaling

In WCDMA system, the mobile station performs intra-frequency measurement and sends measurement report to the Radio Network Controller (RNC), where the final decision is made about which cell to add or remove from the Active Sets [1]. The intra-frequency measurement is done on the downlink P-CPICH [2]. Measurement quantity can be any of the followings; Ec/Io, path loss and the Received Signal Code Power [2].

Consider a model for a network controlled handover filtering (Signal averaging) shown in Figure 1. This model is as recommended in 3GPP specification [2]. Parameter 1 is related to shape of Layer3 filter provided by the network and parameter 2 is related to types of handover, i.e. intra-frequency, inter-frequency, etc and reporting criteria.

In Figure 1, physical layer implementation (inputs A and Layer1 filtering) is not constrained by the standard i.e. the model does not state a specific sampling rate or even if the sampling is periodic or not. What the standard specifies is the performance objectives and reporting rate at point B in the model. The reporting rate is equal to the measurement period, which is 200ms for intra-frequency measurement. The performance objectives for the physical layer measurements are specified in [3]. In addition, the Layer3 filtering is performed according to the following exponential averaging formula to give more accuracy.

$$F_n = (1 - a)F_{n-1} + aM_n \tag{1}$$

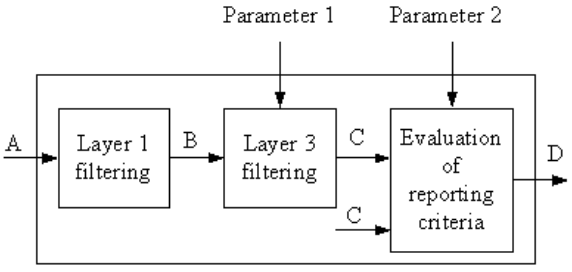


Fig. 1. Model for handover measurements

The variables in (1) are defined as follows; F_n is the updated filtered measurement result. F_{n-1} is the old filtered measurement result. M_n is the latest received measurement result from physical layer measurements. If a is set to 1 that will mean no Layer3 filtering. Also, smaller a will mean that it is giving more weights to past samples. Hysteresis and Time-to-Trigger mechanism on the other hand is important for reducing unnecessary signaling or handover and they complement to averaging mechanism. Evaluation of reporting criteria is based on the measurement results (after appropriate filtering) using the hysteresis and Time-to-Trigger mechanism. The reporting event1A and event1B is defined as;

$$\begin{aligned} &\text{Mean\_Sign} > \text{Best\_Ss - Hyst\_Add} \quad \text{for} \quad \triangle T : \text{event1A} \\ &\text{Mean\_Sign} < \text{Best\_Ss - Hyst\_Drop} \quad \text{for} \quad \triangle T : \text{event1B} \end{aligned} \tag{2}$$

If measured quantity at point C (Meas_Sign) of Figure 1 is continuously larger than the best measured set present in the active set (Best_Ss) minus hysteresis (Hyst_Add) for Time-to-Trigger ($\triangle T$), then measurement reporting message for the eventlA is sent over the air interface to the RNC as shown in Figure 2. Similarly, the eventlB can be reported. The reporting events constitute

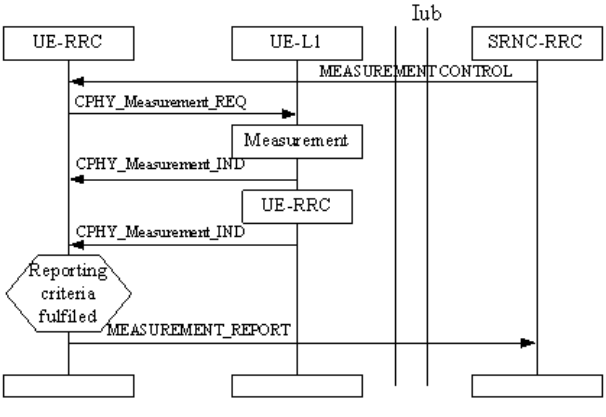


Fig. 2. Reporting of Measurement

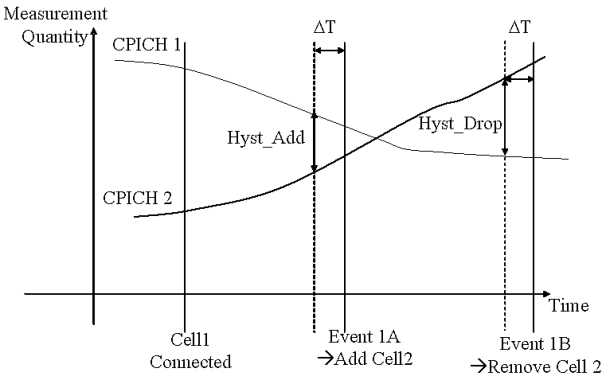


Fig. 3. Simple handover algorithm

basic input to handover algorithm in the RNC, where the handover decision is made (i.e. active set addition, active set removal).

2.2 Handover Scheme

An example of basic handover algorithm is shown in Figure 3, which exploits the hysteresis and Time-to-Trigger mechanism. The event1A and event1B are implemented in this example. Since the soft handover algorithm is performed at RNC, load control strategy and other radio resource management strategy can be exploited for active set updates considering any of the measurement quantities described in [1].

3 Simulation Model

We consider two cells each with radius of 2000m and mobile is allowed to move from BS1 to BS2 in a straight line joining them (Figure 4). Our measurement model is identical to that shown in Figure 1. We apply a basic Layerl filter, which takes CPICH RSCP samples every l0ms at input A of Figure 1 and then 20 samples are averaged over a rectangular block for the duration of measurement period (i.e. 200ms). A typical signal output of Layerl filtering in our model, which in practice depends on the implementation of mobile handset, is shown in Figure 4.

As it can be seen, the basic Layerl filtering of 200ms in our model does not completely average out the signal fluctuation. Relationship between the accuracy and the measurement distance is described in [5]. Our interest is to investigate the effects of the network controlled elements, such as Layer3 filter, hysteresis margin and Time-to-Trigger, on the handover performance after applying a minimal Layerl filtering. The simulation parameters are listed below and the channel model is described in the following subsection.

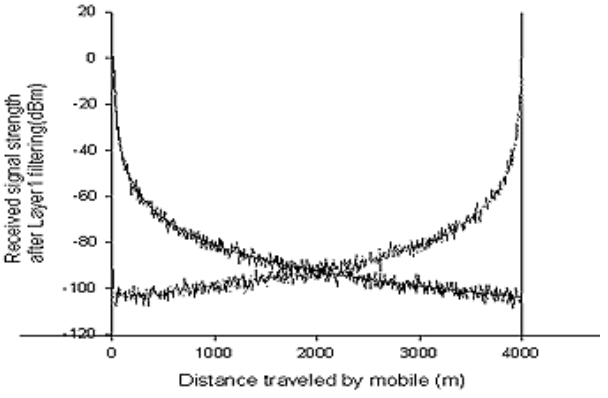


Fig. 4. Model for handover measurements

Table 1. Simulation parameters

Parameter	Value
Propagation Model	128.1+37.6log (R)
Channel Model	ITU-Vehicular A
Standard Deviation of Log-Normal fading	10dB
Decorrelation destance	20ms
CPICH Power	47dBm
Hyst_ADD	1.5dB,3dB,4.5dB,6dB
Hyst_Drop	2.5dB,5dB,7.5dB,10dB
Time-to-Trigger	0ms,200ms,400ms
Measurement Period	200ms
Sampling interval	10ms
Layer3 filter coefficient	0.1∼1

3.1 Propagation Model

The received signal at a mobile consists of three parts; path loss, slow fading and fast fading (Rayleigh distributed). Therefore, the received signal (in dB) as a function of mobile distance is given by,

$$r(d) = K_1 - K_2 \log_{10}(d) + \nu(d) + 20 \log_{10} e(d) \tag{3}$$

The parameters $K1$ and $K2$ accounts for path loss, $\nu(d)$ is the shadow fading process; zero mean, variance 10dB, stationary Gaussian process. The shadowing process is assumed to have the exponential correlation function proposed by [6]. Decorrelation distance is assumed to be 20m in vehicular environments [4].

$$R(\triangle x) = e^{\frac{|\triangle x|}{d_{corr}} \ln 2} \tag{4}$$

For the fast fading, we use ITU Vechicular A model [4]. Received signal after filtering is then given by

$$\tilde{r}(d) = K_1 - K_2 \log_{10}(d) + \tilde{\nu}(d) + 20\widetilde{\log_{10}} e(d) \tag{5}$$

3.2 Performance Measures

Optimal handover is the trade off between the number of unnecessary handover and the handover delay. Many previous literatures have studied the properties of this trade off for various parameters such as the hysteresis margin and the length of averaging distance [8]. In WCDMA, the standards specify the measurement model and the range of parameters like Layer3 filter coefficient, hysteresis and Time-to-Trigger. But, the impacts of these parameter and different choices for the values remain to be clarified. WCDMA uses soft handover mechanism to enhance the coverage and capacity of the network. Soft handover mechanism involves active set update and removal as described in the previous section. Too many reporting events will cause unnecessary active set updates and increase the signaling load. On the other hand, infrequent reporting may cause delay in handover. Optimal size of soft handover depends on loading conditions and, etc. The size of soft handover area can be also controlled by the system parameters.

Number of reporting events triggered for handover and handover delay are key performance measures in this paper. In our simulation, the tradeoff between the number of reporting events and average distance of active set addition/remaval, averaged over 1000 runs, is investigated with different hysteresis margins, Layer3 filter coefficients and Time-to-Trigger. Average distance of reporting event1A is the mean distance at which the active set addition for BS2 takes place.

4 Simulation Results

Figure 5 shows the expected number of reporting event1A for mobile traveling at speeds 50km/h and 120km/h with various hysteresis, not using Time-to-Trigger. Number of reporting events is quite large when Time-to Trigger is not used. It can be observed that the Layer3 filter can reduce the number of reporting events significantly. Especially at low mobile speed, it shows significant improvements. The effect of the hysteresis is also shown in this figure. Figures 7 and 8 show the mean distance at which the mobile sends the reporting event1A for BS2 and event1B for BS1, respectively. It can be interpreted as the expected point where the mobile is entering/leaving the soft handover area. This position depends on the setting of hysteresis levels, but Layer3 filtering also has effects of delaying the distance of entering/leaving the area. Similarly, as show in Figures 9 and 10, the Time-to-Trigger mechanism also delays the reporting events. The gain of soft handover and optimum size depends on many factors including the system loads and the capacity.

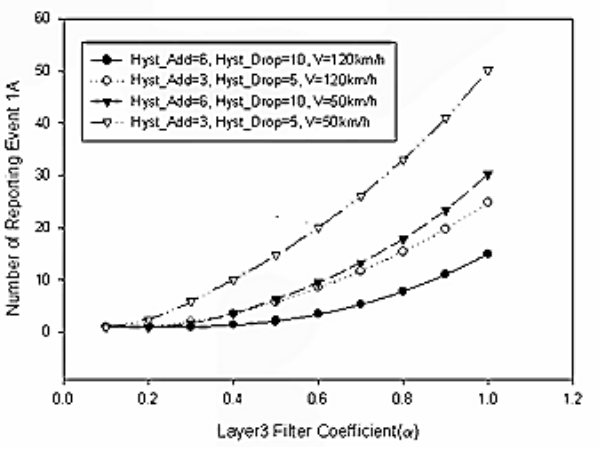


Fig. 5. Average number of reporting event1A when Hyst_add=6 or 3, Hyst_drop=10 or 5, and V=50km/h or 120km/h (Time-to-Trigger=0ms)

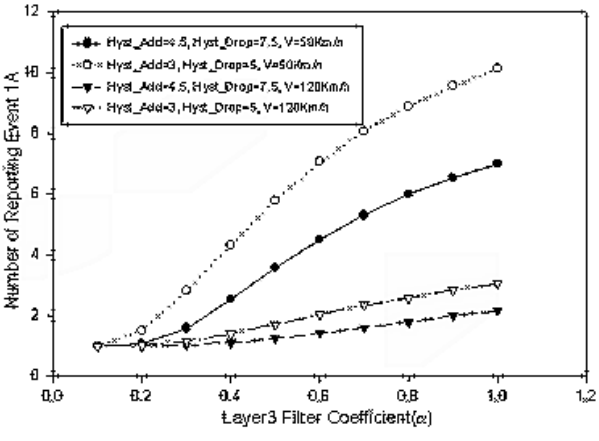


Fig. 6. Average number of reporting event1A when Hyst_add=4.5 or 3, Hyst_drop=7.5 or 5, and V=50km/h or 120km/h (Time-to-Trigger=200ms)

Figure 6 represents the expected number of reporting event1A with Time-to-Trigger of 200ms. It is interesting to observe that the rate of increase of the number of reporting event starts to slow down at some point as the Layer3 filter coefficient is increased. Further, in 400ms Time-to-Trigger, we suppose that the number of reporting actually begin to decrease at some point along the Layer3 filter coefficient. This characteristic is explained as follows. First, with no Time-to-Trigger, the smoother curve will obviously give less

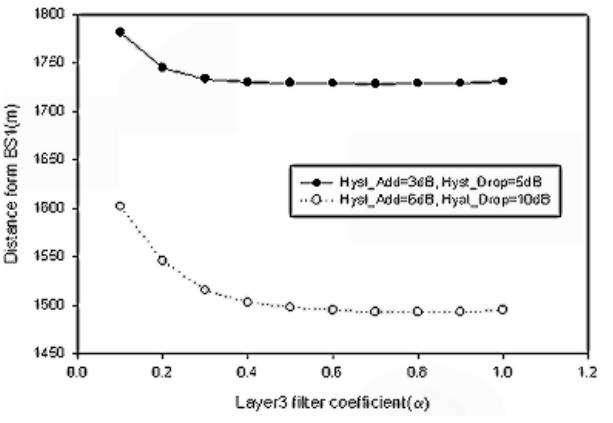


Fig. 7. Mean distance of reporting event1A when Hyst_Add=6 or 3 and Hyst_Drop=10 or 5 (Time-to-Trigger=0ms, 50km/h)

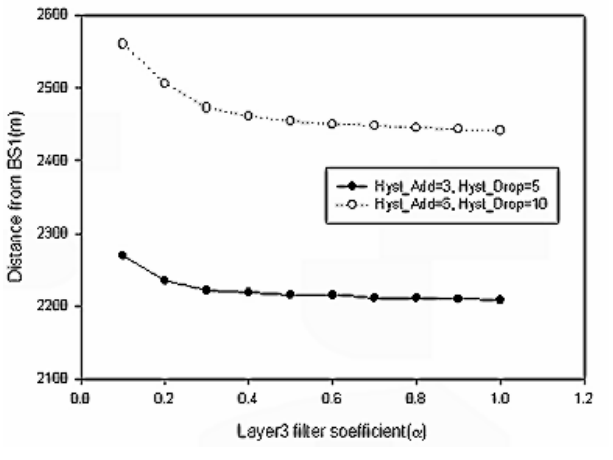


Fig. 8. Mean distance of reporting event1B when Hyst_Add=6 or 3 and Hyst_Drop=10 or 5 (Time-to-Trigger=0ms, 50km/h)

reporting events since it has smaller variations. If the Time-to-Trigger of 200ms and 400ms is used, two and three adjacent samples are subsequently evaluated, respectively. Exponential averaging induces correlations between these samples. Correlated samples will be undesirable than independent samples in extracting the average value. Therefore, there exists a tradeoff between obtaining the stable measurement results and getting independent samples. Consequently, in setting Time-to-Trigger and Layer3 filter constant, we may consider this to be compromised.

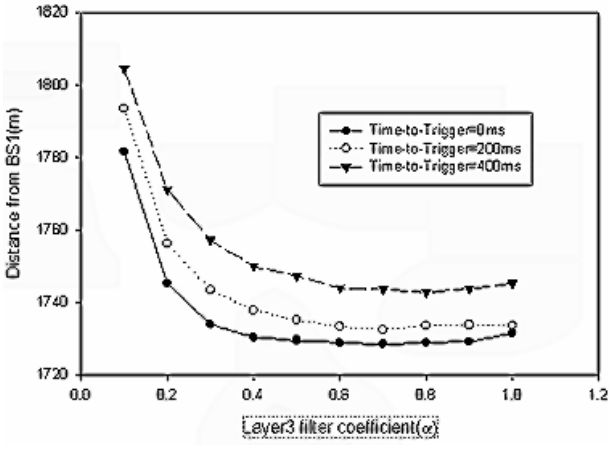


Fig. 9. Mean distance of reporting event1A when Time-to-Trigger=400, 200, or 0ms (Hyst_Add=3,Hyst_Drop=5, 50km/h)

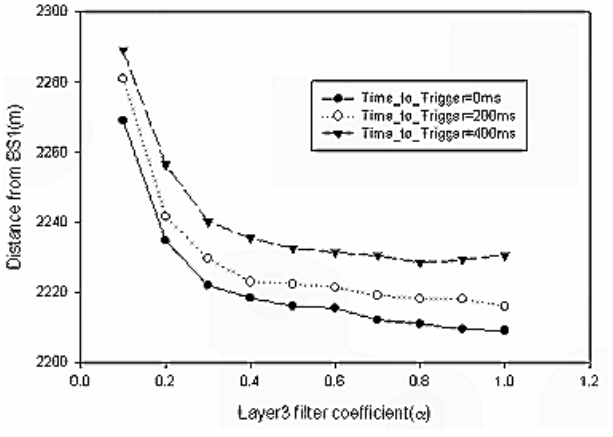


Fig. 10. Mean distance of reporting event1B when Time-to-Trigger=400, 200, or 0ms (Hyst_Add=3,Hyst_Drop=5, 50km/h)

5 Conclusions

This study investigates the impacts of each of the network controlled features (including Layer3 filter, hysteresis margin and Time-to-Trigger) in handover initiation mechanism. The study shows various tradeoffs between the parameters. It was investigated in terms of the number of event reporting and handover delay distance. The performances are also shown to depend on the velocity of the mobile. We have also discovered that the Layer3 filter and Time-to-Trigger mechanism may cause negative effects on each other in some cases anti there is

an optimum combination, when used simultaneously. The results presented in this study may help in understanding the behavior of the features related to triggering of handover measurement reports and in extracting optimum parameter values. Further, these results can be used for developing more efficient handover algorithms.

References

1. 3GPP TS 25.331 Ver 6.4.0, "RRC protocol specification," December, 2004.
2. 3GPP TS 25.302 Ver 6.2.0, "Services prvided by physical layer," December, 2004.
3. 3GPP TS 25.133 Ver 6.8.0, "Requirements for Support of Radio Resource Management," December, 2004
4. ETSI TR 101 112 V3.2.0, "Selection procedures for the choice of radio transmission technologies of the UMTS," April, 1998.
5. C.Y.Lee, "Mobile Communications Engineering," McGrawHill, 1981.
6. M. Gudmunson, "Correlation Model for Shadow Fading in Mobile Radio Systems," Electronics Letter, Vol 27, no23, pp 2145-2146, Nov 1991.
7. R. Vijayan and J. M. Holtzman, "A Model for Analyzing Handoff Algorithms," IEEE Trans. On Vehicular Technology, August 1993.
8. Gregory P. Pollini, "Trends in Handover Design," IEEE Communications Magazine, March 1996.

A New Location Management Scheme for the Next-Generation Mobile Cellular Networks*

Jian-Wu Zhang and Jia-Rong Xi

Telecommunication Engineering Department,
Hangzhou Dianzi University, Hangzhou, 310018, China
hzdzkjdx@zj165.com

Abstract. This paper proposes a location management scheme—combining dynamic location update scheme with static one for the next-generation mobile cellular networks. In the proposed strategy, instead of adopting dynamic location management for all mobile users, classifying them into DLMU(Dynamic Location Management User) and SLMU(Static Location Management User) by a CMR(Call-to- Mobility-Ratio) threshold. Compared with the conventional location update scheme, this strategy can make good use of the dynamic, movement-based location update scheme employed in this paper. Moreover, this paper analyzes how to choose the CMR threshold by simulation. As for the next-generation mobile cellular networks, this proposed scheme can be implemented to reduce the signaling cost evidently, for its facility and feasibility compared with the conventional dynamic location management.

1 Introduction

It's unpractical and almost infeasible for the next-generation mobile cellular networks to adopt a fully dynamic location management scheme, such as distance-based, or movement-based location management, because dynamic location management scheme increases the difficulty of implementation in the mobile communication system.

There are many survey papers on the aspect of location management in mobile cellular networks, such as [1][2][3][4]. In particular, [1] provides a contrast between static and dynamic location management scheme respectively adopted in 3G cellular networks, and [2] provides a comprehensive survey of movement –based location update scheme, [3] and [4] studied the problem to reduce signaling costs in location management through different measures. To sum up, they analyzed signaling costs in location update and paging, or proposed some location management scheme, which is correspondingly complex so that the mobile cellular system has not the ability to support completely.

As a matter of fact, it can be seen from the analysis above, that the next-generation mobile cellular networks may adopt such scheme as combing with dynamic and static location management, thus the mobile cellular system has the ability to provide such location manage scheme, meanwhile, signaling costs can be reduced.

* Supported by the Natural Science Foundation of Zhejiang Province, China, under Grant NO.602136.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 249 – 258, 2005.

In this paper, we propose a new location management scheme, and this new location management scheme combines movement-based location update method with static one.

2 Previous Work

2.1 Location Update Scheme

The current location update scheme adopted by mobile cellular networks, is static one. In such scheme, there are many LAs in a city or a country, each LA includes some cells. The choice of cells in LAs is determined according to the fact of local economy and landform etc. When MT moves out of one LA(recorded by LA1), and into another LA(recorded by LA2), MT must update its location through claiming its current LA2 to the local VLR/MSC and HLR. However, a mass of signaling costs produced in the above static location update scheme, will produce serious influence for the mobile cellular networks, and will not be good for the steady of system performance.

Basically, there are two categories of location management: static and dynamic schemes[1]. In static location management scheme with two-tier mobility databases, the HLR location update and VLR location update are performed when MT enters a LA, and the PA is the same as the LA. Therefore, the LA or PA size is fixed. Similar to the static location update scheme, the HLR location update is performed when MT enters an LA in a dynamic location update scheme. Principally, there are three kinds of dynamic location update schemes in which the LA size is variable [5]: time-based, movement-based and distance-based. Distance-based is the best performance in location management, but its overhead loading on the cellular system, is the highest.

2.2 Aging Scheme

A paging mechanism is needed to search the cells included in a LA registered by the MT, so as to deliver each call to the MT. In current mobile cellular networks, such a paging procedure is carried out by broadcasting paging signals in the LA, once receiving the paging signals, the target MT would send a reply message for the associated system to identify its currently residing cell, subsequently, the connection by radio would be implemented. It is very important to decrease the paging signaling traffic load to the mobile cellular networks by devising efficient paging scheme, in respect that the radio channel resources is limited.

There are many paging scheme, the simplest scheme is “Simultaneous Paging (SiP)”. In such case, all the cells in a LA will be paged simultaneous, thus the paging signaling cost will be dependent on the size of the LA, and the delay time to find the MT is also the shortest. Another important paging scheme is “Sequential Paging (SeP)”, this scheme indicates each cell or each ring of cells is just a PA. On the other hand, if the PA is composed of more than one cell or one ring of cells in SeP, then the scheme may be named “Sequential Group Paging (SeGP)”, by paging in subsection respectively, each subsection is a PA classed by probability. The compare of the performance among the just three paging scheme above, is provided in[6].

If the delay time the system can tolerate, is long enough, SeP is the best scheme one cell by one cell. In fact, such case is impossible, once the delay time is over about 3 seconds or more, the mobile user will be weary of such instance. On the other hand, the paging delay time falling into the scope of constraints on the system, recorded by L, may be accepted by the mobile cellular system and mobile users, therefore SeGP may be more practical and effective, and its paging performance is better. Noted that L is determined by the actual system.

In the next-generation mobile cellular networks, with the appearance and increase of the application on multimedia communication, SeGP may be the best choice to implement to page in constraint to other two, because the character of real time in the procedure of establishing the connection for transmitting data, is not prominent in contrast to transmitting voice, which accounts for almost all the percentage of mobile service application in GSM.

3 System Description

3.1 Model Description

As can be seen from Fig.1, each cell is surrounded by rings of cells. The innermost ring, i.e. the center cell is only ring 0. Ring 1 is surrounded by ring 2 which in turn is surrounded by ring 3, and so on.

The signaling cost C is composed of two parts: location update cost: C_u and paging cost: C_p

$$C = C_u + C_p. \tag{1}$$

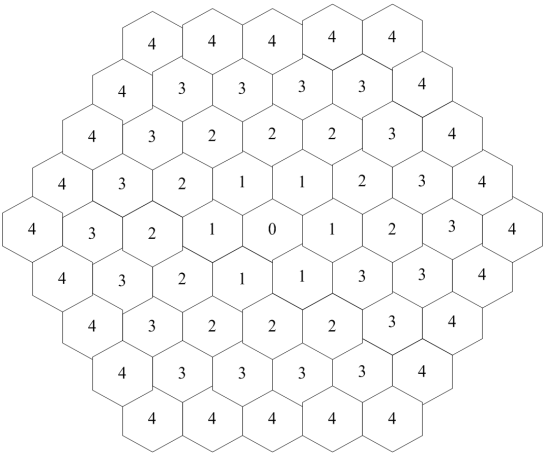


Fig. 1. Cell Rings

When a call to the MT, the cellular system will page every PA in turn, till the target MT would be searched out. Each PA is composed of one or more rings, according to the movement threshold and MT's CMR. In fact, we can think that mobile users can be categorized by the parameter: CMR.

The costs for performing one location update and for paging one cell can be assumed as: U and V, respectively. Let d represents the movement threshold, and $\alpha(j)$ denotes the probability that there are j boundary crossings between two call arrivals. As be in [2][6], the expression for C_u is just as equation (2).Here, $\alpha(j)$ might be given as [2]:

$$C_u = U\sum_{i=1}^{\infty} i \sum_{j=id}^{(i+1)d-1} \alpha(j). \tag{2}$$

Here, $\alpha(j)$ might be given as [2]:

$$\alpha(j) = \begin{cases} 1-1/\rho[1-(\dfrac{1}{\rho+1})], j=0 \\ 1/\rho[1-(\dfrac{1}{\rho+1})]^2(\dfrac{1}{\rho+1})^{j-1}, j>0 \end{cases}. \tag{3}$$

Where ρ is the value of CMR, a quantity that can be achieved by λ_C / λ_m, λ_C is the call arrive rate, and λ_m is the mobility rate (the numbers of crossing cells) during two calls to the MT.

On the other hand, the paging cost C_p can be denoted as follows:

$$C_p = V\sum_{t=1}^{L} P(S_t)S_t. \tag{4}$$

Where S_t is cell numbers during the t times paging, and just the numbers of cells in PA_t, L is the longest constraint delay time the cellular system can tolerate. $P(S_t)$ is the probability that the MT resides in the PA_t. $P(S_t)$ can be achieved by considering the moving as the hexagonal random walk model[2], even adjusting the transfer probability in the hexagonal random walk model through introducing other parameter[7].

We can suppose that PA_t contains rings s_t to e_t as in[2]. If the value of s_t and e_t is also determined by the reference [2]. Then, illogical instance would come into being when d is equal to 7 (or 8) and the constraint delay time L is 5. In such instance, as in [2],

$$s_0 = 0, e_0 = \left\lfloor \frac{7\times 1}{5} \right\rfloor - 1 = 0; s_1 = \left\lfloor \frac{7\times 1}{5} \right\rfloor = 1, e_1 = \left\lfloor \frac{7\times 2}{5} \right\rfloor - 1 = 1;$$

$$s_2 = \left\lfloor \frac{7\times 2}{5} \right\rfloor = 2, e_2 = \left\lfloor \frac{7\times 3}{5} \right\rfloor - 1 = 3;$$

$$s_3 = \left\lfloor \frac{7\times 3}{5} \right\rfloor = 4, e_3 = \left\lfloor \frac{7\times 4}{5} \right\rfloor - 1 = 4;$$

$$s_4 = \left\lfloor \frac{7\times 4}{5} \right\rfloor = 5, e_4 = \left\lfloor \frac{7\times 5}{5} \right\rfloor - 1 = 6; (partition_1)$$

In fact, logical paging area in such case should be:

$$s_0 = 0, e_0 = 0; s_1 = 1, e_1 = 1; s_2 = 2, e_2 = 2;$$

$$s_3 = 3, e_3 = 4; s_4 = 5, e_4 = 6; (partition_2)$$

The same instance would also appear when $d = 8$ and $L = 5$. And other instance is always natural and logical except for the above two cases.

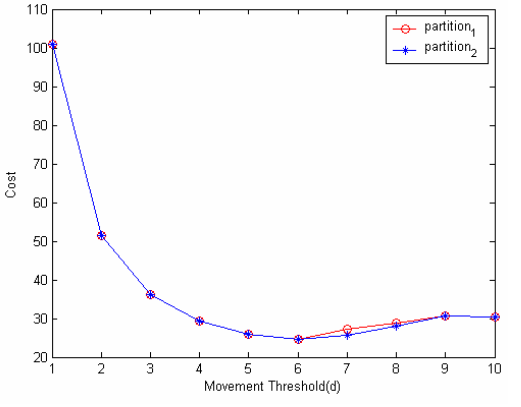


Fig. 2. Compare of two partition ways (U=10,V=1,L=5)

Fig.2 shows the comparisons of the two means: $partition_1$ in[2] and $partition_2$ of this paper, it is evident particularly, to find the difference when d=7 and d=8 as illustrated in Fig.2, which just makes clear that $partition_2$ is better than $partition_1$. In particular, we find that those mobile users whose CMR is in the bound of 0.04~0.05, should adopt to the means of $partition_2$ if the dynamic location management is carried out, and the optimal movement threshold should be 7 and not be 6 in [2], as for those areas where the largest delay time L=5, or some time in one area, when the largest time delay time L=5.

Fig.3. illustrates the change of the total signaling cost C with the increase of movement threshold d. As can be seen from Fig.3, C would increase along with the

decrease of the CMR: ρ . Therefore, we can classify mobile users, by the value CMR, into two parts: static location management mobile users (SLMU) and dynamic location management mobile users (DLMU).

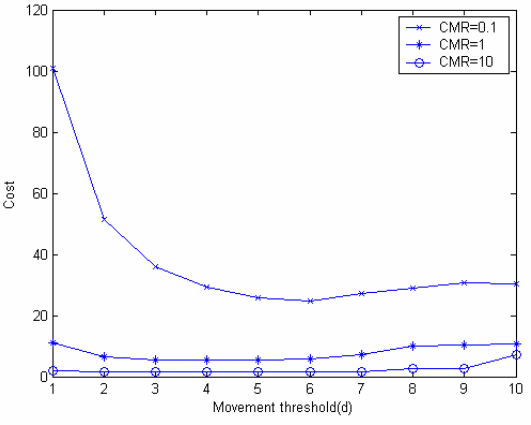


Fig. 3. Comparison of Cost (U=10,V=1,L=5)

For the SLMU, the mobile cellular system may adopt the present static location update scheme, i.e. the PA is the same to LA for the SLMU, which is applied widely in the current mobile cellular networks. As for the partition of the DLMU, the cellular system should adopt the dynamic location update scheme, because the signaling cost for those mobile users' location management is greater than that of SLMU. Furthermore, movement-based location update scheme is the best choice to be implemented in such case.

However, how to judge what kind of mobile users belong to the kind of SLMU and DLMU, respectively? A threshold about the value of CMR, should be determined, to classify mobile users into two parts. Let's analysis the threshold CMR: ρ_0 in the following part.

3.2 Mathematics Analysis and Simulation

Assuming the value of mobile user's CMR submits to Rayleigh distribution as [5], because of the coincidental character.

$$f(\rho) = \begin{cases} \rho / \mu^2 \exp(-\rho^2 / (2\mu^2)), \rho \geq 0 \\ 0, \rho < 0 \end{cases}. \tag{5}$$

We can consider that ρ falls into (0,10], and the signaling cost C is just the function of ρ , i.e. $C(\rho)$, thus the total signaling cost is:

$$C = \int_0^{10} C(\rho) f(\rho) d\rho. \tag{6}$$

Furthermore, as for $C(\rho)$, can be denoted as follows:

$$C(\rho) = \begin{cases} C_{DLMU}(\rho), \rho < \rho_0 \\ C_{SLMU}(\rho), \rho \geq \rho_{0.} \end{cases} \tag{7}$$

In [7], $C_{DLMU}(\rho)$ in (8), is the cost for the mobile users of the DLMU type; and $C_{SLMU}(\rho)$ in (9), is just the cost for SLMU.

$$C_{DLMU}(\rho) = U \sum_{i=1}^{\infty} i \sum_{j=id}^{(i+1)d-1} \alpha(j) + V \sum_{t=1}^{L} P(S_t) S_t. \tag{8}$$

Table 1. COST : CMR Threshold and d

Cost ($\mu = 2$, $L = 5$)							
	CMR threshold: ρ_0						
d	0.1	4	4.8	5.2	6	8	10
1	7.7	7.4	7.3	7.3	7.3	7.3	7.3
2	9.2	5.3	4.8	4.6	4.5	4.4	4.4
5	61.4	13.8	8.3	6.7	5.0	4.1	4.1
8	169.3	32.7	17.1	12.6	7.7	5.3	5.2
10	271.3	52.9	28.2	21.1	13.4	9.6	9.4

$$C_{SLMU}(\rho) = U \sum_{i=1}^{\infty} i \sum_{j=id}^{(i+1)d-1} \alpha(j) + V[3(d+1)^2 + (d+1) + 1]. \tag{9}$$

Thus, the total cost based on the CMR threshold ρ_0 can be expressed as follows:

$$C = \int_0^{\rho_0} C_{DLMU}(\rho) f(\rho) d\rho + \int_{\rho_0}^{10} C_{SLMU}(\rho) f(\rho) d\rho \tag{10}$$

As a matter of fact, if the $\rho_0 = 10$, then the cost will be the minimum, because no SLMU exits, actually. However, such instance is the previous dynamic location management scheme, which can not be come true. Therefore, the optimal threshold, i.e. ρ_0 is not the value that makes the total cost to be minimum, but to lessen the cost C in (6) correspondingly. We are encouraged by simulation result, which shows that the cost C will change slowly when the threshold ρ_0 arrives at a certain value. Hence, we can work out such threshold ρ_0 so as to reduce the signaling cost for those mo-

bile users, i.e. DLMU whose CMR is less than ρ_0 through dynamic location update scheme, because those mobile users contribute to a majority of signaling cost compared with some else mobile users, i.e. SLMU whose CMR is greater than ρ_0.

Fig.4 illustrates C as a function of CMR threshold ρ_0, and movement threshold d, for $\mu = 2$ and the allowable delay time $L = 5$. Fig. 5 gives the results when $\mu = 2$, and $L = 3$. Moreover, the case of $\mu = 3, L = 5$ is shown in Fig.6.

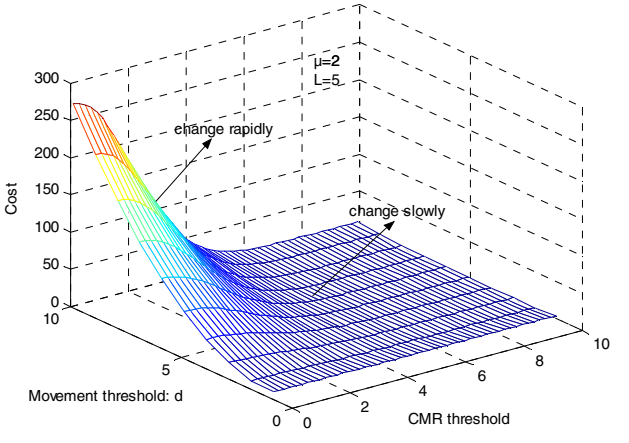


Fig. 4. Cost ($\mu = 2, L = 5$)

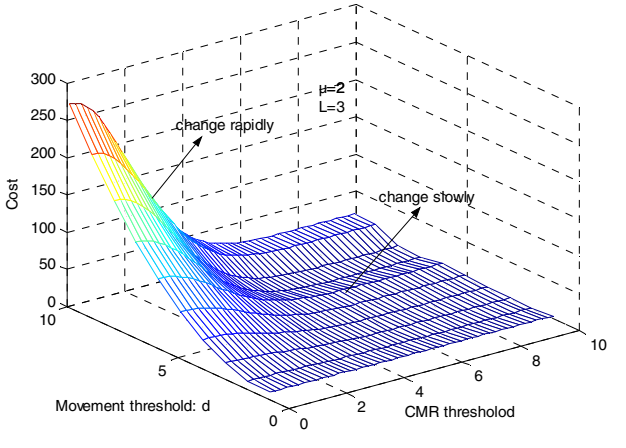


Fig. 5. Cost ($\mu = 2, L = 3$)

As can be seen from Fig.4, Fig.5 and Fig.6, when the CMR threshold is small, and d is great, the value of C changes rapidly, just illustrated by the arrowhead. However, when CMR threshold increases to a certain numerical value, C would changes slowly.

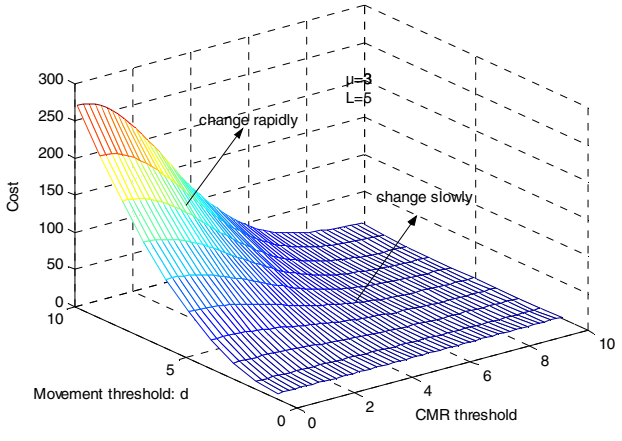


Fig. 6. Cost ($\mu = 3, L = 5$)

Maybe, you would find C is little relatively when CMR threshold is little too and d is 1, just as the instance of $\rho_0 < 2$ and d=1, because the signaling cost for location update is the primary one when the movement threshold d is very small. However, the value of C in such instance, in fact, is much greater than that case whenCMR threshold reaches 4 approximately, here the Cost changes slowly marked by the arrowhead. To illuminate this fact, table I is shown above.

As illustrated in table I above, Cost decreases slowly when ρ_0 increases from 4 to 10. Thus, if the CMR threshold ρ_0 is equal to 4 or about 4, the signaling cost of the cellular system will be less and more controllable correspondingly, moreover the implement of such scheme will feasible because the numbers of DLMU whose CMR is less than ρ_0, is much less.

4 Conclusion

In this paper, we introduced a new location management scheme, which may be applicable for the next-generation mobile cellular networks. This new proposed scheme combines the dynamic movement-based location update scheme with the static one, and classifies the mobile users into SLMU and DLMU, by a CMR threshold, so as to reduce the system signaling cost for location management.

References

1. Yang Xiao, Yi Pan, and Jie Li, “Analysis of Location Management for 3G Cellular Networks”, IEEE Trans. Parallel and Distributed Systems, vol. 15, pp. 339-349, Apr. 2004.
2. IF Akyildiz, JSM Ho and YB Lin, “Movement-Based Location Update and Selective Paging for PCS Networks”, IEEE/ACM Trans. Networking, vol. 4, pp. 629-638, Aug. 1996.

3. IF Akyildiz, Wenye Wang, “A Dynamic Location Management Scheme for Next-Generation Multitier PCS Systems”, IEEE Trans. Wireless Communications vol. 1, pp. 178-189, Jan. 2002.
4. Pablo Garcia Escalle, Vicente Casares Giner, and Jorge Mataix Oltra, “Reducing location Update and Paging Costs in a PCS Network”, IEEE Trans. Wireless Communications, vol. 1, pp. 200-209, Jan. 2002.
5. A. Bar-Noy, I. Kessler, and M. Sidi, “Mobile Users: To Update or Not to Update?”, ACM-Baltzer J. Wireless Networks, vol. 1, pp. 175-186, Jul. 1995.
6. Chang Sup Sung, Hyoun Chul Rim, and Jung Sup Lee, “Effective Paging Procedure for the Optical Feeder Microcellular System” IEEE Trans. Vehicular technology, vol.52, Jul. 2003.
7. Tracy Tung, Abbas Jamalipour, “Adaptive directional-aware location update strategy”, INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS, Int. J. Com. Sys. pp. 141-161. 2004.

Rapid Mobility of Mobile IP over WLAN

Jun Tian and Abdelsalam (Sumi) Helal

Computer & Information Science and Engineering Department,
University of Florida, Gainesville, FL 32611-6120, USA
{jtian, helal}@cise.ufl.edu

Abstract. In this paper, the rapid mobility of MIP/WLAN is emulated on a test-bed. The performance of MIP/WLAN at different moving speeds is evaluated. The result shows that current MIP protocol is not suitable for rapid moving environments. This paper depicts the relationship between the performance and the moving speed and breaks down the handoff latency of MIP/WLAN. A Speed Adaptive MIP extension is proposed and implemented on Hierarchical MIP. The emulation result shows that the Speed Adaptive MIP greatly improves the performance of MIP/WLAN in rapid moving environments.

1 Introduction

Mobile IP [1] is a promising technology to eliminate the barrier of location for the increasing wireless internet usagage. Third generation (3G) wireless networks that are based on a set of radio technology standards such as CDMA2000, EDGE and WCDMA combine high speed mobile access with IP-based services. Mobile IP can be the common macro mobility management framework to merge all these technologies in order to allow mobile users to roam between different access networks.

WLAN provides wireless users with an always-on, wireless connection network. There are currently three major WLAN standards, 802.11b, 802.11a and 802.11g. The performance of WLAN decreases as the distance from the antenna increases. As an example, the bandwidth of 802.11b in an open area will drop from 11, 5.5, 2 to 1 Mbps when the distance increases from 160, 270, 400 to 550 meters. The smaller the cell size the higher the bandwidth, but this indicates more frequent handoffs.

Throughout history, the economic wealth of people or a nation has been closely tied to efficient methods of transportation. A person can drive a car on high way at speed of 12km/h. High speed trains such as France TGV, Japanese bullet, German maglev can travel at speeds of over 320km/h. Could those people surf the internet, communicate with families and enjoy an online movie while traveling at high speeds? In another word, could the current network infrastructure support rapid mobility?

The organization of this paper is as following. Section 2 introduces a rapid mobility emulator. The performance of MIP/ WLAN and its relationship to speeds are shown in section 3. Section 4 breaks down the handoff procedure of MIP/ WLAN and presents a quantitative analysis of the handoff latency. A Speed Adaptive MIP (SA-MIP) is proposed and its performance is evaluated in section 5.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 259 – 267, 2005.

2 Rapid Mobility Emulator

In order to evaluate the performance of MIP/ WLAN, we build up a Rapid Mobile Network emulator, RAMON [2]. RAMON consists of a Pentium II pc as Emulator, a circuit board as Controller, three JFW Industries Attenuators with Antennas, three Cisco 350 Access Points, three FAs, a HA and one or more MNs. The FAs, HA, and MN, which are the major entities of MIP, are running Linux kernel 2.4.20 and are installed with HUT dynamic MIP implementation version 0.8.1[3]. The Attenuators are program controllable device. The Emulator manipulates the Attenuators by the Controller to control the signal strength coming out from the Access Points. By increasing or decreasing the signal strength of one AP, we can emulate the MN moving towards to or away from the AP. By varying the increasing or decreasing speed of the signal strength, we can emulate the speed changes of the MN.

3 Performance of MIP/WLAN in Rapid Moving Environments

Using RAMON, we emulated HUT-MIP in the scenario in Fig.1. In this scenario, a rapid moving MN will travel trough 8 APs. Each AP is wired to a FA. The distance between every two consecutive APs is d=500m or 1000m. The moving speed of MN varies from 10m/s to 80m/s. In our experiments, a large ftp file was transferred from the CN to the MN. The experiment results showed that the time-sequence graph and throughput graph at speed 20m/s and d=1000m is similar to those at 10m/s and d=500m. Also graphs at 80m/s and 1000m are similar to those at 40m/s and 500m.

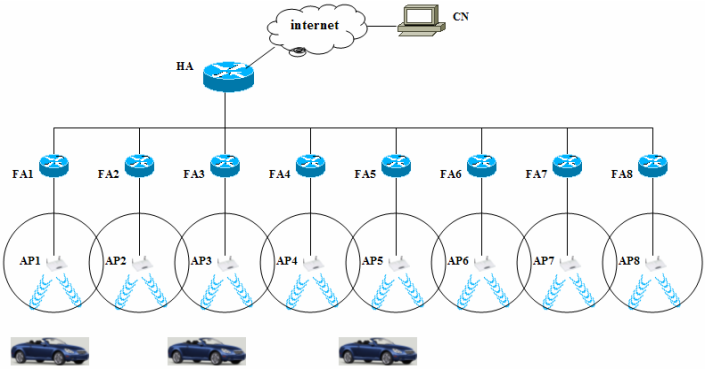


Fig. 1. Emulation scenario for MIP/ WLAN

To compare the performance of MIP/ WLAN at different speeds and different AP distances, we list the experiment data in table 1. In the table, the bytes transferred are the total bytes transferred from when the MN enters the first cell to when it moves out of the last cell. The average throughput is calculated by dividing bytes transferred by travel time. The total handoff time is the summary of the handoff latency of 7 times

handoffs. The effective time is the time for effectively transferring data, which equals to the travel time minus the total handoff time.

Table 1 shows the average throughput drops when the MN's speed goes up. At the same speed of 20m/s, the average throughputs are 92.50kB/s for d=1000m and 76.26kB/s for d=500m. At the speed of 40m/s, the average throughputs are 77.50kB/s for d=1000m and 51.49kB/s for d=500m. If we double the speed and at the same time double the AP distance, the average throughput will stay the same.

Table 1. Throughput at different speedS and AP distances

Speed (m/s)	AP distance (m)	Bytes transferred (kB)	Travel Time (s)	Average throughput (kB/s)	Total hand-off time(s)	Effective time(s)
20	1000	37000	400	92.50	64	336
40	1000	15500	200	77.50	64	136
60	1000	8500	130	65.38	64	66
80	1000	4650	98	48.46	64	34
10	500	36900	397	92.94	64	333
20	500	15100	198	76.26	64	134
30	500	8400	129	65.11	64	65
40	500	5100	101	51.49	64	37

The analysis of table 1 also shows: (1) The handoff time doesn't change with speed. (2) Effective-time/total-travel-time ratio drops when the speed goes up. This is the reason why higher speed has lower throughput. (3) The relationship between the performance of MIP/ WLAN and the moving speed is presented in equation 1:

$$P_{avg} = P_{maxavg}(1 - r_h \times t_{handoff})) \quad (1)$$

Where P_{avg} is the average throughput for the MN; P_{Maxavg} is the average throughput without handoff. $t_{handoff}$ is the average handoff time for each handoff procedure.

We define MN handoff rate as $r_h = v/d$, which is the ratio of the MN's speed and the cell size(AP distance). It means that how many APs or FAs the MN hands over in one second. r_h is also equal to $K_{handoff} / T_{travel}$. Where $K_{handoff}$ is the number of handoffs while traveling and $T_{handoff}$ is the total handoff time while traveling.

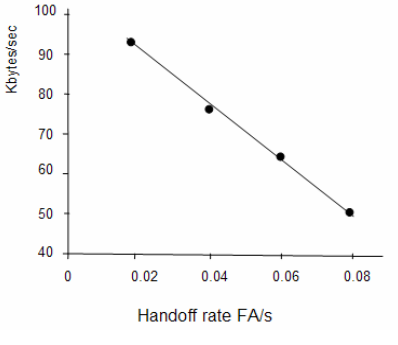


Fig. 2. Average throughput/handoff rate

Since $t_{handoff}$ doesn't change, The change of P_{avg} is caused by handoff rate r_h. Fig.2 shows the relationship between average throughput and handoff rate in equation 1. At handoff rate 0.02 FAs/s, the average throughput is 92.72 kB/s. When the handoff rate goes up to 0.08 FAs/s, the average throughput drops to 49.97 kB/s.

This section shows that the performance of MIP/ WLAN is depending on the MN handoff rate. In section 5, we will propose an idea of how to make use of this throughput/handoff-rate relationship to improve the performance of MIP/ WLAN in rapid moving environment. In the following section, we will take a deep view of the handoff latency by breaking down the handoff procedure of MIP/ WLAN.

4 Quantitative Analysis of the Handoff Latency

MIP, proposed by C. Perkins in RFC3344, is designed independently from all Layer 2 technologies. But such kind of independency also indicates more overhead. Equation 2 gives the life-cycle of MIP/ WLAN handoff procedure:

$$t_{handoff} = t_{L2handoff} + t_{L3handoff} + t_{L4handoff} \quad (2)$$

Where $t_{handoff}$ is the total handoff delay of MIP/ WLAN, $t_{L2handoff}$, $t_{L3handoff}$, and $t_{L4handoff}$ are the handoff cost of Layer2, Layer3, and Layer4 separately.

In the case of IEEE 802.11b WLAN, Layer2 handoff is the change of APs. It causes an interruption of data frame transmission. In our experiment, we split the Layer2 handoff time into three parts and named them as: movement detection, AP searching and reassociation[4]. The detail analysis of three phases of Layer 2 handoff is not given in this paper. The layer2 handoff delay can be expressed in equation 3.

$$t_{L2handoff} = t_{L2detection} + t_{L2seraching} + t_{L2reassociation} \quad (3)$$

Where $t_{L2detection}$, $t_{L2seraching}$ and $t_{L2reassociation}$ are the time costs for Layer2 movement detection, Layer2 AP searching and Layer2 reassociation.

Only after the layer 2 link has been established could the Layer 3 handoff start, because the MN can only communicate with the FA on the same link[6]. The Layer 3 handoff involves 2 phases, agent discovery and registration. The layer3 handoff delay can be splitted into equation 4.

$$t_{L3handoff} = t_{mipagentdicovery} + t_{mipregistration} \quad (4)$$

TCP is a connection-oriented, end-to-end reliable protocol designed to support error recovery and flow control. TCP retransmission follows the exponential back-off algorithm[7]. In our case, during the Layer2 and layer3 handoff, the TCP doubles the retransmission timeout value several times. So even after the layer2 and layer3 handoff is over, TCP still have to wait for RTO to timeout to recover the retransmission. This latency is cost by TCP exponential back-off algorithm. We call it TCP back-off delay $t_{tcp\text{-}back\text{-}off}$.

We define $$t_{L4handoff} = t_{tcp\text{-}back\text{-}off} \quad (5)$$

According the equations 2, 3, 4 and 5, the handoff delay for MIP/ WLAN is shown in equation 6.

$$t_{handoff} = t_{L2detection}+t_{L2seraching}+t_{L2reassociation}+t_{mipagentdicovery}+t_{mipregistration}+t_{tcp\text{-}back\text{-}off} \quad (6)$$

Fig. 3 depicts the handoff latencies of MIP/ WLAN. We used RAMON introduced in section 2 to emulate the same scenario as in Fig.1. We did 20 times experiments to

get the average handoff latency. The experimental result of the handoff latencies of MIP/WLAN is listed in table 2. The handoff latencies are also shown in Fig. 3.

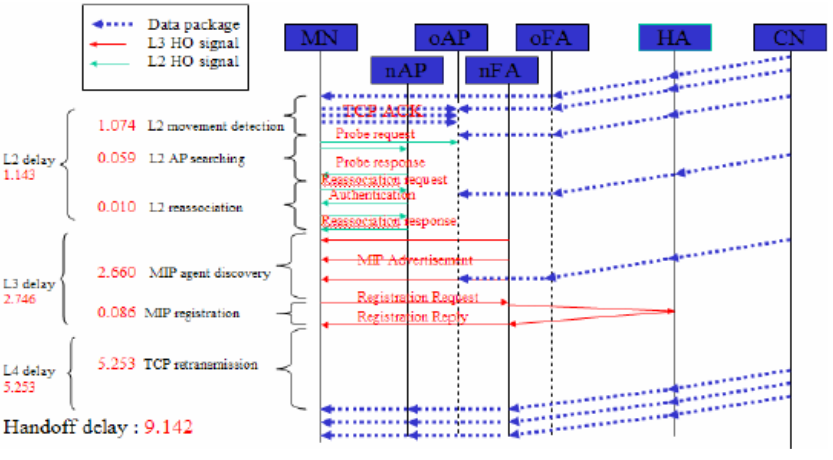


Fig. 3. Handoff latencies with message exchange

Table 2. Handoff latencies of MIP/ WLAN

Delay / Exp#	L2 movement detection	L2 AP searching	L2 reasso-ciation	MIP agent discovery	MIP registration	TCP backoff	Handoff delay
1	1.033	0.061	0.005	2.996	0.073	5.058	9.226
2	1.064	0.044	0.009	1.945	0.042	6.01	9.511
3	1.133	0.063	0.006	3.023	0.052	5.345	9.622
4	1.032	0.100	0.008	2.563	0.050	5.323	9.076
5	1.044	0.065	0.003	2.756	0.052	5.125	9.045
6	1.131	0.057	0.004	2.578	0.043	5.004	8.817
7	1.009	0.056	0.010	2.436	0.060	5.625	9.196
8	1.120	0.060	0.006	3.001	0.704	5.002	9.893
9	1.023	0.059	0.026	2.213	0.054	4.998	8.373
10	1.039	0.076	0.005	3.008	0.053	5.006	9.187
11	1.100	0.045	0.030	2.770	0.041	5.728	9.714
12	1.013	0.049	0.010	2.545	0.042	4.768	8.427
13	1.021	0.051	0.009	3.001	0.065	5.202	8.896
14	1.006	0.043	0.017	2.600	0.046	5.312	9.024
15	1.104	0.069	0.006	2.598	0.047	4.544	8.368
16	1.003	0.064	0.013	2.674	0.062	4.806	8.622
17	1.110	0.054	0.010	2.783	0.054	5.705	9.716
18	1.100	0.064	0.006	3.012	0.057	5.602	9.841
19	1.302	0.056	0.009	2.349	0.070	5.71	9.496
20	1.098	0.044	0.004	2.404	0.062	5.172	8.784
Avg	1.074	0.059	0.010	2.660	0.086	5.253	9.142

Table 2 gives 20 times of experiment data. Each row is one experiment. Each column is the time latency for that handoff phase. The data in the last column are the total handoff latencies for every experiment. The number in the bottom right cell is the average handoff latency.

5 Speed Adaptive MIP

The above quantitative analysis of handoff latencies shows the largest part is the TCP back-off delay $t_{tcp\text{-}back\text{-}off}$. Because of TCP exponential back-off algorithm, if we reduce the L2 and L3 delay, $t_{tcp\text{-}back\text{-}off}$ will be reduced exponentially. The next largest part is L3 latency. In this paper, we first deal with L3 latency, and L2 and L4 latencyies will be considered later. In section3, we define MN handoff rate as $r_h = v / d$. It means how many APs or FAs the MN moved through per second. Equation 1 shows that the performance of MIP/ WLAN depends on the MN handoff rate. r_h is also equal to the ratio of $K_{handoff}/T_{travel}$. Where $K_{handoff}$ is the number of handoffs occurred during the MN traveling. T_{travel} is MN's total travel time. To reduce r_h without changing total travel time, we can reduce the number of handoffs. The optimal is $K_{handoff} = 0$.

Let N be total FA numbers on the way MN traveling. Let's assume somehow M is the number of FAs with whom the MN can communicate without L3 delay. The optimal is M=N. But it costs too many resources, especially when the number of active MNs is large. Also we don't know how long will the MN travel at the beginning.

We call M the size of the FA Set with whom the MN can communicate without L3 handoff delay. From IP level of view, M is the number of FAs that MN has registered to and can communicate with at that moment.

The first problem SA-MIP needs to deal with is to decide FA set size M. In SA-MIP algorithm, M is decided by the following equation.

$$M = \left| t_{handoff} \times r_h \right| + 1 \tag{7}$$

Where $t_{handoff}$ is the handoff time for every handoff procedure, and r_h is the handoff rate. Here, we use the experimental average handoff time 9.142s for $t_{handoff}$. r_h is dynamic. For example, at speed 40m/s, AP distance 500m, M = |9.142 x 40/500| + 1 = 2. At speed 80m/s, AP distance 500m, M = 3.

The second problem for SA-MIP is how to guarantee MN can communicate with a FA set just like it can do with one FA. Our solution is to let MN pre-register M potential FAs along the way MN traveling, at the same time multicast IP packets to those FAs in this FA set. So MN won't feel any handoff delay from the IP level of view. In SA-MIP, the set of FAs that MN can talk to without L3 latency is extended from one point at low moving speed to a line at high moving speed. The length of the line dynamically changes with the MN handoff rate. The behavior of SA-MIP will automatically adapt to the handoff rate of the MN so that the performance of SA-MIP won't decline dramatically in a rapid moving environment. At the same time, SA-MIP only cost reasonable resource that is as much as enough for seamless handoff.

In this paper, we assume the MN has GPS system to detect its location. When the MN moves at speed v, if v < 30m/s (108km/h), it performs a normal registration. If

30m/s < v < 40m/s (144km/h), it initializes registration after receiving two successive agent advertisements. If v > 40m/s, we assume the MN won't change its direction largely in a short distance. It initializes registration once it gets a new agent advertisement. MN's registration message is extended by speed extension. According to Mobile IP Vendor/ Organization-Specific- Extensions[9]. Two kinds of Extensions are allowed for MIP, Critical (CVSE) and Normal (NVSE) Vendor/Organization Specific Extensions. The basic difference is when the CVSE is encountered but not recognized, the message containing the extension must be silently discarded, whereas when a NVSE is encountered but not recognized, the extension should be ignored, but the rest of the Extensions and message data must still be processed. We use the NVSE extension to extend MIP with handoff rate information.

Whenever the MN needs to handoff to a new FA set, after it gets that many times of agent advertisements which is determined by speed(step 1 in Fig. 4), it sends a registration request with up-to-date handoff rate information to the very first FA in a new FA set(step 2). The first FA relays the registration request to upper FA or HA(step 3). Meanwhile, it decapsulates the speed extension, refill the MIP header and authentication extension and then forward it to other FAs(M-1 FAs) in this FA set(step 4). These other FAs relay the registration request to upper FA or HA as well, just like the request comes from the MN (step 5). When the GFA or HA receives these registration requests, it builds up tunnels downwards to each FA and responses with registration reply (step 6 and 7). When the FA receives the registration reply, it builds up tunnel upwards to the GFA or HA. Whenever the MN setups the Link-layer contact with the FA, the later forwards the registration reply to the former (step8, 9 or 10). The MN gets the care-of-address from agent advertisement message (step 10 or 9) or registration reply message (step 9 or 10), and begins data communication. At the same time, it sends registration requests to the new FA with up-to-date speed information (step 11). This new FA decapsulates the message, sets up a new FA set, forwards the request (12,13) and repeats the above process. In Fig.4, the FA set size M changes from 2 to 3 when the MN handoff rate changes from 0.08 to 0.12.

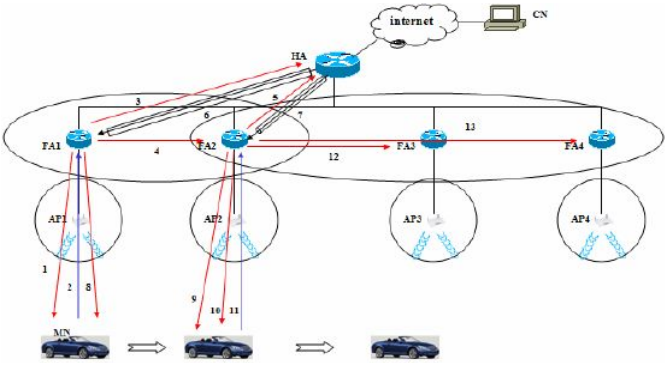


Fig. 4. Speed adaptive handoff procedure

Table 3. Average throughput for SA-MIP

Speed (m/s)	AP distance (m)	Bytes transferred (kB)	Travel Time(s)	Arg throughput (kB/s)
20	1000	40300	399	101.00
40	1000	18400	198	88.38
60	1000	10000	130	76.92
80	1000	6250	99	63.13
10	500	39500	398	99.24
20	500	17000	198	85.86
30	500	9900	131	75.57
40	500	6200	98	63.26

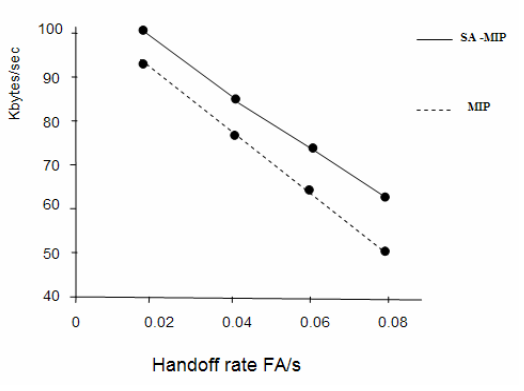


Fig. 5. Performance of SA-MIP

We evaluate the performance of speed-adaptive MIP/ WLAN under the same scenario as in Fig.1 except the SA-MIP is installed. The average throughput at different speed is listed in table 3.

Fig. 5 shows average throughput vs. handoff rate before and after the SA-MIP is installed. At handoff rate 0.02 FA/s, the average throughput is improved by (100.12 – 92.72)/ 92.72 = 7.98%. At handoff rate 0.04, 0.06 and 0.08 FA/s, the average throughput is improved by 12.99%, 16.81% and 26.45% respectively.

6 Conclusion

In this paper, the emulation experiments showed that MIP is not suitable for rapidly moving mobile clients. We depicted the relationship between the performance and the handoff rate of MN and quantitatively analyzed the handoff latencies of the MIP/ WLAN. A Speed Adaptive MIP is proposed and evaluated. The emulation showed that the SA-MIP can improve the performance from 8% to 27% when the handoff rate changes from 0.02 FA/s to 0.08 FA/s. In this paper, SA-MIP only deal with L3 handoff latency. But there is still physical link break from the Layer 2 handoff. And also we noticed that even in SA-MIP, the biggest part of handoff delay was still the layer4 TCP back-off-latency. In future work, we are going to apply the speed adaptive scheme to layer 2 and layer 4 handoff latencies.

References

1. C. Perkins, RFC3344, “IP Mobility Support for IPv4”
2. E. Hernandez and Sumi Helal, "RAMON: Rapid Mobility Network Emulator," Proceedings of the 27th Annual IEEE Conference on Local Computer Networks (LCN), November 2002, Tampa, Florida
3. http://dynamics.sourceforge.net/?page=main
4. Héctor Velayos and Gunnar Karlsson “Techniques to Reduce IEEE 802.11b Handoff Time” IEEE ICC 2004, Paris, France, June 2004.
5. IEEE 802.11F-2003, IEEE Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation
6. N. A. Fikouras, A. J. Könsgen, and C. Görg. “Accelerating Mobile IP Hand-offs through Link-layer Information”. In Proceedings of the International Multiconference on Measurement, Modelling, and Evaluation of Computer- Communication Systems (MMB), Aachen, Germany, September 2001.
7. Pobert Hsieh and Aruna Seneviratne. “A Comparison of Mechanisms for Improving Mobile IP Handoff Latency for End-to-End TCP”. MobiCom’03 San Diego, CA, USA, Sep. 2003.
8. RFC 3115, Mobile IP Vendor/Organization-Specific Extensions.

Least Cost Multicast Spanning Tree Algorithm for Local Computer Network

Yong-Jin Lee[1] and M. Atiquzzaman[2]

[1] Department of Computer Science, Woosong University,
17-2 Jayang-Dong, Dong-Ku, Taejon 300-718, Korea
yjlee@woosong.ac.kr
[2] School of Computer Science, University of Oklahoma,
200 Felgar Street, Norman, OK 73019, USA
atiq@ou.edu

Abstract. This study deals with the topology discovery for the capacitated minimum spanning tree network. The problem is composed of finding the best way to link nodes to a source node and, in graph-theoretical terms, it is to determine a minimal spanning tree with a capacity constraint. In this paper, a heuristic algorithm with two phases is presented. Computational complexity analysis and simulation confirm that our algorithm produces better results than the previous other algorithms in short running time. The algorithm can be applied to find the least cost multicast trees in the local computer network.

1 Introduction

Topology discovery problem [1,2] for local computer network is classified into capacitated minimum spanning tree (CMST) problem and minimal cost loop problem [3]. The CMST problem finds the best way to link end user nodes to a backbone node. It determines a set of minimal spanning trees with a capacity constraint. In the CMST problem, end user nodes are linked together by a tree that is connected to a port in the backbone node. Since the links connecting end user nodes have a finite capacity and can handle a restricted amount of traffic, the CMST problem limits the number of end user nodes that can be served by a single tree. The objective of the problem is to form a collection of trees that serve all user nodes with a minimal connection cost.

Two types of methods have been presented for the CMST problem - exact methods and heuristics. The exact methods are ineffective for instances with more than thirty nodes. Usually, for larger problems, optimal solutions can not be obtained in a reasonable amount of computing time. The reason is why CMST problem is NP-complete [4]. Therefore, heuristic methods [5,6,7] have been developed in order to obtain approximate solutions to the problem within an acceptable computing time. Especially, algorithm [5] is one of the most effective heuristics presented in the literature for performance evaluation.

In this paper, new heuristic algorithm that is composed of two phases is presented. This paper is organized as follows. The next section describes the modeling and algorithm for the CMST problem. Section 3 discusses the performance evaluation and section 4 concludes the paper.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 268 – 275, 2005.

2 Modeling and Algorithm

The CMST problem is represented in Fig. 1. Eq. (1) is the formulation for the CMST problem.

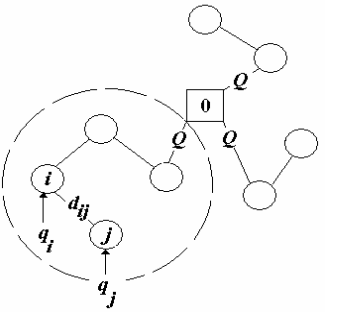


Fig. 1. CMST problem

The objective of the CMST problem is to find a collection of the least-cost spanning trees rooted at the source node. n represents the number of nodes. d_{ij} and q_i are distance between node pair (i, j) and traffic requirement at node i (i=1,..,n) respectively. Q shows the maximum traffic to be handled in a single tree and T_k is the k^{th} tree which has no any cycles.

$$\begin{aligned} &\textit{Minimize} \ \sum_{i,j} d_{ij}\, x_{ij} \\ &\textit{S.T.} \\ &\qquad \sum_{i,j \in T_k} q_i\, x_{ij} \le Q, \quad \forall k \\ &\qquad \sum_{i,j} x_{ij} = n \\ &\qquad x_{ij} = 0 \ or \ 1 \end{aligned} \tag{1}$$

A particular case occurs when each q_i is equal to one. At the time, the constraint means that no more than Q nodes can belong to any tree of the solution. In this paper, we present a heuristic that consists of two phases for the CMST problem. In the first phase, using the information of trees obtained by the EW solution(we will call algorithm [5] as EW solution), which is one of the most effective heuristics and used as a benchmark for performance evaluation, we improve the solution by exchanging nodes between trees based on the suggested heuristic rules to save the total linking cost. In the second phase, using the information obtained in the previous phase, we transfer nodes to other tree in order to improve solutions.

EW solution performs the following procedure: It first compute $g_{ij} = d_{ij} - C_{ij}$ for each node pair (i,j). d_{ij} and C_i represent cost of link (i,j) and the minimum cost between a source node and node set of tree containing node i respectively. At the initialization, it sets $C_i = d_{i0}$. Then, it finds the node pair (i,j) with the minimum negative g_{ij} (we do not consider node pair's with the positive g_{ij} value). If all g_{ij}'s are positive, algorithm

is terminated. Next, it check whether the connecting node i and j satisfies the traffic capacity constraint and forms a cycle together. If no, it sets $g_{ij} = \infty$ and repeats the above check procedure. Otherwise, it connects node i and j and delete the link connecting a source node and tree with the higher cost between C_i and C_j. Since new tree's formation affects C_i in EW solution, g_{ij} values have to be recomputed. When the number of nodes except the source node is n, the EW solution provides the near optimum solution with a memory complexity of $O(n^2)$ and a time complexity of $O(n^2 log\ n)$ for the CMST problem.

We will improve the EW solution by the simple heuristic rules based on the node exchange and transfer between two different trees. Starting from the trees obtained by EW solution, we first exchange nodes between different trees based on the trade-off heuristic rules (ks_{ij}). It is assumed that node i is included in $node(inx1)$, node j is included in $node(inx2)$, and $inx1$ is not equal to $inx2$. $inx1$ and $inx2$ represent indices of trees including node i and node j respectively. In addition, $node(inx1)$ and $node(inx2)$ represent sets of nodes included in tree $inx1$ and $inx2$ respectively. Exchange heuristic rule, ks_{ij} is defined as $C_{inx1} + C_{inx2} - d_{ij}$. C_{inx1} is the least cost from nodes included in tree $inx1$ to the root (source node). That is, C_{inx1} = Min $\{d_{m0}\}$ for $m \in node(inx1)$, $j \in node(inx1)$. Also, C_{inx2} = Min $\{d_{m0}\}$ for $m \in node(inx2)$, $j \in node(inx2)$. If $inx1$ is equal to $inx2$, both node i and node j are included in the same tree, trade-off value is set to $-\infty$. Since the sum of node traffic must be less than Q, Both $\sum_{m \in node(inx1)} q_m + q_j - q_i \le Q$ and $\sum_{m \in node(inx2)} q_m + q_i - q_j \le Q$ must be satisfied. Otherwise, ks_{ij} is set to $-\infty$.

An initial topology is obtained by applying EW solution. For each node pair (i, j) in different trees, heuristic rules (ks_{ij}'s) are calculated and ks_{ij}'s with negative value are discarded. From node pair (i, j) with the maximum positive value of ks_{ij}, by exchanging node i for node j, two new node sets are obtained. The network cost by applying the existing unconstrained minimum spanning tree algorithm [8] to two new sets of nodes is obtained. If the computed cost is less than the pervious cost, the algorithm is repeated after re-computing heuristic rules (ks_{ij}'s). Otherwise the previous ks_{ij}'s are used. If all ks_{ij}'s are negative and it is impossible to extend trees further, we terminate the algorithm.

Node transfer procedure is described as the follows: we improve solutions by transferring nodes from one tree to another tree based on node transfer heuristic rule (ps_{ij}). We first evaluate that the sum of traffics in every tree is equal to Q. If so, the algorithm is terminated. Otherwise, the node pair (i, j) with the minimum negative value of ps_{ij} is found. By transferring node j to the tree including node i, the solution is computed. If $inx1$ is equal to $inx2$ or the sum of traffic is greater than Q, node j can not be transferred to the tree $inx1$. In this case, ps_{ij} is set to ∞. Otherwise, transfer heuristic rule, ps_{ij} is defined as $d_{ij} - d_{max}$. Here, d_{max} = Max $\{C_{inx1}, C_{inx2}\}$.

If each trade-off heuristic rule (ps_{ij}) is positive for all node pair (i, j), and no change in each node set is occurred, we terminate the algorithm. From the above modeling for the CMST problem, we now present the following procedure of the proposed algorithm. In the algorithm, step 2 and step 3 perform node exchange and transfer respectively.

Algorithm: Least-Cost Multicast Spanning Tree
Variable: {$TEMP_{cost}$: network cost computed in each step of the algorithm
EW_{cost}: network cost computed by EW solution
NEW_{cost}: current least network cost
lcnt: the number of trees included in any topology }
Step 1: Execute the EW solution and find the initial topology.
Step 2: *A. Perform the node exchange between two different trees in the initial topology.*
(1) set $TEMP_{cost} = EW_{cost}$. (or set $TEMP_{cost} = NEW_{cost}$ obtained in Step 3)
(2) For each node pair (i, j) in different trees ($i < j$, $\forall$ (i, j)), compute ks_{ij}.
if ($ks_{ij} < 0$), $\forall$ (i, j), goto B.
(3) while ($ks_{ij} > 0$) {
1) For node pair (i, j) with the maximum positive ks_{ij},
exchange node i for node j and create *node*(*inx1*) and *node*(*inx2*).
2) For *node*(*inx1*) and *node*(*inx2*), by applying unconstrained MST algorithm,
compute $TEMP_{cost}$.
3) if ($TEMP_{cost} \geq NEW_{cost}$), exchange node j for node i. set $ks_{ij} = -\infty$ and repeat (3).
else set $NEW_{cost} = TEMP_{cost}$. set $ks_{ij} = -\infty$ and go to (2).
} ;
B. If it is impossible to extend for all trees, algorithm is terminated.
Otherwise, proceed to step 3
Step 3: *A. Perform the node transfer between two different trees obtained in Step 2.*
(1) For all p, (p=1,2,..,*lcnt*), if ($\sum_{i \in p} W_i == Q$), algorithm is terminated.
else set $NEW_{cost} = TEMP_{cost.}$
(2) For each node pair (i, j) in different trees ($i < j$, $\forall$ (i, j)), compute ps_{ij}.
if ($ps_{ij} \geq 0$), $\forall$ (i, j), goto B.
(3) while ($ps_{ij} < 0$) {
1) For node pair (i, j) with the minimum negative ps_{ij},
transfer node j to *node*(*inx1*) and create new *node*(*inx1*) and *node*(*inx2*).
2) For *node*(*inx1*) and *node*(*inx2*), by applying unconstrained MST algorithm,
compute $TEMP_{cost}$.
3) if ($TEMP_{cost} \geq NEW_{cost}$), transfer node j to *node*(*inx2*). set $ps_{ij} = \infty$ and repeat (3).
else set $NEW_{cost} = TEMP_{cost}$. set $ps_{ij} = \infty$ and go to (2).
};
B. If any change in the node set is occurred, goto Step 2. Otherwise, algorithm is terminated.

3 Performance Evaluation

3.1 Property of the Proposed Algorithm

We present the following lemmas in order to show the performance measure of the proposed algorithm.

Lemma 1. Memory complexity of the proposed algorithm is $O(n^2)$.

Proof. d_{ij}, ks_{ij}, and ps_{ij} (i=1,..,n; j=1,..,n) used in step 2 ~ step 3 of the proposed algorithm are two-dimensional array memory. Thus, memory complexity of step 2 and 3 is $O(n^2)$, respectively. Memory complexity of EW solution executed in step 1 of the proposed algorithm is $O(n^2)$. As a result, total memory complexity is $O(n^2)$.

Lemma 2. Time complexity of the proposed algorithm is $O(n^2 log\ n)$ for sparse graph and $O(Qn^2 log\ n)$ for complete graph when the maximum number of nodes to be included in a tree is limited to Q.

Proof. Assuming that q_i=1, $\forall i$, Q represents the maximum number of nodes to be included in a tree. For any graph, $G = (n, a)$, the range of Q is between 2 and n-1. In the Step 2 of the proposed algorithm, trade-offs heuristic rules (ks_{ij}) are computed for each node pair (i, j) in different trees. At the worst case, the maximum number of ks_{ij}'s to be computed is 1/2(n-Q)(n+Q-1) for Q=2,..,n-1. In the same manner, the maximum number of ks_{ij}'s to be computed in the Step 3 is 1/2(n-Q)(n+Q-1) for Q=2,..,n-1. Time complexity of minimum spanning tree algorithm is shown to be $O(E \log Q)$ [8]. E is the number of edges corresponding to Q. Since the proposed algorithm uses minimum spanning tree algorithm for two node sets obtained by exchanging node i for node j in the Step 2 or transferring node j to the tree including node i in Step 3, time complexity of the computation for minimum spanning tree is $2O(E \log Q)$. In the worst case, let us assume that MST algorithms are used maximum number of ks_{ij} (or ps_{ij}) times and EW solution, Step 2 and Step 3 are executed altogether. Time complexity of EW solution is known to be $O(n^2 log\ n)$. Now, let the execution time of EW solution be T_{EW}, that of Step 2 be T_{NEA}, and that of Step 3 be T_{NCA}. Then, for sparse graph ($E = Q$), $T_{NEA} = \text{MAX}_{Q=2}^{n-1} T_Q = \text{MAX}_{Q=2}^{n-1}[1/2(n-Q)(n+Q-1)\ O(E \log Q)] = O(n^2 log\ Q)$. In the same manner, $T_{NCA} = O(n^2 log\ Q)$. Therefore, total execution time = $T_{EW} + T_{NEA} + T_{NCA} = O[\text{MAX}\ (n^2 log\ n, n^2 log\ Q)] = O(n^2 log\ n)$. For complete graph ($E = 1/2Q(Q+1)$), $T_{NEA} = \text{MAX}_{Q=2}^{n-1} T_Q = \text{MAX}_{Q=2}^{n-1}[1/2(n-Q)(n+Q-1)O(E \log Q)] = O(Qn^2 log\ n)$. In the same manner, $T_{NCA} = O(Qn^2 log\ n)$. Hence, total execution time = $T_{EW} + T_{NEA} + T_{NCA} = O[\text{MAX}\ (n^2 log\ n, Qn^2 log\ n)] = O(Qn^2 log\ n)$.

Lemma 3. All elements of trade-off matrix in the algorithm are become negative in finite steps.

Proof. Assume that ps_{ij}'s are positive for some i,j. For node pair(i,j) with the positive ks_{ij}, our algorithm set ks_{ij} to -∞ after exchanging node i for node j. At the worst case, if all node pair(i,j) are exchanged each other, all ks_{ij} are set to -∞. Since trade-off matrix has finite elements, all elements of trade-off matrix are become negative in finite steps.

Lemma 4. The proposed algorithm can improve EW solution.

Proof. Let the solution by the proposed algorithm be NEW_{cost}, the EW solution be EW_{cost}. Also, assume that the number of trees by EW solution is $lcnt$, the set of nodes corresponding to trees j (j=1,2..,$lcnt$) is R_j and the corresponding cost is $C(R_j)$. Then EW_{cost} is $\sum_{j=1}^{lcnt} C(R_j)$. In this case, $\cap_{j=1}^{lcnt} R_j = null$ and $C(R_j)$ is the MST cost corresponding to R_j. in the step 2, NEW_{cost} is replaced by Ew_{cost}. And only in the case that the cost ($TEMP_{cost}$) obtained in step 2 is less than EW_{cost}, $TEMP_{cost}$ is replaced by NEW_{cost}, so, $TEMP_{cost} = NEW_{cost} < EW_{cost}$. Now, one of cases which $TEMP_{cost}$ is less than EW_{cost} is considered. Let two sets of nodes changed after changing nodes in step 2 be R_{sub1}, R_{sub2} and the corresponding sets of nodes obtained by EW solution R'_{sub1}, R'_{sub2}. If cardinalities of R'_{sub1}, R'_{sub2} are $|R'_{sub1}| = |R'_{sub2}| = Q$, at the same time, $|R_{sub1}| = |R_{sub2}| = Q$ where Q is the maximum number of nodes. Assume that link costs, $d_{i1,i2} < d_{i2,i3} <,.. < d_{ik-2,ik-1} < d_{ik-1,ik} = d_{ik-1,jk-1} < d_{ik,jk-2} < d_{ik,jk-1} < d_{ik,jk} < d_{j1,j2} <.. < d_{jk-1,jk} <$ other link cost(d_{ij}), and in EW solution, $g_{i1,i2} < g_{i2,i3} <,..< g_{ik-1,ik} = g_{ik-1,jk-1} < g_{ik,jk-2} < g_{ik,jk-1} < g_{ik,jk} < g_{j1,j2} <..< g_{jk-1,jk} < 0$ (other g_{ij}'s > 0) are obtained. Then, $R'_{sub1} = \{i1,i2,.,ik\}$, the corresponding tree is (0-$i1$-$i2$-..-ik) and $R'_{sub2} = \{j1,j2,.,jk\}$, the corresponding tree is

(0-*j1-j2-..-jk*). Therefore, $C(R'_{sub1}) = d_{0,i1} + d_{i1,i2} +,.. + d_{ik-2,ik-1} + d_{ik-1,ik}$ and $C(R'_{sub2}) = d_{0,j1} + d_{j1,j2} + ,..+ d_{jk-2,jk-1} + d_{jk-1,jk}$. But, edges (*ik*-1, *jk*-1), (*ik*, *jk*-2), (*ik*, *jk*) with the less *g* value are excluded in the solution because $|R_{sub1}|$ is equal to *Q*. By assumption, $d_{ik,jk-2} + d_{ik,jk} < d_{jk-2,jk-1} + d_{jk-1,jk}$ and $d_{ik-1,ik} = d_{ik-1,jk-1}$. If exchanging node(*ik*) for node(*jk*-1), $R_{sub1} = \{i1,i2,..,ik-1,jk-1\}$ and $R_{sub2} = \{j1,j2,..,jk-2,ik,jk\}$. Applying MST algorithm to the above two sets, since $C(R_{sub1})$, $C(R_{sub2})$ are the minimum cost trees, $C(R_{sub1})$ becomes $d_{0,i1} + d_{i1,i2} +,.. + d_{ik-2,ik-1} + d_{ik-1,jk-1}$ and $C(R_{sub2})$ becomes $d_{0,j1} + d_{j1,j2} +,..,+ d_{ik,jk-2} + d_{ik,jk}$. Thus, since $C(R_{sub1}) = C(R'_{sub1})$ and $C(R_{sub2}) < C(R'_{sub2})$, $C(R_{sub1}) + C(R_{sub2}) < C(R'_{sub1}) + C(R'_{sub2})$. Total cost, $TEMP_{cost} = \sum_{j=1,j\neq sub1,sub2}^{lcnt} C(R_j) + C(R_{sub1}) + C(R_{sub2}) < \sum_{j=1,j\neq sub1,sub2} C(R_j) + C(R'_{sub1}) + C(R'_{sub2}) = EW_{cost}$. Hence, there exists the case which $TEMP_{cost}$ is less than EW_{cost}.

Table 1 represents the comparison of several algorithms. In the time complexity of algorithm[7], the practical range of *S* is from *n*/*Q* to *nlog*(*n*/*Q*). Algorithm[6] represents the results when every traffic requirement is one. If the traffic requirements are different or *Q* is not the power of 2, the results are inferior to that of EW solution. For the complete graph, since the time complexity of algorithm[6] is $O(Qn^3)$, the computing time is increased more sharply than the proposed algorithm with the time complexity of $O(Qn^2 log\ n)$ as *Q* is increased.

Table 1. Comparison of memory and time complexity

algorithm	memory complexity	time complexity
EW solution	$O(n^2)$	$O(n^2 log\ n)$
Algorithm[7]	$O(n^2)$	$O(S^3 n log\ n)$
Algorithm[6]	$O(n^2)$	$O(n^3)$
Proposed algorithm	$O(n^2)$	$O(n^2 log\ n)$

3.2 Simulation

In order to evaluate the proposed algorithm, we carried out the computational experiments on IBM-PC. The coordinates of nodes were randomly generated in a square grid of dimensions 100 by 100. The savings rate is defined as Eq. (2).

$$\text{Savings Rate} = (A - B) / A \times 100 \tag{2}$$

In Eq. (2), *A* is the cost by EW solution and *B* is the cost by the proposed algorithm suggested in this paper.

We assumed that the traffic requirements have Poisson distribution. Let exponential random number be *X*, the average traffic rate of network be λ, the following expression is obtained.

$$X = \frac{1}{\lambda} ln \left(\frac{1}{1 - U}\right) \tag{3}$$

In Eq. (3), *U* is uniformly distributed random variable between 0 and 1. Using Eq. (3), *n* Poisson random number are generated and used as the traffic requirements. 20, 30, 40, and 50 as the value of λ and 30, 50, 70 as the number of nodes (*n*) are used respectively. Maximum traffic handled in a single tree is used between the maximum value

among the Poisson random number by the simulation and $n/4$. Fig. 2 shows the mean savings rate. Increasing of λ and n do not affect the results of savings rate. Thus, to derive the expression describing the relation between λ or n and solution exactly is difficult.

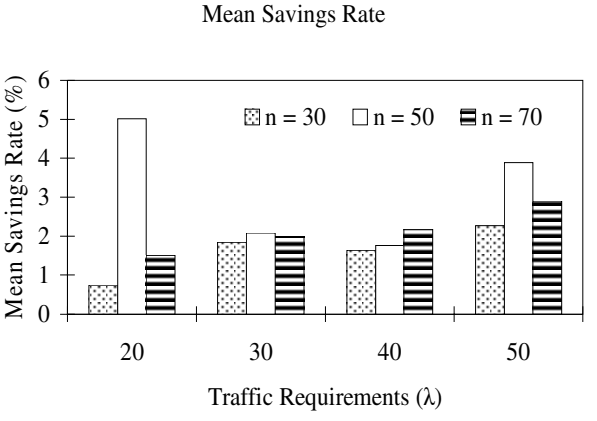


Fig. 2. Mean Savings Rate

As shown in Fig. 2, our proposed algorithm improves the EW solution up to 5 %. Mean savings rate of algorithm[7] is 1.9 % and that of algorithm[6] ranges from 1 % to 5 %. Thus, it is known that no algorithm produces the best result. The reason is because the solution is affected by the generated location of nodes. However, since the time complexity of the proposed algorithm is the least as shown in Table 1, we can state that the proposed algorithm produces reasonable improvements over the EW solution in the short running time in comparison with other heuristics.

4 Conclusions

In this paper, we present new heuristic algorithm and its computational property for the capacitated minimum spanning tree (CMST) problem. The proposed algorithm can be applied to find the least cost multicast trees and topology discovery in the local computer network. It improves solutions by exchanging or transferring nodes between trees based on the suggested heuristic rules. It has the small memory and time complexities and produces good improvements over the benchmark solution in comparison with other existing heuristics. Simulation results show that the proposed algorithm does not limit the type of traffic requirements, has not the fluctuation of solution, and is more efficient when the traffic volume of network is light. Future work includes the more efficient algorithm considering a mean delay constraint.

References

1. Bjerano, Y., Breitbart, M. and Rastogi, R.: Physical topology discovery for large multi-subnet networks. INFOCOM. (2003) 342-352.
2. Huffaker, B., Plummer, D., and Claffy, K.: Topology discovery by active probing. Applications and the Internet (SAINT) Workshops. (2002) 90-96.
3. Lee, Y.: Minimal cost heuristic algorithm for delay constrained loop network. International Journal of Computer Systems Science & Engineering, Vol. 19. CRL Publishing. (2004) 209-219.
4. Papadimitriou, C.H.: The complexity of the capacitated tree problem. Networks, Vol. 8. (1978) 217-230.
5. Esau, L.R. and Williams, K.: On teleprocessing system design, part II. IBM syst. J., Vol. 5. (1966) 142-147.
6. Gavish, B. and Altinkemer, K.: Parallel savings heuristics for the topological design of local access tree networks. Proceedings IEEE-INFOCOM '86. (1986) 139-139.
7. Kershenbaum, K., Boorstyn, R. and Oppenheim, R.: Second-order greedy algorithms for centralized teleprocessing network design. IEEE Trans. on Comm., Vol. 28. (1980) 1835-1838.
8. Sedgewick, R.: Algorithms. Addison-Wesley. (1989) 452-461.

A New Multicast Group Management Scheme for IP Mobility Support*

Miae Woo and Ho-Hyun Park

Sejong University, Seoul, Korea
mawoo@sejong.ac.kr

Abstract. As the number of portable devices roaming across the Internet increases, the problem of routing packets to mobile hosts generates increasing research and commercial interest. To support mobility more effectively for seamless provision of various applications, many multicast-based localized mobility support schemes have been proposed to achieve better performance than the basic solutions provided by the IETF, such as Mobile IP and Mobile IPv6. However, any multicast-based scheme inherently introduces significant bandwidth wastage at the wireless access network due to long leave latency. In this paper, we propose a new multicast group managment scheme that is tailored for managing multicast groups used to support host mobility. It has been shown by simulation that the proposed scheme achieves extremely short leave latency and eliminates bandwidth wastage in the wireless links.

1 Introduction

As Internet places itself as an indispensable factor in today's life, it is expected that future wireless networks will include large number of IP-enabled mobile devices roaming round wireless cells while the devices maintain connections with others using TCP/IP protocol suite. Consequently, providing Internet data services is recognized as an important service in the next generation wireless communication networks. To realize such mobile Internet services that are comparable to the current wired network environment, providing efficient mobility management schemes is essential.

The main problem in supporting mobility in the Internet is created by the location dependency of addresses and by the violation of the layered concept in TCP/IP protocol suite. Location dependent IP unicast address poses problems when an IP enabled host changes its point of attachment from one network to another network. Also, since IP addresses are often used to identify connections in the transport and application layers, the higher layer identifiers are required to be modified and the corresponding connections re-established whenever a node moves [1]. Solutions for this problem made by IETF Mobile IP working group

* This work was supported in part HY-SDR Research Center at Hanyang University, Seoul, Korea, under the ITRC Program of IITA, Korea and in part by grant No. R04-2001-000-00177-0 from the Korea Science & Engineering Foundation.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 276–285, 2005.

are Mobile IP [2] and Mobile IPv6 [3] which use two-tier addressing [4]; one for routing directive and the other for end-point identifier. In these solutions, a mobile node (MN) has a home address. It also has a care-of address (CoA) while it is away from home. Routing of packets heading for the MN is enabled by maintaining mapping between the home address and the CoA at the home agent (HA) and tunneling packets to the proper CoA by the home agent. The service of Mobile IP and Mobile IPv6 is restricted in delivering packets with unicast routable addresses. Therefore, the home address and the CoA used by the mobil node should be globally routable unicast addresses.

Though Mobile IP and Mobile IPv6 provide basic mobility support for the wide area movement, they cannot support fast user mobility efficiently and generate quite number of signaling messages in the Internet backbone. Since IP multicast provides a mechanism for location independent addressing and packet delivery to multicast group members, it has been considered as a mechanism for providing IP mobility. Many multicast-based proposals [1,5,6,7,8,9] have been made for supporting host mobility. One of main reasons to use multicast in Internet host mobility is its shorter delay in location registration than Mobile IP or Mobile IPv6. However, utilizing multicast for the host mobility support accompanies signaling traffic to maintain multicast tree and multicast group menbership status as well as bandwidth wastage due to leave latency.

A multicast group address to be used for mobility support can be dynamically allocated or automatically configured. For the dynamic allocation, a mobile node should consult a server to lease a multicast group address. Such procedure introduces delays in mobile node's location registration. As a result, automatic configuration by mobile nodes [8,9] is a better solution for the host mobility support. If multicast is used for mobility support to compensate the drawbacks of Mobile IP or Mobile IPv6, the multicast group address that is used by a mobile node should be unique in the multicast domain. In other words, each multicast group formed for mobility support has only one member in the multicast domain. The reason for that is to preserve the objective of Mobile IP and Mobile IPv6 to serve packets with unicast routable addresses. In IPv4, a mobile node cannot configure a unique multicast address automatically because of the limitation in the IP address space. On the other hand, there is no ambiguity in automatic configuration of multicast group addresses in IPv6. So, this paper addresses issues and solutions in terms of IPv6. However, the basic idea can easily adapted in IPv4 if a uniqueness of multicast group addresses is guaranteed.

One of the main concerns in using multicast for mobility management scheme is leave latency. Leave latency is the time between the moment the last node on a link ceases listening to a particular multicast address and the moment the routing protocol is notified that there is no longer any group memeber for that address. Since any host mobility support scheme using multicast results multicast groups with a single member, mobile node's movement from one subnet to anouther triggers a leave process. In this case, long leave latency can result significant bandwidth wastage in the link between the multicast router and the mobile node, since leave latency can be from several seconds to several minutes. So, we

propose a new group management scheme that can efficiently handle mutlicast groups with only one member. The main purpose of the proposed scheme is to reduce leave latency in order to efficiently utilize valuable bandwidth in the wireless links.

The remainder of the paper is organized as follows. We first overview multicast group management protocol in Section 2. Section 3 describes the motivation for our study. The proposed scheme is given in Section 4. In Section 5, the performance of the proposed scheme is evaluated. Finally, Section 6 concludes this paper.

2 Overview of Multicast Group Management Protocol

Multicast group memberships at the leaf router in the multicast tree are maintained using the information obtained via the Internet Group Management Protocol (IGMP) [10] for IPv4 or Multicast Listener Discovery (MLD) [11] for IPv6. Since this paper addresses issues in terms of IPv6, we present an overview of MLD.

IPv6 router uses MLD to discover the presence of multicast listeners (members) on its directly attached links, and to discover specifically which multicast addresses are of interest to those neighboring nodes. There are three types of MLD messages; Multicast Listener Query, Multicast Listener Report, and Multicast Listener Done. Multicast Listener Query is used by the router to learn whether there is any multicast listener on the attached link. General Query and Multicast-Address-Specific Query are subtypes for Multicast Listener Query. Multicast Listener Report messages are used primarily by multicast hosts to signal their local multicast router when they wish to join a specific multicast group and begin receiving group traffic. Hosts may also signal to the local multicast router that they wish to leave an IP multicast group and, therefore, are no longer interested in receiving the multicast group traffic using Multicast Listener Done messages. A multicast group membership is active on an interface if at least one host on that interface has signaled its desire, via MLD, to receive multicast group traffic.

Using the information obtained via MLD, routers maintain a list of multicast group membership on a per interface basis. The list contains entries for identifying which multicast addresses have listeners on that link, and a timer associated with each of that listeners for a given multicast address are present on a link. The router does not need to learn the identity of those listeners and the number of listeners present.

When a host wants to join a multicast group, the host will send one or more unsolicited Multicast Listener Report for the multicast group immediately it desired to join. For a leave process, a host may just go away to leave a multicast group, or may explicitly send a Multicast Listener Done to its local multicast router. Without a Multicast Listener Done message, the router continues to forward multicast traffic onto the local subnet for several minutes after the last host leaves the group. If a Multicast Listener Done message is received from

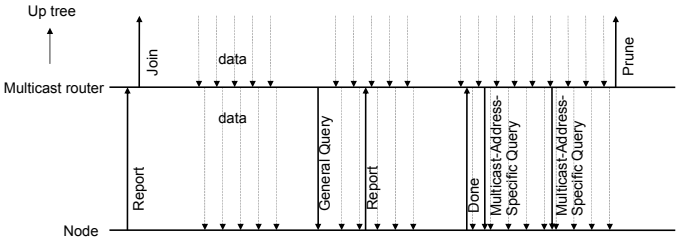


Fig. 1. An example of message flows between a host and a multicast router in MLD

a link, the router sends Multicast-Address-Specific Queries to that multicast address for "Last Listener Query Count" times. If no Reports of the last query has passed, the routers on the link assume that the address no longer has any listeners there. The address is then deleted from the list and its disappearance is made known to the multicast routing component. Fig. 1 shows the typical message flows occurred between a host and a multicast router in MLD when Multicast Listener Done message is used for a leave process.

3 Motivation

If multicast is used for mobility management, the wasted bandwidth is strongly correlated with the leave latency. In the multicast-based schemes, multicast packets are delivered to the subnet to which a mobile node was previously attached during the leave latency. Such multicast packets incur bandwidth wastage in the wireless access network.

The amount of leave latency varies whether the access network provides soft handoff or hard handoff mechanism. If soft handoff mechanism is not provided at the access network, a mobile node just goes away when it is moves into an adjacent subnet. It is not able to send any Multicast Listener Done message to the previously attached router. In this case, the router in the previously attached subnet continues to forward multicast traffic for several minutes after the mobile node leaves the subnet. If soft handoff mechanism is available in the access network, a mobile node can explicitly send a Multicast Listener Done message to the router in the subnet from where it is moving out.

Based on MLD, leave latency for the hard handoff (HH), denoted by l_{HH}, can be described as follows:

$$l_{HH} = r \cdot I_Q + I_{QR} \tag{1}$$

Leave latency for the soft hanoff (SH), denoted by l_{SH}, is

$$l_{SH} = c_{LQ} \cdot I_{LQ} \tag{2}$$

Table 1 provides the meaning of the variable used in Eq. 1 and Eq. 2 and their default values defined in MLD. Subsequently, the leave latency for the hard handoff is 260 seconds, and that for the soft handoff is 2 seconds.

Table 1. Variables and their default values used for multicast group management

Variables	Parameters	Values
r	Robustness Variable	2 ea
I_Q	Query Interval	125 sec
I_{QR}	Query Response Interval	10 sec
I_{LQ}	Last Listener Query Interval	1 sec
c_{LQ}	Last Listener Query Count	Robustness Variable

For Mobile IPv6, the bandwidth wastage occurs during registration delay. Registration delay constitutes time span from when a mobil node discovers that it is moved to a new subnet to when its binding update is delivered to and accepted by its home agent. Considering that typical transpacific propagation delay is about 0.2 seconds [13] and router advertisement interval is between 0.03 seconds and 0.07 seconds [3], registration delay in Mobile IPv6 is typically less than 1 second. Subsequently, multicast-based schemes waste valuable wireless bandwidth more than Mobile IPv6.

One way to reduce the leave latency in the multicast based scheme is to reduce the Query Interval or Last Listener Query Interval. However, one of drawbacks of reducing these intervals is increased signaling traffic for multicast membership management inside the access network, resulting wireless bandwidth wastage. Especially when Query Interval is reduced, Query messages are generated more frequently. In response to the Query messages, at least one membership report per group is delivered through the access network. Consequently, the bandwidth required to deliver signaling messages is increased proportional to the number of groups in the subnet. On the other hand, the effect of reducing the Last Listener Query Interval on the bandwidth usage is somewhat less than that of Query Interval, since Last Listener Query only requires the listeners who are members of the specified group to respond with Report message.

Based on the analysis given in this section, it can be concluded that an access network without capability of soft handoff is not appropriate to use a multicast-based mobility support scheme for mobility management. Also, a mechanism for efficient utilization of precious wireless bandwidth is needed even in the access network which is provisioned with soft handoff.

4 The Proposed Group Management Scheme

In this section, we propose a new group management scheme to efficiently handle multicast groups with only a single listener to shorten the leave latency for the multicast based localized mobility support schemes. Our proposal is an extension of MLD to handle multicast groups with only one listener.

As stated in the previous section, we assume that soft handoff is provisioned at the access network to realize an efficient leave group process. Accordingly, a mobile node can send a Multicast Listener Done message to the router in the subnet to which it was attached previously.

```
Procedure Handling_Report_Message {
  if (single_listener flag in the Report is set)
    if (there is a reported multicast address in the router's list that is same as the multicast address in the Report)
      if (listener's address in the list ≠ source address of the Report)
        Reset single_listener's flag in the router's list;
    else
      Add <multicast address in the Report, timer, single_listener flag = 1, source address of the Report> to the router's list;
  else
    if (there is a reported multicast address in the router's list that is same as the multicast address in the Report)
      Reset single_listener's flag in the router's list;
    else
      Add <multicast address in the Report, timer, single_listener flag = 1, source address of the Report> to the router's list;
  Go to Listener Present State;
}
```

Fig. 2. Procedure at the multicast router to handle the received Report message

```
Procedure Handling_Done_Message {
  if (there is a reported multicast address in the router's list that is same as the multicast address in the Report)
      if (single_listener flag in the list is set)
        Report to the router's multicast routing component;
    else
      Send multicast-specific query;
      Go to Checking Listener's State;
}
```

Fig. 3. Procedure at the multicast router to handle the received Done message

In the proposed scheme, a mobile node sends a Multicast Listener Report with a *single_listener flag* set to indicate it belongs to a multicast group with just one listener. This mechanism is only used to receive unicast packets destined to the mobile node. For example, the Code field in the Multicast Listener Report message can be used for the single_listener flag. Since the value of Code field is ignored by the receivers in the standard MLD, any value can be used for Code field and our scheme is operable in any network with multicast capability.

To manage groups in the interfaces, the router maintains a list of multicast address for each interface to define multicast address having listeners on that link. In MLD, the list of multicast address is defined to have tuples of <reported multicast address, timer>. In addition to the defined tuples in MLD, we extend tuples of the list of multicast address to <reported multicast address, timer, single_listener flag, listener's address> to incorporate the management of single listener group. Listener's address is only effective for tuples with single_listener flag set. Listener's address records the unicast address of the listener. For the tuples without single_listener flag set, listener's address field should be empty.

The procedure for handling the received Report message at the access router is depicted in Fig. 2. First, the router checks whether the single_listener flag is set or not. If the flag is set, it checks whether there is an entry that has the same multicast address given in the Report message. If the router does not have any corresponding entry, it inserts the received information to its multicast list, set the single_listener flag, and record the source address of the Report message to the listener's address field. If the reported multicast address is already in the multicast list, the router compares the source address of the Report message

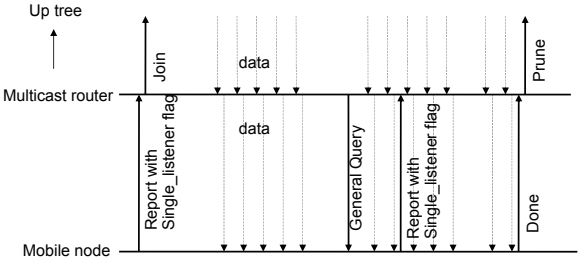


Fig. 4. An example of the message flows between a mobile node and a multicast router using the proposed multicast group managment scheme

with the listener's address in the multicast list. If they are different, then the router reset the sigle_listener flag in the multicast list since there are multiple listeners for the reported multicast address. If the received Report message is for an ordinary multicast member, then the router handles the message according to the procedure defined in MLD.

When the access router receives a Done message from a link, if the Multicast Address identified in the message is present in the list of addresses having listener on that link, the router checks whether the Done message is for single listener group. If that is the case, the router deletes the entry specified by the address from the list. It also notifies the multicast routing component about the disappearance of the address immediately. Otherwise, the router follows the procedure specified in MLD. Fig. 3 illustrates explained procedure for leave processing at the router.

Fig. 4 illustrates the messages exchanged between the mobile node and the multicast access router while the mobile node is in the coverage area of the multicast router. Comparing with Fig.1, Fig. 4 shows how the proposed multicast group management scheme can shorten leave latency and thus can prevent unnecessary data delivery on the access link.

5 Performance Evaluation

To evaluate the performance of the proposed scheme via experiments, the proposed scheme uses the multicast-based mobility support scheme proposed in [9] and the group management scheme proposed in Section 4. We evaluate the performance of the proposed scheme in terms of packet delivery efficiency and bandwidth wastage. The results are compared with those of Mobile IPv6 (MIPv6) [3], Hierarchical Mobile IPv6 (HMIPv6) [12], and a multicast-based scheme with ordinary MLD [9].

For the simulation, network environment shown in Fig. 5 is used. The gateway in the visited domain is used for a rendezvous point (RP) for the proposed scheme and a mobility anchor point (MAP) for the HMIPv6. Links in the Internet backbone network to which the correspondent node (CN), home agent (HA),

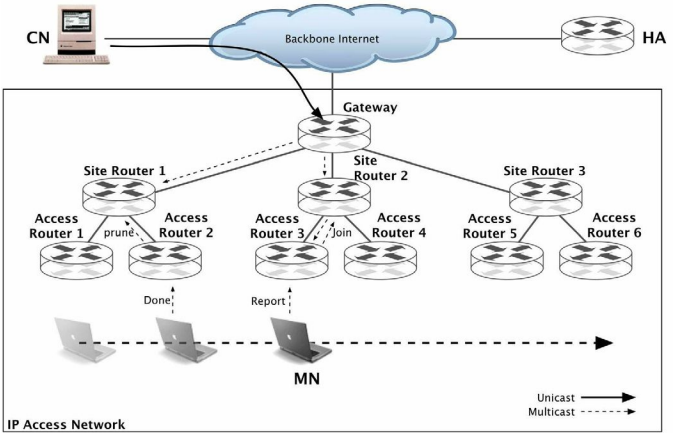


Fig. 5. Network architecture

and gateway of the foreign domain are connected are assumed to have same bandwidth of 2 Mbps. Inside the foreign domain, all the links are assumed to have same bandwidths of 10 Mbps. Since 10 msec is a typical MAN delay [13], we assign 5 msec between an access router (AR) and a site router and between a site router and a gateway respectively. For the WAN delay, 50 msec of propagation delay is assumed. All the routers in the IP access network, such as the gateway, site routers and access routers, are assumed be multicast routers.

In the simulation, the distance between ARs in the foreign domain is assumed to be 400 meters and smooth handoff mechanism [14] is adopted. The velocity of a MN is set to 40 meters/sec according to the simulation time. Also, router advertisement interval is set to 0.03 sec.

Packet delivery efficiency and bandwidth wastage in the wireless access links is evaluated through UDP performance. It is assumed that the CN transmits packets to the MN in a constant bit rate with fixed packet size of 256 bytes. In the simulation, we apply various transmission rates to see their effect on the packet loss and bandwidth wastage.

Fig. 6 shows the results of packet loss during handoffs. Since packet delivery efficiency is same for a multicast-based scheme with ordinary MLD and the proposed scheme, we only draw one legend for both schemes. As depicted in the figure, the average number of packet loss per handoff increases as the transmission rate increases. On average, the proposed scheme gives the least packet loss which is about 91% less than that of Mobile IPv6, and about 74% less that that of HMIPv6.

Next, we measure the number of packets delivered to the wireless link of the previous subnets after movement of a mobile node to analyze the wasted bandwidth due to handoffs. The result given in Fig. 7 shows that the amounts of wasted bandwidth for various transmission rate of UDP traffic for Mobile IPv6, HMIPv6, the multicast-based scheme with MLD, and the proposed scheme. For

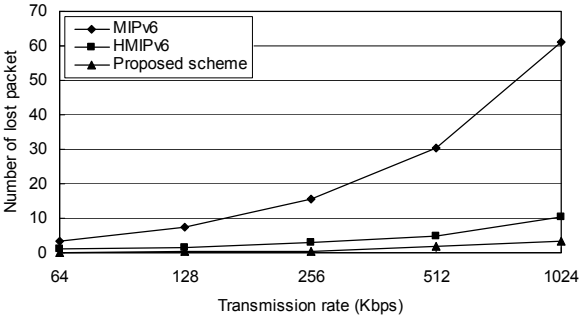


Fig. 6. Average number of packet loss during handoffs for various transmission rates

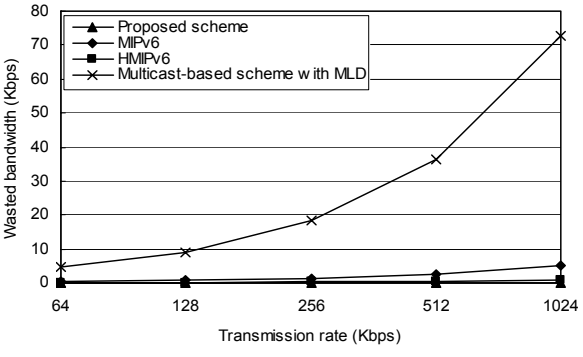


Fig. 7. Average bandwidth wastage during handoffs

the multicast-based scheme with MLD, the default value of the Last Listener Query Interval is used. The multicast-based scheme with MLD results in about 14 times of bandwidth wastage than Mobile IPv6 on average. However, the proposed scheme eliminates bandwidth wastage. Such result can be obtained since the leave latency is only the propagation time of the Multicast Listener Done message in the wireless link and that propagation delay is negligible.

6 Conclusion

In this paper, we propose a new multicast group management scheme. The proposed scheme is tailored to manage groups with only one listener per each group in order to minimize the leave latency. Although leave latency is not a big issue in the wired network since the network bandwidth in the wired network is abundant, it is very critical issue in the wireless network because of the scarcity of the radio spectrum. Using the proposed scheme, the leave latency is extremely small, and the precious wireless bandwidth can be saved.

One of characteristics of the proposed scheme is that it only requires slight processing overhead at the multicast router to handle multicast groups with a single listener. The number of entries in multicast lists maintained by the multicast router in the proposed scheme is same as that in MLD. As far as deployment issue concerned, the proposed scheme is compatible with ordinary group management protocol such as MLD for IPv6, and it can be applied to the actual network incrementally.

References

1. C. Castelluccia, "A Hierarchical Mobility Management Scheme for IPv6," in Proceedings of ISCC '98, pp. 305-309, 1998.
2. C. Perkins, Ed., "IP Mobility Support for IPv4," RFC 3344, Aug. 2002.
3. D. B. Johnson and C. Perkins, "Mobility Support in IPv6," RFC3775, Jun. 2004.
4. P. Bhagwat, C. Perkins, and S. Tripathi, "Network Layer Mobility: An Architecture and Survey," IEEE Personal Communications, pp. 54–64, Jun. 1996.
5. J.P. Mysore and V. Bharghavan, "A New Multicasting-based Architecture for Internet Host Mobility," in Proceedings of ACM Mobicom, 1997.
6. A. Mihailovic, M. Shabeer and A.H. Aghvami, "Multicast For Mobility Protocol(MMP) For Emerging Internet Networks," in Proceedings of PIMRC 2000, Vol. 1, pp. 327-333, 2000.
7. A. Stephane, A. Mihailovic, and A. H. Aghvami, "Mechanisms and Hierarchical Topology for Fast Handover in Wireless IP Networks," IEEE Communication magazine, Vol. 38, No. 11, pp. 112-115, Nov. 2000.
8. A. Helmy, M. Jaseemuddin, and G. Bhaskara, "Multicast-Based Mobility: A Novel Architecture for Efficient Mobility," IEEE Journal on Selected Areas in Communications, Vol, 22, No, 4, pp. 677-690, May 2004.
9. H. Jun and M. Woo, "Performance Analysis of Multicast-based Localized Mobility Support Scheme in IPv6 Networks," in Proceedings of CNSR2004, pp. 243-248, May 2004.
10. B. Cain, S.Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan, "Internet Group Management Protocol, Version 3," RFC 3376, Oct. 2002.
11. S. Deering, W. Fenner, and H. Haberman, "Multicast Listener Discovery (MLD) for IPv6," RFC 2710, Oct. 1999.
12. H. Soliman, C. Castelluccia, K. El-Malki, and L. Bellier, "Hierarchical MIPv6 Mobility Management (HMIPv6)," Internet draft, draft-ietf-mobileip-hmipv6-08.txt, Jun. 2003.
13. S. Das, A. Misra, and P. Agrawal, "TeleMIP: Telecommunications-Enhanced Mobile IP Architecture for Fast Intradomain Mobility," IEEE Personal Communications, pp. 50-58, Aug. 2000.
14. C. Perkins, Kuang-Yeh Wang, "Optimized smooth handoffs in Mobile IP," Proceedings in IEEE International Symposium on Computers and Communications, pp. 340-346, 1999.

On the Minimization of the Number of Forwarding Nodes for Multicast in Wireless Ad Hoc Networks*

Chen-guang Xu, Yin-long Xu, and Jun-min Wu

Department of Computer Science & Technology,
University of Science & Technology of China,
National High Performance Computing Center at Hefei, 230027 P.R. China
gerrard@ustc.edu, {ylxu, jmwu}@ustc.edu.cn
Tel/Fax: ++86-551-3601013

Abstract. Ad-hoc networks are collections of mobile nodes communicating using wireless media and without any fixed physical infrastructure. Multicast is an important application in wireless ad hoc networks. Most of the existing protocols construct a VMB (*Virtual Multicast Backbone*) to provide multicast services. In this paper, we will use MSCDS (*Minimum Steiner Connected Dominating Set*) in UDG (*Unit Disk Graph*) to model the optimal VMB, which aims at minimizing the number of forwarding nodes and present a centralized approximation algorithm with a PR (*Performance Ratio*) approaching $2c+1$, where c is the PR of *edge weighted Steiner tree* algorithm, currently with $c=1.55$. It is an improvement of the previous best approximation guarantee 10.

1 Introduction

Ad hoc networks attract more and more attentions in recent years, due to its potential applications in many areas. A wireless ad hoc network consists of a collection of radio devices located in a region, without any fixed physical backbone infrastructures. Two devices u and v can communicate if and only if u is in v*'s* transmission range and v is in u*'s* transmission range. In this paper, we assume that all the nodes have the same transmission radius and the wireless ad hoc network can be modeled as UDG (*Unit Disk Graph*), where two nodes are neighbors if and only if their distance is no more than 1.

A wireless ad hoc network has no physical backbone, so routing protocol design is very challenging. Virtual backbone based routing is a promising approach, though there are no physical infrastructures in wireless ad hoc networks. To minimizing the virtual backbone, it is a common approach to model it as the MCDS (Minimum Connected Dominating Set) problem. There are several approximation algorithms for MCDS [4, 5, 6, 7, 8, 9, 15]. MCDS is still NP-hard in UDG [14], and *Chen* and *Du* [10] proposed a PTAS for MCDS in UDG.

* This Paper is supported by National Natural Science Foundation of China [No. 60173048].

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 286 – 294, 2005.

However, virtual backbone cannot perform efficiently when it serves for multicast. Most protocols [1, 2, 3] construct a VMB (*Virtual Multicast Backbone*) to provide multicast services. In [11], *Wu* and *Xu* firstly used MSCDS (*Minimum Steiner Connected Dominating Set*), which is proposed by *Guha* and *Khuller* in [12], to model the optimal VMB aiming at minimizing the number of forwarding nodes for multicast. *Guha* and *Khuller* reduced Set-Cover problem to MSCDS in general graph [12]. So unless $NP \subseteq DTIME$ $[n^{O(\log\log n)}]$, there is no polynomial algorithms with approximation ratio better than $O(\log n)$ for general graph, where n is the number of nodes in the graph. So far, the best approximation algorithm for MSCDS in general graph based on the *greedy scheme* has a performance ratio of $(c+1)\mathrm{H}(k)+c-1$[12], where k is the maximum degree of the nodes in the specified node set M, and c is the PR of the *edge weighted Steiner tree* algorithm, currently with $c=1.55$ [13]. MSCDS is a generalization of MCDS [12], so MSCDS is still NP-hard in UDG. In [11], *Wu* and *Xu* proposed a distributed approximation algorithm for MSCDS in UDG, with a performance ratio of 10, which is the previous best result. In this paper we will give a centralized algorithm with the approximation ratio approaching $2c+1$. This may be an evidence to show that currently existing implemented approximations can be greatly improved.

The rest of the paper is organized as follows. In Section 2, we introduce the model of MSCDS. In Section 3, we present the approximation algorithm for MSCDS and analyze its performance. Section 4 serves as concluding remarks.

2 Preliminaries

Due to the scarce resource of ad hoc networks, we focus on construct the optimal VMB for multicast. Since the forwarding cost is much more than the receiving cost, we aim at minimizing the number of forwarding nodes in VMB for multicast to decrease overhead and cost. The MSCDS only consists of the forwarding nodes and connects all the multicast nodes [11]. So we use MSCDS instead of MST, which may consist of receiving nodes, to model the optimal VMB. See in Fig. 1. We give the definition of SCDS and MSCDS first.

Definition 1 [12]. *Let $G(V, E)$ be a graph, and $M \subseteq V$ is a subset of nodes. If there is a subset $K \subseteq V$, such that*

1) $\forall m_i \in M \ \exists\ k_j \in K$, *$m_i$ is dominated by k_j, i.e. m_i is one of the neighbors of k_j,*
2) *The subgraph induced by K is connected.*

K is called an SCDS (Steiner Connected Dominating Set) of M. The SCDS of M with the minimum number of nodes is called MSCDS (Minimum Steiner Connected Dominating Set).

The MSCDS connectedly dominate all the nodes in the subset M. We may let the subset M be the set of the multicast nodes, then all the nodes in MSCDS form the Virtual Multicast Backbone and only the nodes in MSCDS need to forward during multicasting [11]. See in Fig. 1.

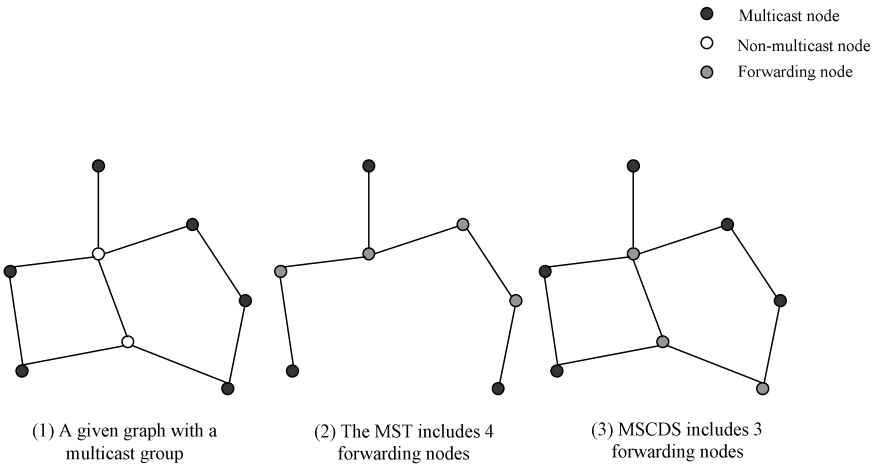


Fig. 1. MSCDS and Virtual Multicast Backbone

Definition 2. *A UDG* (*Unit Disk Graph*) *is a graph induced by a set of points in the Euclidean plane such that two points have an edge in between if and only if their distance is no more than* 1.

Definition 3. *Let G* (*V, E*) *be a Unit Disk Graph and node u*∈ *V. The* 1*-disk of u is defined to be a round area centered at u and of radius* 1. *The* 1/2*-disk of u is defined to be a round area centered at u and of radius* 1/2.

Assume that all the nodes in the wireless ad hoc networks have the same transmission radius, and thus we can model the ad hoc networks as a UDG. According to above analysis, we format the problem of constructing a minimal Virtual Multicast Backbone as the problem of MSCDS in UDG. Before providing the algorithm, we prove two lemmas. We prove that for an area of 2*L**2*L* (*L* is a positive integer), the MDS (*Minimum Dominating Set*) can be computed in polynomial time, which will be useful for the analysis of the algorithm in Section 3.

Lemma 1. *Let* $G(V,E)$ *be a Unit Disk Graph, and* $M \subseteq V$ *be a subset of nodes located in a* 2*L**2*L square. Then the MIS* (*Maximal Independent Set*) *of M is with at most* $4(2L+1)^2/\pi$ *nodes.*

Proof: In *Unit Disk Graph,* two nodes *u*, *v* are neighbors if and only if *d* (*u, v*)≤1. It means that the 1/2-disk of *u* and the 1/2-disk of *v* intersect. Thus, each node covers an area of $\pi/4$, and the areas covered by independent nodes do not intersect. Since all the nodes in a 2*L**2*L* area can cover an area of (2*L*+1)*(2*L*+1) at most,

$$\left|MIS_M\right| \le (2L+1)*(2L+1)/(\pi/4) = 4(2L+1)^2/\pi . \qquad \square$$

Lemma 2. *Let $G(V,E)$ be a Unit Disk Graph, and $M \subseteq V$ be a subset of nodes located in an area of 2L*2L. Then the MDS (Minimal Dominating Set) of M can be computed in time $O(n^{4(2L+1)^2/\pi})$, where n is the number of nodes in G.*

Proof: The MIS (*Maximal Independent Set*) of *M* is the DS (*Dominating Set*) of *M* [18].From Lemma 1, we have:

$$\left|MDS_M\right| \le \left|MIS_M\right| \le (2L+1)*(2L+1)/(\pi/4) = 4(2L+1)^2/\pi .$$

Using the brutal search, computing MDS for M can be done in time

$$\sum_{k=1}^{4(2L+1)^2/\pi} C_n^k = O(n^{4(2L+1)^2/\pi}).$$ □

3 The Algorithm for MSCDS in UDG

3.1 The Algorithm

In this subsection, we will give the algorithm, which is mainly based on the *edge weighted Steiner tree* algorithm [13] and the *shifting strategy* [16, 17].

Algorithm Cell Dominating
Input: a UDG *G* (*V, E*), and a set $M \subseteq V$ of nodes. Let |*M*| = *p*, |*V*| = *n*.
Output: a *Steiner Connected Dominating Set* for *M*.

Case 1: If there are two nodes $u, v \in M$, such that their distance $d(u, v) > p$, we only use the *edge weighted Steiner tree* algorithm to connect *M*. Assume that tree *T* is the result of the algorithm. Return *V*(*T*), where *V*(*T*) is the set of nodes of *T*.

Case 2: If $\forall u, v \in M$, $d(u,v) \le p$, we will take another way, which consists of two phases.

Phase 1: We use the *shifting strategy* [16, 17]. Assume that all the nodes in *M* are located in an area *I*.

1.1) Divide *I* into vertical and horizontal strips of width 2. Label each of the boundary lines of the strips orderly, say 0, 1, 2, 3,…, and 0, 1, 2, 3,…, for vertical and horizontal respectively.

1.2) Let *L* be a small positive integer, and $0 \le j \le L-1$, $0 \le k \le L-1$. Define *partition*(*j, k*) to be the partition of the area *I*, with the horizontal boundary lines labeled $j, j+L, j+2L, \ldots$, and the vertical boundary lines labeled $k, k+L, k+2L, \ldots$. See in Fig. 2. Those lines form the boundary lines of *partition* (*j, k*).

1.3) A cell in *partition* (*j, k*) is defined to be a 2*L**2*L* area, with the boundary lines of *partition* (*j, k*). For each cell *e* in *partition* (*j, k*), we compute the MDSe (*Minimum Dominating Set*) for the nodes belonging to *M* in cell *e* by brutal search. And OPT_{DS}*(*j, k*) is defined to be the union of MDSe for all cells in *partition* (*j, k*).

1.4) For $0 \le j \le L-1$, $0 \le k \le L-1$, choose the OPT_{DS}* (*j, k*) with the minimal number of nodes, and denote it as OPT_{DS}*.

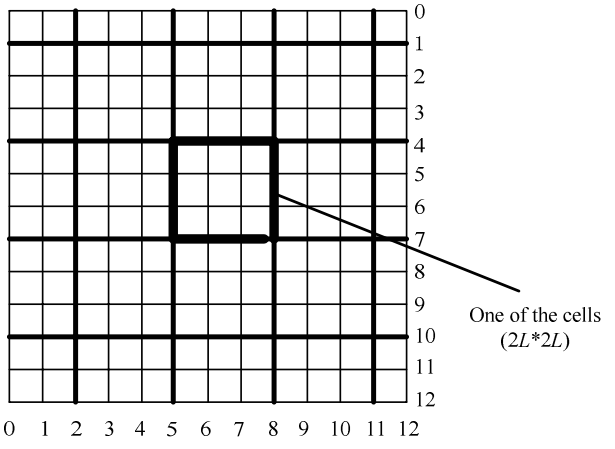


Fig. 2. *L*=3, *partition* (2, 1)

Phase 2: Connect the dominating set OPT_{DS}*

2.1) For every node $v \in OPT_{DS}$*, arbitrarily choose one of it dominating nodes in *M*, and all those nodes form the set *S*. Then we have $|S| \leq |OPT_{DS}^*|$.

2.2) Use the *edge weighted Steiner tree* algorithm to connect all the nodes in *S*. Let *T* be the result.

2.3) Return $V(T) \cup OPT_{DS}$*, where $V(T)$ is the set of nodes of *T*.

End.

3.2 Performance Analysis

Let OPT_{SCDS} be the optimal result of the MSCDS problem. We will analyze the efficiency of the *Cell Dominating* algorithm in this subsection. We analyze the results in both Case 1 and Case 2, and prove that the *Cell Dominating* algorithm has a PR approaching $2c+1$.Time complexity analysis is also given.

Lemma 3. *In Case 1, $V(T)$ is an SCDS of M, and with at most 2c times of the number of nodes in MSCDS for M.*

Proof: $V(T)$ is clearly an SCDS of *M*. OPT_{SCDS} can connectedly dominate *M*, so the spanning tree of $OPT_{SCDS} \cup M$ is a Steiner tree of *M*.

$$|V(T)| - 1 \leq c * (|M| + |OPT_{SCDS}| - 1)$$

where *c* is the PR of the *edge weighted Steiner tree* algorithm, currently with $c = 1.55$ [13].Since OPT_{SCDS} connects *u* and *v,* we have

$$|OPT_{SCDS}| \geq p = |M|$$

$$|V(T)| \leq 2c * |OPT_{SCDS}| - c + 1.$$

The performance ratio in Case 1 is:

$$|V(T)|/|OPT_{SCDS}| \le 2c. \quad \square$$

Lemma 4. *In Case 2,* $OPT_{DS}^{*}(j,k)$ *is a Dominating Set of M, and can be computed in polynomial time.*

Proof: $OPT_{DS}^{*}(j,k)$ is clearly a dominating set of *M*. Note that for each cell of 2*L**2*L*, we only have to search a region of $(2L+2)*(2L+2)$. From Lemma 2, we can prove that the MDS of every cell of 2*L**2*L* can be computed in time $O(n_e^{4(2L+1)(2L+1)/\pi})$, where n_e is the number of nodes in the extended cell $(2L+2)*(2L+2)$. So the total time for computing $OPT_{DS}^{*}(j,k)$ is at most:

$$\sum_{e \in partition(j,k)} n_e^{4(2L+1)^2/\pi} = O(n^{4(2L+1)^2/\pi}). \quad \square$$

In the following, we will say node *v* intersects line *l*, if and only if the 1-disk of *v* intersects line *l*. Let OPT_{DS} be the optimal result of the *Minimum Dominating Set* of *M* in *G*.

Definition 4. See in Fig. 3.

$OPT_{DSX}(j,k) = \{v \mid v \in OPT_{DS}(j,k)$, and *v* intersects the horizontal boundary line of *partition* (*j, k*) , and *v* doesn't intersect the vertical boundary line of *partition* (*j, k*)}.

$OPT_{DSY}(j,k) = \{v \mid v \in OPT_{DS}(j,k)$, and *v* doesn't intersect the horizontal boundary line of *partition* (*j, k*), and *v* intersect the vertical boundary line of *partition* (*j, k*)}.

$OPT_{DS}(j,k) = \{v \mid v \in OPT_{DS}(j,k)$, and *v* intersects the horizontal boundary line of *partition* (*j, k*) , and *v* intersect the vertical boundary line of *partition* (*j, k*)}.

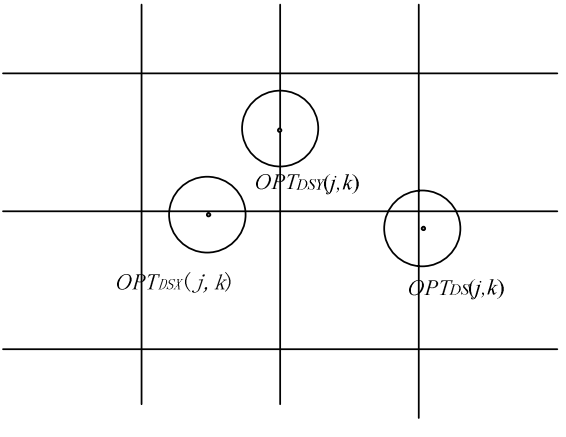


Fig. 3. Boundary lines of *partition* (*j, k*)

Lemma 5. $\left|OPT_{DS}^{*}(j,k)\right| \le \left|OPT_{DS}\right| + \left|OPT_{DSX}(j,k)\right| + \left|OPT_{DSY}(j,k)\right| + 3\left|OPT_{DS}(j,k)\right|$

Proof: Since each node belonging to $OPT_{DSX}(j,k)$ or $OPT_{DSY}(j,k)$ intersects two cells of $partition(j,k)$, and it is counted twice in the right side of the inequation. And each node belonging to $OPT_{DS}(j,k)$ intersects 4 cells in $partition(j,k)$, so it is counted 4 times. Then it will form the Dominating Set for every cell in $partition(j,k)$.

$$\left|OPT_{DS}^{*}(j,k)\right| \le \sum_{e\in partition(j,k)} \left|MDS_e\right| \le \left|OPT_{DS}\right| + \left|OPT_{DSX}(j,k)\right| + \left|OPT_{DSY}(j,k)\right| + 3\left|OPT_{DS}(j,k)\right|. \quad \square$$

Lemma 6. $\left|OPT_{DS}^{*}\right| \le (1+1/L)^2 * \left|OPT_{DS}\right|$, *where* OPT_{DS} *is the optimal result of the Minimum Dominating Set for M.*

Proof: From Lemma 5,

$$\sum_{j=0}^{L-1}\sum_{k=0}^{L-1} \left|OPT_{DS}^{*}(j,k)\right|$$
$$\le \sum_{j=0}^{L-1}\sum_{k=0}^{L-1} \left(\left|OPT_{DS}\right| + \left|OPT_{DSX}(j,k)\right| + \left|OPT_{DSY}(j,k)\right| + 3\left|OPT_{DS}(j,k)\right|\right).$$

For $v \in OPT_{DS}$, if the 1-disk of v intersects horizontal strips boundary line m and vertical strips boundary line n (defined in Case 2, Phase 1), v will be counted in $OPT_{DSX}(m \bmod L, X)$ for $0 \le X \le L-1, X \ne (n \bmod L)$, $L-1$ times in all. In the same way, v will be counted in $OPT_{DSY}(X, n \bmod L)$ $L-1$ times in all. And v will be counted only once in $OPT_{DS}(m \bmod L, n \bmod L)$. So

$$\sum_{j=0}^{L-1}\sum_{k=0}^{L-1} \left|OPT_{DS}^{*}(j,k)\right|$$
$$\le L^2\left|OPT_{DS}\right| + (L-1)\left|OPT_{DS}\right| + (L-1)\left|OPT_{DS}\right| + 3\left|OPT_{DS}\right|$$
$$= (L+1)^2\left|OPT_{DS}\right|.$$

Since OPT_{DS}* is the one with the smallest cardinality of OPT_{DS}*(j, k) for all $0 \le j, k \le L-1$,

$$\left|OPT_{DS}^{*}\right| \le \sum_{j=0}^{L-1}\sum_{k=0}^{L-1}\left|OPT_{DS}^{*}(j,k)\right| / L^2 \le (1+1/L)^2\left|OPT_{DS}\right|. \quad \square$$

Lemma 7. $V(T) \cup OPT_{DS}^{*}$ *in Case 2 is an SCDS of M. And* $V(T) \cup OPT_{DS}^{*}$ *is* $\left(c + (1+c)(1+1/L)^2\right)$*- approximation for the Minimal Steiner Connected Dominating Set of M.*

Proof: T connects OPT_{DS}* and OPT_{DS}* dominate *M*, so $V(T) \cup OPT_{DS}^{*}$ is an SCDS of *M*. Since OPT_{SCDS} can connectedly dominate all the nodes in *S*, the spanning tree of $OPT_{SCDS} \cup S$ is a Steiner tree of *S*. So

$$|V(T)|-1 \le c(|OPT_{SCDS}|+|S|-1) \le c(|OPT_{SCDS}|+|OPT_{DS}^{*}|-1).$$

From Lemma 6,

$$\begin{aligned}|V(T)\cup OPT_{DS}^{*}| &\le |V(T)|+|OPT_{DS}^{*}| \\ &\le c*(|OPT_{SCDS}|+|OPT_{DS}^{*}|)-c+1+|OPT_{DS}^{*}| \\ &\le c*|OPT_{SCDS}|+(1+c)*|OPT_{DS}*|-c+1.\end{aligned}$$

Since $|OPT_{DS}| \le |OPT_{SCDS}|$,

$$|V(T)\cup OPT_{DS}^{*}| \le (c+(1+c)(1+1/L)^2)*|OPT_{SCDS}|-c+1.$$

$$PR = |V(T)\cup OPT_{DS}^{*}|/|OPT_{SCDS}| \le c+(1+c)(1+1/L)^2. \quad \square$$

From Lemma 3 and Lemma 7, the algorithm based on *Cell Dominating* has a PR verging into $2c+1$, when $L \to \infty$. Phase 1 in Case 2 takes most of the time, so we only consider the time complexity of Phase 1 in Case 2. There are *L*L* partitions, and in each partition it takes $O(n^{4(2L+1)^2/\pi})$ to compute $OPT_{DS}^{*}(j,k)$. So the time complexity is $O(L^2 * n^{4(2L+1)^2/\pi})$.

4 Conclusions

This paper addresses the MSCDS problem, which is used to model optimal VMB for multicast in wireless ad hoc networks. An algorithm based on *Cell Dominating* is proposed for MSCDS in UDG, with a PR approaching $2c+1$. This may be an evidence to show that currently existing implemented approximations can be greatly improved. An open problem is whether it is possible to design a PTAS for MSCDS in UDG. The future work is to design more efficient distributed algorithms for the MSCDS problem.

References

[1] J. J. Garcia-Luna-Aceves and E.L. Madruga, "The Core-Assisted Mesh Protocol," *IEEE JSAC*, Aug. 1999, pp. 1380–94.

[2] K. Chen and K. Nahrstedt, "Effective Location-Guided Tree Construction Algorithms for Small Group Multicast in MANET," Proc. INFOCOM, 2002, pp. 1180–89.

[3] M. Gerla, S. J. Lee, and W. Su, "On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks," *Internet draft, draft-ietf-manet-odmrp-02.txt,* 2000.

[4] J.Wu, M. Gao, I. Stojmenovic, "On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks ", *International Conference on Parallel Processing*, 2001 , pp. 346 –354.

[5] Fei Dai, Jie Wu, "An Extended Localized Algorithm for Connected Dominating Set Formation in Ad Hoc Wireless Networks", *IEEE Trans on Parallel and Distributed Systems*, Vol. 15, No 10 October 2004.

[6] B. Das and V. Bharghavan, "Routing in ad-hoc net- works using minimum connected dominating sets," *IEEE International Conference on Communications*, pp. 376-380, June 1997.
[7] J. Wu, "Extended Dominating-Set-Based Routing in Ad Hoc Wireless Networks with Unidirectional Links," *IEEE Trans on Parallel and Distributed Systems*, Vol. 9, no. 3, pp. 189-200, Sept., 2002.
[8] Sergiy Butenko, Xiuzhen Cheng, Ding-Zhu Du, and Panos M. Pardalos, "On the construction of virtual backbone for ad hoc wireless networks", volume 1 of *Cooperative Systems*, chapter 3, pages 43--54. Kluwer Academic Publishers, January 2003.
[9] P.-J. Wan, K. M. Alzoubi, and O. Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks", In *IEEE INFOCOM,* June 2002.
[10] X. Cheng, X. Huang, D. Li, W. Wu, and D.-Z. Du., "Polynomial-time approximation scheme for minimum connected dominating set in ad hoc wireless networks", *Networks,* Vol. 42(4), 202-208 2003.
[11] Yafeng Wu ,Yinlong Xu ,Guoliang Chen, Kun Wang, "On the Construction of Virtual Multicast Backbone for Wireless Ad Hoc Networks", In *IEEE MASS* October 2004.
[12] S. Guha and S. Khuller, "Approximation algorithms for connected dominating sets," *Algorithmica,* Vol. 20(4), April 1998, pp. 374-387.
[13] Gabriel Robins and Alexander Zelikovsky, "Improved Steiner tree approximation in graphs",. In *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms,* pp. 770--779, 2000.
[14] B.N.Clark, C.J.Colbourn and D.S.Johnson, "Unit Disk Graphs", *Discrete Mathematics,* Vol.86, 1990, pp.165-177.
[15] S. Guha and S. Khuller,"Improved methods for approximating node weighted Steiner trees and connected dominating sets." *Inform and Computing*, 150 (1999), pp. 57—74.
[16] Hochbaum, D.S., and Maass, W, "Approximation schemes for covering and packing problems in image processing and VLSI". *Journal of ACM* 32(1985), pp. 130-136.
[17] Xiang-yang Li and Yu Wang, "Simple heuristics and PTASs for intersection graphs in wireless ad hoc networks". *ACM Dial-M*, September 28, 2002 Atlanta, Georgia, USA.
[18] Shu-he Wang, "*Graph Theory and its algorithm*", pp. 127, USTC press, 1990.

The Impact of Mobility Modeling in Mobile IP Multicast Research*

Guoliang Xie[1,2], Mingwei Xu[1], Kwok-Yan Lam[2], and Qian Wu[1]

[1] Department of Computer Science and Technology, Tsinghua University, Beijing, P.R. China, 100084
{xgl, xmw, wuqian}@csnet1.cs.tsinghua.edu.cn
http://netlab.cs.tsinghua.edu.cn
[2] School of Software, Tsinghua University
lamky@tsinghua.edu.cn

Abstract. Since the deployment of mobile IP multicast protocols is still relatively rare, research in this area is typically based on simulations. However, many of the previous evaluations of mobile IP multicast protocols were based on a single, simple mobility model, and thus fail to capture the variety of mobility patterns likely to be exhibited by mobile IP multicast applications. The impact of different mobility models on the performance of mobile IP multicast protocols, including RS, BT and MoM, is evaluated in the paper. Results show that the protocol performance may vary drastically across mobility models and performance rankings of protocols may vary with the mobility models used. Two key mobility metrics that help to explain these performance variations are also demonstrated.

1 Introduction

Providing multicast support for mobile hosts in an IP network is a challenging issue [1]. Many researchers have been engaged in valuable research in this area and a number of approaches for mobile IP multicast have been proposed in the past few years. For instance, the current version of Mobile IP [2] proposes two approaches to support mobile multicast, which are called Remote Subscription (RS) and Bi-directional Tunneling (BT). For the sake of improving the multicast delivery to mobile receivers in Bi-directional Tunneling, a Mobile Multicast protocol (called MoM) is proposed in [3]. An earlier survey on mobile IP multicast can be found in [1], which covers well the approaches proposed until 2004. Researchers often use simulation to validate their algorithms and to evaluate the performance of their protocols. In order to thoroughly simulate a new protocol, it is imperative to use a mobility model that accurately represents the mobility

* This research is supported by the National Natural Science Foundation of China (No. 60373010), the National 973 Project Fund of China (No 2003CB314801), and Cooperative research project on Mobile IPv6 Multicast between Hitachi (China) and Tsinghua University.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 295–304, 2005.

pattern of the mobile nodes (MNs). Only in this type of scenario is it possible to provide realistic performance measurements.

The purpose of this paper is to investigate the impact of mobility models on mobile IP multicast protocols performance. Our study focuses on simulation-based evaluation because mobile IP multicast is still an emerging area and deployment is relatively uneventful, and simulation can provide researchers with a number of significant benefits, including repeatable scenarios, isolation of parameters, and exploration of a variety of metrics. It must be considered that communication patterns also have a significant impact on routing performance and merit a study on their own. However, in this study we want to isolate the effects of mobility, hence we fix the communicating traffic pattern to constant bit-rate audio streams with long enough session times.

In this paper, our interest lies in examining the importance of choosing a mobility model in the simulation of mobile IP multicast protocols. While doing so, a rich set of parameterized mobility models is primarily introduced including Random Walk, Random Waypoint, Gauss-Markov, City Section and Exhibition models. We then examine the performance of RS, BT and MoM under each of the mobility models. Our results show that mobility patterns do affect multicast routing performance and that the observed performance variations can be explained by two key mobility metrics - number of link changes and multicast agent density.

This paper is organized as follows. Section 2 briefly introduces some previously used mobility models and motivation. Then, in Section 3, five mobility models with detailed description are presented. In Section 4, the simulation environment and methodology are introduced. In Section 5, a set of experiments is carried out and analytical results are presented. Finally, in Section 6, the conclusions from this study and planned future work are listed.

2 Motivation

Motivation for this research originates from the fact that many of the previous performance evaluations of mobile IP multicast protocols were based on a single, simple mobility model, and thus fail to capture the variety of mobility patterns likely to be exhibited by mobile IP multicast applications. For example, [4] used a mobility model in which hosts can be in one of two states (ignoring travel time): at the home network or at a foreign network. [5] performed a comparison study of the two improved protocols: MoM and RBMoM, using the mobility model of random direction, in which MN moves at each clock in any direction uniformly. [6] used another simple mobility model, in which each MN can either stay in the same IP subnet or move into one of the neighboring IP subnets with equal probabilities in any time unit. Most of the other studies [7], [8], [9] even fail to specify the choice about the mobility model and parameters in use. The work of this paper is hence motivated to illustrate the importance of choosing a mobility model in the simulation of mobile IP multicast protocols.

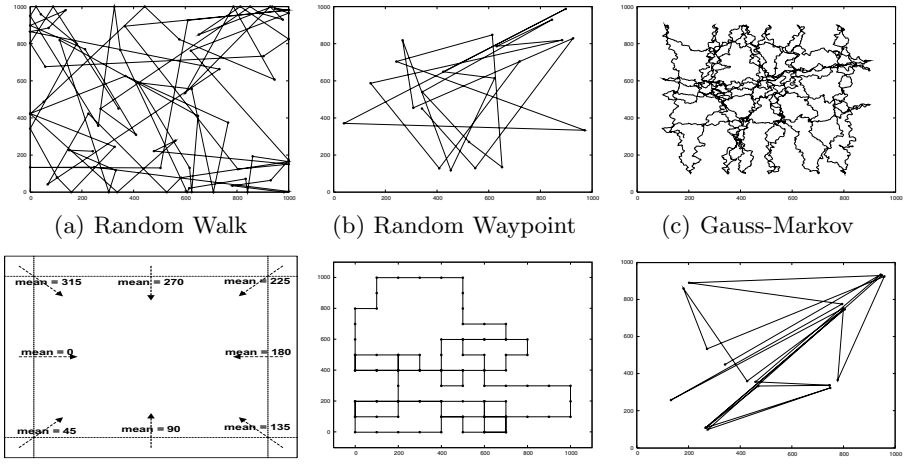
3 Mobility Models

Extensive research has been done in mobility modeling. Researchers in this area can choose from a variety of models that have been developed in the wireless communications and mobile computing community during the last decades. In this work, we use a variety of mobility models designed to capture a wide range of mobility patterns for mobile IP multicast applications. We choose models from different classes of motion, including both statistical models and constrained topology based models that simulate real-world scenarios [10]. The models we use are listed in Table 1 and described below:

Table 1. Mobility models and their application

Model	**Application**
Random Walk	Movement in extremely unpredictable ways
Random Waypoint	Wandering in an area
Gauss-Markov	Random movement without sudden stops and sharp turns
City Section	Movement in one section of a city
Exhibition	Visitors to a museum

- **Random Walk**[11]**:** Figure 1(a) shows an example of the movement observed from our implementation. The MN begins its movement in the center of the 1000m x 1000m simulation area or position (500, 500) moves for 1000 seconds. At each point, the MN randomly chooses a direction between 0 and 2π and a speed between 0 and 30 m/s. The MN is allowed to travel for 30 seconds before changing direction and speed.
- **Random Waypoint**[12]**:** In our implementation, an MN using the Random Waypoint mobility model starting at a randomly chosen point; the speed of the MN in the figure is uniformly chosen between 0 and 30 m/s, and the pause time is uniformly chosen between 0 and 10 s.Figure 1(b) shows an example of the travelling pattern.
- **Gauss-Markov**[13]**:** Figure 1(c) illustrates an example travelling pattern of an MN using the Gauss-Markov mobility model. In our simulation, the α parameter is fixed at 0.75, and $s_{x_{n-1}}$ and $d_{x_{n-1}}$ are chosen from a random Gaussian distribution with $mean = 0$ and $standard\ deviation = 1$. The value of $\bar{s}$ is fixed at 10 m/s; the value of $\bar{d}$ is initially 90 degrees but changes over time according to the edge proximity of the node as showed in Figure 1(d).
- **City Section:** Our City Section model Figure 1(e) is based on previous grid-based models [14] and represents path-based motion with low spatial dependence. In our implementation, the grids are placed 100 meters apart, and the speed of each node is set to a fixed value of 30 m/s.
- **Exhibition**[15]**:** As shown in Figure 1(f), Our implementation uses 10 centers placed uniformly. When a node travels to a center, it stops when it is within 20 meters of the center and then pauses for 30 seconds. The speed of a node is random between 0 and 30 m/s, as with Random Waypoint.



(a) Random Walk (b) Random Waypoint (c) Gauss-Markov

(d) Change of Mean Angle Near the Edges (e) City Section (f) Exhibition

Fig. 1. Travelling patterns of an MN using different mobility models

4 Simulation Environment and Methodology

To determine the impact of mobility models on mobile IP multicast protocol performance, we simulated RS, BT, and MoM. We chose RS and BT because almost all the mobile IP multicast protocols are based on these two protocols. MoM was also chosen because it is an improved protocol based on BT.

The simulation environment was built on OMNET++, a discrete event simulator written in C++ [16]. With the Mobility Framework extension [17], it provides a set of independent modules that implement node mobility, dynamic connection management and a wireless channel model. We extended the framework to include the Random Walk, Random Waypoint, Gauss-Markov, City Section and Exhibition mobility models. To verify our mobility model implementations, we ran simulations identical to those reported in [18] that surveys mobility models for ad hoc network. Our results are very close, with slightly difference due to the different input parameters. Then we wrote our own implementation of RS, BT and MoM based on a specification published as an Internet draft [2] and the original MoM publication [3].

The network topology in our simulation is based on an 8 * 8 mesh network which is showed in Figure 2. Each node acts as a multicast router of a local network and also acts as a base station. The power range is set to a square for simplicity, and the distance between two nearby base stations is 125 meters long. Multicast group communication is also simulated. We premise one multicast group with a single source. Multicast messages are generated in a fix host, using an audio like constant bit-rate of 12kbps.

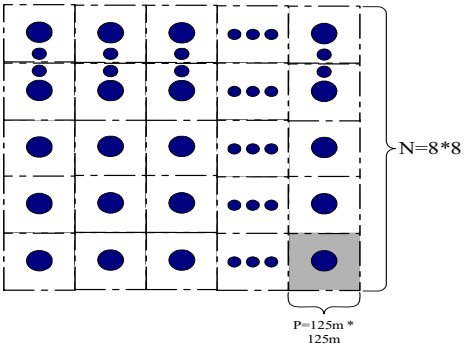


Fig. 2. Network model used in simulation

We run each simulation for 1000 seconds. For each simulation we use 50 nodes, randomly placed over a square field, of which length and width is 1000 meters. The mobility patterns used and their parameters were the same as those used to Section 3, except that the maximum speed Vmax was set to 1, 5, 10, 20, 30, 40, 50 and 60 m/sec to generate different movement patterns for the same mobility model. We summarize all above parameters in Table 2.

Table 2. Simulation Parameters

Parameter	Description	Values
N	Number of LANs	8 * 8 = 64
D	Distance between two nearby BS (meters)	125
P	Power range of a BS (sq.m.)	125 * 125
M	Number of multicast group	1
g	Multicast group size	50
λ	Multicast packet generating rate (pkts/ sec)	5
s	Size of Multicast packet (byte)	300
V	Mobile hosts speed (m/s)	$0 \cdots 60$
T	Total simulation time (seconds)	1000

To remove any effects due to randomness of the traffic pattern, we used different random seeds to generate 3 different traffic patterns having the same number of sources and connections. The results for each model (for a given Vmax) are averaged over simulation runs using these 3 different traffic patterns.

5 Results

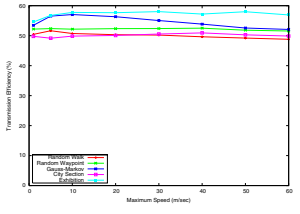
In this section, we explore the impact of different mobility patterns on three different mobile IP multicast protocols - RS, BT and MoM. The results presented illustrate that the choice of a mobility model can have a significant effect on the

performance investigation of a mobile IP multicast protocol. We also analyze the relationship between the mobility metrics and the performance metrics, which helps us gain a lot of insight to answer the question "Why mobility affects protocol performance".

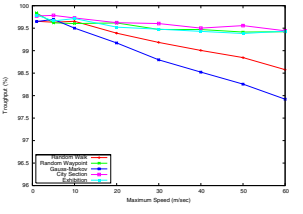
5.1 How Mobility Affects Protocol Performance

We have evaluated the performance of RS, BT and MoM across this rich set of mobility models and observed that the mobility models may drastically affect protocol performance. Lets take RS as an illustrative example. RS shows a difference of almost 10% in transmission efficiency from Random Walk to the Exhibition model as seen from Figure 3(a). As shown in Figure 3(c), the tree maintenance overhead produced by RS in Gauss-Markov is nearly 8 times as much as in City Section. Also, there is an order of magnitude difference in the throughput and number of nodes transmitting multicast data of RS across the various models as shown by Figure 3(b) and 3(d). Similar differences in performance were also observed for other protocols used in our study.

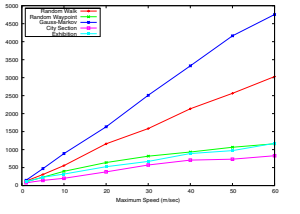
It can be observed that RS achieves the highest transmission efficiency, MoM takes the second place, and the least is that of BT in most cases (Figure 3(a), 3(e), 3(f)). This result is somewhat expected, because our simulation is carried out with comparatively denser topology, which is in favor of optimizing the performance of RS. HoweverWe also observed that BT achieves a higher transmission efficiency than MoM in Gauss-Markov as shown in Figure 4(d). Thus, though MoM is an improved protocol based on BT, it is not always true that MoM performs better than BT.



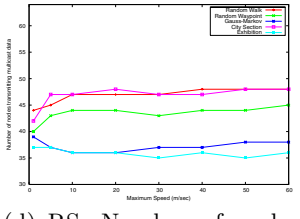
(a) RS: Transmission efficiency



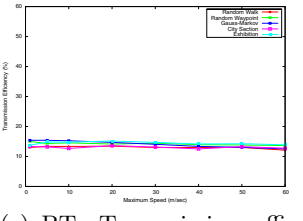
(b) RS: Throughput



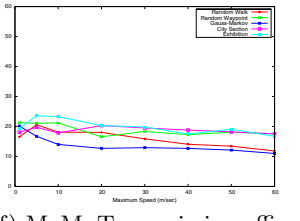
(c) RS: Tree maintenance overhead



(d) RS: Number of nodes transmitting multicast data

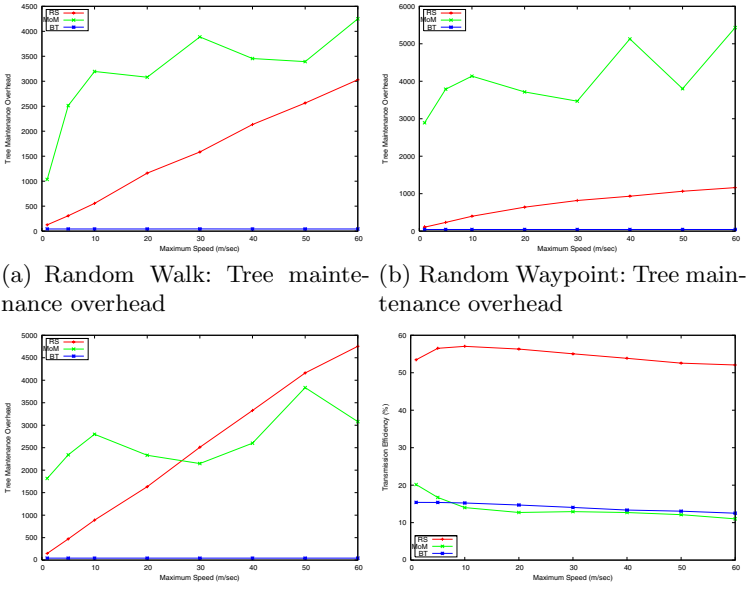


(e) BT: Transmission efficiency



(f) MoM: Transmission efficiency

Fig. 3. Performance of protocols using different mobility models



(a) Random Walk: Tree maintenance overhead

(b) Random Waypoint: Tree maintenance overhead

(c) Gauss-Markov: Tree maintenance overhead

(d) Gauss-Markov: Transmission efficiency

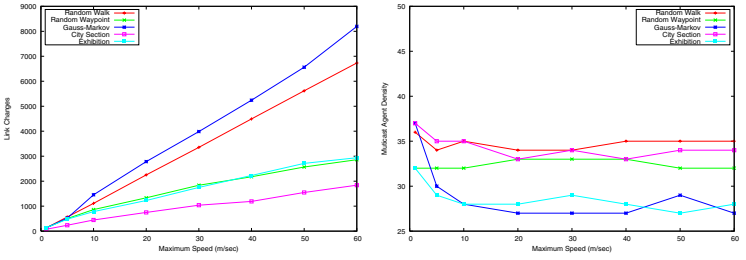
Fig. 4. Relative performance rankings of protocols

It seems that relative rankings of protocols may vary with the mobility model used. We observed that BT results in the least tree maintenance overhead in most cases, while the relative ranking of RS and MoM in terms of tree maintenance overhead seems to depend on the underlying mobility model as shown in Figure 4(a), 4(b), and 4(c): RS has a lower overhead than MoM in the Random Walk and Random Waypoint, but produces higher overhead than MoM in Gauss-Markov.

5.2 Why Mobility Models Affects Performance

In order to find out the correlation between mobility and protocol performance, we implemented some mobility metrics, including direct mobility metrics, like host speed or relative speed, and derived mobility metrics [10], like link changes, link duration, multicast agent density, and etc. Of these metrics, we found that the number of link changes and multicast agent density are able to differentiate between our mobility models and help to explain multicast routing performance. They are defined as:

– **Number of link changes:** The total number of link changes seen during the course of a simulation. A link change is defined as an event when MN comes within radio range of a multicast agent (or BSs) when previously they had not been able to communicate directly.



(a) Number of link changes (b) Multicast agent density

Fig. 5. Values for mobility metrics

– **Multicast agent density:** The average number of multicast agents that there are MNs within their radio range.

Figure 5(a) and 5(b) show values for each of the metrics, as a function of speed. We collect link change information and sample multicast agent density each time a node moves during the course of a simulation. For multicast agent density, the final result is averaged over all the sampling times.

In general, it was observed that RS, BT and MoM had a higher transmission efficiency and lower number of nodes transmitting multicast data for the Exhibition models than that of the Random Waypoint model. Especially for RS, transmission efficiency (Figure 3(a)) and number of nodes transmitting multicast data (Figure 3(d)) depend on the model and the ordering from worst-to-best is roughly predicted by the multicast agent density shown in Figure 5(b). At the same time, all the protocols had a higher throughput and lower tree maintenance overhead for the City Section than for the Gauss-Markov model. As show in Figure 3(c) and 5(a), the ordering of tree maintenance overhead in RS is again similar to that of number of link change. The above observation can be explained as follows:

1. With similar speed, between Random Waypoint and Exhibition, low multicast agent density (for Exhibition) means MN is much more concentrated, thus fewer packets will be needed to forward, and finally it will result in higher transmission efficiency and lower number of nodes transmitting multicast data. We also notice that with the Gauss-Markov model the transmission efficiency declines rapidly as speed increases, this is because the number of link changes increases rapidly as speed increases as seen from Figure 5(a), which finally affects the performance of transmission efficiency.
2. For a given speed, if a mobility pattern has a high number of link changes (for Gauss-Markov), handoff between different multicast agents will happen more frequently. Thus, more packets will be dropped due to link breakage and this will lead to a lower throughput. At the same time, the cost of tree maintenance overhead would be higher since greater effort is required to quit and rejoin the multicast group.

6 Conclusions and Future Work

It has been found out that the performance of a mobile IP multicast protocol significantly varies with different mobility models. Using in the same mobility model, different parameters also lead to widely different performance. The performance of mobile IP multicast protocol should be evaluated with the mobility model which matches best with the expected real-world scenario. If the expected real-world scenario is not available, researchers should make it clear their chosen mobility model and parameters.

Our results show that Mobility metrics are able to differentiate mobility models and help to explain multicast routing performance. The multicast agent density imposed by a particular mobility model is a good predictor of transmission efficiency. Even when the multicast agent density is small, high number of link changes can also degrade transmission efficiency. And the number of link changes is a good predictor of throughput and tree maintenance overhead, while greater amounts of link changes indicating worse performance.

In the future, we plan to study the impact of mobility models on the performance of more other mobile IP multicast protocols, such as multicast with fast handover support and hierarchical routing protocols. We believe that several parameters such as traffic patterns, node power range and initial placement pattern of nodes may affect the routing performance and should be investigated further.

References

1. I. Romdhani, M. Kellil, H-Y. Lach, A. Bouabdallah, H. Bettahar: IP Mobile Multicast: Challenges and Solutions. IEEE Communications Surveys & Tutorials, vol.6, No.1. (2004) 18-41
2. D. Johnson, C. Perkins, and J. Arkko: Mobility Support in IPv6. RFC 3775. (2004)
3. T. Harrison, C. Williamson, W.L. Mackrell and R.B. Bunt: Mobile Multicast (MoM) protocol: Multicast support for mobile hosts. ACM International Conference on Mobile Computing and Networking (MOBICOM). (1997) 151-160
4. C. L. Williamson, T. Harrison, W. Mackrell and R. Bunt: Performance Evaluation of the MoM mobile Multicast: Design Issues and Proposed Architecture, Protocol. Mobile Networks and Applications. (1998) 189-201
5. C. R. Lin and K.M. Wang: Mobile Multicast Support in IP Networks. IEEE INFOCOM 2000. (2000) 1664-1672
6. Y. Wang and W. Chen: Supporting IP Multicast for Mobile Hosts. ACM/Kluwer Mobile Networks and Applications, Special Issue on Wireless Internet and Intranet Access, vol.6, no.1. (2001) 57-66
7. Young-Joo Suh, Hee-Sook Shin, and Dong-Hee Kwon: An Efficient Multicast Routing Protocol in Wireless Mobile Networks. ACM Wireless Networks, vol.7, no.5. (2001) 443-453
8. S. J. Yang and S. H. Park: An Efficient Mobile Multicast Routing for QoS Guarantee in IPv6 Based Networks. World Wireless Congress (WWC 2002). (2002)
9. H. Omar, T. Saadawi, and M. Lee: Multicast Support for Mobile IP with Hierarchical Local Registration Approach. 3rd ACM Int'l. Wksp. Wireless Mobile Multimedia. (2000) 55-64

10. Qunwei Zheng, Xiaoyan Hong, Sibabrata Ray: Recent advances in mobility modeling for mobile ad hoc network research. ACM Southeast Regional Conference 2004. (2004) 70-75
11. I. Rubin and C. Choi: Impact of the Location Area Structure on the Performance of Signaling Channels in Wireless Cellular Networks. IEEE Communications Magazine. (1997) 108-115
12. D. Johnson and D. Maltz: Dynamic source routing in ad hoc wireless networks. In T. Imelinsky and H. Korth, editors, Mobile Computing. (1996) 153-181.
13. B. Liang and Z. Haas: Predictive distance-based mobility management for PCS networks. Proceedings of the Joint Conference of the IEEE Computer and Communications Societies (INFOCOM). (1999) 1377-1384
14. V. Davies: Evaluating mobility models within an ad hoc network. Master's thesis, Colorado School of Mines. (2000)
15. P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark: Scenario Based Peformance Analysis of Routing Protcols for Mobile Ad Hoc Networks. IEEE MobiCom. (1999)
16. A. Varga: The omnet++ discrete event simulation system. Proc. of ESM, Prague, Czech Republic. (2001)
17. W. Drytkiewicz, S. Sroka, V. Handziski, A. Koepke, and H. Karl: A mobility framework for omnet++. 3rd International OMNeT++ Workshop (2003)
18. T. Camp, J. Boleng, and V. Davies: A Survey of Mobility Models for Ad Hoc Network Research. Communication & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, vol.2, no.5. (2002) 483-502

Broadcast in the Locally k-Subcube-Connected Hypercube Networks with Faulty Tolerance[1]

Fangai Liu and Ying Song

Department of Computer Science, Shandong Normal University,
Ji'nan 250014, China
sssongying@126.com

Abstract. It is of great interests to develop algorithms to find the routing path for broadcast in a hypercube with a large number of faulty nodes while the network remains connected. In this paper, we introduce a broadcasting algorithm on locally k-subcube-connected hypercube networks under the above situation, based on the conception of locally k-subcube-connected hypercube. Our algorithm is distributed and local-information-based, namely, in the network each node knows only its neighbors' status and no global information is required in the network. Our broadcasting algorithm can tolerant the upper bound 2^{n-1}-2^{n-k} for faulty nodes under the condition of locally k-subcube-connected hypercube. This is a much larger bound on the number of faulty nodes compared to the previous broadcast algorithms which the number of faulty nodes be bounded by O(n) in hypercube H_n. More over, our algorithm can find nearly optimal length path in hypercube H_n in linear time.

1 Introduction

The efficiency of communication and computation on the networks depends heavily on the efficiency of routing algorithms. As the sizes of interconnection networks become larger and larger, reliability problems arise with increasing frequency. The node fault tolerant routing has become one of the central issues in the interconnection network [2-9]. From the research on networks with faulty nodes, now there are two kinds of emphasis points in the world. Most people pay attention to try their best to select a shortest path to establish a communication between the two given nodes in order to improve the efficiency of the fault tolerant routing [2-4]. I.e. Lee and Hayes proposed the concept of unsafe node and gave a simple routing algorithm [3]. While other researchers value the fault-tolerant quality of the network even more [1,5,6], such as the concept of forbidden set [5,6] and cluster fault tolerance model (abbreviated as CFT routing in what follows)[7,8], have been proposed for this purpose.

[1] This work supported by the Natural Science Foundation of China under Grant No.60373063.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 305 – 313, 2005.

Although all of the above improve the fault-tolerant quality to a certain extent, their algorithms for a fault tolerance model that doesn't allow the number of faulty nodes to be larger than O(n). In [1] they introduce a simple and natural condition, the local subcube-connectivity, which is identified under which hypercube networks with a very large number of faulty nodes still remain connected. They developed efficient routing algorithms on locally subcube-connected hypercube networks. For a locally subcube-connected hypercube network that may contain up to 37.5 percent faulty nodes, their algorithms run in linear time and for any two given nonfaulty nodes, find a routing path of length bounded by four times the Hamming distance between the two nodes.

Now the majority of researches on the broadcasting are focus on the time complexity other than the fault tolerance in the network [3,9]. To the author's knowledge, there are no broadcasting algorithms what can tolerant more than O(n) faulty nodes in the hypercube. In this paper we develop a broadcasting algorithm based on the conception of locally k-subcube-connected hypercube introduced in [1]. Our algorithm can tolerant the upper bound 2n-1-2n-k for faulty nodes under the condition of locally k-subcube-connected hypercube. More over, our algorithms can find near the shortest path in hypercube Hn within linear time.

2 The Locally k-Subcube-Connectivity

A n-dimensional hypercube H_n (or the n-cube for short) consists of 2^n nodes. Each binary string $b_1b_2 \ldots b_{n-k}$ of length n -k corresponds to a k-dimensional subcube H_k in H_n (or a k-subcube for short) of 2^k nodes whose labels are of the form $b_1b_2 \ldots b_{n-k}x_{n-k+1} \ldots x_n$, where each x_j is either 0 or 1. The subcube H_k will also be written as $H_k = b_1b_2 \ldots b_{n-k}$ **. It is easy to see that each k-subcube of H_n is isomorphic to the k-cube.

Definition 2.1 [1]
The n-dimensional hypercube network H_n is locally k-subcube-connected if, in each k-dimensional subcube H_k of H_n, less than half of the nodes in H_k are faulty and the nonfaulty nodes of H_k make a connected graph.

Lemma 2.1 [1]
The nonfaulty nodes in a locally k-subcube-connected n-dimensional hypercube H_n make a connected graph.

Lemma 2.1 shows a nice property for the hypercube networks: Local connectivity in a hypercube network implies global connectivity of the network. In many cases, the local connectivity of a hypercube network can be easily verified and Lemma 2.1 can be conveniently used to ensure the global connectivity of the hypercube network.

3 Broadcast in Locally k-Subcube-Connected Hypercube Networks

In networks, broadcasting often been used in many applications. Based on the above definition and lemma, we designed a new broadcast algorithm as follows. Our

algorithm has well tolerance and time complexity. More over our algorithm can find near the shortest broadcast path between the source node and any nonfaulty nodes in the hypercube.

In order to introduce our algorithm, we give two definitions first.

Definition 3.1
If there is only one bit difference of two nodes in the first k bits, other all bits are same, we call the two nodes pre-k-neighbor nodes to each other.

Definition 3.2
If there is only one bit difference of two nodes and the different bit is the ith bit, other all bits are same, we call the two nodes ith-neighbor nodes to each other.

Now, we give our algorithm as follows:

Algorithm Broadcast

Input: an n-dimensional hypercube Hn with faults and one nonfaulty node u in Hn.
Output: the broadcast paths of non-faulty nodes in Hn that connects u.
From the source node u:

1. Since the k-subcube is connected, it is easy to construct a tree taking this node as the root within the k-subcube and at the same time broadcast its information through the tree.

2. At the same time, it sends the information to its pre-k-neighbor nodes

If some of its pre-k-neighbor node (i.e.x) has already received the information, discard the information and don't pass it on;

else

if there are faulty nodes in its pre-k-neighbor nodes

① for example x is a faulty node and it is its source node's ith-neighbor node, find two adjacent non-faulty nodes w and v, w in the same k-subcube with the source node of x, v in the same k-subcube with x, pass information from w through v to this k-subcube, toward v return to step 2.

② other nonfaulty nodes: parallel broadcast information in the k-subcube of each through it's tree, then return to step 2;

3. go on n-k times so, and the following lemma3.1 can prove that every k-subcube has received this information.

Remark. If at any point the algorithm could not proceed, then stop: the n-cube H_n is not locally k-subcube-connected. The above-mentioned course of constructing a tree taking node x as the root within the k-subcube and broadcasting its information through the tree can be described as concretely: x send the information to its neighbor nodes, these nodes form the first layer of sub nodes taking x as root. At the same time these nodes pass the information to their neighbor nodes (exclude the nodes which send the information to them), if the node, i.e. y, which receive the information has already received the same information, then do not change the tree and discard the information and don't pass it on. Go on so, and until a certain moment, each all of the neighbor nodes have received this information, namely, all the nonfaulty nodes have received the information. If there are two neighbor nodes send the information to each other at the same time in the link between them, collision arise in this link. The solution is discard the information in this link and do not change the tree. This will

not influence the broadcast of the information. We may introduce the course of constructing a tree taking node a as the root within the 3-subcube and broadcasting its information through the tree as the following fig.1 (suppose h is a faulty node).

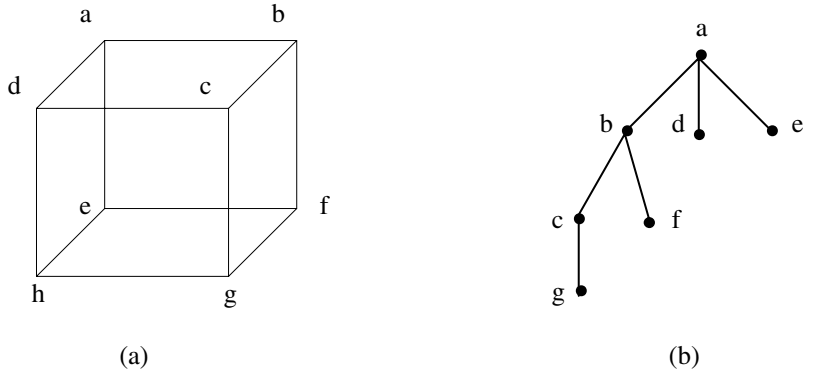


Fig. 1. The course of constructing a tree within the 3-subcube and broadcasting its information through it. (b) is the constructed tree of (a) taking node a as the root

In order to understand algorithm Broadcast, we explain it by the following fig.2:

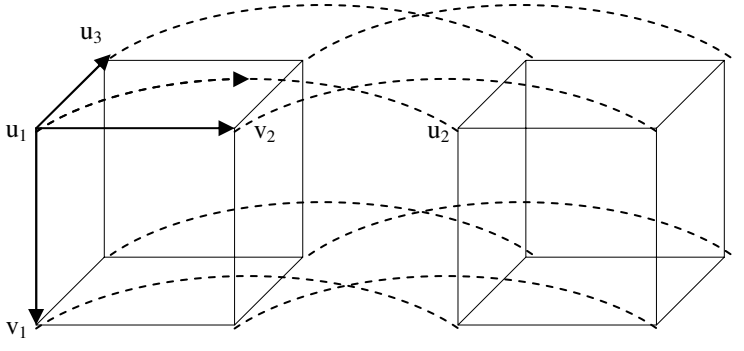


Fig. 2. The application of algorithm Broadcast

From the above illustrate, we can see: if n=4,k=2, the node u_1 broadcasts the information in its 2-subcube through the neighbor node (v_1,v_2), at the same time, send the information to its pre-2-neighbor nodes(namely its 1st-neighbor and 2nd-neighbor nodes u_2,u_3), after u_2 and u_3 receive the information, they broadcast it in the 2-subcube of each. Till all the nonfaulty nodes receive the information.

Remark. This algorithm has solved the problem of the route conflict. When broadcasting within a k-subcube, it broadcast the information along the tree and we

have carried on rational settlement in producing the situation conflicting. This has prevented repetition of the information from transmitting. At the same time, it guarantees that there is only once information transmission during two neighbor nodes. When broadcasting in different k-subcube, it will not bring the conflict. When transmitting information among the k-subcubes, if the node which has already received the information, discard it and don't pass it on. This make each k-subcube been transmitted to only once.

Lemma 3.1. There are n-k steps for transmitting information among k-subcubes by using algorithm Broadcast.

Proof. Using mathematical induction.

① If k=n, namely, n-k=0,then only need broadcast information within k-subcube, there is no transmitting among k-subcubes, so it need 0 step for transmitting information among k-subcubes;

If k=n-1, namely, n-k=1, then there are $2^1=2$ k-subcubes, so it only need 1 step for transmitting information from one k-subcube to another;

② Suppose when it comes to k=t, this conclusion is still right, namely, it need n-t steps for transmitting information among k-subcubes and now there are 2^{n-t} k-subcubes.

Next let's consider k=t-1, and there are $2^{n-t+1}=2\bullet 2^{n-t}$ k-subcubes. Regard 2^{n-t} k-subcubes among them as a large subcube, so there are 2 large subcubes of such. First, transmit information from the source node u (it belongs to a certain large subcube) to another large subcube, this need 1step. Then, according to algorithm Broadcast-I, parallel broadcast information in the two large subcubes respectively. From the suppose we can see, each need n-t steps; So, when it comes to k=t-1,it need n-t+1 steps for transmitting information among k-subcubes.

From the induction and suppose we can learn, lemma 3.1 is established.

So there are n-k steps for transmitting information among k-subcubes by using algorithm Broadcast. □

Remark. Although it needs different time for information to transmit between different subcubes, information take n-k the subcube transmit when transmitting along some route. When the information finished n-k subcube transmitting along all route, it has already been spread all over all the k-subcube in H_n.

Theorem 3.2. If the input n-cube H_n is locally k-subcube-connected, then the algorithm broadcast constructs paths of nonfaulty nodes from u to all the other nonfaulty nodes in time $O(2^{k+1}(n-k))$.

Proof. This algorithm can divide broadcast routing into two parts to consider. One part is transmitting information among k-subcubes, another is broadcasting information within k-subcubes. First, let's consider broadcasting information within k-subcubes: there are 2^k nodes in a k-subcube, since k-subcube is connected and the information is broadcasted along the tree, this takes $O(2^k)$ times. Next, we'll think over transmitting information among k-subcubes: When a node (we mark the k-subcube which include this node K_1) transmit the information to its pre-(n-k)-

neighbor nodes, if the neighbor node is nonfaulty, it only need 1 time unit; else (we mark the k-subcube which include this faulty node K_2) it need to find a pair of non-faulty nodes w and v, let w and v are adjacent and in the k-subcube K_1 and K_2 respectively, this takes $O(2^k)$ time. From lemma3.1 we can see, there are n-k steps for transmitting information among k-subcubes. Corresponding to the two parts, each need $O(2^k(n-k))$time. So *this algorithm can construct paths of nonfaulty nodes from u to all the other nonfaulty nodes in time* $O(2^{k+1}(n-k))$.

Remark. In case k is small, for example k=3, the routing algorithm Broadcast runs in linear time. Even for large k, our algorithm still spends only necessary computational time. For each large k-subcube Hk, we may put no constraints on the structure of Hk. The only thing we can assume is that the subgraph consisting of the nonfaulty nodes in Hk is connected and contains more than half of the nodes in Hk.

Theorem 3.3. If the input n-cube H_n is locally k-subcube-connected, then the algorithm broadcast constructs routing paths of no longer than the shortest length+2^{k+1}-3 of nonfaulty nodes from u to all the other nonfaulty nodes in the faulty tolerance networks.

Proof. Using algorithm Broadcast, the worst case is: the source node u transmit information to the node x (x is a node of k-subcube K) through n-k steps transmitting information among k-subcubes. Each step takes 1 time unit, so it need n-k time unit for all, the length of this path is n-k; another path which also can reach K is: u and K click and links to each other through a faulty node, so it need 2^k time unit to find two nodes w and v and transmit information from u through w to v, and then it takes 1 time unit to send the information to node y (y is a node of k-subcube K), let's suppose the length of path between u and w is m($m\geq 1$),so it need 2^k+1 time unit for all, the length of this path is m+2. Since we consider the worst case, we suppose the length of the path from x to y is 2^k-1 might as well. Now from u to y, the length of the first path is n-k+2^k-1 while that of the second route is m+2.Now, we suppose: $n-k\leq 2^k+1$. Then routing to the k-subcube K through the first path is earlier than the second one, so our algorithm may find the first path. Yet the second path to reach the k-subcube K is the shortest one obviously. So, we can see: the length of the path constructed by algorithm broadcast–the length of the shortest path$\leq n-k+2^k-1-2-m\leq 2^k+1+2^k-3-m=2^{k+1}-2-m\leq 2^{k+1}-3$; so, the length of the path constructed by algorithm broadcast$\leq$the length of the shortest path+2^{k+1}-3. □

In order to understand the worst case of the above proof, we illustrate a simple example (for reducing space, we only consider a useful part in understanding the proof.):

Let n=6,k=2, the source node is 110101. 100101 and 111101 are faulty nodes, other nodes is nonfaulty. Using algorithm Broadcast can find a path from the source node to the node 101111 which is in the 2-subcube 1011**:

110101→ 110001→ 111001→ 101001 → 101101→ 101111(the time to the 2-subcube 1011** is 4 time unit; the length of the path to the node 101111 is 5); yet the shortest path from the source node to 2-subcube 1011** is:

110101→ 110111→ 100111→ 101111(the time to the 2-subcube 1011** is 2^k+1=5 time unit; the length of the path to the node 101111 is 3).

Remark. Our algorithm Broadcast is distributed and local-information-based, when k is small, it can find nearly optimal length path in hypercube H_n. The above case is the worst case, under most situations, this algorithm can find nearly optimal length path.

We illustrate the algorithm Broadcast by an example. Let the source node u=110101 be a nonfaulty node in the 6-cube H_6 and suppose that the nodes 010101, 011110 and 101101 in H_6 are faulty, and k=3. Then the routing path constructed by the algorithm Broadcast-I is (~~010101~~ indicates the faulty nodes):

First the source node u broadcasts the information in its 3-subcube 110**, at the same time transmit it to its pre-3-neighbor nodes 111101, 100101 and 010101. Among them 010101 is faulty node and it is u's 1st-neighbor node. Find two adjacent nonfaulty nodes 110111 and 010111, 110111 is in u's 3-subcube, 010111 and 010101 are in the same 3-subcube 010**. Transmit the information from the node 110111 through 010111 to 3-subcube 010**; as to other two nonfaulty nodes 111101 and 100101, parallel broadcast the information in their 3-subcube 111** and 100** respectively, at the same time, each through nodes 010111□111101 and 100101 transmit the information to their own pre-3-neighbor nodes(exclude the node which send the information to it). Let's skip the other nodes only along the node 100101, first it broadcasts the information in its 3-subcube 100**, at the same time transmit it to its pre-3-neighbor nodes 000101 and 101101. 010101 is faulty node, the information is passed to the node 101111 through 100111. Since this 3-subcube has already received the information, it discards the information. After the node 000101 receive the information, it broadcast the information in its 3-subcube 000**, and at the same time, transmit the information to its pre-3-neighbor nodes 001101 and 010101. Node 010101 is faulty, through node 000111 the information is transmitted to 010111. The 3-subcube 010** has already received the information, so discards it; as to the node 001101, since the 3-subcube 001** has already received the information, it discards the information. Till now, all the 3-subcube has received the information, namely, the source node u has already broadcasted the information to all the nonfaulty nodes.

According to the above analysis, the graph of the process for constructing broadcast routes is as follows Fig.3:

It's easy to see that in this example, the broadcast routes constructed by algorithm Broadcast are exactly the shortest broadcast routes.

Next let's analysis the tolerance of the locally subcube-connected hypercube H_n.

Theorem 3.4. Broadcasting algorithms in locally k-subcube-connected hypercube H_n can tolerant the upper bound $2^{n-1}-2^{n-k}$ for faulty nodes.

Proof. From the definition of locally k-subcube-connected we can see, the faulty nodes are less than the nonfaulty nodes in H_n. So, under the condition of H_n is locally k-subcube-connected, the number of the faulty nodes which can be tolerant by H_k is $2^k/2-1=2^{k-1}-1$.

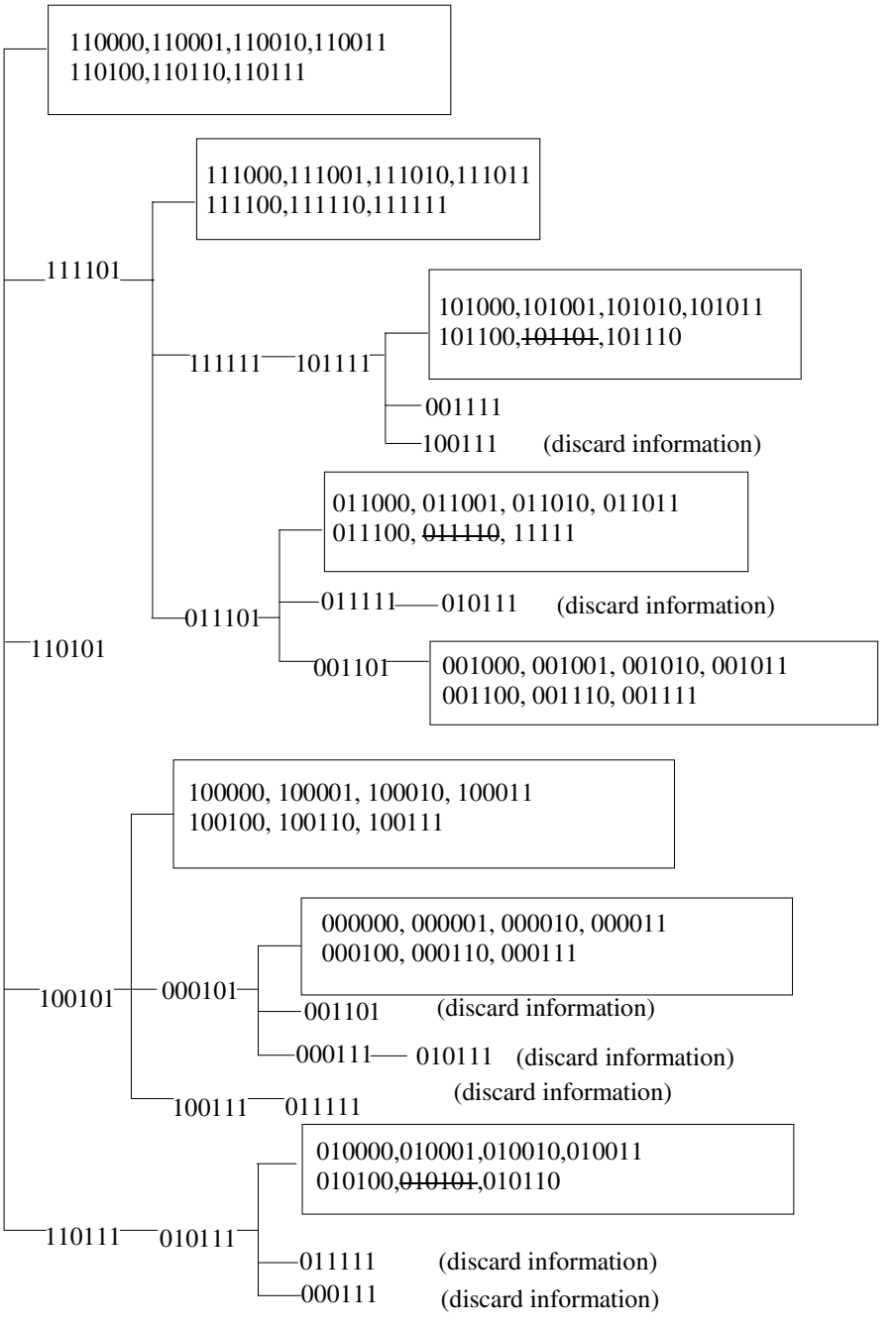


Fig. 3. The graph of the process for constructing broadcast routes by algorithm Broadcast

The percentage of faulty nodes that can be tolerant by H_k compared to all nodes in H_k is $(2^{k-1}-1)/2^k=1/2-1/2^k$.

The number of faulty nodes that can be tolerant by H_n is $(1/2-1/2^k)*2^n=2^{n-1}-2^{n-k}$. □

From the above we can see, with the increasing of k, the number of faulty nodes that can be tolerant by H_n is larger. In case k is small, k=3, the number of faulty nodes that can be tolerant by H_n is$2^{n-1}-2^{n-3}=3*2^{n-3}$. When k=n, the number of faulty nodes that can be tolerant by H_n is $2^{n-1}-1$.

4 Conclusion

By studying the definition of locally k-subcube-connected hypercube, we designed a broadcasting algorithm in networks with faulty nodes. If the input n-cube H_n is locally k-subcube-connected, then the algorithm Broadcast runs in linear time $O(2^{k+1}(n-k))$. And it could construct broadcasting paths of nearly optimal length. We analyzed that our broadcasting algorithm can also contain $2^{n-1}-2^{n-k}$ faulty nodes for the upper bound under the condition that H_n is locally k-subcube-connected. To the authors' knowledge, there has not been any development of efficient broadcasting algorithms in a hypercube networks for a faulty tolerance model that allows the number of faulty nodes to be larger than O(n). Our algorithm has greater improvement in this respect, it brings faulty-tolerant ability up to $2^{n-1}-2^{n-k}$ from O(n)on condition that H_n is locally k-subcube-connected.

References

1. Jianer Chen, Guojun Wang and Songqiao Chen. Locally Subcube-Connected Hypercube Networks: Theoretical Analysis and Experimental Results. IEEE Transactions on Computers, vol.51, No.5, pp.530-540, MAY.2002.
2. T.C.Lee and J.P.Hayes. A Fault-Tolerant Communication Scheme for Hypercube Computer. IEEE Trans. Computers,vol.41, No.10, pp.1242-1256, OCT.1992.
3. G.-M. Chiu and S.-P. Wu. A Fault-Tolerant Routing Strategy in Hypercube Multicomputers. IEEE Trans. Computers, vol.45, No.2, pp.143-155, FEB.1996.
4. M.-S Chen and K. G. Shin. Adaptive Fault-Tolerant Routing in Hypercube Multicomputers. IEEE Transactions On Computers, vol.39, No.12, DEC.1990
5. A.H.Esfahanian. Generalized Measures of Fault Tolerance With Application to n-Cube Networks. IEEE Trans. Computers, vol.38, No.11, pp.1586-1591, Nov.1989.
6. S.Latifi. Combinatorial Analysis of the Fault Diameter of the n-cube. IEEE Trans.Computers, vol.42, No.1, pp.27-33, Jan.1993.
7. Q.-P.Gu and S. Peng. Optimal Algorithms for Node-to-Node Fault Tolerant Routing in Hypercubes. The Computer J., vol.39, No.7, pp.626-629,1996.
8. Q.-P.Gu and S. Peng. k-Pairwise Cluster Fault Tolerant Routing in Hypercubes. IEEE Trans. Computers, vol.46, No.9, pp.1042-1049, 1997.
9. J.Wu and E.B.Fernandez. Reliable broadcasting in faulty hypercube computers. Microprocessing and Microprogramming, vol.39, pp.43-53, 1993.

Performance Analysis of Route Discovery in Wireless Ad Hoc Networks: A Unified Model

Xian Liu and Yupo Chan

Department of Systems Engineering, University of Arkansas at Little Rock,
Little Rock, AR 72204-1099, USA
{xxliu, yxchan}@ualr.edu

Abstract. Consider a mobile ad hoc network with source-routing. The frequency of route discovery is an important performance metric. Such a metric measures the overhead of transmitting data in the network. Conventionally, this metric was derived from the exponential distribution. While appropriate to light-tailed traffic, the exponential model is inadequate in characterizing heavy-tailed traffic. In this paper, we propose a unified model to take both traffic types into account. The proposed model is based on the reliability analysis of series and parallel connections. Analysis is conducted with the compound Weibull distributions. Network performance is evaluated by the first and second moments.

1 Introduction

In *mobile ad hoc networks* (MANET), routing strategies based on the *source-initiated on-demand* (SIOD) approach [10] are more appropriate than the *table-driven* approach. The SIOD approach consists of two main phases: *route discovery* and *route maintenance*. One of the representative SIOD methods is *dynamic source routing* (DSR) [5]. For a particular *source-destination* (SD) pair in SIOD, the source node is responsible to find a *route* if it wants to send the packets to the destination node. Here the route is defined to be an alternating sequence of nodes and links, beginning with the source node and ending with the destination node. The route discovery phase is typically conducted by *broadcasting* (*flooding*) a probe packet to the whole network. Once a route is established, each packet will carry the route information in its header. Since the intermediate nodes do not have to keep the routing tables, the overhead in the table driven approach is eliminated. However, the route-discovery phase with broadcasting may take up a substantial amount of network bandwidth. Therefore, the frequency of route discovery needs to be reduced as much as possible. One of the representative methods is to explore multiple routes from a single broadcasting. This leads to the *multipath* routing paradigm [6]. In this paradigm, given a particular SD pair, there are a primary route and several secondary ones. Usually, the packet stream goes through the primary route. A secondary routes is put into use if the primary one was broken up. A new route discovery is needed only after all routes are broken.

In the MANET, a route can be broken for the following reasons: (a) A link disappears when the wireless signals fades away as the transmitter-receiver (T-R) distance increases or the interferences become dominant. (b) A link disappears when its end

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 314 – 323, 2005.

nodes quit their participation in networking. Cause (a) is a common problem in any wireless systems. It is typically analyzed by the space-propagation models for large-scale fading or the *Rayleigh* and *Ricean* distributions for small-scale fading. Cause (b) seems to be unique to MANET as it is directly related to the "ad hoc" nature of the participations. To develop analytical yet tractable models, some recent studies have tried to investigate causes (a) and (b) separately (e.g., [6], [8]). For the latter, a random variable X_{Li} is usually employed to describe the lifetime of a wireless link L_i $(i = 1,2,\cdots,n)$. For example, in the model presented in [6], X_{Li} is assumed to follow the *negative exponential distribution* (NED) and all X_{Li}'s are supposed to be *independently* and *identically distributed* (i.i.d.). The authors compared the multi-route approach with the single route approach in terms of route-discovery frequencies. The results, however, are mainly useful for voice traffic (rather than data traffic). The reason is that the model is based on the NED, and the NED is adopted to characterize voice traffic in circuit-switching networks. For data traffic in which burstiness inheres, models other than NED should be established. Here, we propose a unified model applicable to both voice and data traffic. Our model mainly characterizes cause (b). with moderate enhancement, however, it can also take cause (a) into account.

2 Background and Motivation

Mathematically, the distribution of a random variable is *heavy-tailed* if it decays slower than the NED. In this paper, we propose a Weibull-distribution model to evaluate MANET performance. Formally, a random variable X is said to follow the Weibull distribution if its *probability density function* (PDF) is:

$$f_X(x) = \frac{ax^{a-1}}{b}\exp\left(\frac{-x^a}{b}\right), \qquad (a,b,x>0)$$

where a is the *shape* parameter and b is the *location* parameter. Accordingly, the *cumulative distribution function* (CDF) and the kth-order moment respectively are:

$$F_X(x) = 1-\exp\left(\frac{-x^a}{b}\right), \qquad E\left(X^k\right) = b^{k/a}\Gamma\left(1+\frac{k}{a}\right),$$

where $\Gamma(\cdot)$ is the Gamma function. It is can be shown that the Weibull distribution is heavy-tailed if $a<1$, while it becomes the NED if $a=1$. It is more heavy-tailed (i.e., the tail becomes longer) as parameter a gets smaller. We adopt the Weibull distribution to conduct our analysis for the following reasons: (1) As a general function, it approximates voice, data, and wireless communications ([1], [3], [11]); (2) Its two-parameter structure provides the flexibility to characterize various types of traffic: (a) It is a member of the heavy-tailed distribution family with appropriate parameters; (b) It includes the exponential and Rayleigh distribution as a special case; (3) It will characterizes other traffic primitives such as the *Transmission Control Protocol* (TCP) inter-connection times (Chapter 15 in [7]) and may be used to enhance the reported work on TCP operated in MANET ([2], [4]); and (4) It is one of the most representative distributions in reliability analysis—a key issue in MANET.

3 The Compound Weibull Distributions

In MANET a route is defined as a series of multiple links. Thus a route fails when any one of these links breaks. Let X_{Li} $(i = 1,2,.....,n)$ be the lifetime of the *i*th link, then the critical lifetime of a route *P* with *n* links is simply that of its weakest link:

$$X_P = \min(X_{L1}, X_{L2},....., X_{Ln}).$$

Proposition: The life-time of a route, X_p, follows the Weibull distribution if all link life-times X_{Li} $(i = 1,2,.....,n)$ are i.i.d. Weibull random variables.

Proof: According to probability theory (e.g., [9]),

$$F_{Xp}(x) = P(X_P \le x) = 1 - \prod_{i=1}^{n}\left[1 - F_{X_{Li}}(x)\right] = 1 - \prod_{i=1}^{n}\exp\left(\frac{-x^a}{b}\right) = 1 - \exp\left(\frac{-nx^a}{b}\right). \quad (1)$$

The result is the Weibull CDF with location or scale parameter b/n and the same shape parameter *a* as appeared in the individual link random-variables. Q.E.D.

The mean and variance can readily be obtained from Eq. (1):

$$E_1(X_p) = \left(\frac{b}{n}\right)^{\frac{1}{a}}\Gamma\left(1+\frac{1}{a}\right) = \left(\frac{1}{\lambda}\right)^{\frac{1}{a}}\Gamma\left(1+\frac{1}{a}\right), \quad \sigma_1^2 = \left(\frac{1}{\lambda}\right)^{\frac{2}{a}}\left[\Gamma\left(1+\frac{2}{a}\right) - \Gamma^2\left(1+\frac{1}{a}\right)\right] \quad (2)$$

where $\lambda = n/b$ and the subscript "1" on the left-hand-side suggests that this is the *single-route* case.

3.1 The Parallel-of-Series (PoS) Compound Weibull Distribution

Now let us investigate the multi-route case. There are two basic types of topology: *parallel-of-series* (PoS) and *series-of-parallel* (SoP). The former represents a bank of serial routes laid side-by-side, while the latter suggests a route made up of redundant elements. The former is considered in this section. Suppose that there are *m* disjoint routes connecting an SD pair. Each disjointed route is a series of n_q $(q = 1,2,\cdots,m)$ links. In this configuration, a new route discovery is needed only after all *m* routes broke. As a result, the time between successive route discoveries *Z* is dependent upon the most robust path. In other words, no route discovery is needed until the "toughest" serial route breaks: $Z = \max(X_1, X_2,....., X_m)$.

Proposition: The PDF of *Z* takes the following form:

$$f_Z(z) = \left\{\prod_{p=1}^{m}\left[1 - \exp\left(-\lambda_p z^a\right)\right]\right\}\left(\frac{a}{b}\right)z^{a-1}\sum_{q=1}^{m}\frac{n_q \exp\left(-\lambda_q z^a\right)}{\left[1 - \exp\left(-\lambda_q z^a\right)\right]},$$

where $\lambda_p = n_p / b$ and $\lambda_q = n_q / b$.

Proof: According to probability theory:

$$F_Z(z) = \prod_{p=1}^{m} F_{Xp}(z) = \prod_{p=1}^{m} \left[1 - \exp\left(-\lambda_p x^a\right)\right] \tag{3}$$

$$f_Z(z) = \sum_{q=1}^{m} \left[\frac{f_{Xq}(z)}{F_{Xq}(z)} \prod_{p=1}^{m} F_{Xp}(z) \right] = \prod_{p=1}^{m} F_{Xp}(z) \left[\sum_{q=1}^{m} \frac{f_{Xq}(z)}{F_{Xq}(z)} \right]$$

$$= \left\{ \prod_{p=1}^{m} \left[1 - \exp\left(-n_p z^a / b\right)\right] \right\} \left\{ \left(\frac{a}{b}\right) z^{a-1} \sum_{q=1}^{m} \frac{n_q \exp\left(-n_q z^a / b\right)}{\left[1 - \exp\left(-n_q z^a / b\right)\right]} \right\}.$$

Q.E.D.

Definition: A random variable Z is said to follow the *parallel-of-series (PoS) compound Weibull distribution* if its CDF is given by Eq. (3).

In the case of $m = 2$, the time-between-discovery PDF becomes:

$$f_Z(z) = \left\{1 - \exp\left(-\lambda_1 z^a\right)\right\}\left\{1 - \exp\left(-\lambda_2 z^a\right)\right\}\left(\frac{a}{b}\right) z^{a-1} \left[\frac{n_1 \exp\left(-\lambda_1 z^a\right)}{1 - \exp\left(-\lambda_1 z^a\right)} + \frac{n_2 \exp\left(-\lambda_2 z^a\right)}{1 - \exp\left(-\lambda_2 z^a\right)} \right]$$

$$= a z^{a-1} [\lambda_1 \exp(-\lambda_1 z^a) + \lambda_2 \exp(-\lambda_2 z^a) - \lambda_{12} \exp(-\lambda_{12} z^a)].$$

where $\lambda_{12} = \lambda_1 + \lambda_2$.

3.2 The Series-of-Parallel (SoP) Compound Weibull Distribution

In the SoP configuration, we define a component as m parallel one-hop links that have the same end nodes. An SoP topology is referred to as a series of n such components. Accordingly, a component's life-time X_S is dependent on its most robust link: $X_S = \max(X_{L1}, X_{L2}, ..., X_{Lm})$. According to probabilistic mathematics we have:

$$F_{Xs}(x) = P(X_s \le x) = \prod_{j=1}^{m} F_{XLj}(x) = \left[1 - \exp\left(\frac{-x^a}{b}\right)\right]^m.$$

In general, the components are allowed to have different number of parallel links. Specifically, for a series of n components, the ith component includes m_i parallel links, where m_i is not necessarily the same as m_j. Considering the resultant life-time Z of a series of n components, the most vulnerable component dictates: $Z = \min(X_1, X_2, ..., X_n)$, hence

$$F_Z(z) = P(Z \le z) = 1 - \prod_{i=1}^{n} \left[1 - F_{Xs}(x)\right] = 1 - \prod_{i=1}^{n} \left\{1 - \left[1 - \exp\left(\frac{-x^a}{b}\right)\right]^{m_i}\right\}. \tag{4}$$

Definition: A random variable *Z* is said to follow the *series-of-parallel* (SoP) *compound Weibull distribution* if its CDF is given by Eq. (4).

It can be shown that the mean of random variable Z, $E(Z)$, of both PoS and SoP networks is of a closed-form, including a Gamma function. The derivation and the resulted expressions, however, are somewhat tedious. In the following, we consider two special cases to gain more insights.

4 Case Study 1

Consider a case of the PoS topology in which all m routes have the same length, i.e., $n_1 = n_2 = \cdots = n_m = n$. From Eq. (4), it can be shown that:

$$f_Z(z) = \left(\frac{a}{b}\right) z^{a-1} \left\{ \prod_{p=1}^{m} [1 - \exp(-nz^a / b)] \right\} \sum_{q=1}^{m} \frac{n \exp(-nz^a / b)}{[1 - \exp(-nz^a / b)]}$$

$$= (a\lambda z^{a-1})\, m \exp(-\lambda z^a) \left\{ \sum_{q=0}^{m-1} \binom{m-1}{q} [-\exp(-\lambda z^a)]^{m-1-q} \right\},$$

where $\lambda = n / b$. Note that we used the binomial theorem to obtain the last line above. Consequently, the mean is:

$$E_m(Z) = \int_0^\infty z\, f_Z(z) dz = a\lambda m \int_0^\infty z^a \exp(-\lambda z^a) \left\{ \sum_{q=0}^{m-1} \binom{m-1}{q} [-\exp(-\lambda z^a)]^{m-1-q} \right\} dz$$

$$= \frac{m}{\lambda^{1/a}} \left[\sum_{q=0}^{m-1} \binom{m-1}{q} \frac{(-1)^{m-1-q}}{(m-q)^{1+1/a}} \right] \Gamma\left(1 + \frac{1}{a}\right).$$

Here subscript m on the left hand side highlights the m-route paradigm. Furthermore, the variance is:

$$\sigma_m^2 = \frac{m}{\lambda^{2/a}} \left[\sum_{q=0}^{m-1} \binom{m-1}{q} \frac{(-1)^{m-1-q}}{(m-q)^{1+2/a}} \right] \Gamma\left(1 + \frac{2}{a}\right) - \frac{m^2}{\lambda^{2/a}} \left[\sum_{q=0}^{m-1} \binom{m-1}{q} \frac{(-1)^{m-1-q}}{(m-q)^{1+1/a}} \right]^2 \Gamma^2\left(1 + \frac{1}{a}\right).$$

In order to estimate the relative merit of the multi-route connection over single-route, we first evaluate the ratio of $E_m(Z)$ to $E_1(Z)$, as shown respectively in Eq. (2) and the $E_m(Z)$ equation above:

$$h(a,m) = \frac{E_m(Z)}{E_1(Z)} = m \sum_{q=0}^{m-1} \binom{m-1}{q} \frac{(-1)^{m-1-q}}{(m-q)^{1+1/a}}.$$

By definition, this ratio is bigger than unity. It is important to recognize that this ratio is independent on route length n. Note that h represents the normalized average time between successive route discoveries. Therefore, the larger h values should be sought whenever possible. The numerical profile of five instances for m = 2 to 6 are illustrated in Figure 1.

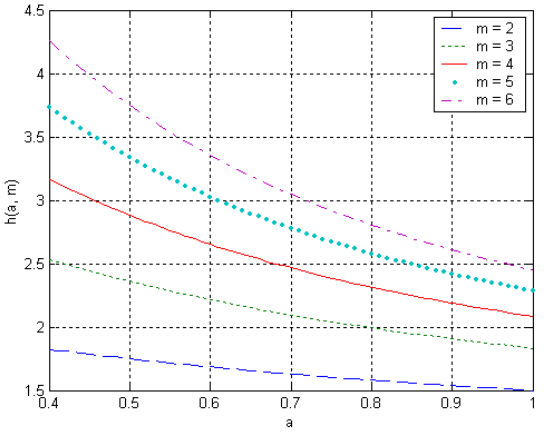


Fig. 1. The mean ratio of *m*-route to 1-route (with identical route lengths)

The merit of employing m routes with an identical length can easily be observed from Figure 1. However, the merit tends to diminish when the shape parameter a gets larger, assuming its smallest gain when the Weibull degenerates into an exponential distribution (a = 1). For example, the relative performance gain for m = 6 is about 4.25 when $a = 0.4$ and less than 3.0 when $a > 0.75$. In other words, the longer the tail, the better. So far we have focused on comparing the average values, i.e., the first-order moments. In order to characterize the deviations between the single route and m-route configurations, we evaluate the ratio of their variances:

$$r(a,m) = \frac{\sigma_m^2}{\sigma_1^2} = m\left\{\Gamma\left(1+\frac{2}{a}\right)\left[\sum_{q=0}^{m-1}\binom{m-1}{q}\frac{(-1)^{m-1-q}}{(m-q)^{1+2/a}}\right] - m\Gamma^2\left(1+\frac{1}{a}\right)\left[\sum_{q=0}^{m-1}\binom{m-1}{q}\frac{(-1)^{m-1-q}}{(m-q)^{1+1/a}}\right]^2\right\}\left[\Gamma\left(1+\frac{2}{a}\right)-\Gamma^2\left(1+\frac{1}{a}\right)\right]^{-1}.$$

The profile of the ratio r vs. a is illustrated in Figure 2. Unlike the ratio for the first moment, here the smaller r values should be sought. It is clear from Figure 2 that the m-route approach is unfavorable in terms of the variances, as the ratio is consistently greater than unity. The situation gets worse as parameter a decreases. Recall that a smaller a corresponds to a heavier tail. This observation is consistent with the original characteristics of the heavy-tail distributions: their variances are usually large.

Since the average performance is opposite to the variation in performance, we wish to evaluate their joint effects. The ratio of the mean to the standard deviation is used for this purpose: $w(a,m) = E_m(Z)/\sigma_m$. The profile of w vs. a is illustrated in Figure 3. It is noted that the merit of employing m routes becomes clear—for a fixed m, the standard deviation decreases faster than the mean as a increases. It is more important to watch the mean in comparison to the variance. Thus the gain in average performance outweighs the cost of service variation.

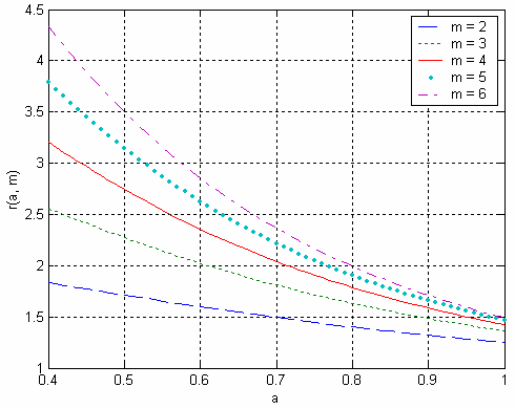


Fig. 2. The variance ratio of *m*-route to 1-route (with identical route lengths)

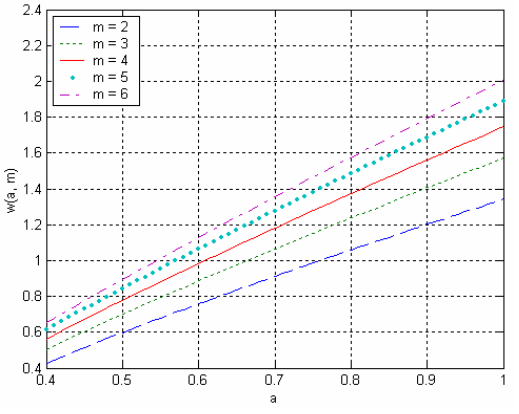


Fig. 3. The ratio of mean to deviation (identical route length)

5 Case Study 2

Now consider anther case of the PoS topology. In this case, we investigate two routes (i.e., $m = 2$) where the secondary route is longer than the primary, i.e., $n_2 > n_1$. It follows from the general PDF expression that for $m = 2$:

$$f_Z(z) = az^{a-1}\{\lambda_1 \exp(-\lambda_1 z^a) + \lambda_2 \exp(-\lambda_2 z^a) - (\lambda_1 + \lambda_2)\exp[-(\lambda_1 + \lambda_2)z^a]\},$$

$$E_2(Z) = \left[\left(\frac{1}{\lambda_1}\right)^{\frac{1}{a}} + \left(\frac{1}{\lambda_2}\right)^{\frac{1}{a}} - \left(\frac{1}{\lambda_1 + \lambda_2}\right)^{\frac{1}{a}}\right]\Gamma\left(1 + \frac{1}{a}\right),$$

$$\sigma_2^2 = \left[\left(\frac{1}{\lambda_1}\right)^{\frac{2}{a}} + \left(\frac{1}{\lambda_2}\right)^{\frac{2}{a}} - \left(\frac{1}{\lambda_1 + \lambda_2}\right)^{\frac{2}{a}}\right]\Gamma\left(1 + \frac{2}{a}\right) - \left[\left(\frac{1}{\lambda_1}\right)^{\frac{1}{a}} + \left(\frac{1}{\lambda_2}\right)^{\frac{1}{a}} - \left(\frac{1}{\lambda_1 + \lambda_2}\right)^{\frac{1}{a}}\right]^2 \Gamma^2\left(1 + \frac{1}{a}\right),$$

where subscript "2" on the left-hand-side highlights the double-route case. To estimate the relative merits of the double-route connections, we evaluate the ratio of $E_2(Z)$ to $E_1(Z)$:

$$h(a,u) = \frac{E_2(Z)}{E_1(Z)} = 1 + \left(\frac{\lambda_1}{\lambda_2}\right)^{\frac{1}{a}} - \left(\frac{1}{1+\lambda_2/\lambda_1}\right)^{\frac{1}{a}} = 1 + \left(\frac{1}{u}\right)^{\frac{1}{a}} - \left(\frac{1}{1+u}\right)^{\frac{1}{a}}, \tag{5}$$

where $u = \lambda_2/\lambda_1 = n_2/n_1 > 1$, or the secondary route is longer than the primary route. Note that u can be interpreted as the normalized length of the secondary route. To show the effect of route length, the profile of h vs. u is illustrated in Figure 4.

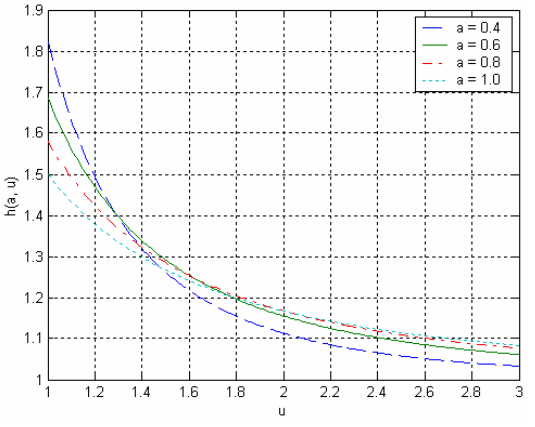


Fig. 4. The mean ratio of 2-route to 1-route (varying route lengths)

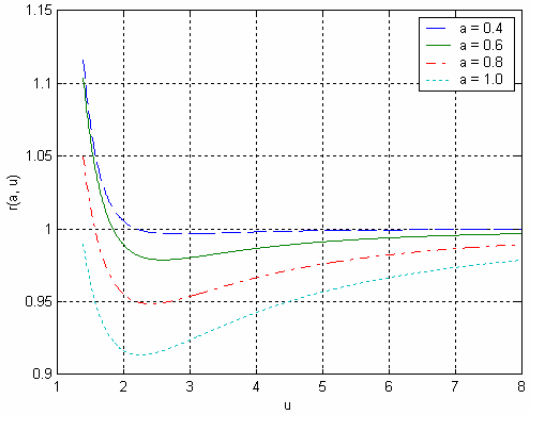


Fig. 5. The variance ratio of 2-route to 1-route (varying route lengths)

It is observed there is little advantage in employing two routes when the secondary route is a lot longer. Actually, the two-route approach contributes only marginally

when $u > 2.5$. These results suggest that, in the MANET, it is only worthwhile to introduce a second route of comparable length. The above represents average performance comparisons. In order to characterize variation in performance, we evaluate the ratio of their variances:

$$r(a,u) = \frac{\sigma_2^2}{\sigma_1^2} = \left\{ \left[1 + \left(\frac{1}{u}\right)^{\frac{2}{a}} - \left(\frac{1}{1+u}\right)^{\frac{2}{a}} \right] \Gamma\left(1+\frac{2}{a}\right) - \left[1 + \left(\frac{1}{u}\right)^{\frac{1}{a}} - \left(\frac{1}{1+u}\right)^{\frac{1}{a}} \right]^2 \Gamma^2\left(1+\frac{1}{a}\right) \right\} \times \left[\Gamma\left(1+\frac{2}{a}\right) - \Gamma^2\left(1+\frac{1}{a}\right) \right]^{-1}.$$

To show the effect of route length, the profile of r vs. u is illustrated in Figure 5. Unlike average performance, here the smaller r values are preferable, justifying the secondary route. Clearly, when $u > 1.5$ the advantages are clear—the r values are less than 1.1. In other words, a long secondary route results in performance variations comparable to the primary route. A distinct characteristic of r is that it is not monotonic. For instance, a minimum of $r \approx 0.95$ is found at $u \approx 2.4$ for $a = 0.8$. Put it in other way, if the length of the second route is two to three times longer than the primary route, the performance variation of the 2-route case is at its smallest. Furthermore, it is observed from Figure 5 that, as the shape parameter a gets larger, this minimum shifts left. When the Weibull degenerates into an exponential function, for example, the ratio r is at its smallest: 0.92, which occurs when $u = 2.1$. Thus the local convexity of r serves to find the optimal length of the secondary route.

6 Conclusions

Toward a unified framework, the proposed model takes the heavy-tailed traffic into account, extending the conventional model that mainly considers light-tailed traffic. Built upon the Weibull distribution, the model encompasses both heavy-tailed and light-tailed traffic. With the established model, the multi-route approach—both in SoP and PoS networks—has been compared with the single-route approach. Although the former is generally better than the latter, it seems neither necessary to introduce a great number of secondary routes, nor a few long secondary routes. The above conclusions are applicable to both bursty data-traffic (with the Weibull shape parameter $a < 1$) and conventional voice traffic (with the shape parameter $a \approx 1$).

References

1. Alouini, M., Simon, M.: Performance of Generalized Selection Combining over Weibull Fading Channels. Proc. IEEE Vehicular Technology Conference, Vol. 3 (2001) 1735–1739
2. Choi J., Yoo C.: TCP-Aware Source Routing in Mobile Ad Hoc Networks. Proc. IEEE International Symposium on Computers and Communication (2003) 69-75

3. Chuah C., Katz R.: Characterizing Packet Audio Streams from Internet Multimedia Applications. Proc. IEEE ICC, Vol. 2 (2002) 1199 –1203
4. Holland G., Vaidya, N.: Impact of Routing and Link Layers on TCP Performance in Mobile Ad Hoc Networks. Proc. WCNC, Vol. 3 (1999) 1323–1327
5. Johnson D. and Maltz D.: Dynamic Source Routing in Ad Hoc Wireless Networks. Mobile Computing, edited by T. Imielinski and H. Korth, Kluwer Academic Publishers (1996) 153-181
6. Nasipuri, A., Das, S.: On-Demand Multipath Routing for Mobile Ad Hoc Networks. Proc. IEEE ICCCN (1999) 64-70
7. Park, K., Willinger W. (eds.): Self-Similar Network Traffic and Performance Evaluation. Wiley, NY (2000)
8. Pham P., Perreau, S.: Performance Analysis of Reactive Shortest Single-Path and Multi-Path Routing Mechanism with Load Balance. Proc. IEEE INFOCOM (2003)
9. Ross, S.: Introduction to Probability Models. 4th Ed. Academic Press, CA (1989)
10. Toh, C.: Ad Hoc Mobile Wireless Networks: Protocols and Systems. Prentice Hall, NJ (2001)
11. Tzeremes G., Christodoulou, C.: Use of Weibull Distribution for Describing Outdoor Multipath Fading. Proc. IEEE Antennas and Propagation Society International Symposium, Vol. 1 (2002) 232 –235

A Load-Balancing Control Method Considering Energy Consumption Rate in Ad-Hoc Networks

Hyun Kyung Cho[1], Eun Seok Kim[2], and Dae-Wook Kang[1]

[1] Department of Computer Science, Chonnam National University,
300 Yongbong-dong Buk-gu, Kwangju 500-757, Republic of Korea
{gscho,dwkang}@chonnam.chonnam.ac.kr
[2] Department of Multimedia Contents, Dongshin University,
252 Daeho-dong, Naju, Jeollanamdo 520-714, Republic of Korea
Tel:+82-61-330-3456, FAX:+82-61-330-3452
eskim@dsu.ac.kr

Abstract. The existing on-demand routing protocol in Ad-Hoc network is suitable to the wireless network, which has frequent movement of nodes. But, traffic is concentrated into the particular node where the mobility of the nodes is low because it cannot find new route till the network topology alters (even when the acting route is complicated). Besides, when network is stable, data is transmitted after choosing shortest path without any consideration of any particular node's traffic, and then traffic is concentrated into a particular node, which raises the problem of delay of transmission and huge energy consuming. We suggest a load-balanced routing method, which considers energy consumption. Our method improves the function of route discovery by adding energy factor to the existing DSR(Dynamic Source Routing).

1 Introduction

Ad-Hoc network is a collection of wireless mobile nodes forming a temporary network without the aid of any established infrastructure or centralized administration. The topology of connections between nodes in Ad-Hoc network may be quite dynamic. Ad-Hoc networks require a highly adaptive routing scheme to deal with the frequent topology changes[1].

Ad-Hoc network has the characteristics that all nodes move spontaneously to arbitrary positions by the time. Therefore, there exist technical difficulties in route discovery and route maintenance. All nodes in network spend additional energy because they perform routing or hosting functions as well as the existing transmitting functions. This can directly affect the lifespan of Ad-Hoc network, which raises many problems that have to be resolved for the practical use. Ad-Hoc network establishment necessarily requires minimizing of the energy consumption and performing the optimal route discovery and maintenance. That is to say, with the minimization of the energy consumption, network lifespan can be elongated. Route will be efficiently maintained by choosing the optimal route, and higher rate of packet transmission can be achieved by the minimization of the link[2,8]. To solve these problems, various types of algorithm are being suggested on which the characteristics

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 324 – 333, 2005.

of network are well reflected[8,6,7,10]. We are here proposing upgraded protocol, which is advanced more properly than existing routing protocols.

This paper is organized as follows. In section 2, we describe the characteristics of Ad-Hoc network and problems. Section 3 provides considerable insight into ECLB which is the suggested method, and experimental results are presented in section 4. Finally, we induce the conclusion and propose the future research works.

2 Previous Work

2.1 The Characteristics of Ad-Hoc Networks and Problems

Ad-Hoc network is a temporary network that consists of mobile nodes that can communicate with each other without relying on any infrastructure.

The wireless Ad-Hoc network has features as follows:

Firstly, Mobile nodes, which use wireless interfaces, are restricted by a distance between them as their data transmission rate gets lower in proportion to it. Secondly, as the node moves, the network topology also shows its dynamic change. Lastly, mobile nodes have their limitation in their energy supplies because of utilizing capacity-limited batteries.

Ad-Hoc network requires offering communication services constantly regardless of the topology changes that are induced by its frequent changing. Effective algorithm and mechanism are also required to prohibit the nodes in the network from consuming of resources excessively[1,2]. Therefore, it is necessary to research on the routing protocol, which can minimize control packet overload and energy consumption in its route discovery process resulting in improving the network efficiency.

Ad-Hoc network is recommended to uses limited amount of energy to support most of the node mobility. The energy consumption that determines the efficiency of Ad-Hoc network occurs in dealing with data, transmitting various control messages, and communicating[9,11,12,13]. That is, to improve the efficiency of whole the network, it is necessary to design energy-concerning protocol.

Of the route within Ad-Hoc network, route discovery and recovery process frequently occurs because of the dynamic change of topology of nodes. If the required time can be shortened, it is possible not only to cope promptly with the topology change of nodes but also to transmit data as fast as it can, and the reliability of network can also be heightened by reducing missing data during the process of route discovery and route recovery. DSR, one of the on-demand method routing protocol, can shorten the required time to spend in reforming since it maintains Route Cache of all nodes within network, beginning investigation for route reformation not from source node but from the node in which error occurs[4]. In this paper, more efficient a DSR-based routing method is suggested.

2.2 Energy Conservation and Load-Balancing

The question is that the Ad-Hoc network generates control traffic overhead or causes transmission delay since network topology changes dynamically and it also performs route rediscovery when route is cut off because of single path in routing protocol.

In existing on-demand routing method, message transmissions occur after forming the optimal route. Successive message transmissions, however, occur with particular nodes acting as routers when the network topology alteration is small.

As a result, excessive traffic makes transmission delay and excels the energy consumption in the node used as a router, which means that most of energy is spent on the routing function. As it were, traffics are concentrated into a particular node when the mobility of nodes is low[1,2].

A number of routing proposals for ad hoc networks took energy conservation into consideration so as to extend the lifetime of the wireless nodes by wisely using their battery capacity [7,10]. Minimum battery cost routing (MBCR) [7] utilizes the sum of the inverse of the battery capacity for all intermediate nodes as the metric upon which the route is picked. However, since the summation must be minimal, some hosts may be overused because a route containing nodes with little remaining battery capacity may still be selected. Min-max battery cost routing (MMBCR) [7] treats nodes more fairly from the standpoint of their remaining battery capacity. Smaller remaining battery capacity nodes are avoided and ones with larger battery capacity are favored when a route is chosen. However, more overall energy will be consumed throughout the network since minimum total transmission power routes are no longer favored. In [10], MTTP is used when all the nodes forming a path (note that one path is sufficient) have remaining battery capacity that is called battery protection threshold, and MMBCR is used if no such path exists. The combined protocol is called conditional max-min battery capacity routing(CMMBCR).

In existing on-demand routing method transmission of message occurs right after the formation of an optimal route from source node to destination node. However, if it is in stable condition that network topology change is relatively small, it successively transmits messages with a particular node in the route acting as a router.

Consequently, excessive traffic causes transmission delay, increases the energy consumption to router, and makes most of energy wasted in acting route functions. In low mobility of node, traffics are concentrated into a particular node[1,2]. To solve this problem, SLAP(Simple Load-balancing Ad-hoc routing Protocol) and LBAR(Load-Balancing wireless Ad-hoc Routing) method are suggested.

In SLAP[5], a node judges that excessive traffic is concentrated on it when the traffic amount reaches to a upper threshold value. And then, it avoids participating in the route by transmitting GIVE_UP messages. However, if the upper threshold is set high, SLAP is similar to AODV or DSR. On the other hand, if the upper threshold is set low, GIVE_UP messages are highly transmitted. LBAR[6] is a routing protocol that finds the route with the minimum traffic load considering load balancing under the Ad-Hoc network circumstance. Since LBAR uses traffic load information not only from its own node but also from neighbor nodes, additional overhead occurs for the regular collection of the information. Also, in case of disconnection of linkage, this protocol opts for an alternative bypass based on the route information collected in the past. Therefore, the probability of errors becomes higher in wireless Ad-Hoc network circumstance where network topology (the nodes) changes frequently.

3 ECLB (Energy Consumption Based Load Balancing Method)

For solving the problems mentioned above, we propose ECLB (Energy Consumption Load Balancing), a routing method that concerns energy consumption rate. ECLB makes balanced energy consumption available by calculating energy consumption rate of each node and choosing alternative route accordingly in order to exclude the overburdened-traffic-conditioned node in route discovery. The point is that not only main path but also alternative path can be formed on the basis of the measure energy consumption rate using present packet amount per unit time and mean packet throughput of the past. By forming route in advance and conversing into preformed alternative path when route impediment occurs, transmission for route rediscovery and control traffic overhead can be decreased.

In other words, when main path cannot be restored because of cut-off, data are transmitted through existing alternative path without re-performing source-initial route discovery. When network topology is relatively stable, the energy-deficient nodes are included in the routing path, which could shorten the lifespan of whole network. To solve this problem, balancing energy consumption algorithm based on DSR is suggested in which a few parameters and several simple functions are added.

To be formed inversely proportionate to packet throughput, energy threshold is calculated as follows:

$$Th_0 = E - \alpha \tag{1}$$

$$Th_{t+\Delta t} = Th_t \cdot \left(1 - \frac{P}{MaxP}\right) \cdot k \tag{2}$$

According to the control coefficient α, initial threshold Th_0 is established as follows to have the value little smaller than E. E refers the initial energy value of each node.

The threshold of $t + \Delta t$ can be calculated as formula (2), where P refers the packet numbers treated until time t, $MaxP$ is the experientially gained mean value of maximum packet throughput of Ad-Hoc network which has the similar circumstance to the present network. In formula (2), k is the control coefficient to accelerate the adjustment of threshold. When the control of threshold is slow, lowering k can accelerate adjustment of threshold. The calculated threshold would be renewed every Δt and transmit through the present formed route.

The method of multipath formation and route formation using transmitted threshold is as follows:

Every node adds energy_remainder which is the surplus energy storing parameter. energy_remainder is initialized as E which is the initial energy value of each node, and decreased according to the packet throughput. In each packet, energy_packet for transmitting energy_threshold has been taken into consideration in addition. In source node, energy_packet is to be set as Th_0. energy_threshold can be calculated according to formula (2). When source node generates RREQ packet, energy_packet in packet is to be set as energy_threshold of itself. When intermediate nodes receive

RREQ, they determine whether it would participate in routing or not by comparing its own energy_remainder with energy_packet of packet.

That is, when energy_remainder is larger than energy_packet, like existing DSR, RREQ is to be broadcast. But if energy_remainder is smaller than energy_packet, then RREQ is discarded and it makes the node not participate in routing. When data packet is being transmitted, intermediate node chooses appropriate alternative path by sending RERR to source node to find that its own energy_remainder is smaller than the energy_packet.

ECLB Algorithm : Energy Consumption based Load Balancing Method.

For node N

```
energy_remainder = E;
when receives a packet {
   if( RREQ packet ) {
         if( packet's RREQ ID != RREQ ID in node cache ){
          if( addresses in RREQ's route record !=
                                                 node's address ){
             if( RREQ's destination address != node's address ){
                if( energy_remainder > energy_packet in RREQ ){
                   attaches to node's address in route record;
                   broadcast the network;
                }else{
                   discard packet;
                }
             }else{ //destination node
                send RREP;
             }
          }else{ // already included in the path
             discard packet;
          }
       }else{ // already received RREQ
          discard packet;
       }
   }else if (RREP packet ){
      if( node N is the source node ){
         select route;
      }else{
         forward a packet to the source node;
      }
   }else if( ERROR packet ){
      if( node N is the source node ){
         if( the source node needs the route ){
            if( a alternative path exists ){
               select route;
            }else{
               energy_packet = energy_Threshold;
               initiate the route discovery;
            }
         }
      }else{
         remove error node's address in route cache;
         forward error packet to the source node;
      }
   }else{
      if( energy_remainder < energy_packet ){
         forward RERR packet to the source node;
      }else{
         process the packet using the underlying
              routing protocol;
      }
   }
   reduce energy_remainder;
}
```

4 Performance Evaluation of ECLB Routing

We have constructed a packet-level simulator that allows us to observe and measure the protocol's performance under a variety of conditions. The model is similar to that in [14]. Our simulations are run using ad hoc networks of 50 nodes under a nominal bit rate of 2 Mbps. Mobile terminals move with a speed that is uniformly distributed between 0 and 20 m/sec. In addition, mobility is varied by means of varying the pause/rest period. For every variation of the traffic sources, the experiments are run for a set of pause periods. The smaller the pause period, the higher the mobility, and, the greater the pause period, the lower the mobility. This implies that varying the length of the pause period is equivalent to varying the mobility model. Each and every mobile node alternately rests and moves to a new random location within the rectangular grid.

Experiments were run for pause period of 0, 10, 20, 40 and 100 seconds in case of 50 nodes. Mobile nodes can communicate only within a constant range of 200m. The experiments use different number of sources with a moderate packet rate and changing pause times. We use 10, 20, 30 and 40 traffic sources and a packet rate of 4 packets/sec. Mobile nodes are free to move in a 500m x 500m topology boundary and simulation time of 100 sec.

The experiments were run for two different initial energy of node : 5 and 10. And the energy values spent when nodes receive and transmit the packets are set to 0.3 and 0.4 respectively.

4.1 Performance Metrics

Three important performance metrics are evaluated:

Packet delivery fraction – The ratio of the data packets delivered to the destinations to those generated by the CBR sources.

Average end-to-end delay of data packets – This includes all possible delays caused by buffering during route discovery latency, queuing at the interface queue, retransmission delays at the MAC, and propagation and transfer times.

Normalized routing load – The number of routing packets transmitted per data packet delivered at the destination. Each hop-wise transmission of a routing packet is counted as on transmission.

4.2 Simulation Results

Figures 1 and 2 show the packet delivery fractions for variations of the pause time for ECLB, AODV, and DSR. Note that the packet delivery fractions for ECLB, AODV, and DSR are very similar for both 10 and 20 sources. With 30 and 40 sources, however, ECLB outperforms AODV and DSR. In fact, ECLB achieves the highest packet delivery fraction for all pause time values. For 30 sources, ECLB achieves up to 20% higher packet delivery fractions than both AODV information that is stored in destination node to provide aid in routing of route discovery. Similarly, ECLB has superior performance to both AODV and DSR in the case of 40 sources, in terms of the packet delivery fraction.

Especially, in case of pause period value of 100secs, which has low node mobility, ECLB shows a better efficient performance twice as high as the DSR does. Furthermore, the less the value of initial energy is, generally the better the performance is. Through the simulation, we got the result that where the value of initial energy is high (bigger than 40), ECLB has almost the same performance as DSR. Also, ECLB has a better average end-to-end delay than both AODV and DSR(see Figure 3 and 4). For 30 and 40 sources, ECLB achieves significantly lower delay than AODV and DSR. Moreover, the delays decrease with lower mobility for ECLB in all four cases while it increase with 30 and 40 sources for both AODV and DSR. This is due to a high level of network congestion and multiple access interference in certain regions of the ad hoc network.

The routing load results see Figures 5 and 6, show that the routing load of all three protocols increases with increasing the number of sources. This is because the increase in the number of source nodes causes a greater number of request message flooding. ECLB demonstrates a lower routing load than both AODV and DSR. In summary, ECLB outperformed the AODV and DSR. ECLB achieves a higher packet delivery fraction, a lower average end-to-end delay, and a lower normalized routing load.

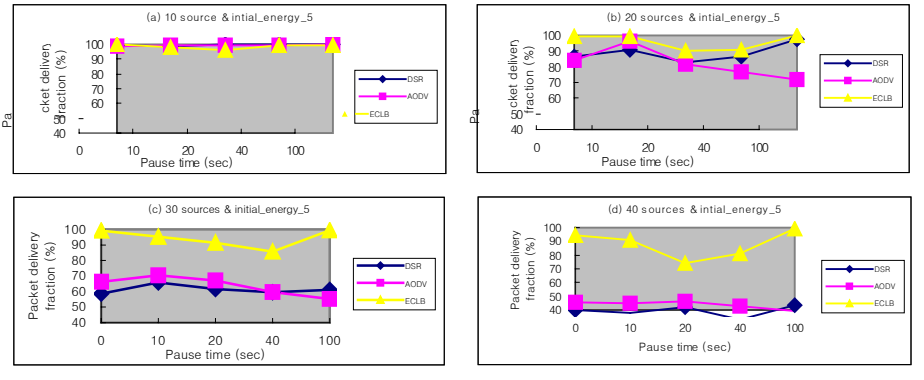


Fig. 1. Packet delivery fraction (Initial Energy = 5)

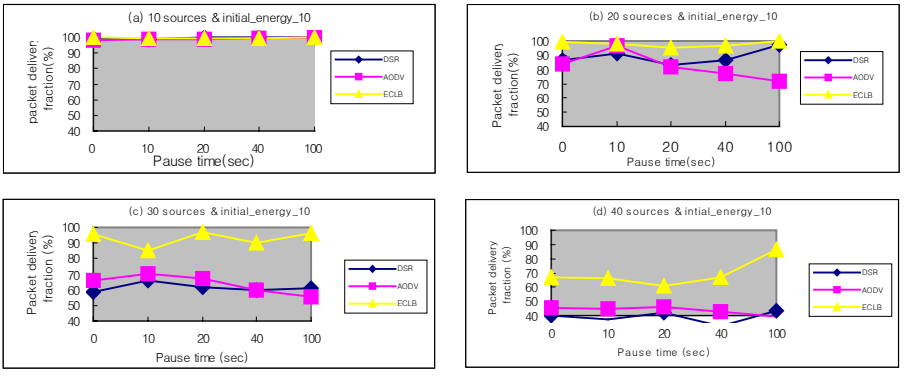


Fig. 2. Packet delivery fraction (Initial Energy = 10)

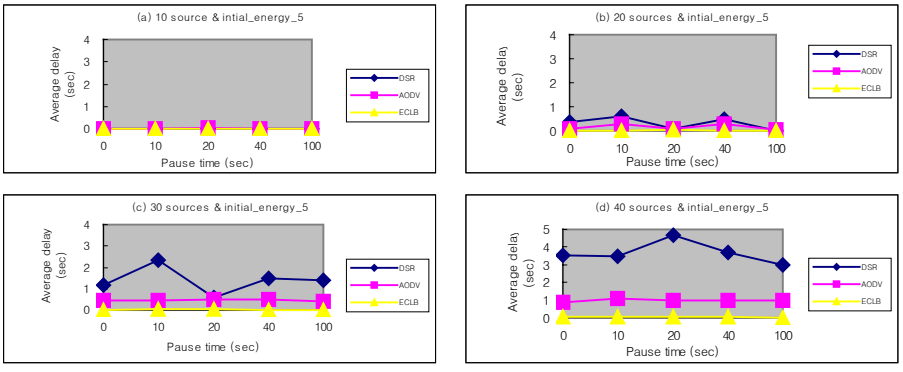


Fig. 3. Average end-to-end delay of data packets (Initial Energy = 5)

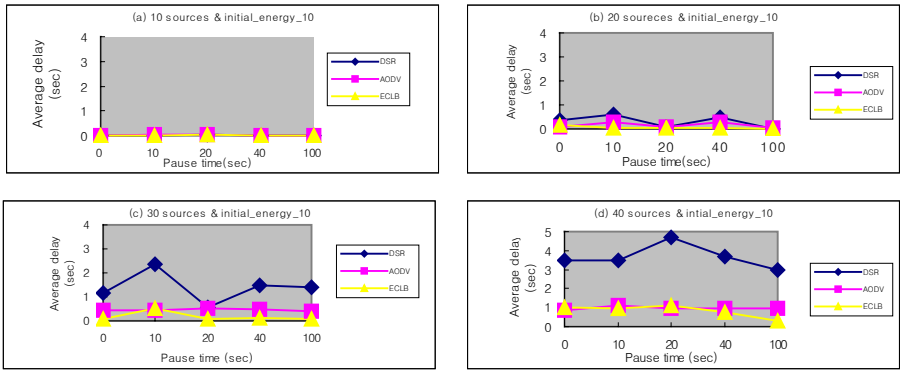


Fig. 4. Average end-to-end delay of data packets (Initial Energy = 10)

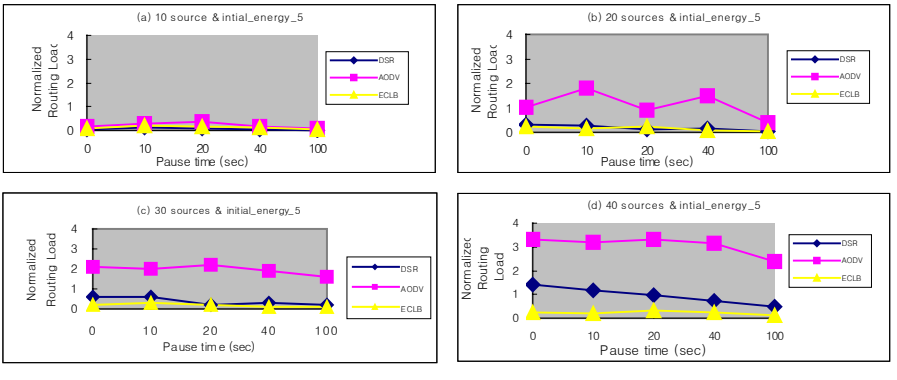


Fig. 5. Normalized routing load (Initial Energy = 5)

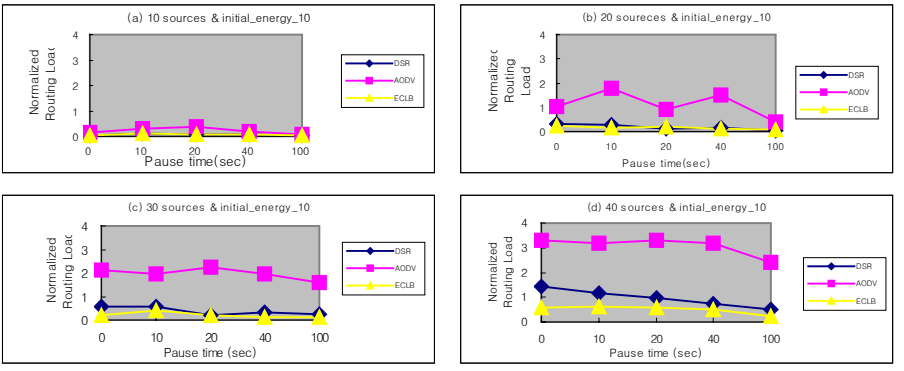


Fig. 6. Normalized routing load (Initial Energy = 10)

5 Conclusions and Future Work

In this paper, we proposed a novel on-demand routing scheme, namely the Energy Consumption Load Balancing (ECLB) protocol. In ECLB, routing policy concerning the energy efficiency on the basis of DSR has been proposed. The higher efficiency of packet delivery could be achieved by determining the participation on routing according to the present energy leftover, because the excessive energy consumption of particular node is avoided in Ad-Hoc networks of low node mobility.

The performance of the proposed ECLB protocol has been studied through a simulation study. Simulation results have clearly shown the advantages of ECLB over DSR and AODV in terms of packet delivery fraction. The simulation is performed with NS-2 version of 2.26, the proposed method achieved the double efficiency performance of DSR, in relatively stable Ad-Hoc network, which is composed with many nodes and has low mobility of the nodes. When the initial energy is very low, generally much better efficiency has been achieved, which let us expect that it will show the better efficiency in necessary application using of terminal node with low electricity. The future work of this study will be additional model for a performance improvement on the basis of load-balancing in the networking environment of higher power level.

References

1. C-K Toh, "*Ad Hoc Mobile Wireless Networks protocols and systems*", Prentice Hall PTR, pp. 13-25, 2002.
2. E. M. Royer, C-K Toh, "A review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks", IEEE Personal Communications, pp. 46-55, April 1999.
3. Charles E. Perkins, Elizabeth M. Royer, and Samir R. Das, "Ad Hoc On-demand Distance Vector Routing", IETF Draft, Oct. 1999.
4. D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks", Mobile Computing, Chapter 5, pp. 153-181, Kluwer Academic Publishers, 1996.

5. Sanghyun Anh, Yujin Lim, and Kyoungchun Kim, "An Ad-hoc Routing Protocol for Load Balancing : Simple Load-balanced Ad hoc routing Protocol", Proceeding of KISS, April 2002.
6. Hossam Hassanein and Audrey Zhou, "Routing with load Balancing in wireless ad hoc network", Proc. the 4th ACM international workshop on modeling, analysis and simulation of wireless and mobile systems, pp. 89-96, July 2001.
7. S. Singh, M. Woo, and C. Raghavendra, "Power-aware routing in mobile ad hoc networks", Proceedings of Mobicom '98, pp. 181-190.
8. Mehran Abolhasan, Tadeusz Wysocki, and Eryk Dutkiewicz, "A review of routing protocols for mobile ad hoc networks", Telecommunication and Information Research Institute, March 2003.
9. Ahmed Safwat, Hossam Hassanein, and Hussein Mouftah, "Energy-Aware Routing in MANETs : Analysis and Enhancements", Proceedings of MSWiM '02, pp. 46-53, September 2002.
10. C. K. Toh, "Maximum battery Life Routing to Support Ubiquitous Mobile Computing in Wireless Ad hoc networks", IEEE Communication Magazine, June 2001.
11. J-H chang and L. Tassiulas, "Energy conserving routing in wireless ad-hoc networks", Proceeding of INFOCOM 2001.
12. C. Schurgers and M. B. Srivastava, "Energy efficient routing in wireless sensor networks", IEEE Wireless Communications and Networking Conference, 2002.
13. M. Maleki, K. Dantu, and M. Pedram, "Power-aware Source routing in Mobile Ad hoc networks", Proceedings of ISLPED '02, Monterey, CA.

Efficient Node Forwarding Strategies via Non-cooperative Game for Wireless Ad Hoc Networks

Mingmei Li[1], Eiji Kamioka[2], Shigeki Yamada[2], and Yang Cui[3]

[1] National Institute of Informatics,
The Graduate University for Advanced Studies,
2-1-2 Hitotsubashi, Chiyoda Ku, 101-8430, Tokyo, Japan
amynana@grad.nii.ac.jp
[2] National Institute of Informatics,
2-1-2 Hitotsubashi, Chiyoda Ku, 101-8430, Tokyo, Japan
{kamioka,shigeki}@nii.ac.jp
[3] Institute of Industrial Science, University of Tokyo,
4-6-1 Komaba, Meguro, 153-8505, Tokyo, Japan
cuiyang@imailab.iis.u-tokyo.ac.jp

Abstract. In multi-hop ad hoc networks, communications rely heavily on cooperation of each node. Albeit a good teamwork will run the wireless networks well, some selfish behaviors could definitely break them down. This paper examines the theoretical aspect of selfish nodes through a non-cooperative game framework. Depending on the tradeoff between the nodes packet generating requirements and forwarding preferences, we introduce a unique "cost and compensation" scheme: the nodes first select their initial packet generating rates, in order to attain their desired values, they adjust the rates according to the associated cost reflected by network status; and they are also compensated once they forward packets for other nodes. We then propose a distributed algorithm to achieve optimal point for individual node–Nash Equilibrium(NE). Finally, simulation results show that proposed scheme is effective to enforce the potentially selfish nodes to co-operate.

Keywords: Ad Hoc Networks, Non-cooperative Game, Nash Equilibrium(NE).

1 Introduction

Wireless Ad Hoc networks are growing increasingly due to the fact that they offer unique benefits for certain applications. Wireless ad hoc network tries to pull all the nodes participation in network function, but some nodes use a probabilistic "waiting and see" approach - try to avoid the forwarded packets by waiting for some other nodes to take it up, with a certain probability. Consider that if "one packet" is held in an intermediate node, and that node feels no interest in forwarding the packet after a long time, then how can we do with that?

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 334–343, 2005.

Earlier work [1,2,3] has shown that such non-cooperative behavior could easily jeopardized the network performance to severely degrade. However, the dynamic interactions arising in ad hoc networks make it difficult to analyze and predict node performance, inhibiting the development of the wireless ad hoc networks.

Recently, the idea of using pricing scheme based on Game theory to stimulate node cooperation rushes in wireless ad hoc networks [2,4,5,6]. An efficient pricing mechanism makes decentralized decisions compatible with overall system efficiency by encouraging less aggressive sharing of resources rather than the aggressive competition of the purely noncooperative game. A pricing policy is called incentive compatible if pricing enforces a Nash equilibrium that improves the sum of all players utilities. Although those pricing schemes achieve the whole system maximal throughput or power control purposes, (here, pricing does not refer to monetary incentives, can be treated as a credit level) some policies are extreme, which we think do not account for the relative preferences for individual nodes. Typically, pricing should be motivated by two different objectives: 1) it generates revenue for the system and 2) it encourages players to use system resources more efficiently [7]. However most previous work focus on the first aspect of this problem. In this work, pricing rather refers to motivate individual node to adopt a social behavior from gaining more benefit for themselves.

In this paper, we use pricing policy in such a way: we introduce a "cost and compensation" scheme as a less-aggressive way to avoid such non-cooperative behavior. We assume that once a packet is sent from a source node, the packet is associated with a cost, i.e, when node i needs sending packets as a source node, it is required a cost(e.g. reasonably some money). The cost is adjustable according to the network status, whereas the node can also accept or reject the cost. In order to induce voluntary forwarding, the network will also compensate the nodes who consume energy in forwarding packets for other nodes. If we think of the implied costs as the penalties to be paid by the source nodes and the compensation as the encouragement to relay nodes then local optimization of the node, for example, the desired performance plus the compensation then minus the cost to be paid, will yield an optimal point. Each node can optimize only its packet generate strategy (However the final utility is determined by the strategy set constituted by all other nodes). The "cost and compensation" in this context could be regarded as the network credits, which do not necessarily relate to real money.

The remainder of the paper is organized as follows. Section 2 we describe the basic framework and definitions. In Section 3 we propose an algorithm to find Nash Equilibrium in the game and discuss the implementation issues. Section 4 is the illustration of 3-Node case study. In section 5, we analyze the simulation results. Finally section 6 concludes the paper and the illustrates the future work.

2 Basic Framework

Given a N-node wireless ad hoc network, the transmission radius is assumed to be identical for all nodes. A node can only directly communicate with the nodes

which are inside its transmission range. Each node cannot receive more than one packets or cannot transmit and receive a packet simultaneously and we do not consider channel errors.

The basic setting of the game is as following: There are N nodes in ad hoc networks. Here the nodes are non-cooperative in the sense that they have no means of contributing to others, each node wishes to optimize its usage of the network independently. Each node $i, (i \in \{1, \ldots, N\})$ has strategy x_i as: the rate of the packets generated by node i as a source node. $\underline{x}$ could represent the space of x_i vectors. And utility function U_i is taken to be increasing and concave in accordance with dynamic topology. Utility Function models user i desired normalized throughput depending on both its willingness to pay and the network status, defined on a subset R^N of termed $\underline{x}$. The nonnegative packet generating rate x_i generated by node i satisfies the bounds $0 \leqslant x_i \leqslant MR$. P_i^{sd} is the probability the assigned packets are forwarded by i from node s to node d. $S\{i\}$ is the set of sessions in which node i is a source node. $R\{i\}$ is the set of sessions in which node i is a relay node. α_i is cost factor of node i which represents the cost incurred per unit of packet rate generated by node i as a source node. λ_i is compensation factor of node i which represents the compensation associated with the contribution that node i made.

2.1 Node Problem

The objective of each node is to maximize its net utility which is, for a particular rate, the difference between the network utility and the cost of accessing the network, considered as Lagrangian of system problem Q,

$$max_{\{\underline{x}\}} \begin{cases} x_i \prod\limits_{j \in S\{i\}} {P_j}^{sd} - \sum\limits_{i \in S\{i\}} \alpha_i ln x_i + \prod\limits_{j \in R\{i\}} \lambda_i x_s {P_j}^{sd} \\ 0 \leqslant x_i \leqslant MR \end{cases} \tag{1}$$

2.2 Network Problem

The objective of network is that to determine the optimal packets generating rates to nodes that maximizes its total revenue, based upon the difference between charging and compensation for nodes. We also consider it as Lagrangian

$$max_{\{\underline{x}\}} \begin{cases} \sum\limits_{i \in S\{i\}} \alpha_i ln \underline{x} - \prod\limits_{j \in R\{i\}} \lambda_i x_s {P_j}^{sd} \\ 0 \leqslant A\underline{x} \leqslant MR \end{cases} \tag{2}$$

In this paper, we assume all the nodes are "rational", which means nodes' behavior are totally determined by themselves. In the game, the nodes control their packet generating rates $\underline{x}$ and forwarding preferences $\underline{p}^{sd}$to optimize their utilities; the network controls cost coefficient α_i and compensation coefficient λ_i to maximize its revenue.

2.3 Nash Equilibrium

Definition 1. The situation $x^* = (x_1^*, \ldots, x_i^*, \ldots, x_n^*)$ [1] is called the Nash Equilibrium in the Game Γ, if for all nodes give strategies $x_i \in X_i$ and $i = 1, \ldots, n$ there is

$$U_i(x^*) \geqslant U_i(x^* \parallel x_i) \tag{3}$$

Remark. It follows from the definition of the Nash equilibrium situation that none of the nodes i is interested to deviate from the strategy x_i^*, (when such a node uses strategy x_i instead of x_i^* , its payoff may decrease provided the other nodes follow the strategies generating an equilibrium x^*). Thus, if the nodes agree on the strategies appearing in the equilibrium then any individual non-observance of this agreement is disadvantageous to such a node. In this paper, we will simplify Nash Equilibrium as NE.

3 The Distributed Algorithm

In this section, we give an algorithm to compute NE of non-cooperative node game, and illustrate the implementation issue on ad hoc networks.

3.1 The Distributed Algorithm

As mentioned above, the algorithm could easily be implemented as a local procedure (optimization of $U_i(\cdot)$). For the case of more general networks, we need to calculate the derivative of the utility function of Equation 1. Then the problem is reduced to a single variable optimization problem: a node does an iterative step to compute its optimal packet generating rate. Thus, we compute the derivative with respect to equation 1,

$$\frac{dx_i}{dt} = \dot{x_i} = \frac{\alpha_i}{x_i} - \prod_{j \in S\{i\}} {P_j}^{sd} \tag{4}$$

Note that in the above expression we first assume that the packet forwarding probabilities ($\underline{p}$) and cost and compensation factor of all the source nodes in the network are same initially and then compute the derivative with respect to this ($\underline{x}$). This is because during the computation the node must take both cost and compensation into account to get the optimal strategies.

Note that in the above expression we first assume that the packet forwarding probabilities ($\underline{p}$) and cost and compensation factor of all the source nodes in the

[1] Note $(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n)$ is an arbitrary nodes' strategy set in cooperative game, and x_i is a strategy of node i. We construct a nodes' strategy set that is different from x only in that the strategy x_i of node i has been replaced by a strategy $x_i^{'}$. As a result we have a nodes' situation $(x_1, \ldots, x_{i-1}, x_i^{'}, x_{i+1}, \ldots, x_n)$ denoted by $(x \| x_i^{'})$. Evidently, if node i's strategy x_i and $x_i^{'}$ coincide, then $(x \| x_i^{'}) = x$.

network are same initially and then compute the derivative with respect to this ($\underline{x}$). This is because during the computation the node must take both cost and compensation into account to get the optimal strategies.

Thus, solving the problem is reduced to a single variable optimization issue. A node does an iterative ascent to compute its optimal packet generating rate. Thus, in its k^{th} computation, a node i uses the iteration

$$x_i(k+1) = x_i(k) + \xi(k)(\frac{\alpha_i}{x_i(k)} - K \prod_{j \in S\{i\}} {P_j}^{sd}) \tag{5}$$

where $\xi(k)$ is a sequence of positive numbers satisfying the usual conditions imposed on the learning parameters in stochastic approximation algorithms, i.e., $\Sigma_k a(k) = \infty$ and $\Sigma_k a(k)^2 < \infty$.

Note that it is possible that different nodes settle to different local maxima. We define here that all the nodes settle Nash Equilibrium (Nash Equilibria)in the highest packet generate rate. We are going to discuss the implementation issue of this algorithm in the following description.

3.2 Network Implementation

Above algorithm requires a node to know neighborhood status around itself. In order to get effective knowledge about the network status in topology-blind ad hoc networks, feedback signals are included in the packet header to measure or estimate the network status. Simply to say, the feedback signals reflects the node willingness to pay α_i and network compensation factor λ_i. The iterations can be run at each network node using local information. In the following, we describe the local procedures associated with the scheme only with parameter α_i, because that compensation factor λ_i could be integrated in the packet header in a similar way.

Source Procedure:
A source i sends a forward packet and inserts P_i^{0d} in the corresponding fields. Then, it sends the packet to the destination. At the reception of a backward packet with α_i, i adjusts its P_i^{d0} according to α_i contained in the backward packet. We consider that a source has a variable called P_i^{sd} which is updated as follows: $P_i^{d0} \longrightarrow P_i^{d1}$.

Relay Node Procedure:
1: Let $\underline{x}(0)$ be the initial N-vector of nodes' generating rates.
2: Source node i is associated with a cost factor $\alpha_i(0)$ according to its packets generating rate. This is a global parameter of the system.
3: At the k iteration step of the game, the node i will choose a new packets generating rate according to equation 5.
4: Node i broadcasts the new packet generating rate to its neighbor.
5: All other nodes in the same session will likewise update their choice of forward probability strategies according to step 3.

6: Those nodes advertise their new forward probability according to their neighbors $\underline{x}(1)$.
7: Source node S checks the currently active number of participating nodes, n_j;
8: Broadcast the value of optimal strategy $\underline{x}*$ to all the participating nodes;
9: If the session has changed(e,g, topology changed) go to back to 2; otherwise go back to step 3.

4 Case Study

As a simplified example, let us firstly consider an ad hoc network with 3 nodes, denoted by N_1, N_2, N_3. Transmission could be finished through one intermediate node or to the destination directly. N_1 has one unit packet to send to N_3, it sends its packet to other nodes and keeps its desired cost. N_2 also has packet to send to N_3. N_3 has no knowledge of whether N_1 or N_2 will send the packet directly to it or using a relay node. (Suppose the network cannot verify any claims the nodes might make about their strategies.)

We let $\underline{x}\{1, 2\}$ represents the set of possible strategies that $N1$, $N2$ originally generate. The disagreement outcome is $U*(0, 0)$, where the network gets neither contribution nor utility from the node, and the node gets no utility from the network. That is, each other could guarantee itself a payoff of 0 by refusing the cooperation. Then we have optimal strategies for $N1$, $N2$, the network separately as,

Depending on the value of x_3, $\frac{\partial U}{\partial \underline{x}}$ takes on different values:

$$\frac{\partial U_2}{\partial x_2} = \frac{\alpha_2}{x_2} - P_1^{23} \tag{6}$$

$$\frac{\partial U_1}{\partial x_1} = 1 - \frac{\alpha_1}{x_1} \tag{7}$$

Then we draw the conclusion that the strategy combination achieves a Nash Equilibrium $(x_1, x_2) = (\frac{\alpha_1}{1+\alpha_1}, \frac{\alpha_2}{1+\alpha_2})$ in the 3-node game, which means neither N_2 or N_3 can benefit by unilaterally deviating from this strategy combination.

5 Evaluation Results

In this section, we evaluate the performance of "cost and compensation" scheme in a more general setting, which is closer to the realistic topology scenario of wireless ad hoc networks, we conducted the following simulation on glomosim [10].

5.1 Scenario

We studied a given network 20 nodes (Fig.1) located randomly according to a uniform distribution within a geographical area of 1000m by 1000m. The simulations we investigate has the main design parameters listed in table 1. We illustrate our results for various parameters. For each parameter, the default value and their varying range are provided. In our simulation, the studied scenario is high density and the speed mobility of the nodes is rather low, so we

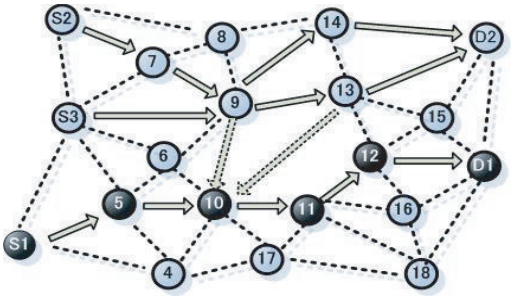


Fig. 1. The random scenario for 20 nodes case

Table 1. Main Simulation Parameters

Parameters	Value
Space	$1000m \times 1000m$
Number of Nodes	20
MAC	802.11b
Cost Factor	0.1, 0.2, 0.3, 0.4, 0.5
Compensation Factor	0.3, 0.5
Packet Generating Rate	Initial Value= 0.6packet/s
(the packet size is fixed	set as 1024k byte)
Packet Forward Probability	Initial Value=0.5
Strategy Updating Interval	1s
Simulation Time	300s

could ignore the packets drop rate. Also, we consider only the number of packets that are generated and forwarded, ignore the size of the packets.

The following process is repeated: nodes randomly choose a destination, and generate packets according to a Poisson process with the initial value 0.6packet/s. At each updating step, relay nodes decide whether to forward the packets as before, or to cease forwarding for a while. The decision is taken on the base of their current payoff function (equation 1): Relay nodes observe the updating cost associated with the former packet generating rate for the new destination node. The new packet forward probability is chosen randomly. Comparing the costs and compensation the nodes choose in the nest step whether to generating own packet or to forward packet for other nodes. For each node, we determined NE that results in the highest packet generate rate.

5.2 Metrics

The main metrics of the overall simulation cycle is:

- *Convergence of the global scheme*: computes the time required for convergence of the scheme.

- *Packet Forward Probability*: computes the probability that assigned packets are successfully forwarded by node i to correct relay node or destination in 5s intervals.

- *Individual Throughput*: Individual throughput is determined by logging the accumulative traffic originating form the node in 5s intervals.

5.3 Analysis of Results

It can be observed from Fig.2a that small values of α lead to low iteration time. This is due to the fact that if the number of sessions is low at the same time, nodes will operate far from the central region and their strategies will not be

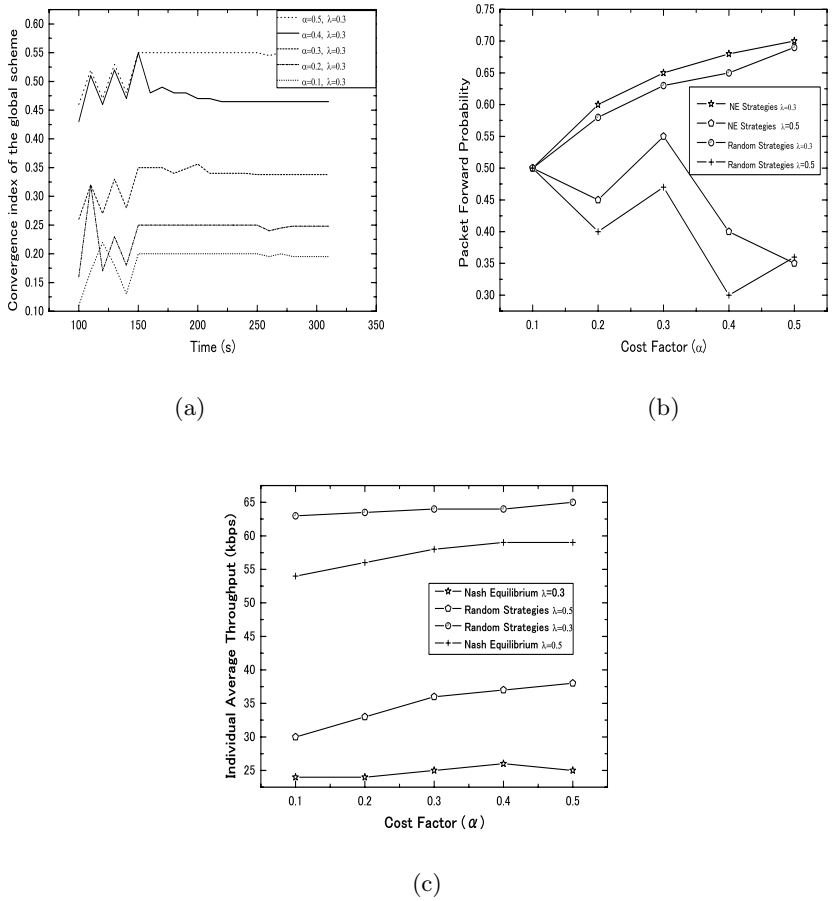


Fig. 2. (a) Convergence of the global scheme; (b) Comparison of NE strategies with random node strategies for individual throughput; (c) Comparison of NE strategies with random node strategies for packet forwarding probabilities

strongly coupled. However, as the value of α increases, the convergence speed also increases. This is due to the fact that as α increases, the cost is heavier, more negotiation time is needed to compensate for the packet forward probability strategies. Accordingly, there is less incentive for the nodes behave selfishly.

From Fig.2b we found that as the high cost α, the packets are forwarded with higher probability. This is due to the fact that the cost factor α increases, the packet generate rate at the NE point for node i decreases. It is shown that the scheme guarantees the optimality for individual node. The value of Equilibria on packet forward probability for node i can be selected to find the best tradeoff point.

The Fig.2c presents payoff as a function of the cost factor α, here different λ values are used. We see that in the NE strategies, individual throughput for node i is improved compared with common random strategies. Thus choosing cooperation is more beneficial with respect to non-cooperative behavior. This figure also compares payoff with different λ value. We see that through the introduction of the compensation, the individual throughput on NE strategies for node i is also improved compared with only using cost strategies. Thus choosing cooperation is more beneficial with respect to non-cooperative behavior. Also, individual throughput for node i increases with the high compensation factor.

6 Conclusion and Future Work

We established a framework using game theory to provide incentives for non-cooperative nodes to collaborate in the case of wireless ad hoc networks. The incentive scheme proposed in the paper is based on a simple "cost and compensation" mechanism via pricing that can be implemented in a completely distributed system. Using non-cooperative game model, we showed network has a steady state and such optimal point — NE exist in the system, the algorithm we provided helps to find the NE. From the simulation results, we showed that node behavior could be influenced through the introduction of "cost and compensation" system. The advantage of this proposed scheme is to lead to a less aggressive way in the sense that it does not result in a degenerate scenario where a node either generates all the own traffic, not forwarding any of the request, or forwards all the other nodes packets. As far as we know, this is the first work that introduce "cost and compensation" concept that has formal framework for encouraging nodes to cooperate.

In terms of future work, we will investigate the effect of different packet sizes on our scheme, and take the dynamic number of arrival and departure nodes into consideration. However, in this paper we do not discuss the conditions under which integration of nodes are interested in forming small non-cooperative groups, this will need a strong NE exist in the system, but it rarely happens. We think this problem will be a part of our future work. Our future work will also want to address the issues of the algorithmic implementation in the context of different measurement scenarios.

References

1. Levente ButtyLan, Jean-Pierre Hubaux, "Stimulating Cooperation in Self-Organizing Mobile Ad Hoc@Networks" ACM/Kluwer Mobile Networks and Applications (MONET), 8(5), Oct., 2003
2. Vikram Srinivasan, Pavan Nuggehalli, Carla F. Chiasserini, Ramesh R. Rao, "Cooperation in Wireless Ad Hoc Networks" Proc. of IEEE INFOCOM, San Francisco March 30 April 3, 2003
3. S. Buchegger, J.-Y. Le Boudec,"Performance Analysis of the CONFIDANT Protocol: Cooperation Of Nodes - Fairness In Distributed Ad-hoc NeTworks" MobiHoc 2002, Lausanne, 9-11 June 2002
4. Sheng Zhong, Jiang Chen, and Yang Richard Yang,"Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks" Pro. of IEEE INFOCOM 2003, San Francisco, CA, April 2003.
5. Luzi Anderegg and Stephan Eidenbenz,"Ad hoc-VCG: a Truthful and Cost-Efficient Routing Protocol for Mobile Ad hoc Networks with Selfish Agents", Mobicom San Diego,California,Sep.2003.
6. J. Crowcroft, R. Gibbens, F. Kelly and S. Ostring, "Modelling Incentives for Collaboration in Mobile Ad Hoc Networks" Proc. of Modeling and Optimization in Mobile Ad Hoc and Wireless Networks (WiOpt)2003, March 3-5, 2003, INRIA Sophia-Antipolis, France
7. C. Saraydar, N. Mandayam, D. Goodman, "Efficient power control via pricing in wireless data networks", IEEE Trans. Communications, Vol. 50, No. 2, Feb 2002
8. Haikel Yaiche, Ravi R. Mazumdar, Catherine Rosenberg,"A game theoretic framework for bandwidth allocation and pricing in broadband networks", IEEE/ACM Transactions on Networking, 2000, Volume 8 , Issue 5 , Pages: 667 - 678.
9. J.B.Rosen, "Existence and uniqueness of equilibrium points for concave n-person game," Econometrica, vol.33,pp.520-534, Jul.1965.
10. http://pcl.cs.ucla.edu/projects/glomosim/

A Cluster-Based Group Rekeying Algorithm in Mobile Ad Hoc Networks*

Guangming Hu [1], Xiaohui Kuang[2], and Zhenghu Gong[1]

[1] School of Computer Science, National University of Defense Technology, Changsha Hunan, China
{gm_hu, gzh}@nudt.edu.cn
[2] Beijing Institute of System Engineer, Beijing, China
xiaohui_kuang@hotmail.com

Abstract. Many emerging mobile wireless applications depend upon secure group communication, in which secure and efficient group rekeying algorithm is very important. In this paper, a novel rekeying algorithm is proposed, which is based on the Distributed Group Key Management Framework and secure clustering. This algorithm involves two phases: (a) Virtual Backbone Management phase: a dynamic virtual backbone is formed below the routing layer such that each node in the network is either a part of the backbone or one hop away from at least one of the backbone nodes. (b) Group Rekeying phase: backbone nodes form group rekeying forest, in which each group rekeying tree can generate a new and same group-key. Because this algorithm generates group key with local secrete information, it is very fit for mobile ad hoc networks. Simulation shows that the algorithm performs better than many existing group key management protocols in terms of the success ratio and average delay of group rekeying.

1 Introduction

Group communication is one of the most important services in a mobile ad-hoc network[1], in which data confidentiality and integrity is realized by encrypting data with group key. In order to meet the forward-secrecy membership and the backward-secrecy polices, any change in the group membership will induce group rekeying. So how to update group-key securely and efficiently is a crucial problem in secure group communication. A lot of work has been done on this problem in wired network. However, in the case of mobile ad-hoc network, the level of difficulty of the problem increases due to the characteristics of the network, such as highly dynamic, multi-hop, and infrastructure-less.

The Distributed Group Key Management Framework (DGKMF)[3] is based on threshold secret sharing, secret share update and RSA encryption technique, in which no single entity in the network knows or holds the complete system secret. Instead, each entity only holds a secret share of the complete system secret. Multiple entities, Say K, locality jointly could generate new group key. So organizing k members is the key problem in DGKMF.

* This work is supported in part by National 973 Program(Grant No. 2003CB314802) and National 863 Program(Grant No. 2003AA142080).

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 344 – 353, 2005.

Utilization of virtual backbones or clusters has been proven to be effective in solving of several problems in mobile ad-hoc networks [3], such as minimizing the amount of storage for communication information (e.g. routing and multicast tables), reducing information update overhead, optimizing the use of network bandwidth, service discovery, network management and security etc. It is highly desirable to have a tighter relation between different layers of communication to reduce the redundancies associated with repeating similar tasks in different layers which results with increased control message overhead. Provided that virtual backbone formation and maintenance mechanisms exist below the network layer, upper layer protocols i.e. routing and group rekeying algorithm, can exploit this backbone together.

In this paper, a novel rekeying algorithm named CBDR*(Cluster Based Distributed Rekeying Algorithm)* is proposed based on the above Distributed Group Key Management Framework. Our solution involves two phases: (a) Virtual Backbone Management (VBM) phase: a dynamic virtual backbone is formed below the routing layer such that each node in the network is either a part of the backbone or one hop away from at least one of the backbone nodes. (b) Group Rekeying (GRK) phase: backbone nodes form group rekeying forest, in which each group rekeying tree can generate a new and same group-key.

The rest of the paper is organized as follows: in Section 2, a brief summary of previous related work is presented. Section 3 describes network model and notation used. Section 4 illustrates the VBM and the GRK phase in detail. Performance measures, simulation framework and results are presented in Section 5; and in section 6, we conclude the paper.

2 Previous Work

Several group key management approaches have been proposed for wired network in the last decade. These approaches generally fall into three categories: 1) centralized, 2) distributed and 3) contributory.

Centralized group key management protocols such as GKMP[4] are conceptually simple as they involve a single entity (or a small set of entities) that generates and distributes keys to group members. But it is not suitable for mobile ad hoc because of its dynamic topology and limited bandwidth. Further more, the single point of failure is another restricting factor.

Distributed group key management protocols such as CKD[5] dynamically select a group member as key sever, and are more suitable to unreliable networks. Although robust, this approach has a notable drawback in that it requires the key server to maintain long-term pairwise secure channels with all current group members to distribute group keys. In mobile ad hoc networks it is a hard task.

In contrast, contributory group key agreement [6] requires each group member to contribute an equal share to the common group key (computed as a function of all members' contributions). This approach avoids the problems with the single points of trust and failure. But this method heavily depends on network topology and connectivity and can not be applied to mobile ad hoc network.

It is difficult to obtain good performance in mobile ad-hoc networks using presence group key management protocols algorithms. Secure group communication becomes

one of research hotpots in mobile ad-hoc network. S. Griffin et. al. characterize the impact of mobility on secure rekeying of group communication in a hierarchical key-distribution framework [7] and propose several rekeying algorithms (SR, BR, IR and FEDRP) [8] that preserve confidentiality as members move within the hierarchy. But all of them depend on fixed node to generate and distribute group key. A novel key management protocol is specifically designed for the distributed sensor network environment in [9], including Identity-Based Symmetric Keying and Rich Uncle. However, their work has focused heavily on energy consumption during key management, and mobility is not actually considered. S. Basagni et. al.[10] consider the problem of securing communication in large ad hoc networks and propose a group rekeying algorithm which update group key periodically by combining mobility-adaptive clustering and an effective probabilistic selection of the key-generating node. This algorithm imposes temper-resistance properties to protect the network from adversaries trying to insert malicious nodes in the network after capturing honest ones. Besides, this algorithm assumes that left node cannot expose any secret information of group and did not update group key when a member leaves.

3 Network Model and Notation

In this section, we present the network model and the notation used throughout the paper.

3.1 Network Model

All the nodes in a mobile ad hoc network are assumed to have an omni-directional antenna and have the same transmission power. All links are bi-directional, i.e. if node A can hear B, then node B also can hear node A. Nodes share the same communication channel (e.g. same frequency band, same spreading code or frequency hoping pattern) to transmit and receive at the same time. Nodes are generally deployed from a common source and the opportunity for the pre-deployed security parameter exchange often exists.

Security attacks on wireless network's physical layer are beyond the scope of this paper. Spread spectrum has been studied as a mechanism for securing the physical layer against jamming [11]. Denial-of Service (Dos) attacks against MAC layer protocols are not considered also; we assume that the wireless network may drop, corrupt, duplicate or reorder packets. We also assume that the MAC layer constrains some level of redundancy to detect randomly corrupted packets; however, this mechanism is not designed to replace cryptographic authentication mechanism.

3.2 Notation and Definitions

We model a mobile ad hoc network by an undirected graph $G=(V,E)$ in which V is the set of wireless nodes and $|V|=N$. There is an edge $\{u,v\}\in E$ if and only if u and v can mutually receive each other's transmission, in this case u and v are neighbors. An RSA-based design is used, which is currently the most prevalent public

cryptosystem. The system RSA key pair is denoted as $\{SK, PK\}$, where SK is the system private key and PK is the system public key. SK is used to sign certificates for all group members in the network, which can be verified by the well-known system public key PK.

Assume that global unique network identifier for node i is ID_i, $i \in 1,2,\cdots N$ and each node knows its own ID. The pre-deployed security parameters include $\{sk_i, pk_i\}$: a personal RSA private and public key pair of node i. It is used in end-to-end security to realize cipher key exchange, message privacy, message integrity and non-repudiation.

$(cert_j)_{SK}$ [1]: Certificate of node i, it is certified that the personal public key of node i is pk_i and node i is a member of group.

GCK_i: Secret share hold by node i, which is generated by the centralized secret share dealer at group bootstrapping phase. The centralized secret dealer obtains the RSA secret key $SK = (d,n)$ and randomly selects a polynomial $f(x) = d + \sum_{i=1}^{k-1} f_i x^i$, where $1 < k < N/2$. Each member ID_i $i \in 1,2,\cdots N$ holds a secret share $GCK_i = f(ID_i) \bmod n$.

$HASH\left(\left(g(m)\right)_{SK}\right)$: Initial group key generated by the centralized secret share dealer at group bootstrapping phase, where $m \in 1,2,\cdots N$ and increases by 1 when group key updates, $g(x)$ is the seed generating function, and *HASH* is a kind of hash function. $g(x)$ and *HASH* are known by all nodes in network. When choosing m', any coalition of *K* members can compute the group key $HASH\left(\left(g(m')\right)_{SK}\right)$ [12].

The notation and the definitions used in virtual backbone management phase are as follows:

$N(i)$: Set of neighbors of node i.

$M(i)$: Set of cluster members. Its initial value is empty and only if node i is clusterhead, should it refresh this set.

In virtual backbone management phase, the node can be one of three states: *Undecided*, *Member* and *Clusterhead*.

The notation and the definitions used to form group-rekeying forest are as follows:

$G(i)$: Set of nodes, which i used to generate group key.

$layer_i$: The layer of node i in group-rekeying tree.

F_i: The father of node i.

$C(i)$: Set of child nodes of i.

Round: Cooperative range of clusterhead.

T_{cop}: Waiting timer for respond message of cooperation request, whose value is relative with *Round*.

---

[1] $(m)_{SK}$: *m* encrypt by *SK*

4 A Cluster Based Distributed Group Rekeying Algorithm

The CBDR algorithm consists of two parts. The first part, VBM phase, selects a subset of the network nodes to form a relatively stable dominating set securely, and discovers the paths between dominating nodes and adapts to topology change by adding or removing network nodes into this dominating set.

After the first part is successfully carried out, the second part is used to efficiently generate and distribute group key. In GRK phase, clusterheads form group-rekeying forest, in which each group-rekeying tree can generate a new and same group-key.

When a node wants to join, we can distribute new group key encrypted by old group key. This method need not construct forest but a group-rekeying tree. The group-rekeying forest needs construct when node leaves. Because construction of a group-rekeying tree is an example of construction of group-rekeying forest, so we only discuss the construction of group-rekeying forest.

4.1 Virtual Backbone Management (VBM) Phase

The goal of the VBM algorithm is to obtain a small size and relatively stable backbone securely. The algorithm is highly distributed and based on local decisions, which makes it fast to react back to the changes in the network topology. VBM algorithm can be described in three components: a) neighbor discovery, b) clusterhead selection, c) finding path between neighboring clusterheads.

The detailed descriptions of VBM can be found in our another paper [13].

4.2 Group Rekeying (GRK) Phase

After the VBM phase is successfully carried out, the virtual backbone is formed, and all clusterheads are connected. As describe in 3.2, any coalition of K members can compute the group key. So, in GRK phase, clusterheads form group-rekeying forest, in which root of each group-rekeying tree is the center of group rekeying. There are two kinds of child node in tree: cooperation child and non-cooperation child. Cooperation child cooperates with its father to generate new group-key; non-cooperation child only receive new group key from his father. Besides, $|G|$ of each group-rekeying tree is bigger than K, and if m' is equal, each group-rekeying tree generates the same new group key $HASH\left(\left(g\left(m'\right)\right)_{SK}\right)$. Based on virtual backbone, GRK phase can be described in two components (a) group rekeying tree formation, (b) group key update.

Group-Rekeying Tree Formation

When clusterhead receive the broadcast of group rekeying, they decided their state (*ROOT, T_UNDECIDED*) by the number of set N and M. Then clusterheads belong to *T_UNDECIDED* start cooperation process to form group-rekeying tree. The detail is followed:

Assume clusterhead is i, if

1. $|M(i)| \geq k-1$, then clusterhead i changes to *ROOT* state, and $G(i) = M(i) \cup \{i\}$.
2. $|M(i)| < k-1 \wedge |N(i)| \geq k-1$, then clusterhead i changes to *ROOT* state, and $G(i) = N(i) \cup \{i\}$.
3. $|N(i)| < k-1$, clusterhead i needs to cooperate with neighboring clusterheads to construct group-rekeying tree. i broadcasts cooperation message, neighboring clusterheads react according its state. Figure 1 shows different scenarios of cooperation.

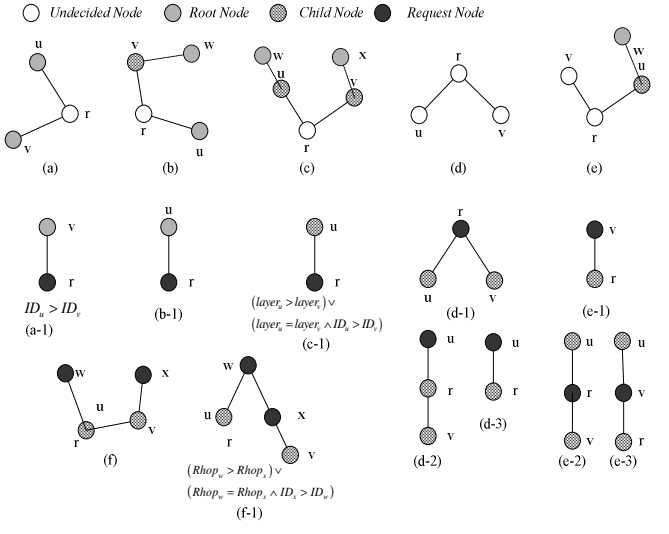


Fig. 1. Scenarios that cooperation occurs and its responding Group-Rekeying Tree

When the number of neighbors is less than *k*-1, clusterhead i sets $C(i)$ to empty, *Round* to 1 and $G(i)$ to $N(i) \cup \{i\}$, then it broadcasts cooperation message (*CBDR_ COOP_REQ*) to *Round* hop clusterheads, at the same time start timer T_{cop}.

1) If there is the neighboring clusterhead in *ROOT* state, it sends *CBDR_ROOT* message to i. When T_{cop} is timeout, i chooses the root node which has the lowest ID as its father, and sends *CBDR_JOIN* to its father. The father node, say l, receives this message and adds i to $C(l)$ as a non-cooperation child. (a-1) and (b-1) in figure 1 shows this Scenario.

2) If there is no neighboring clusterhead in ROOT state, the neighboring clusterheads, which are children of other group-rekeying tree, send *CBDR_ MEMBER* message to i, containing its layer in group-rekeying tree. When T_{cop} is timeout, i chooses its father from these responding node according their *layer* and ID, then sends

CBDR_JOIN to its father. The father, say l, receives this message and adds i to $C(l)$ as a non-cooperation child. (c-1) and (e-2) in figure 1 shows this Scenario.

3) If $|N|$ of each neighboring clusterhead is less than k-1, the neighboring clusterhead, say l, sets $F_l = i$ and changes to be a cooperation child of i, when it receives *CBDR_ COOP_REQ* from i and l does not send cooperation message. Then it sends *CBDR_COOP_JOIN* message to i, which contains $G(l)$. i adds l to $C(i)$ and unites $G(l)$ to $G(i)$. If $|G(i)| > k$, clusterhead i changes to *ROOT* state. (d-1), (d-3) and (e-1) in figure 1 shows this Scenario. If $|G(i)| < k$, i increases *Round*, and repeats cooperation process, when the non-cooperation child receives the cooperation message, it sends *CBDR_COOP_JOIN* message to its father, its father changes the state of this child to cooperation child and send G in the message up to i when the father receives this message. (d-2) and (e-3) in figure 1 shows this Scenario.

When many clusterheads (bigger than 1) start cooperation process, clusterheads decide their relation by their Ids. In (f-1) of figure 1, r and v are children of w and x separately and start cooperation process. When w receives cooperation message from x, it does not react this message because of $ID_x > ID_w$. When T_{cop} of w is timeout, it increases its *Round* and re-broadcasts cooperation message. x chooses z as its father (because z is the nearest node to x in $C(w)$) and changes to a cooperation child, then send *CBDR_COOP_JOIN* to z. w unites $G(x)$ to $G(w)$ when receiving $G(x)$ from z.

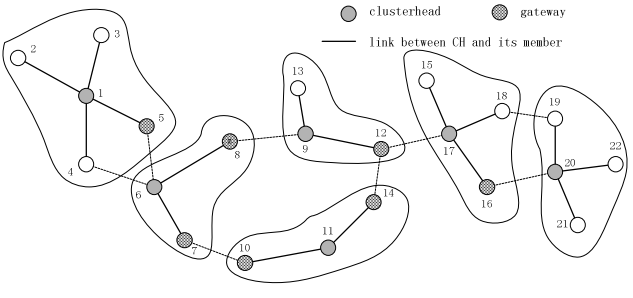


Fig. 2. A mobile ad-hoc network partitioned into clusters

Group Key Update

Once group-rekeying forest has been constructed, in which root of every tree, say i, broadcasts request of group rekeying to its neighbors containing G, its cooperation children and m. If $j \in G(i)$, it send back part group key through secure channel: $GCK_j(m) = (m)^{GCK_j * l_j(0) \bmod n}$, where $l_j(x) = \prod_{m=1, m\neq j}^{|G|} (x-m)/(j-m)$ and $m \in G(i)$. If $j \in G(i)$ is the gateway to cooperation child, it relay the request to the child which contains $G(i)$. The cooperation child rebroadcast the request to its children, and all the

nodes in $G(i)$ that will send back their part group keys to the root. Because $|G(i)| \geq k$, root can combine part group keys to new group key by k-bounded coalition offsetting algorithm [21]. The root distributes new group key by tree: fathers distribute new group key to its neighbors in G and children.

5 Performance Evaluation

We evaluate the performance of our design through simulation, using the network simulator ns-2[14] with wireless extensions. We implement the other group key management protocol such as CKD, GDH v.2 and BD in ns-2. Their performances are then examined and compared.

5.1 Simulation Environment

The signal propagation model uses TwoRayGround model. At the link layer, the IEEE 802.11 standard Medium Access Control (MAC) protocol Distributed Coordination Function (DCF) is implemented. Each mobile node has a position and a velocity and moves about over a rectangular flat space. Nodes move according to the "random waypoint" model.

The performance metrics we observe are:

Group rekey success ratio $S_{ratio} = N_s / N_r$, where N_s is the number of nodes that update group key successfully and N_r is the number of nodes that receive the request of rekey.

Group rekey delay is the time used for all members to update group key.

5.2 Performance with Group Size

We first examine the performance as the group size increases from 40 to 100 when the error rate becomes 10%, the transmission range is 150m, maximum speed is 5 m/s and threshold is 3. As it is shown in Figure 4, the success ratio of CBDR is almost 100% no matter node join or leave group, while other group key management protocols fails. From the figure, average delay almost remains unchanged as group size grows. However, other group key management protocols incur much higher delay, which also greatly fluctuates.

The performance of CKD□GDH v2 and BD between node joins and leaves change dramatically, while the performance of CBDR almost unchanged.

Node leave

From the detail of CBDR algorithm, the higher the density of node is, the better the performance of algorithm is, because every clusterhead can computer new group key separately if the number of one-hop neighbors is bigger than k. Figure 3 shows the performance of CBDR algorithm when the value of threshold varies, node leaves group and average neighbors is 4.

As it is shown in Figure 5, the success ratio of CBDR is almost 100% and average delay is nearby 40s when threshold is 3 and 5 because the threshold approximates average neighbors, and the layer of group-rekeying tree is very small. When threshold

increases to 10, the performance of CBDR goes to the bad because it need more time to constructs the forest and multi-hop communication increases.

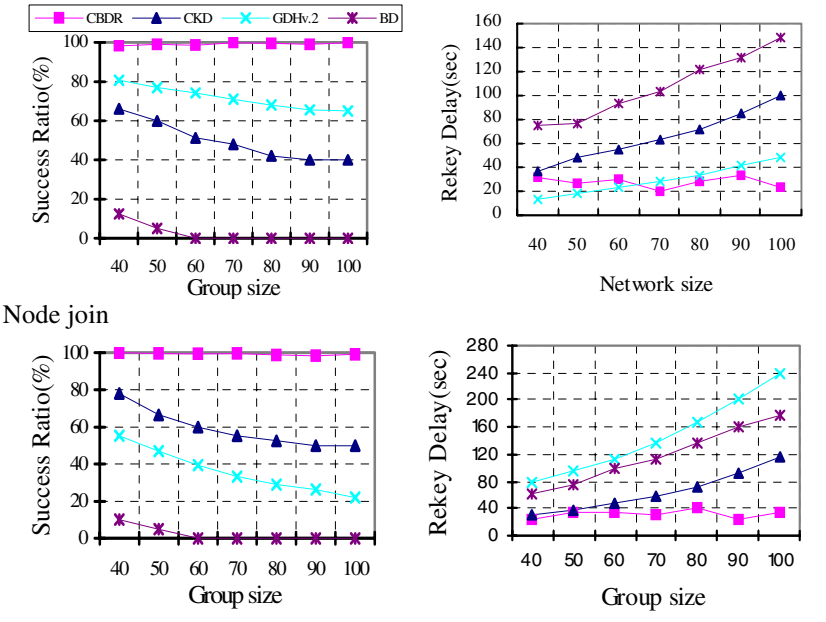


Fig. 3. Performance comparison with respect to group size

6 Conclusion

In this paper, a novel algorithm to update group key in mobile ad hoc networks is presented. This algorithm has been motivated by these main factors: (a) Clustering is an effective method in the solution of several problems in mobile ad-hoc networks. Therefore, secure clustering algorithm is used for group rekeying. (b) Group key can be generated locally by threshold secret sharing, which makes group rekeying decentralizing to operate in a large-scale. To this end, we have addressed how to organize the members of group to update group key. The network issues, including mobility, scalability and network dynamics such as channel interference and node failures are also taken into consideration. Simulation shows positive results for our algorithm in terms of the success ratio and average delay of group rekeying.

References

1. Z.J. Haas, J. Deng and B. Liang. Wireless Ad Hoc Net-works. Encyclopedia of Telecommunications, John Proakis, editor, John Wiley, 2002.
2. X Kuang, H Hu and X Lu. A New Group Key Man-agement Framework for Mobile Ad-Hoc Networks. Journal of Computer Research and Development China, 2004,41(4): 704~710.

3. M. Chatterjee, S.K. Das and D. Turgut. WCA: A Weighted Clustering Algorithm for Mobile Ad hoc Networks. Journal of Clustering Computing IEEE Vol. 5, No. 2, April 2002 pp.193-204
4. H. Harney and C. Muckenhirm. Group key management protocol (GKMP) architecture. RFC 2093 IETF, July 1997
5. Y. Amir, G. Ateniese, D. Hasse, Y. Kim, C. Nita-Rotaru, T. Schlossnagle, J. Schultz, J. Stanton and G. Tsudik. Secure Group Communication in Asynchronous Net-works with Failures: Integration and Experiments. IEEE ICDCS 2000: 330~343
6. Y. Kim, A. Perrig, and G. Tsudik. Simple and fault-tolerant key agreement for dynamic collaborative groups. In Proceedings of 7th ACM Conference on Computer and Communications Security, pp. 235~244, ACM Press, November 2000.
7. S. Griffin, B. DeCleene, L. Dondeti, R. Flynn, D. Ki-wior, and A. Olbert. Hierarchical Key Management for Mobile Multicast Members. Submitted to NDSS 2002
8. C. Zhang, B. DeCleene, J. Kurose and D. Towsley. Comparison of Inter-Area Rekeying Algorithms for Se-cure Wireless Group Communications. Submitted to ACM Symmetric 2002.
9. Carman. Constraints and Approaches for Distributed Sensor Network Security. dated September 1, 2000. NAI Labs Technical Report #00-010
10. S. Basagni, K. Herrin, D. Bruschi, and E. Rosti. Secure pebblenets. In Proceedings of the 2001 ACM Int. Symp. on Mobile Ad Hoc Networking and Computing. ACM Press, October 2001:156~163
11. Raymond L. Pickholtz, Donald L. Schilling and Laur-ence B. Miltein. Theory of Spread Spectrum Communi-cation – A Tutorial. IEEE Transactions on Communica-tions, 30(5):855~884, May 1982
12. J Kong, P Zerfos, H Luo, S Lu and L Zhang. Providing robust and ubiquitous security support for mobile ad-hoc networks. In Ninth International Conference on Network Protocols (ICNP'01): 251~260, 2001.
13. Hu Guangming, Huang Zunguo, Hu Huaping, Gong ZhengHu, SLID: A Secure Lowest-ID Clustering Algorithm, WUHAN UNIVERSITY JOURNAL OF NATURAL SCIENCES, 2005 Vol.10 No.1
14. Wireless and Mobility Extensions to ns-2. http://www.monarch.cs.cmu.edu/

Enhanced Positioning Probability System for Wireless Ad Hoc Networks*

Insu Jeong[1], Yeonkwon Jeong[1,*], Joongsoo Ma[1], and Daeyoung Kim[2]

[1] School of Engineering, Information and Communications University,
Daejeon, 305-714, Korea
{bijis, ykwjeong, jsma}@icu.ac.kr
[2] Dept. of InfoCom Engineering, Chungnam National University,
Daejeon, 305-764, Korea
dykim@cnu.ac.kr

Abstract. This paper presents an enhanced positioning probability algorithm. This algorithm is a completely improved algorithm for locating mobile nodes in defective environment of trilateration location system. The positioning node can find its location with just two reference nodes and their neighbor information. Using numerical analysis model and ad hoc network model, this paper analyzes the effects of positioning probability enhancement and control messages overhead. And this algorithm is also implemented on Linux systems and we test the system at playground. In the pure lateration location system, when a locating node has just two reference nodes, the probability of its positioning is zero but our system is about 80% positioning rate with the average control messages less than 0.5 at 100 reference nodes.

1 Introduction

Location awareness may be a mandatory function in most applications. The popular applications are the building automation for ease of installation and maintenance, home automation, inventory in hospital, warehouses, and file tracking, people tracking for resource optimization in offices, efficiency optimization and security in factories and so on[1]. The positioning of a fixed or mobile user is a fundamental and crucial issue. Especially the location of nodes which are not equipped with GPS receivers has recently attracted much interest in the wireless research community[2][3].

A device running on the lateration-location-sensing system[4] must have more than two reference nodes, whose position information is known, to locate it. This mechanism is that a mobile node measures the distances from the three reference nodes which already have their location information and then calculates its

* This research was supported by the MIC(Ministry of Information and Communication), Korea, Under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment).

** Corresponding Author.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 354 – 363, 2005.

coordinates from the three distances. The important weak point of above trilateration-location sensing technique is that because the reference nodes have to be fixed at very accurate positions, the management of the reference points require a great deal of labor. Therefore reducing these troublesome reference nodes is effective impact.

However if the positioning node has only two reference nodes, the node obtains two locations (one is real position and the other is imaginary position) and cannot determine its position. This happening can break out from followings: One of the three reference nodes may be out of order because of occurrence of accidents such as pouring rain, storm of snow and lightning or shut-down by the malicious attacks. And this event can also come from exhaustion of the nodes' batteries. In these situations, the location system abruptly breaks down. And there can be another case where from the beginning, the location system does not work, that is, one of the three reference nodes is absent at the first time. The border of location service area is a good example. And, reference nodes also cannot be installed in certain areas such as restricted and private areas.

In this paper, we present an enhanced positioning probability algorithm. The positioning node can find its location with just two reference nodes and added one parameter. The added one parameter may obtain from each routing table of two reference nodes. With the location information of the neighbors and effective communication range information, the positioning node is able to distinguish the imaginary position from two coordinates. By removing the imaginary position, this algorithm can locate the node.

Using numerical analysis model and ad hoc network model, this paper discusses that how much the positioning probability increase and how much the control messages overhead is requested to reach positioning probability enhancement goal. We also implement this algorithm on Linux systems and integrate it into enhanced positioning probability system. Lastly, we verify its effectiveness through experimentations in ad hoc network having IEEE 802.11 network interface.

The rest of the paper is organized as follows: Section 2 gives related work. Section 3 proposes our algorithms for positioning probability enhancement. Section 4 presents numerical evaluations of our algorithms and section 5 shows implementation and test results. Finally, section 6 makes conclusions of this paper.

2 Related Work

2.1 Positioning Techniques

Until now, many researchers have studied many methods to know the current location of mobile users. When attempting to determine a given location, we can choose from three major techniques as follows[4]: scene analysis, proximity, trilateration, and triangulation.

Scene analysis is a method to know the position by way of using the characteristics of the scene of mobile users. A merit of this method is only the use of observed scene, no necessity of geographical features. However, the radical defect is that location systems have to know the scene of all observing positions beforehand. RADAR[5][6] is representative example of this class. Microsoft developed RADAR with signal strength to measure distance between the mobile host and AP.

Proximity is to know the location of things can be known by way of sensing contact with the already located things. It just can presume the location and know the vicinity. The Active Badge of AT&T Cambridge[7] is one example.

The idea of trilateration is that if each distance from three points can be taken, the point of intersection is the current position of things in two dimensions. Lateration is used in GPS(Global Positioning System)[8], Cricket System which is developed by MIT[9].

Contrary to trilateration, triangulation obtains the location obtains by way of measuring not distances but angles from the base direction. Angulation is used in VOR(VHF Omnidirectional Ranging) aircraft navigation system. The strong point of this method is the load for calculation is low and the reliance and accuracy are very higher than others. By the way, because measuring an angle is more difficult than measuring a distance and triangulation needs the system for measuring not only an angle but also a distance, but trilateration needs only the system for measuring a distance, trilateration is more popular than triangulation. So, we are going to consider trilateration method for locating things.

But unfortunately, the important weak point of this method is that because the datum points are fixed at very accurate positions, the installations, operations and maintenance of the fiducial points require a great deal of labor. Therefore reducing these troublesome reference nodes is effective impact.

2.2 Routing Protocols for Ad Hoc Network

The routing has been the most active research field in ad hoc networks. Generally, the routing protocol is divided into proactive and reactive protocol.

Protocols that keep track of routes for all destinations in the ad hoc network have the advantage that communications with arbitrary destinations experience minimal initial delay from the point of view of the application. When the application starts, a route can be immediately selected from the rout table. Such protocols are proactive protocols or table driven protocols.

On the contrary, on-demand, or reactive, protocols have been designed so that routing information is acquired only when it is actually needed.

The topology changes from time to time in the ad hoc networks. Because it is impossible to keep track of routes for all destinations in the ad hoc network, proactive protocols are not good for ad hoc networks. As examples of reactive protocols, AODV[10] and DSR[11] are well-known.

3 Positioning Probability Enhancement

3.1 Enhanced Positioning Probability Algorithm

If a positioning node has only two references, the node recognizes its position as two coordinates. The two positions -$\{x_1, y_2\}$ and $\{x_2, y_2\}$- are two points of intersection of two circles. One is the real position and the other is the imaginary position like Figure 1. The node, say X, does not know which one is the real position.

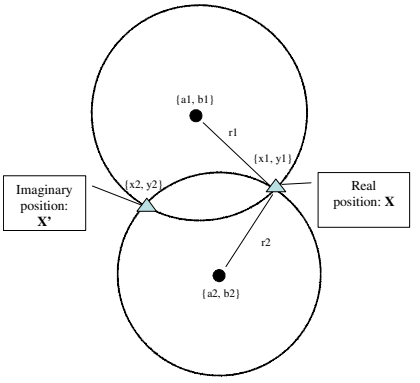


Fig. 1. Positioning with Two References

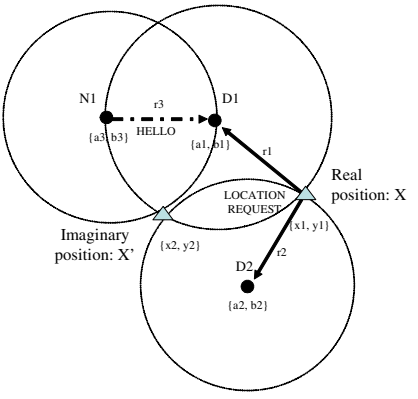


Fig. 2. Elimination of Imaginary Position

In here, if the node X can discriminate between the real position and the imaginary position, and eliminate the imaginary position with a certain method, the node X can locate its position with only two references.

The certain method is just each routing table of two references. Using the neighbor's location and effective communication range of the references, our algorithm can remove the imaginary position of the node X.

Let assume there is at least a node, N1. The node N1 has to meet one of two following conditions for usage in our algorithm.

(1) The node N1 can be directly reachable from node X if it is neighbor of node X.
(2) Otherwise, if the node N1 is located on two hop distance from node X, then the node X can be reachable through at least one of two reference nodes.

In the case of (1), the node X has thee reference nodes and we can get its positioning with legacy trilateration. However, the case (2) has only two reference nodes around node X and the positioning is impossible. The reason is two positions exist there. So, we make rules to distinguish the imaginary position from two positions and would remove it.

Our algorithm use following rule: If there is a position satisfying the condition of equation (1), the position is real position. Otherwise, the position is imaginary position.

$$(x_i - a_3)^2 + (y_i - b_3)^2 > r_3^2,\ i \in \{1,2\} \tag{1}$$

Where (a_3, b_3) means the position of node X and r_3 means the effective communication range of node X.

For example, in the figure 2,

$$\begin{aligned} (x_1 - a_3)^2 + (y_1 - b_3)^2 &\geq r_3^2 \\ (x_2 - a_3)^2 + (y_2 - b_3)^2 &< r_3^2 \end{aligned} \tag{2}$$

The point $\{x_2, y_2\}$ runs counter to equation (1). Therefore, the point $\{x_2, y_2\}$ is the imaginary position and has to be removed.

3.2 Routing Protocol Modification

AODV[12] has a little modifications to apply our algorithm. At first, two new fields are added into the routing table: One is location and the other is effective communication range. Reference nodes have to periodically notify this information to neighbor nodes for helping their location finding. Location and effective communication range fields are encapsulated into the HELLO message of AODV and exchanged.

As an example, in the Figure 2, D1 and D2 are references and N1 is the neighbor of D1, not D2. In those configurations, D1 can know its neighbor through the HELLO message of the AODV. So, the D1 can have location information of neighbor N1 via HELLO message of the AODV as table 1.

Table 1. Location Table of Neighbors of D1

Neighbor	Location	Effective Communication range
N1	$\{a_3, b_3\}$	r_3

When a node, say X, running on ad hoc routing protocol comes in the small fraction of the location-based network, the node X can know its neighbors automatically through its ad hoc routing protocol. When the node X wants to know its position, if the node X has more than two references in its neighbors, the node X can know its location without the help of neighbors of the references. If not, the node X requests location tables of neighbors to the references to remove the imaginary position$\{x_2, y_2\}$ after ranging between itself and each reference. Then the two references respond with the information of themselves and their neighbors. So, the node X has the table as table 2, eliminates its mirror (x_2, y_2) via table 1, and finally finds its location.

Table 2. Table of Node X

Reference	Location	Neighbor	Neighbor location	Effective Communication range
D1	$\{a_1, b_1\}$	N1	$\{a_3, b_3\}$	r_3
D2	$\{a_2, b_2\}$	.	.	.

4 Evaluation of Enhanced Positioning Probability Algorithm

4.1 Numerical Results

We make a system model to evaluate our algorithm easily and we define three parameters to study how the positioning probability changes over topology of the three points:

(1) q is the distance between one reference node and neighbor of it.
(2) s is defined as distance between two reference nodes.
(3) θ is $\angle N1D1D2$, that is the angle among three reference nodes, one of which is not a reference here, but can be a reference in another fraction of the location area.

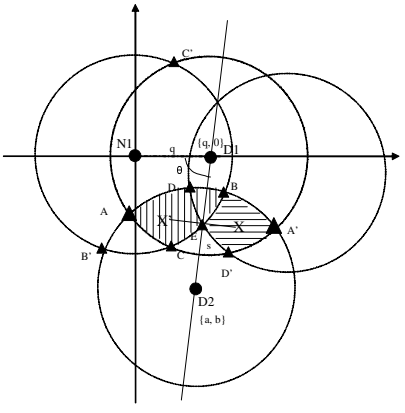


Fig. 3. Analysis Model

We analyze the positioning probability of node X when the node X is moving in the area where node X is in the common radio range of two reference nodes D1 and D2. Then we iterate the above job with changing q, s and θ. If q and s and θ are well adjusted, that is, reference points are well located, the probability of positioning of mobile nodes with our location algorithm can be maximized. Refer to other paper[13] for detailed numerical analysis and its improvements.

4.2 Impact of the Algorithm in Network Model

We simulated two topologies, which are uniform random and grid topologies. Evaluated network area is configured 100m by 100m. The network models have 100 reference nodes which are distributed in random or grid and 3000 blindfolded nodes are distributed uniformly in the network. We assume the communication link is reliable (no delay, no packet drop) and all references are reachable to each other via ad hoc routing protocol (there is no isolated reference).

Figure 4 shows an example of grid topology. Figure 5 and 6 show each of the positioned nodes at grid topology by the pure location system and enhanced ositioning probability system. As we see, the positioning area increases in the network with improved location system, especially at the outer ring.

The more the number of references increase, the more the positioning rate increase. But, to reduce the number of references is important because the maintenance of them is a big labor of work.

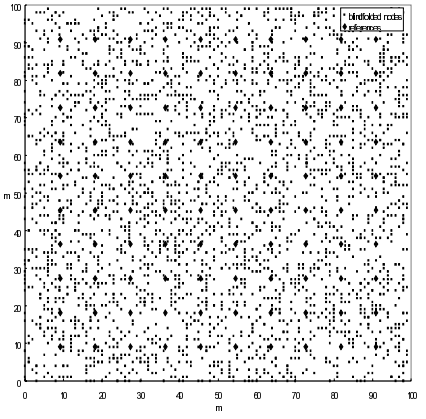


Fig. 4. Grid Topology

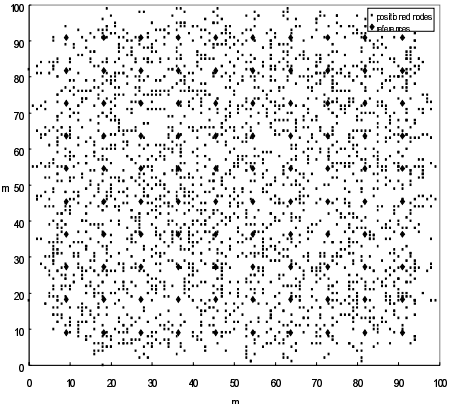


Fig. 5. Pure Location System

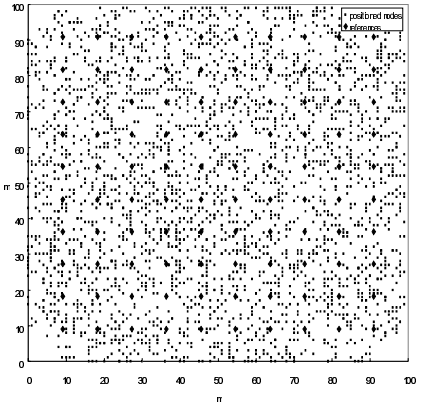


Fig. 6. Enhanced Positioning System

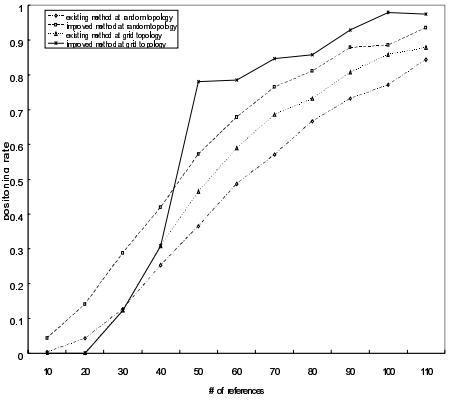


Fig. 7. Positioning Rate over References

Figure 7 shows the positioning rate of nodes changing the number of reference nodes in the two network models with effective communication range of 13m. At about 80% of positioning rate, the number of reference nodes at grid topology with enhanced positioning probability systems is the lowest as 50. Other mechanisms need the number of reference nodes above 70. The number of reference nodes in the topologies with enhanced systems always is lower than that with pure location systems. At the low reference nodes, the rate of the nodes at random topology is larger due to the randomness of the network. However, because the rate is very small, it is worthless.

4.3 Control Overhead

In section 3, if the number of reference nodes of a node is two, the node sends the request for neighbor tables of two references. Then the two references respond to the

request. The number of control messages for a node to send and receive for location is three, that is, as follow equation:

$$Avg.\ control\ msg. = \frac{3n_{2b}}{n_b + n_r} \quad (7)$$

$$where,\ n_b : \text{the number of blindfolded nodes}$$
$$n_r : \text{the number of references}$$
$$n_{2b} : \text{the number of location-requsted nodes}$$

Figure 8 shows average control messages per a node at random and grid topologies with 100 references. Up to 60 reference nodes, the average control messages per a node at grid topologies are more than those at random topologies because the case that a node has two reference nodes as neighbors in the grid topology is more frequent than that in the random topology. But, over 60 reference nodes, the case that a node has more than two references is dominant at both topologies. At this time, the control messages are not generated because the nodes use pure location system. We can see the average control messages are less than 0.5 at about 100 reference nodes with which the positioning rate is over 80%. This is relatively low overhead over the network.

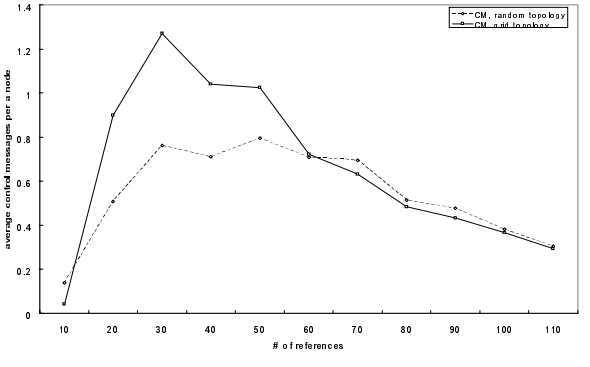


Fig. 8. Average Control Messages per a Node over Reference Nodes

5 Experimentation and Results

Figure 9 gives the software architecture of our positioning system. AODV-UU-0.7.2 is modified to support the enhanced positioning probability algorithm. The HELLO message is used to deliver the location and effective communication range. In addition, the routing table of AODV is also modified to cope with exchanging neighbor table which includes the positioning information due to node mobility. The range estimation technique of the current system is RSS (Received Signal Strength) with well-recognized log distance path loss model[15].

Figure 10 shows the test-bed configuration for mirror elimination algorithm. All reference nodes are Samsung V30 Laptops, the mobile node is LG-IBM X31 sub-notebook, and transmission power is 1*mw*. We started to measure the location of the

mobile node at (40, 20) 100 times. We moved the mobile node toward the left of the Figure 10. At every 10*m*, we measured the location of the mobile node 100 times up to (-20, 20).

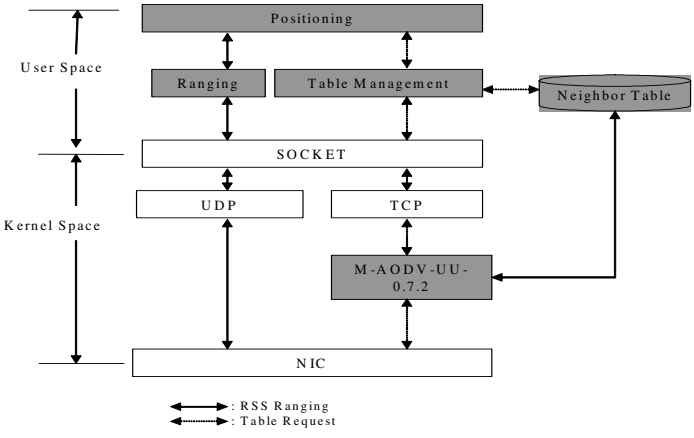


Fig. 9. Software Architecture of Prototype of Improved Positioning

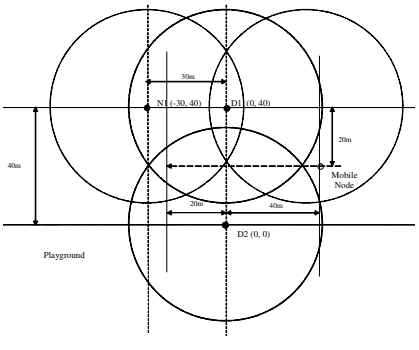


Fig.10. Test-bed Configuration

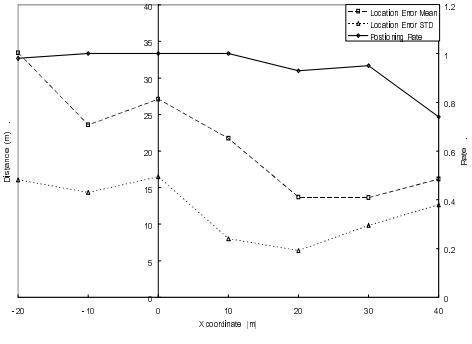


Fig. 11. Positioning Success Rate & Location Error

At 20*m,* 30*m* and 40*m* of axis where are outside of the communication range of *N1*, that is, where the mobile node has just two references, positioning rate is about 80% with about the location error 21*m* from Figure 11. The more the distance increases, the less the positioning rate is because the more the distance increases, the less the communication rate is, so the probability of the success of neighbor table exchange for enhanced positioning probability system decreases.

6 Conclusions

This paper presents an enhanced positioning probability algorithm. This algorithm is a completely improved algorithm for locating mobile nodes in defective environment of

trilateration location system. The positioning node can find its location with just two reference nodes and their neighbor information. Using numerical analysis model and ad hoc network model, this paper analyzes the effects of positioning probability enhancement and control messages overhead. And this algorithm is also implemented on Linux systems and we test the system at playground. We can summarize the results as follows:

(1) At the analysis model, if we configure reference nodes well in terms of three parameters (q, s and θ), the probability of positioning of mobile nodes with our location algorithm can be maximized.
(2) In the pure lateration location system, when a locating node has just two reference nodes, the probability of its positioning is zero but our system is about 80% positioning rate.
(3) Then the average control messages are less than 0.5 at about 100 reference nodes. That is relatively low overhead over the network.

References

1. G. Chen and D.Kotz, "A Survey of Context-Aware Mobile Computing Research," Dartmo-uth Computer Science Tech. Report TR2000-381, 2000
2. S. Capkun, M. Hamdi, J. P. Hubaux, "GPS-free Positioning in Mobile Ad-Hoc Networks", Proc. of HICSS, pp.10, Jan. 2001
3. D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS)", Proc. of GLOBECOM pp.2926-2931, Nov. 2001
4. J. Hightower and G. Borriello, "A survey and Taxonomy of Location Systems for Ubiquitous Computing", IEEE Computer, 34(8), pp57-66, Aug. 2001
5. P. Bahl and V. Padmanabhan. "RADAR: An in-building RF-based user location and tracking system". Proc. of INFOCOM, vol. 2, pp.775-784, Mar. 2000.
6. P. Bahl and V. Padmanabhan, "Enhancement to the RADAR User Location and Tracking System", Technical Report MSR-TR-2000-12, Microsoft Research, Feb. 2000
7. R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The Active Badge Location System", A CM Trans. on Information Systems, 10(1), pp.91-102, Jan. 1992
8. Getting, "The Global Positioning System", IEEE Spectrum, 30(12), p36-47, Dec. 1993
9. N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System", Proc. of 6th ACM MOBICOM, Aug. 2000
10. C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-demand Distance Vector (AODV) Routing," RFC 3561, July 2003
11. D. B. Johnson, D. B. Maltz, and Yih-Chun Hu, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," draft-ietf-manet-dsr-10.txt, July 2004
12. Uppsala University, http://user.it.uu.se/~henriki/adov/
13. I. Jeong, N. Kim, Y. Jeong, and J. Ma, "A Positioning Probability Enhancement Algorithm Using Ad Hoc Routing Protocol", Proc. of CIC, Oct. 2004
14. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, ANSI/IEEE Std 802.11 1999 Edition
15. P. Bhagwat, B. Raman, and D. Sanghi "Turning 802.11 Inside-Out", Second Workshop on Hot Topics in Networks (HotNets-II), Nov. 2003

A Virtual Circle-Based Clustering Algorithm with Mobility Prediction in Large-Scale MANETs

Guojun Wang[1,2], Lifan Zhang[2], and Jiannong Cao[1]

[1] Department of Computing, Hong Kong Polytechnic University,
Hung Hom, Kowloon, Hong Kong
[2] School of Information Science and Engineering, Central South University,
Changsha, Hunan Province, P.R. China 410083

Abstract. The design of routing algorithms in MANETs is more complicated than that in traditional networks. Constructing a virtual dynamic backbone topology is a general approach for routing in MANETs. In our previous work, we have proposed a logical Hypercube-based Virtual Dynamic Backbone (HVDB) model, which has the capabilities of high availability and good load balancing in order to support various communication modes such as unicast, multicast and broadcast in large-scale MANETs. In this paper, we propose a novel virtual Circle-based clustering Algorithm with Mobility Prediction (CAMP) in order to form a stable HVDB for effective and efficient routing.

1 Introduction

Mobile Ad hoc NETworks (MANETs) [4] is a very hot research topic in recent years because of their self-organizing, rapidly deployable, dynamically reconfigurable properties. The concept of *Virtual Dynamic Backbone* (VDB) has been proposed to seek for similar capabilities of the high speed and broadband backbone in the Internet. Two major techniques are used to construct a VDB, i.e. Connected Dominating Set (CDS) and Clustering. Routing based on the VDB scales better, since the number of nodes concerned with routing can be reduced to that of the backbone nodes. But the scalability is not automatically guaranteed if too many tiers exist in the VDB. One generally uses a backbone with only a few tiers (say, two) [22], for several reasons, such as the maintenance of multi-tier routing, traffic load of higher tier nodes.

In this paper, we propose a novel *virtual Circle-based clustering Algorithm with Mobility Prediction (CAMP)*, to form a stable logical Hypercube-based Virtual Dynamical Backbone (HVDB) in large-scale MANETs, which is presented in [21]. The CAMP algorithm is based on the location information and mobility prediction, and elects a Mobile Node (MN) as a Cluster Head (CH) when compared to others: (1) It has the longest stay time within the predefined circle region that called Virtual Circle (VC), based on mobility prediction. (2) It has the minimum distance to the center of the VC, based on location information. The HVDB model is derived from an n-dimensional hypercube. An n-dimensional hypercube has $N=2^n$ nodes. We generalize the *incomplete hypercube* [9] by assuming that any number of nodes/links can be absent. Hypercube is originally proposed as an efficient interconnection

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 364 – 374, 2005.

network topology for Massively Parallel Processors (MPPs). Recently the hypercube has been applied to other network environments, such as the Internet [5] [12], the P2P networks [16] [19] [23], and the overlay networks for P2P computing [17].

The motivation for us to introduce hypercube into MANETs is that, the hypercube networks have four kinds of desirable properties, i.e. fault tolerance, small diameter, regularity and symmetry, which help to achieve high availability and load balancing of the network that are prerequisites for economical communications.

The remainder of the paper is organized as follows. Section 2 presents some related works on some clustering techniques and the VDB in MANETs. The proposed HVDB model is introduced in Section 3. Section 4 describes the proposed clustering algorithm. Section 5 gives the analysis and Section 6 concludes the paper.

2 Related Works

This section shows some related works in two aspects: (1) the clustering techniques; and (2) the techniques to form the VDB structure.

2.1 Clustering Techniques

Four kinds of typical clustering algorithms available in the literature are as follows and numerous other algorithms are their variations.

The *lowest identifier* (Lowest-ID) algorithm [8]: each MN has a unique identifier. The node with the smallest identifier in its neighborhood is elected as the CH. The algorithm needs simple computation and easy execution. However, a highly mobile CH with the lowest ID will cause severe re-clustering; and if it moves into another region, it may unnecessarily replace an existing CH, causing transient instability.

The *maximum-connectivity* algorithm [6]: the node with the maximum degree in its neighborhood is elected as the CH. And the smallest identifier is the second criterion. The algorithm has small number of clusters, resulting in short delay of packet transmission, but causing the problem of small channel spatial reuse probability. Node mobility will influence the degree of the node, resulting in the change of the CH.

The *weighted clustering* algorithm [3]: it uses a combined weighted metric, which takes into account several system parameters like node degree, node transmission power, node mobility, and node energy. The number of nodes in a cluster is a pre-defined threshold to facilitate the optimal operation of the MAC protocol. Election of a CH is on-demand. The algorithm has good load balancing. The difficulty is how to compromise the weighted parameters to consume relatively low control overhead.

The *passive clustering* algorithm [11]: it executes clustering upon the information that is listened, and it needs no periodical control information and no initial process like the above three algorithms. It is realized in the MAC layer and can be used in a diversity of on-demand routing protocols. However, the collected information may not be integrated and up-to-date because of its passive characteristic.

A zonal algorithm for clustering in [4] is proposed to divide the graph into regions by using a spanning forest, to construct a weakly-CDS for each region, then to produce a weakly-CDS of the entire graph. A distributed clustering algorithm called MOBIC is proposed in [1], which is based on the ratio between successive

measurements of received power at any nodes from its neighbors. It is similar to the Lowest-ID algorithm by using mobility metric instead of the ID information. Clustering schemes based on mobility prediction in [15] [18] have some similarities to our proposed CAMP algorithm, and we discuss them in Section 5 in more details.

2.2 The Virtual Dynamic Backbone (VDB)

The VDB is a sub-graph of the entire network topology and it exploits a hierarchical structure. It must be stable enough to facilitate to use the constant routes on the backbone. And the size of the VDB should be small to avoid the scalability problem.

In [13], a cluster-based backbone infrastructure is proposed for broadcasting in MANETs. For each broadcast, the backbone that is actually a CDS can be formed either statically or dynamically. The static backbone consists of fixed CHs, which are selected from source-independent gateways. The dynamic backbone selected fixed CHs from the gateways dynamically. This algorithm is message-optimal, i.e. the communication complexity and time complexity of such a backbone are both $O(n)$, where n is the network size. But the stability of the backbone can not be guaranteed.

A distributed virtual backbone scheme is presented in [14], which uses clustering, together with labeling and heuristic Steiner tree techniques. With the labeling scheme, nodes with higher connected degree will have higher probability to be CHs. Heuristic Steiner tree algorithm is used to generate the optimal virtual backbone with small size. However, this algorithm is rather complicated; and the CH is changeable due to considering the connected degree as the clustering criterion.

In [2, 10], a VDB approximates a minimum CDS. It is an NP-complete problem to find a minimum CDS. In [2], it computes a spanning tree with as many leaves as possible; all the non-leaf nodes form a CDS. In [10], relatively stable and high degree nodes have higher priority to be backbone nodes. Both of the two use the marking process to reduce the size of a CDS. The marking process is effective and it supports localized maintenance, but it has a high computation cost since each node needs to check all the coming messages in [2] and to check all pairs of its neighbors in [10].

In [20], it uses two mechanisms: clustering and adjustable transmission range to form a VDB. The basic idea is to first reduce the network density through clustering using a short transmission range. Then neighboring CHs are connected using a long transmission range and form a CDS. It extends the marking process in [2] to reduce the size of the CDS. The approaches bring out effectiveness but with high computation cost, especially in dense networks.

3 The Logical Hypercube-Based VDB Model

The 3-tier HVDB model is shown in Fig. 1, which is presented in [21]. In the model, the large-scale MANET is deployed in a rectangular region, which is divided into square regions of equal size. The Virtual Circles (VCs) of equal size that are overlapping to each other in a systematic way, as in [18], are formed based on these square regions. The center of each VC is called a Virtual Circle Center (VCC).

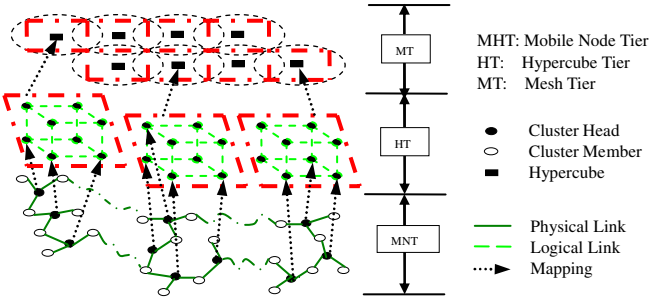


Fig. 1. The Virtual Dynamic Backbone Model

The MN Tier consists of MNs in the network. The nearby MNs are grouped into a cluster. Each cluster has a CH and some cluster members. The CHs are responsible for communicating between clusters, and for managing their member nodes. Each MN can determine the VC where it resides by its location information. If there is a CH in a VC, then we view the VCC as the CH; if not, then it is only a placeholder.

The Hypercube Tier consists of several logical *k*-dimensional hypercubes, whose nodes are CHs actually, and *k* is relatively small in our consideration, e.g., 3, 4 or 5. A logical hypercube node becomes an actual one only when a CH exists in the VC. The CHs located within a predefined region build up a logical hypercube, most possibly an incomplete hypercube due to the distribution and high mobility of MNs. The hypercube is logical in the sense that the logical link between two adjacent logical hypercube nodes possibly consists of multihop physical links.

The Mesh Tier is a logical 2-dimensional mesh network, by viewing each hypercube as one mesh node. In the same way, the 2-dimensional mesh is possibly an incomplete mesh; the link between two adjacent mesh nodes is logical. A mesh node becomes an actual mesh node only when a logical hypercube exists in it.

In Fig. 1, the mesh tier is drawn in circle regions, and the other tiers aren't done for clarity. In particular, the HVDB has the *non-virtual* and *non-dynamic* properties, which are similar to *reality* and *stability* properties of the backbone in the Internet respectively: (1) To realize the non-virtual property, we assume each MN can acquire its location information by using some devices such as a GPS. Then each MN can determine the VC where it resides. (2) To realize the non-dynamic property, we assume the MNs have different computation and communications capabilities to form a stable HVDB, with the *super MNs* having stronger capabilities such as multilevel radios than the *normal MNs*. It is reasonable in practice, e.g., in a battlefield, a mobile device equipped on a tank can have stronger capability than that on a foot soldier.

The HVDB model has high availability and good load balancing properties which are new QoS requirements in large-scale MANETs, due to four kinds of properties of the hypercube. Firstly, the fault tolerance property provides multiple disjoint paths for QoS routing, and small diameter facilitates small hop count of path on logical links. It brings the network with high availability. Secondly, hypercube is regular and symmetrical, and in our architecture no leader exists in a hypercube, so no single node is more loaded than any others. It is easy to achieve load balancing in the network.

4 The CAMP Algorithm

We extend the formerly defined VC to three terms, (1) *Basic virtual circle* is defined as a circle region that takes the center of a square region as its center and the diagonal of the square region as its diameter; (2) *Extended virtual circle* is defined as a circle region that takes the center of a basic virtual circle as its center, and the radius of the basic virtual circle added by a parameter value as its radius; and (3) *Reduced virtual circle* is defined as a circle region that takes the center of a basic virtual circle as its center, and the radius of the basic virtual circle subtracted by a parameter value as its radius. Since the three circles based on the same square region are concentric circles, we call the integrity of them a Virtual Circle (VC).

4.1 The Clustering Algorithm

The proposed CAMP algorithm is based on the mobility prediction and location-based clustering technique used in [18], which has been shown to form clusters much more stably than other schemes. In [18], it uses two criteria to elect an MN: longest stay time and closest to the VCC, which have been mentioned in Section 1. And our CAMP algorithm makes some improvements: (1) It uses a simple mobility prediction scheme compared with [18]. (2) It assumes that the CHs have multilevel radios other than the unit disk model in [18]. (3) It uses three kinds of VCs.

We assume that there are enough super MNs evenly dispersed in the network. This algorithm has two strategies of CH and cluster members as follows (the pseudo code is shown in Fig.2).

1. As a *candidate CH*, the MN must be a super MN, together with the above two criteria. MNs must be located within the range of the reduced VC, or it will move to the range in a certain time, which is a small system parameter value for mobility predication. If no MNs can be elected as a new CH in a VC, then the normal MNs should find their proxy CHs in their neighboring VCs. The CH will be replaced by a candidate CH while it leaves out of the reduced VC or it will do so in a certain time by mobility prediction. If no candidate CH exists in either of the two situations, then it triggers the CH re-electing process. Candidate/current CHs can determine themselves whether they are new CHs or be degraded as candidate CHs, through the mobility prediction information they received. This fact simplifies the electing process.
2. As a *cluster member*, it must be located within the basic VC, or it is currently within the extended VC and it will move to the range of the basic VC in a certain time based on mobility predication. A certain MN may belong to more than one VC at the same time for more reliable communications, as those VCs overlap with each other.

4.2 The Mobility Prediction Scheme

We assume the CHs have 2-level radios. The short radio is used for communications between normal MHs. The long radio is used for direct communications among candidate/current CHs. And we assume a natural and realistic random mobility model, which characterizes the movement of MNs in a 2-dimensional space. The movement of MNs consists of a sequence of random length mobility epochs. The speed and

direction of MNs keep constant in epochs, and vary from epoch to epoch. The acceleration γ of the same direction of previous speed exists due to the inertia effect.

The CAMP Algorithm:

1. For each candidate CH (which is a super MN):
   The MN computes its stay time and distance to the VCC;
   If the MN is located or will move to the reduced VC in a certain time **Then**
      If the MN has the longest stay time and it is closest to the VCC **Then**
         The MN is elected as the new CH;
      Else The MN is regarded as a candidate CH;
   Else If no MNs can function as candidate CHs in the VC **Then**
      The normal MNs in the VC find proxy CHs in their neighboring VCs;
   If the CH leaves the basic VC, or it will do so in a certain time **Then**
      If candidate CHs exist in the VC **Then**
         The CH is replaced by a certain candidate CH;
      Else Start to re-elect a new CH;
2. For each cluster member (either a super MN or a normal MN):
   If the MN is located within the VC, or the MN is located within the extended VC and it will move to the basic VC in a certain time **Then**
      The MN is a cluster member of the VCC;
   If the MN is located within more than one VC simultaneously **Then**
      The MN is cluster members of all these VCCs.

Fig. 2. The virtual Circle-based clustering Algorithm with Mobility Prediction (CAMP)

Each MN detects the change of its speed and moving direction. If they change, i.e., if a new epoch starts, then it records its current 2-dimentional coordinate (x, y), speed V and direction θ, current time t', and the changing duration time Δ. Then the interval time T between two epochs can be got by $T_i=t'_{i+1}-t'_i$, the acceleration γ_i can be got by $(V_{i+1}-V_i)/\Delta$, where $1\le i\le P$ and P is the number of sampling points. We call this recording process as *sampling*, the recording place as *sampling point*. P is determined by each MN based on their mobility degree. As MN moves faster, P becomes larger.

Here we give some system parameters: network center coordinate $C(x_c, y_c)$, network length L, network width W, diameter of VCs D, difference value α of VC's diameter and reduced VC's diameter, β of VC's diameter and extended VC's diameter. Fig. 3 simply illustrates the HVDB model on a 2-dimentional plane.

Theorem 1: Closest to the VCC.
Given an MN m that at time t_0 its coordinate is (x_m, y_m), m is within the reduced VC. The distance from m to its corresponding VCC $d(m)$ can be attained by Formula 4.1.

$$d(m)=\sqrt{\left(\frac{\sqrt{2}}{2}D\left\lceil\frac{(x_m-x_c+\frac{L}{2})}{\frac{\sqrt{2}}{2}D}\right\rceil-\frac{\sqrt{2}}{4}D-x_m\right)^2,\left(\frac{\sqrt{2}}{2}D\left\lceil\frac{(y_m-y_c+\frac{W}{2})}{\frac{\sqrt{2}}{2}D}\right\rceil-\frac{\sqrt{2}}{4}D-y_m\right)^2} \quad \ldots (4.1)$$

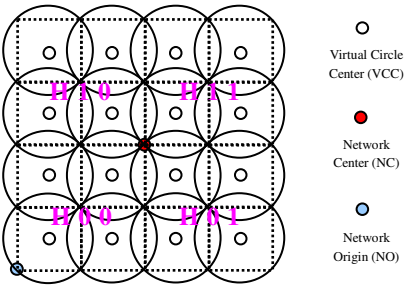


Fig. 3. The HVDB model of a 2-dimentional plane

Proof: The coordinate of network origin point (NO) $O\ (x_o, y_o)$ is $\left(\left(x_c - {}^{L}/_{2}\right), \left(y_c - {}^{W}/_{2}\right)\right)$. The length of each square is $\frac{\sqrt{2}}{2}D$, and the coordinate of VCC in which m is located is:

$$\left(\frac{\sqrt{2}}{4}D\left(2\left\lceil\frac{x_m - x_o}{\frac{\sqrt{2}}{2}D}\right\rceil - 1\right), \frac{\sqrt{2}}{4}D\left(2\left\lceil\frac{y_m - y_o}{\frac{\sqrt{2}}{2}D}\right\rceil - 1\right)\right).$$

The extrapolation method is used to predict the stay time. Extrapolation is used in the situation that function $f(x)$ is known at $a \le x \le b$, while we compute the value of $f(x)$ when $x<a$ or $x>b$. Extrapolation is always not very accurate, but it is relatively simple. Since the extrapolation in low-power multinomial is more accurate than that in high-power multinomial [7], we will use the subsection low-power interpolation method to get an approximate function.

Theorem 2: Longest stay time.
Given an MN m and a quaternion (P, v_x, v_y, T), v_x and v_y denote the speed of m on the direction of X-axis and Y-axis, got by $v_x = V\cos(\theta)$ and $v_y = V\sin(\theta)$. Suppose v_x', v_y', T', and γ' denote the future value by prediction, and current coordinate is (x_p, y_p), current speed is (v_{xp}, v_{yp}), then the future coordinate (x', y') at time t can be attained by Formulae 4.2 and 4.3 respectively, based on Newton's Laws.

$$x' = f_x(t) = \left(\frac{V_x' + V_{xp}}{2}\right)\left(\frac{V_x' - V_{xp}}{\gamma'}\right) + V_x'\left(t - \frac{V_x' - V_{xp}}{\gamma'}\right) = V_x' t - \frac{(V_x' - V_{xp})^2}{2\gamma'} \quad \ldots (4.2)$$

$$y' = f_y(t) = \left(\frac{V_y' + V_{yp}}{2}\right)\left(\frac{V_y' - V_{yp}}{\gamma'}\right) + V_y'\left(t - \frac{V_y' - V_{yp}}{\gamma'}\right) = V_y' t - \frac{(V_y' - V_{yp})^2}{2\gamma'} \quad \ldots (4.3)$$

Proof: The function value of $v_{xk} = f_x(t_k)$ at point $t_k = t_0 + \sum_{i=0}^{k} T_i$ (k=0,1,2,…,P-1) is known by sampling. We use Lagrange Multinomial to do secondary power interpolation at each section consisting of three points: (v_{xr0}, T_{r0}), (v_{xr1}, T_{r1}), (v_{xr2}, T_{r2}), where r is the section number and $0 < r \le \frac{P}{3}$, and $t_{rk} = t_{r0} + \sum_{i=0}^{k} T_{ri}$ (k=0,1,2). At each section, we get approximate sub-function from Formula 4.4:

$$v_x = f_x'(t) = v_{xr0}\frac{(t - t_{r1})(t - t_{r2})}{(t_{r0} - t_{r1})(t_{r0} - t_{r2})} + v_{xr1}\frac{(t - t_{r0})(t - t_{r2})}{(t_{r1} - t_{r0})(t_{r1} - t_{r2})} + v_{xr2}\frac{(t - t_{r0})(t - t_{r1})}{(t_{r2} - t_{r0})(t_{r2} - t_{r1})} \quad (4.4)$$

Suppose v_x' is the value got from the latest subsection's sub-function by Formula 4.4. Then the value of v_y', T', γ' can be got through the same method. So the future coordinate (x', y') at time t can be attained by Formulae 4.2 and 4.3. Suppose the VCC in which MN m is located is (X, Y). Then, the stay time t when m is located in the VC can be attained by Equation 4.5:

$$\sqrt{(x'-X)^2+(y'-Y)^2}=\frac{D-\alpha}{2} \quad \ldots (4.5)$$

If the stay time t is longer than T', that is, MN m still exists in the reduced VC when the predicted speed or move direction changes, the next quaternion of (v_x', v_y', T', γ') is predicted and the prediction process is continued. Furthermore, we predict those parameters not only based on the latest subsection's function, but also considering the varying rule of whole function $v_x = f_x(t)$ if the rule can be seen easily.

A simple mapping function is used to map each CH to a hypercube, e.g., see Fig. 3, four 2-dimentional logical hypercubes exist, HID is given as H00, H01, H10, and H11. Assume W and L are 8 km, the NO coordinate is (0, 0), then the NC coordinate is (4, 4). Given CHs' coordinate as (x, y), then the HID of the hypercube they belong to can be got by function 4.6:

$$\text{HID}=\begin{cases}\text{H00} & \{(x,y)\mid 0\le x<4, 0\le y<4\}\\ \text{H01} & \{(x,y)\mid 4\le x<8, 0\le y<4\}\\ \text{H10} & \{(x,y)\mid 0\le x<4, 4\le y<8\}\\ \text{H11} & \{(x,y)\mid 4\le x<8, 4\le y<8\}\end{cases} \quad \ldots (4.6)$$

Using the same method, we can map each CH to a hypercube node, or map each hypercube to a mesh node. Finally, a stable HVDB structure can be constructed.

5 Performance Analysis

Here we compare the proposed scheme to those ([15], [18]) with similarities that are mentioned in subsection 2.1. An (α, t) cluster framework proposed in [15] sets the criteria relying directly on path availability. Its random mobility model is similar to ours, but no acceleration exists from a velocity to another velocity. It gives pure random movements, such as the sudden stop, turn back, and sharp turn, etc., which are physically impossible in the real world for the MNs.

In [15], every node in an (α, t) cluster has a path to every other node in the cluster that will be available at time t_0+t with a probability $\geq \alpha$. Each node maintains routes to the set of adjacent clusters. But in our scheme only the CH maintains these kinds of routes, which leads to easy management. In [15], α controls the minimum level of cluster stability, and t determines the maximum cluster size. Large t that is necessary to maintain routes will reduce the path availability, thus resulting in instability. Large t also results in small number of clusters, causing the problem of small channel spatial reuse probability. It leads to ambiguity as to how large t is. But, clusters formed by our scheme are stable: firstly, a CH is not changed until it moves out of the reduced VC. Secondly, the cluster can be seen as unchanged during the alternation of CHs.

The (p,t,d) clustering model used in [18] hasn't considered the fact that the CH is relatively heavy loaded than cluster members, and then it is easily to be broken down,

which seldom occurs in our scheme due to strong capabilities of the CH. Furthermore, considering only the virtual cluster other than our proposed three kinds of VCs makes it not flexible and potentially not adaptive to node mobility. In our scheme, the close degree can be controlled by a threshold: radius of the reduced VCs, which is determined by a system parameter α. If the MNs move fast, the radius of the reduced VC should be large, that is, α should be small, to make the CH stay longer time within the cluster. Otherwise, we set a larger α to form a small reduced VC, to make sure a good CH to be closer to the VCC. In addition, the CHs can broadcast messages to other candidate CHs in the same VCs directly, due to their multilevel radios.

Theorem 3: Assume h is the number of super MNs in the MANETs, the message complexity of forming the cluster topology is $O(h)$.

Proof: To elect the CHs, only super MNs need to compute their stay time and distance to the VCC, and broadcast the result, this needs to transmit messages h times. After the CHs being elected, it should broadcast the result to notify their cluster members. The number of messages needed is the number of CHs, which is at most the number of VCs: $\frac{2LW}{D^2}$. Thus the total number of messages needed is only $h+\frac{2LW}{D^2}$.

Theorem 4: To form a cluster topology, the number of MNs needed to do mobility prediction is $O(h+\beta^2+\beta D)$.

Proof: Based on the CAMP algorithm, only super MNs have the title being candidate CHs, all the super MNs should do mobility prediction. The number of super MNs is h. The normal MNs which are located in the extended VC should predict whether it will move to the basic VC in a certain time or not. Since the MNs are evenly dispersed in the network, the number of MNs located in the extended VC is in proportion to the area of the extended VCs, which is $\frac{1}{4}\pi(D+\beta)^2-\frac{1}{4}\pi D^2=\frac{1}{4}\pi\beta^2+\frac{1}{2}\beta D$.

But in [18] each MN has to predict its mobility behavior, consuming too much memory and processing power, and generating too much traffic load.

6 Conclusions

The proposed CAMP algorithm facilitates to form a stable HVDB in large-scale MANETs. Our previous work in [21] shows that the desirable characteristics of hypercube networks, especially for the fault tolerance, small diameter, regularity and symmetry, bring high availability and good load balancing to QoS routing in large-scale MANETs. In our future work, we will develop effective and efficient QoS routing algorithms by combining the proposed CAMP algorithm with more traditional QoS mechanisms such as resource reservation and admission control.

Acknowledgment

This work is supported in part by the Hong Kong Polytechnic University Central Research Grant *G-YY41*, and in part by the University Grant Council of Hong Kong under the CERG Grant PolyU *5170/03E*.

References

1. P. Basu, N. Khan, T.D.C. Little, "A mobility based metric for clustering in mobile ad hoc networks," *Proc. Distributed Computing Systems Workshop*, 2001.
2. S. Butenko, X. Cheng, D.Z. Du, P. Pardalos, "On the construction of virtual backbone for ad hoc wireless network", *Proc. 2nd Conference on Cooperative Control and Optimization,* 2001.
3. M. Chatterjee, S.K. Sas, D. Turgut, "An on-demand weighted clustering algorithm (WCA) for ad hoc networks," *Proc. IEEE GLOBECOM 2000,* Vol. 3, pp. 1697-1701, Nov.-Dec. 2000.
4. Y.P. Chen, A.L. Liestman, "A zonal algorithm for clustering ad hoc networks," *International Journal of Foundations of Computer Science,* Vol. 14, No. 2, pp. 305-322, 2003.
5. R. Friedman, S. Manor, K. Guo, "Scalable stability detection using logical hypercube," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 13, Issue 9, pp. 972-984, Sept. 2002.
6. M. Gerla, J.T.-C. Tsai, "Multicluster, mobile, multimedia radio network," *ACM/Baltzer Wireless Networks*, Vol. 1, Issue 3, pp. 255-265, 1995.
7. R.W. Hornbeck, "Numerical Methods," Quantum publishers Inc, 1975.
8. T.-C. Hou, T.-J. Tsai, "An access-based clustering protocol for multihop wireless ad hoc networks," *IEEE Journal on Selected Areas in Communications*, Vol. 19, Issue 7, pp. 1201-1210, Jul. 2001.
9. H.P. Katseff, "Incomplete hypercubes," *IEEE Transactions on Computers*, Vol. 37, No. 5, pp. 604-607, May 1988.
10. U.C. Kozat, G. Kondylis, B. Ryu, M.K. Marina, "Virtual dynamic backbone for mobile ad hoc networks," *Proc. IEEE ICC 2001*, Vol. 1, pp. 250-255, Jun. 2001.
11. T.J. Kwon, M. Gerla, V.K. Varma, M. Barton, T.R. Hsing, "Efficient flooding with passive clustering-an overhead-free selective forward mechanism for ad hoc/sensor networks," *Proceedings of the IEEE,* Vol. 91, Issue 8, pp. 1210-1220, Aug. 2003.
12. J. Liebeherr, B.S. Sethi, "A scalable control topology for multicast communications," *Proc. INFOCOM 1998*, Vol. 3, pp. 1197-1204, Mar.-Apr. 1998.
13. W. Lou, J. Wu, "A cluster-based backbone infrastructure for broadcasting in MANETs," *Proc. Workshop on Wireless, Mobile, and Ad Hoc Networks (in conjunction with IPDPS)*, Apr. 2003.
14. J.-H. Lin, C.-R. Dow, S.-F. Hwang, "A distributed virtual backbone development scheme for ad-hoc wireless networks," *Wireless Personal Communications* (Kluwer Academic), Vol. 27, pp.215-233, Sept. 2003.
15. A.B. McDonald, T.F. Znati, "A mobility-based framework for adaptive clustering in wireless ad hoc networks," *IEEE Journal on Selected Areas in Communications*, Vol.17, pp. 1466-1487, Aug. 1999.
16. A. Rowstron, P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware),* Heidelberg, Germany, pp. 329-350, Nov. 2001.
17. M. Schlosser, M. Sintek, S. Decker, W. Nejdl, "A scalable and ontology-based P2P infrastructure for semantic web services," *Proc. Second International Conference on Peer-to-Peer Computing (P2P 2002)*, pp. 104-111, Sept. 2002.
18. S. Sivavakeesar, G. Pavlou, A. Liotta, "Stable clustering through mobility prediction for large-scale multihop intelligent ad hoc networks," *Proc. IEEE WCNC 2004,* Vol. 3, pp. 1488-1493, Mar. 2004.

19. I. Stoica, R. Morris, D. Liben-Nowell, D.R. Karger, M.F. Kaashoek, F. Dabek, H. Balakrishnan, "Chord: A scalable peer-to-peer lookup protocol for Internet applications," *IEEE/ACM Transactions on Networking*, Vol. 11, Issue 1, pp. 17-32, Feb. 2003.
20. J. Wu and F. Dai, "A distributed formation of a virtual backbone in ad hoc networks using adjustable transmission ranges," *Proc. IEEE ICDCS 2004*, pp. 372-379, 2004.
21. G. Wang, J. Cao, L. Zhang, K.C.C. Chan, J. Wu, "A novel QoS multicast model in mobile ad hoc networks," *Proc. 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2005),* Denver, Colorado, USA, pp. 206-213, Apr. 2005
22. K. Xu, X. Hong, M. Gerla, "Landmark routing in ad hoc networks with mobile backbones," *Journal of Parallel and Distributed Computing*, Vol. 63, pp. 110-122, 2003.
23. B.Y. Zhao, L. Huang, J. Stribling, S.C. Rhea, A.D. Joseph, J.D. Kubiatowicz, "Tapestry: A resilient global-scale overlay for service deployment," *IEEE Journal on Selected Areas in Communications,* Vol. 22, Issue 1, pp. 41-53, Jan. 2004.

Mobility-Aware On-demand Global Hosts for Ad-Hoc Multicast

Chia-Cheng Hu[1], Eric Hsiao-Kuang Wu[2], Gen-Huey Chen[1], and Chiang Jui-Hao[1]

[1] Department of Computer Science and Information Engineering, National Taiwan University, No. 1, Sec. 4, Roosevelt Rd., Taipei, Taiwan
jjhwu@ksts.seed.net.tw, ghchen@csie.ncnu.edu.tw
[2] Department of Computer Science and Information Engineering, National Central University, Chung-Li, Taiwan
hsiao@csie.ncu.edu.tw

Abstract. Recent routing protocols and multicast protocols in large-scale mobile ad-hoc networks (MANETs) adopt two-tier infrastructures to avoid the inefficiency of the flooding. Hosts with a maximal number of neighbors are often chosen as backbone hosts (BHs) to forward packets. Most likely, these BHs will be traffic concentrations/ bottlenecks of the network. In addition, since host mobility is not taken into consideration in selecting BHs, these two-tier schemes will suffer from more lost packets if highly mobile hosts are selected as BHs. In this paper, a new multicast protocol is proposed for multicast services in a large-scale MANET. In the proposed protocol, hosts with fewer hops and longer remaining connection time to the other hosts will be selected as BHs. The objective is not only to obtain short multicast routes, but also to construct a stable two-tier infrastructure with fewer lost packets.

1 Introduction

Recently, several multicast protocols for mobile ad-hoc networks (MANETs) have been proposed in the literature [1-7]. They can be classified into two categories: tree-based protocols and mesh-based protocols. Tree-based protocols build a tree for each multicast group, whereas mesh-based protocols create a mesh of hosts for forwarding packets between multicast members. The protocols proposed in [1-5] belong to tree-based protocols, and the other two proposed in [6, 7] belong to mesh-based protocols. For both, adding a new member into an existing multicast group will cause the flooding of a join request message over the entire network. The flooding process is time-consuming and bandwidth-consuming, especially, for a large-scale MANET.

To avoid the inefficiency of flooding, two-tier infrastructures [8-12] were adopted for routing/multicasting in large-scale MANETs. Some of hosts, named backbone hosts (BHs), were selected and responsible for managing the flooding, maintaining the infrastructures and determining the routes. The protocols proposed in [8-12] all selected BHs by finding dominating sets with a maximal number of neighbors. Since most of packets are initiated and processed by BHs, a proper way to select BHs is

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 375 – 384, 2005.

crucial to a two-tier infrastructure. Since only one host is allowed to broadcast within a transmission range at a time, the BHs determined in [8-12] are likely to be traffic concentrations/bottlenecks of the networks. They also suffer from frequent changes in highly mobile MANETs. Frequent changes of BHs adversely affect the performance of the networks. Once a host with high mobility is selected as a BH, the infrastructure may become fragile and the performance may decline dramatically. Therefore, host mobility should be considered an important factor when the infrastructure is being constructed. The two-tier protocols proposed in [8-12] all suffer from more lost packets caused by highly mobile BHs.

We adopt different approaches for selecting BHs compared with [8-12] in which the hosts with a large number of neighbors are selected as BHs. First to determine shorter multicast routes, we select the BHs with minimal hops distant to other hosts. To obtain shorter routes is one of the major concerns for most existing routing/multicast protocols. It reduces the number of hosts participating in packet forwarding so as to lower bandwidth-consumption and shorten transmitting latencies from sources to destinations. Particularly for large scale multicast services, the issue of shorter routes must be carefully scrutinized since the BHs attached by a server are necessary to transmit packets to multiple destinations over the network. Second, the selected BHs should be stable to the member of multicast groups so as to reduce the frequencies of re-selecting BHs, re-attaching BHs and re-determining multicast routes.

In this paper, we select BHs with shorter hops and longer remaining connected time to the other hosts. The problem of selecting BHs is formulated as a 0/1 integer linear programming (ILP). Second, we propose a new multicast protocol, named On-demand Global Hosts for Ad-hoc Multicast (OGHAM for short). The protocol is proposed to approximate the above optimal solution by dynamically organizing the network into hierarchical infrastructures in which some BHs are selected and highly cooperative for multicast applications. Once the infrastructure is constructed on-demand and dynamically by some multicast group, these selected BHs are globally available for other multicast groups. Therefore, follow-up multicast groups are not necessary to flood again for constructing additional infrastructures.

2 Link Prediction

Since the motion of the hosts will induce serious effects to the infrastructure, BHs with less mobility are desired. The performance of OGHAM is expected to become better if it is enhanced with a mobility prediction scheme. Below we describe the mobility prediction method of [6], where the locations, speeds and directions of hosts are provided by GPS. Suppose that v_i and v_j are two hosts. Define h : the radius of the transmission range (all hosts are assumed to have the same transmission range), (x_i, y_i) $((x_j, y_j))$: the location of v_i (v_j), s_i (s_j) : the speed of v_i (v_j), θ_i (θ_j) : the moving direction of v_i (v_j) $(0 \le \theta_i \le 2\pi$ and $0 \le \theta_j \le 2\pi)$, and $t_{i,j}$: the amount of time that v_i and v_j will stay connected. Let $a = v_i\cos\theta_i - v_j\cos\theta_j$, $b = x_i - x_j$, $c = v_i\sin\theta_i - v_j\sin\theta_j$, and $d = y_i - y_j$. When v_i and v_j are neighboring, i.e., they can hear each other, $t_{i,j}$ can be estimated by

$$t_{i,j} = -(ab+cd) - \sqrt{(a^2+c^2)h^2 - (ad-bc)^2} \Big/ a^2 + c^2$$

3 Determining BHs

Suppose that there are n hosts, denoted by v_1, v_2,…, v_n, and let $d_{i,j}$ and $t_{i,j}$ be the hops and the amount of remaining connecting time from v_i to v_j, where $1 \le i \le n$ and $1 \le j \le n$. We have $d_{i,j}=0$ if $i=j$, and $d_{i,j}>0$ an integer if $i \ne j$. We assume that given constant $d_{i,j}$ is finite and $d_{i,j}=d_{j,i}$ for all pairs of i and j. We use variable $x_i=1$ ($x_i=0$) to denote that v_i is (is not) chosen to be a BH. The hosts that are not BHs are called NBHs.

To achieve the aim in which the hosts with longer remaining connecting time and shorter hops to the other hosts are selected as BHs, some host v_i with smaller weighted value of $\sum_{1 \le j \le n} d_{i,j}/t_{i,j}$ is preferred. Letting $w_i = \sum_{1 \le j \le n \text{ and } j \ne i} d_{i,j}$ and $\tau_i = \sum_{1 \le j \le n \text{ and } j \ne i} t_{i,j}$ are to sum the hops and the remaining connecting time from v_i to the other $n-1$ hosts. We aim to minimize $\sum_{1 \le i \le n} x_i w_i/\tau_i$. For the sake of hierarchical infrastructures, we require that each host is attached to exactly one BH that is at most r hops distant, where $r \ge 1$ is a predefined integer. Let $y_{i,j}=1$ ($y_{i,j}=0$) denote that host v_i is (is not) attached to BH v_j. There are two constraints induced: $\sum_{1 \le j \le n} y_{i,j} = 1$ for all $1 \le i \le n$ and $x_j - y_{i,j} \ge 0$ for all $1 \le i \le n$ and $1 \le j \le n$. Besides, we let $y_{i,j}=0$ initially if $d_{i,j}>r$. Mathematically, the problem of finding BHs can be expressed as the following 0/1 ILP.

$$\text{Minimize} \quad \sum_{1 \le i \le n} x_i w_i/\tau_i$$

subject to
$$\sum_{1 \le j \le n} y_{i,j} = 1 \text{ for all } 1 \le i \le n$$
$$x_j - y_{i,j} \ge 0 \text{ for all } 1 \le i \le n \text{ and } 1 \le j \le n$$
$$x_i \in \{0, 1\} \text{ for all } 1 \le i \le n$$
$$y_{i,j} = 0 \text{ if } d_{i,j} > r \text{ and } y_{i,j} \in \{0, 1\} \text{ if } d_{i,j} \le r \text{ for all } 1 \le i \le n \text{ and } 1 \le j \le n$$

The 0/1 ILP problem in general is known to be NP-hard. If $x_i \in \{0, 1\}$ and $y_{i,j} \in \{0, 1\}$ are relaxed to $0 \le x_i \le 1$ and $0 \le y_{i,j} \le 1$, then LP results. In the LP problem, we set $y_{i,j}=0$ if $d_{i,j}>r$, initially. The LP problem can be solved in polynomial time. Suppose that x_i^*s and $y_{i,j}^*$s are the optimal solution to the LP, where $1 \le i \le n$ and $1 \le j \le n$. In the following, we present a rounding algorithm that can round x_i^* and $y_{i,j}^*$ to x_i' and $y_{i,j}'$, respectively, where $x_i' \in \{0,1\}$ and $y_{i,j}' \in \{0,1\}$. Initially, we set $x_i' = x_i^*$ for all $1 \le i \le n$ and $X=\{ x_i^* \mid 0 < x_i^* < 1\}$. The solution (i.e., x_i' and $y_{i,j}'$) obtained by the rounding algorithm is feasible to the original 0/1 ILP with r relaxed to $2r$.

(1) Determine $x_c^* \in X$ so that $(1-x_c^*)w_c = \min\{(1-x_i^*) w_i/\tau_i \mid x_i^* \in X\}$.
/* x_c^* is determined so that the increment of the objective function induced by $x_c^*=1$ is minimum. */
(2) Set $x_c'=1$ and delete x_c^* from X.

(3) Determine $Y=\{ x_j^* \mid x_j^* \in X$ and $d_{c,j} \le r\}$.

(4) Set $x_j^{'}=0$ and delete x_j^* from X for all $x_j^* \in Y$.

/* Each v_j that is at most r hops distant from v_c is determined as an NBH. */

(5) Compute $\Delta^- = \sum_{x_j^* \in Y} x_j^* w_j / \tau_j$.

/* Δ^- is the decrement of the objective function induced by $x_j^*=0$. */

(6) Compute $\Delta^+ = (1-x_c^*) w_c / \tau_c$.

/* Δ^+ is the increment of the objective function induced by $x_c^*=1$. */

(7) If $\Delta^+ - \Delta^- > 0$ and X is not empty, determine $0<f<1$ satisfying $\sum_{x_l^* \in X} \left(x_l^* - f x_l^*\right) w_l / \tau_l \ge \Delta^+ - \Delta^-$ and then set $x_l^* = f x_l^*$ for all $x_l^* \in X$.

/* When $\Delta^+ - \Delta^- > 0$, the increment (i.e., Δ^+) of the objective function exceeds the decrement (i.e., Δ^-) of the objective function. To offset the difference (i.e., $\Delta^+ - \Delta^-$), the objective function further decreases by reducing each x_l^* to $f x_l^*$. */

(8) If X is not empty, then go to (1).

(9) For each host v_i, determine a BH, say v_p, that is closest to v_i and then set $y_{i,p}^{'}=1$ and $y_{i,q}^{'}=0$ for all $1 \le q \le n$ and $q \ne p$.

/* Each host is attached to a BH that is closest to it. */

As a consequence of step (9), if v_i is a BH, then $y_{i,i}^{'}=1$ and $y_{i,q}^{'}=0$ for all $1 \le q \le n$ and $q \ne p$. In other words, when $y_{i,j}^{'}=1$ ($i \ne j$), we have $x_i^{'}=0$ and $x_j^{'}=1$. Suppose that x_i^{**}s and $y_{i,j}^{**}$s are the optimal solution to the 0/1 ILP, where $1 \le i \le n$ and $1 \le j \le n$. Let $S^{**} = \sum_{1 \le i \le n} x_i^{**} w_i / \tau_i$, $S^* = \sum_{1 \le i \le n} x_i^* w_i / \tau_i$, and $S^{'} = \sum_{1 \le i \le n} x_i^{'} w_i / \tau_i$. Clearly, $S^{**} \le S^*$. In the following lemma, we show that the approximation ratio, which is defined as $S^{'} / S^{**}$, is bounded above by $1 + \max\{ w_i / \tau_i \mid 1 \le i \le n\} / S^{**}$.

Lemma 1. $S^{'} / S^{**} \le 1 + \max\{ w_i / \tau_i \mid 1 \le i \le n\} / S^{**}$.

Proof. We note that the rounding algorithm is iterative. We assume that there were n_k iterations executed, where $1 \le n_k \le n$. We use Δ_t^-, Δ_t^+ and S_t to denote the values of Δ^-, Δ^+ and $\sum_{x_i^* \in X} x_i^* w_i / \tau_i + \sum_{x_j^* \notin X} x_j^* w_j / \tau_j$, respectively, evaluated at the tth iteration, where $1 \le t \le n_k$. We have $S^{'} = S_{n_k}$. We first consider $1 \le t \le n_k - 1$. When $\Delta_t^+ - \Delta_t^- \le 0$, we have $S_t = S^* + \Delta_t^+ - \Delta_t^- \ge S^*$ if $t=1$ and $S_t = S_{t-1} + \Delta_t^+ - \Delta_t^- \le S_{t-1}$ if $2 \le t \le n_k - 1$. When $\Delta_t^+ - \Delta_t^- > 0$, further decrement (i.e., $\sum_{x_l \in X} \left(x_l - f x_l\right) w_l / \tau_l$) of the objective function was

made. We have $S_t=S^*+\Delta_t^+-\Delta_t^--\sum_{x_l^*\in X}\left(x_l^*-fx_l^*\right)w_l/\tau_l \le S^*$ if t=1 and $S_t=S_{t-1}+\Delta_t^+-\Delta_t^--\sum_{x_l^*\in X}\left(x_l^*-fx_l^*\right)w_l/\tau_l \le S_{t-1}$ if $2\le t\le n_k-1$. Hence we have $(S^{**}\ge)S^*\ge S_1\ge S_2\ge \ldots \ge S_{n_k-1}$. At $t=n_k$, we have $S'=S_{n_k}=S_{n_k-1}+\Delta_{n_k}^+-\Delta_{n_k}^-$ (X is empty after step (4)). If $\Delta_{n_k}^+-\Delta_{n_k}^-\le 0$, then $S'\le S_{n_k-1}$, which implies $S'/S^{**}\le 1$. On the other hand, if $\Delta_{n_k}^+-\Delta_{n_k}^->0$, then $S'\le S^{**}+\Delta_{n_k}^+\le S^{**}+\max\{(1-x_i^*)\,w_i/\tau_i \mid 1\le i\le n\}\le S^{**}+\max\{w_i/\tau_i \mid 1\le i\le n\}$, which implies $S'/S^{**}\le 1+\max\{w_i/\tau_i \mid 1\le i\le n\}/S^{**}$.

Next, we show that every NBH is attached to one BH that is at most 2r hops distant.

Lemma 2. $d_{p,q}\le 2r$ if $y'_{p,q}=1$, where $1\le p\le n$, $1\le q\le n$, and $p\ne q$.

Proof: It suffices to show that for every NBH v_p, there exists a BH that is at most 2r hops distant from v_p. Since $y'_{p,q}=1$ and $p\ne q$, we have $x'_p=0$, which further implies either $0<x_p^*<1$ or $x_p^*=0$. We first consider $0<x_p^*<1$. Suppose that there are n_k iterations executed, and let $x_{c_t}^*$ and Y_t denote the x_c^* and Y determined at the tth iteration, where $1\le n_k\le n$ and $1\le t\le n_k$. Clearly, $x_p^*\in X$. Since $x'_p=0$, we have $x_p^*\in Y_l$ for some $1\le l\le n_k$. The distance from v_p to v_{c_t}, which is a BH ($x_{c_t}^*=1$), is at most r hops. Then we consider $x_p^*=0$. Let $Z_p=\{x_j^* \mid d_{p,j}\le r,\ x_j^*>0,$ and $1\le j\le n\}$. We have Z_p nonempty, as a consequence of the constraints of the LP. Suppose $x_z^*\in Z_p$. If $x'_z=1$, the distance from v_p to v_z, which is a BH, is at most r hops. If $x'_z=0$, we have $0<x_z^*<1$. With the same arguments as the previous situation (i.e., $0<x_p^*<1$), there exists a BH that is at most r hops distant from v_z. Consequently, the distance from v_p to the BH is at most 2r hops.

4 Protocol

OGHAM adopts hierarchical architecture by selecting some BHs for multicast applications. The members of the multicast group are attached to the selected BHs which are responsible for determining multicast routes, forwarding multicast data packets, handling dynamic group membership (the clients can dynamically join or leave the group) and updating multicast routes due to host movements.

When a server (client) v_i attempts to create (join) a multicast group, v_i first tries to find a BH within a region with a radius of 2r hops centered at v_i, where $r\ge 1$ is a predefined integer. If such a BH is found, then v_i is attached to it. Otherwise, v_i broadcasts a message over a larger region, called *multicast region*, with a radius of γ hops centered at v_i for collecting neighboring information, where $\gamma\ge 2r$ is a predefined integer. Then, v_i selects BHs for the multicast region and determines the attachment from NBHs to

BHs by the method of Section 3. Also, v_i sends the list of all BHs and the neighboring information to each BH.

The BHs determine multicast routes with the neighboring information. The BH attached by the client computes the routes to the other BHs for querying the location of the server. The BHs attached by the server replies and determines the multicast routes. For determining stable multicast routes, the link entries will be discarded during the determination if the estimated connected time is expired.

In the case, the BH attached by a client cannot find the server and then executes a merging process by broadcasting its topology information to the whole network. The BH attached by server replies its topology information to client's BH. In this way, client's region and server's region are merged into a larger one and the multicast routes can be determined. The follow-up clients in this larger region will not trigger another merging.

For robustness, BHs utilize the topology information to generate a substitute route whenever a particular disconnected multicast route is detected. At the stage of multicast region creation, each BH receives and stores the topology information from the initiator host who creates the region. With the topology information, BHs can generate a substitute route if some route is disconnected.

In OGHAM, the BHs are selected as Section 2 and are evenly distributed within the constructed multicast region. If some NBHs are disconnected with their BHs, they will have higher probabilities to be re-attached other BH in $2r$ hop distance. If not, these NBHs create their own multicast region where some BHs will be selected. On the other hand, a threshold is defined to evaluate the transmission quality (the value is based on the multicast application) from the members of the multicast groups to the attached BHs. If the packet delivery ratios have dropped behind the threshold, the members also try to re-attach to other BHs. Without loss of generality, the accuracy of the neighboring information decreases gradually as the hosts move. We propose three methods to address the problem. First each host piggybacks its newly neighboring relationship onto the transmitting or forwarding packets. To raise the accuracy of the determined multicast routes, BHs update their topology information by the piggybacked information while receiving the packets. Second once the members of groups or BHs detect a disconnected route, they generate a substitute route to replace the old one. Third, the BHs attached clients re-broadcast a broadcast packet to all other BHs for re-querying server location. Since the BHs are global, all multicast groups can benefit from the correction.

When a multicast route is determined, the links constituting the route may be disconnected. We present a distinct approach. Our proposal is to rearrange the multicast routes along the transmission. The intermediate host v_f determines a new route $np_{f,d}$ to the destination v_d and compares $np_{f,d}$ with the received one $rp_{f,d}$. The shorter and more stable route ($np_{f,d}$ or $rp_{f,d}$) will be chosen to forward the received packet. In the following two cases, $np_{f,d}$ will be chosen. (1) Some hosts v_i and v_j are disconnected where v_i and v_j are adjacent in $rp_{f,d}$. (2) The hops of $np_{f,d}$ is shorter than the hops of $rp_{f,d}$ and $\min\{t_{i,j} \mid v_i \text{ and } v_j \text{ are adjacent in } np_{f,d}\} > \min\{t_{i,j} \mid v_i \text{ and } v_j \text{ are adjacent in } rp_{f,d}\}$. Generally, the hosts with fewer hops distant from the destination are likely to keep the more accuracy routing information to the destination. Figure 1(a) shows the described scenario of case (1). The source host s generates a route s-f-a-b-d to the destination d. Unfortunately, a is moving out of the transmission rang to f before f forwards the

packet to *a*. *f* rearranges the substitute route *f-c-b-d* for *f-a-b-d* such that the lost packet is avoided and *s* is unnecessary to re-route. The other scenario of case (2) is depicted in Figure 1(b). *f* determined a better route *f-c-b-d* to replace *f-e-a-g-b-d*.

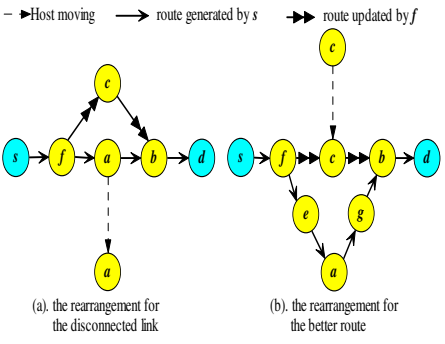


Fig. 1. Rearranging multicast routes

Table 1. Simulated parameter values

Parameter name	Meaning	Value		
		OGHAM	MCEDAR	ADB
r	Maximum hops from NBH to BH	3	1	3
γ	Maximum hops from a host to create a multicast region	7		
maxhopcount	Maximum hops to flood	20	20	20
transmission quality	The number of continuous lost data packets	3		
Neighbor discovery process period	Period for each host to broadcast a neighboring control packet		1.5 sec	1.5 sec
R	Robustness factor		2	
$Nlff$	Time window for computing normalizing link failure frequency			3 sec
α	Smoothing factor			0.6

5 Simulation

The simulation is implemented using the Network Simulator 2 package (ns-2). The simulation environment models a large- scale MANET of 200 hosts which are randomly spread in a 2000m×2000m area. IEEE 802.11 is used as the MAC layer protocol. Each host is equipped with a radio transceiver which is capable of transmitting up to approximately 250 meters over a wireless channel. The mobility of each host is based on *random waypoint* model.

We compare OGHAM with MCEDAR and ADB. We assume that the MAC layer in each host estimates the remaining time between neighboring hosts periodically by the methods of Section 2 and the accuracy of the estimated remaining connecting time is ±10%. One server and seven clients in a multicast group are randomly chosen from 200 hosts. The simulations proceed for 300 seconds and the speed is varied from 0 to 30 meter per second. Table 1 summarizes the essential parameter values for the three protocols in these simulations. The values for MCEDAR and ADB are the same as CEDAR [9].

We use the following metrics to investigate the effectiveness, the stability of constructed hierarchical infrastructures from multicast members to the attached BHs, and receiving data packet ratios for the three protocols. (1) Number of control packets. (2) Two kinds of latencies (Transmission latency: from the server to the clients and Transmission time: between two neighboring hosts). (3) Number of data packets (the number of data packets transmitted by servers and forwarders. (4) Number of the messages *init_group* and *join_request*: It represents a measure of stability for the attachment between the members and the attached BHs. (5) Packet delivery ratios: The ratio of the number of data packets received by clients versus the number of data packets delivered from the server.

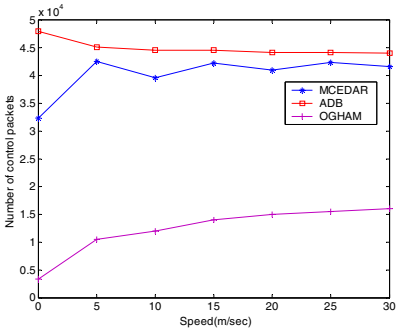


Fig. 2. Number of control packets

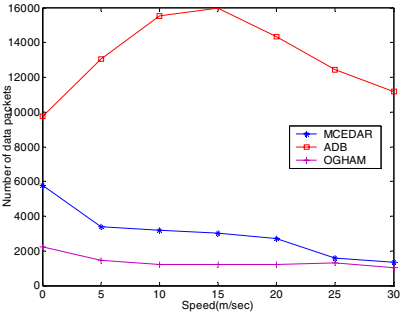


Fig. 3. Number of data packets

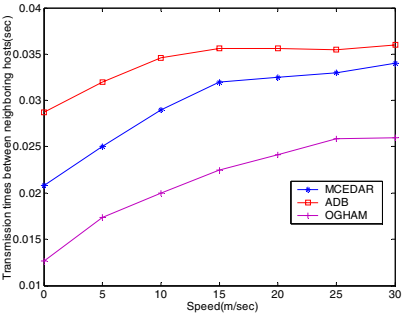


Fig. 4. Transmission times between neighboring hosts

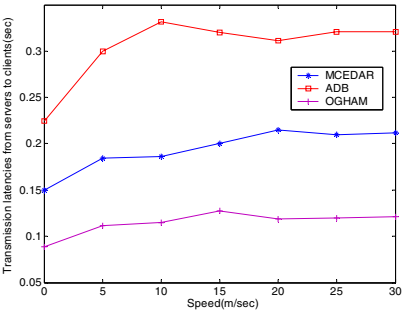


Fig. 5. Transmission latencies from servers to clients

First, we compare the effectiveness of the three protocols: (1) *To avoid flooding to the entire network:* Figure 2 illustrates that OGHAM generates fewer control packets than MCEDAR and ADB. In Figure 3, ADB transmits more data packets caused by ADB flooding data packets to all BHs. (2) *To lower traffic concentrations and bottlenecks of the network:* In MCEDAR and ADB, more control and data packets are transmitted according to Figures 2 and 3. Further, the selected BHs have move neighbors to contend channel. Figure 4 demonstrates that MCEDAR and ADB have spent more transmission latencies between neighboring hosts than OGHAM. Due to applying the ADB flooding scheme, ADB has longer latencies than MCEDAR as well. (3) *To minimize multicast relays (transmission latencies from servers to clients):* In Figure 5, OGHAM has less transmission latencies from servers to clients caused by less transmission latencies between neighboring hosts in Figure 4 and the objective of the OGHAM 0/1 ILP in which the hops from BHs to the other hosts are minimized.

Second, since MCEDAR and ADB select the BHs with maximum neighbors, the neighboring relationships are changed as hosts move. A more highly mobile environment leads the more frequency of re-selecting BHs, re-attaching to new BHs from the

multicast members and re-determining multicast routes. Figure 6 demonstrates and validates the points. Third, in Figure 7 the ratio of receiving data packets demonstrates the following two observations. First, when the speed of hosts is low, the packet delivery ratios of OGHAM are superior to MCEDAR and ADB caused by the better effectiveness (fewer packets, shorter transmission latencies) and more stable attachments in OGHAM. Second, as the speed of hosts increases, the packet delivery ratios of these three protocols decline. In a highly mobile environment, OGHAM and ADB are superior to MCEDAR. Further, OGHAM and ADB have close packet delivery ratios even though ADB floods more data packets to all BHs.

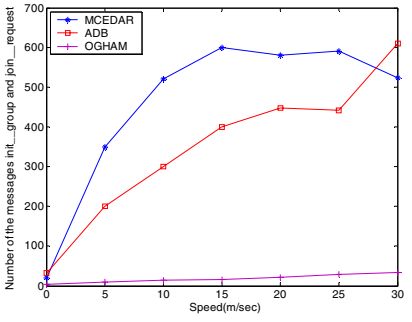


Fig. 6. Number of the messages *init_group* and *join_request*

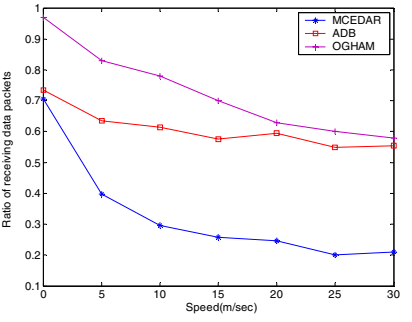


Fig. 7. Ratio of Receiving data packets

6 Conclusions

This paper has addressed a distinct methodology for selecting stable BHs by a 0/1 ILP to minimize the multicasting routes. A protocol OGHAM is proposed to approximate the optimal solution of the 0/1 ILP. The proposed protocol has the following advantages. First, contrary to the existing two hierarchical protocols MCEDAR and ADB that broadcast control packets periodically to maintain the infrastructure for all hosts, OGHAM constructs the hierarchical infrastructure on-demand. Consequently, the overheads for constructing and maintaining the infrastructure will be decreased substantially in OGHAM. Second, the selected BHs are determined by solving a 0/1 ILP. Lemma 1 shows that our 0/1 ILP solution has approximation ratio equal to $1+\max\{ w_i/\tau_i \mid 1 \le i \le n\}/S^{**}$. Thus the multicast routes, which attach NBHs to BHs, become shorter and stable. On the other hand, lemma 2 shows that each host in a constructed multicast region is at most $2r$ hops distant from a BH, which avoids a flooding to the entire network. Simulation results show that OGHAM is superior to the two existing multicast protocols (MCEDAR and ADB). Third, in opposition to the strategy of selecting BHs with maximum neighbors, OGHAM has more stable attaching relationship such that the frequencies of re-selecting BHs, re-attaching BHs and re-determining multicast routes is decreased.

Acknowledgement

This work was supported by Ministry of Economic Affair. under the "Service-oriented Information Management" project (93-EC-17-A-02-S1-029). This work was also supported by National Science Council of Taiwan under the NSC93-2524-S-008-002 Integrated knowledge Management Project.

References

1. M. S. Corson and S. G. Batsell, "A reservation-based multicast (RBM) routing protocol for mobile networks_ initial route construction phase," *ACM/Baltzer Wireless Networks*, vol. 1, no. 4, pp. 427-450, 1995
2. E. M. Belding-Royer and C. E. Perkins, "Transmission range effects on AODV multicast communication," *ACM/Kluwer Mobile Networks and Applications*, vol. 7, pp. 455-470, 2002.
3. J. Xie, R. R. Talpade, A. Mcauley, and M. Liu, "AMRoute: adhoc multicast routing protocol," *ACM/Kluwer Mobile Networks and Applications*, vol. 7, pp. 429-439, 2002.
4. S. K. S. Gupta and P. K. Srimani, "Cored-based tree with forwarding regions (CBT-FR), a protocol for reliable multicasting in mobile ad hoc networks," *Journal of Parallel and Distributed Computing*, vol. 61, no. 9, pp. 1249-1277, 2001.
5. K. Chan and K. Nahrstedt, "Effect location-guided tree construction algorithms for small group multicast in MANET," *Proceedings of the* 21st *International Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, 2002, pp. 1180-1189.
6. S. J. Lee and M. Gerla, "On-demand multicast routing protocol in multihop wireless mobile networks," *ACM/Kluwer Mobile Networks and Applications*, vol. 7, pp. 441-453, 2002.
7. J. J. Garcia-Luna-Aceves and E. L. Madruga, "The core-assisted mesh protocol," *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 1380-1394, 1999.
8. U. C. Kozat, G. Kondylis, B. Ryu, and M. K. Marina, "Virtual dynamic backbone for mobile ad-hoc networks," *Proceedings of the IEEE International Conference on Communications*, vol. 1, 2001, pp. 250-255.
9. P. Sinha, R. Sivakumar, and V. Bhanghavan, "CEDAR: a core-extraction distributed ad-hoc routing algorithm," *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 1454-1465, 1999.
10. R. Sivakumar, B. Das, and V. Bharghavan, "Spine routing in ad-hoc networks," *Cluster Computing*, a special issue on mobile computing, vol. 1, no. 2, pp. 237-248, 1998.
11. P. Sinha, R. Sivakumar, and V. Bhanghavan, "MCEDAR: multicast core-extraction distributed ad-hoc routing," *Proceedings of the IEEE Wireless Communications and Networking Conference*, 1999, pp. 1313-1317.

Bottom Up Algorithm to Identify Link-Level Transition Probability

Weiping Zhu

ADFA, The University of New South Wales, Australia
weiping@cs.adfa.edu.au

Abstract. Network tomography aims to obtain network characteristics by end-to-end measurements. Most works carried out in the past focused on the methods and methodologies to identify some of the characteristics, such as loss rate, delay distribution, etc. which are typical *static* statistical variables showing long-term network behaviors. In contrast to the previous works, we in this paper turn our attention to dynamic characteristics, e.g. transition probability of each link, which unveil the temporal correlation of traffic flows. Those dynamic characteristics could be more important than those static ones since the temporal information can be used in prediction. Apart from that, those characteristics are essential to many other issues, including the models used in network tomography. To identify transition probabilities by end-to-end measurements and in a real-time manner is a challenging task although the problem can be formulated by a hidden Markov model (HMM). Instead of using Baum-Welch algorithm to identify the transition probabilities because it needs a long execution time, we propose a new method that consider the correlations observed by receivers to obtain the transition probabilities in a simple and real-time manner. The proposed method is equal to a closed form solution that makes it a candidate for real-time network control.

1 Introduction

Network characteristics, such as loss rate, delay distribution, available bandwidth, are important to network design and performance evaluation. Due to the distributed management of the Internet, an organization cannot access the network devices (switches or routers) of another organization to obtain traffic related data. Apart from that, commercial interests often prohibit ISPs to exchange this type of data with their competitors. All of those make it hard, if not impossible, to study the traffic interactions between networks in a large scale. Apart from those, some characteristics, such as the delay of a path, cannot be obtained from network devices even with the privilege to access all related devices. Network tomography tends to develop methods or methodologies to obtain various characteristics by end-to-end measurements. A number of studies have been carried out to identify link-level loss rates, delay distribution, etc. in the past few years [1], [2], [3], [4], [5]. In contrast to the previous works, our attention in this paper is turned to dynamic network characteristics and in particular focused on

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 385–394, 2005.

identifying link-level transition probability by end-to-end measurements since this information unveils the temporal correlation of a traffic flow. If this information can be obtained in real time, it can be used in traffic engineering, such as load balancing. To achieve this goal, an innovative method is proposed in this paper to estimate the transition probability that is fast and consistent.

It has been widely agreed that the losses occurred on a link have strong temporal correlation, e.g. if a packet is lost on a link, the immediately followed packet on that link has a higher probability to be lost than a packet that follows a passed packet. The transition probability that shows the temporal dependence of consecutive packets on losses or other events can provide an accurate model to model the losses of a link, and provide the ability to predict future state. Obtaining the transition probabilities of losses by end-to-end measurements is a challenge problem although it can be formulated by a hidden Markov model (HMM) since the complexity of network structures. Baum-Welch algorithm is one of the most popular methods used in HMMs to find transition probabilities, which uses an iterative procedure to search for a solution in a multi-dimensional space that can take considerable amount of time. In addition, Baum-Welch algorithm may converge at a local maximum since it is based on the expectation and maximization (EM) algorithm.

The contribution of this paper are two folds. Firstly, in order to obtain the transition probability in real time, we propose a simple and fast method that considers the unique feature of the correlations created by multicast on receivers attached to leaf nodes, and apply it to infer the transition probability. Rather than using iterative approximation as Baum-Welch algorithm, the proposed method can find the transition probability of a link directly from observations and is equivalent to a closed form solution that makes it a candidate for real-time network control. Secondly, the proposed method takes a divide-conquer approach to complete its estimation, it factories the multicast tree into a number of pieces, one for a link. Each piece is modelled by a factorial HMM (FHMM), its transition probability can be sought independently. Therefore, the process can be carried out in a parallel and distributed manner, which further speed up the processing.

The rest of the paper is organized as follows. In Section 2, we introduce HMM in loss tomography and the principle used in statistical inference. In Section 3, we detail the bottom up approach to infer the transition probability. Section 4 presents the related work in network tomography. The last section is devoted to concluding remark, it also contain our current and future work in line of measuring network performance.

2 Problem Formulation

There are two methods to send probes to receivers, i.e., multicast and unicast, and the multicast approach is more scalable than the unicast one since it avoids the repeated transmission of the same packet on the same link. Due to this, we in this paper only consider the multicast approach to create related data. However, this method can be easily extended to the unicast situation.

The multicast tree used to send probes to receivers can be abstracted by a three-element tuple (V, E, Θ). The first two elements represent the nodes and links that have the same definitions as that in graph theory, i.e., $V = \{v_0, v_1, ...v_n\}$ is a set of nodes, which correspond to routers and switches in a network, $E = \{e_1, ..., e_n\}$ is a set of links that connect the elements of V to form a network. As a regular tree, we assign a unique number to each link, starting from 1 to n, we also assign a unique number to each node, starting from 0 to n. The two sets of numbers map each other as follows: link 1 connects node 1's parent (node 0) to node 1, link 2 connects node 2's parent to node 2, and so on. To assist the following discussion, let F_x denote the parent node of node x, these two nodes are called the parent and child nodes of link x in the rest of the paper.

When probes are sent to receivers via the network, their arrivals at receivers can be treated as samples collected from the traffic that is flowing on the path connecting the source to the receivers. If the probes are periodically sent from the source located at the root, even-spaced samples are collected at receivers, although network delay can lead some receivers observe the probes earlier than others, even some of the probes can be lost on the way to the receivers. Those observations forms a Markov chain as shown in Figure 1, which are incomplete with regards to the internal nodes. We aim to identify the transition probability of each link, including those that cannot be directly observed, from the incomplete observations. This problem can be modeled by a HMM, in which the internal states are not observable and needed to be estimated from observations conducted at receivers. As pointed in [6], network traffic has some type of temporal locality, i.e., a link would not change its state in a short period. If the probing frequency is enough, the locality can be caught correctly, and used to create accurate loss model and predict the trend of network traffic.

When a probe traverses through a network to receivers, each node in the network has only two possible outcomes: observed or missed. Let 1 denote the observed outcome, and 0 denote the other. The loss rate of link x that connects F_x to x is a conditional probability defined as

$$P(x = 0|F_x = 1)$$

Given the network topology, an observation obtained at receivers may lead to several explanations, one for a possible cause that leads to the observation.

When temporal correlation is taken into account, the situation becomes more interesting because of the impact of past states of the network, including every link, on current observation. HMM is a tool in this situation to identify the temporal correlation from incomplete observations, which is a five-tuple (Ω_X, Ω_O, A, B, π), where

1. $\Omega_X = \{q_1, ..., q_N\}$ denotes all possible states of the network;
2. $\Omega_O = \{v_1, ..., v_M\}$ denotes all possible observations that can be obtained by receivers;
3. $A = \{\alpha_{ij} = P(X_{t+1} = q_j|X_t = q_i)\}, (1 \leq i, j \leq N)$ denotes all possible transition probabilities;

4. $B = \{b_i(k) = P(o_t = v_k | X_t = q_i, t)\}, (1 \leq i \leq N)$ and $(1 \leq k \leq M)$ denotes the observation probability that links network states to observations. For the above example, B quantitatively specifies the probability of each state that creates the observation; and
5. $\pi = \{\pi_i\}(1 \leq i \leq N)$ denotes the initial distribution of states.

Given the multicast tree and possible states of each link, we have Ω_X and Ω_O. Then, our task is to determine the other three, i.e. A, B, and π, from observations and let $\lambda = (A, B, \pi)$. Baum-Welch algorithm is one of the most popular methods used in this situation to estimate λ, which takes an iterative approximating approach to search for a solution that satisfy an objective function. Unfortunately, the time spent on the search increase very quickly as the increase of the number of links. In addition, all observations collected by receivers must be sent to a central node for processing. All those make Baum-Welch algorithm unscalable and unsuitable to real-time control.

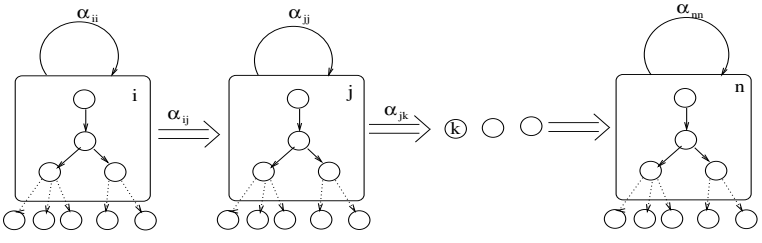


Fig. 1. Hidden Markov Model for Network Tomography

2.1 Factorial HMM

What we are concerned here is whether there are other alternatives to speed up the estimation of the transition probability, which are simple, efficient and accurate, in particular if we want to use it for network controls. Considering the correlations created by multicast, we propose a method that factories the multicast tree into a number of FHMMs, one for a link. Each of the FHMMs can be evaluated independently. Owning to this factorizing, the time spent on determining λ is substantially reduced and can be processed distributely, which makes it possible to use the identified λ to control network traffic.

When probes sent consecutively from a source to receivers, via a network, every link that forwards the probes to receivers can be modeled by a Markov chain (discrete) if its parent node observes those probes. Let x_t be the state of link x at time t. If x_t depends on the previous n-states of x, the Markov model can be described by an n-order Markov model:

$$P(x_t | x_{t-1} x_{t-2} \cdot \cdot x_{t-n}) \tag{1}$$

Note that no matter which state x is, from $t - n$ to t, a pre-condition for this Markov chain to be valid is $F_{x_i} = 1$, $i \in t - n, .., t$ for an non-homogeneous

Markov system. Otherwise, the Markov chain built on a link is broken because of $P(x = 0|F_x = 0) = 1$, which means no matter what the previous states are link x and the receivers attached to it must not observe the probe. To include the pre-condition, the above formula should be written as

$$P((x_i|x_{t-1}x_{t-2} \cdot \cdot x_{t-n})|F_{x_i} = F_{x_{t-1}} = \cdot\cdot = F_{x_{t-n}} = 1) \tag{2}$$

This shows that the transition probability of a link can only be estimated when the parent of the link receives n consecutive probes. The above formula can be simply rewritten as:

$$P((x_i x_{t-1}x_{t-2} \cdot \cdot x_{t-n})|F_{x_i} = F_{x_{t-1}} = \cdot\cdot = F_{x_{t-n}} = 1) \tag{3}$$

If the Markov chain is broken because $F_x = 0$, the Markov effect should be rebuilt from stage 0 in an non-homogeneous Markov system. Alternatively, if assuming a homogeneous system, the above restriction can be removed; in which link x remains in its previous state if $F_x = 0$ until its parent receives a probe. For a homogeneous 2-state Markov chain, its transition diagram is shown in Figure 2, and (3) can be written as:

$$P(x_i, x_{i-1}|F_{x_i} = 1, F_{x_{i-1}} = 1) \tag{4}$$

which will be mainly used in our analysis in the rest of this paper. However, it can be extended to an n-stage ($n > 2$) model.

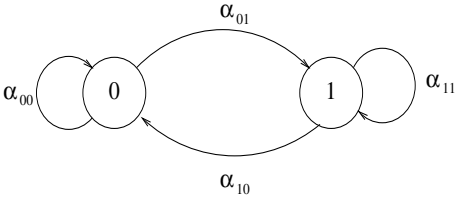


Fig. 2. State Transition Diagram

3 Bottom Up Estimation

The proposed algorithm adopts a bottom up approach to estimate the transition probability of a link, thereby, it is named after this approach.

3.1 Leaf Link

The bottom up algorithm starts its estimation from those leaf links that have all their sibling brothers' observations available. Let B_x denote the sibling brothers of link x, which is a binary set recording the observations of those receivers attached to those brothers. Each element in the set represents the observation

of a receiver for a probe as previously mentioned, 1 means the receiver observed the probe, 0 means otherwise. Let S_{B_x} represent the observation of the sibling brothers of x, which is defined as

$$S_{B_x} = \begin{cases} 1, \exists i,\ i \in B_x,\ i = 1 \\ 0, \forall i,\ i \in B_x,\ i = 0 \end{cases} \tag{5}$$

Formula (5) shows if at least one of the elements in B_x is 1, $S_{B_x} = 1$, that also implies the parent of x observed the probe. Since S_{B_x} is independent from x, the transition probability of link X can be derived:

$$\begin{aligned} \alpha_{00}(x) &= P(x_i = 0, x_{i-1} = 0 | F_{x_i} = 1, F_{x_{i-1}} = 1) \\ &= P(x_i = 0, x_{i-1} = 0 | F_{x_i} = 1, F_{x_{i-1}} = 1, B_{x_i} = 1, B_{x_{i-1}} = 1) \\ &= P(x_i = 0, x_{i-1} = 0 | B_{x_i} = 1, B_{x_{i-1}} = 1) \\ &= \frac{n(x_i = 0, x_{i-1} = 0, B_{x_i} = 1, B_{x_{i-1}} = 1)}{n(\cdot, B_{x_i} = 1, B_{x_{i-1}} = 1)} \end{aligned} \tag{6}$$

$$\begin{aligned} \alpha_{01}(x) &= P(x_i = 1, x_{i-1} = 0 | F_{x_i} = 1, F_{x_{i-1}} = 1) \\ &= P(x_i = 1, x_{i-1} = 0 | F_{x_i} = 1, F_{x_{i-1}} = 1) \\ &= P(x_i = 1, x_{i-1} = 0 | F_{x_i} = 1, F_{x_{i-1}} = 1, B_{x_i} = 1, B_{x_{i-1}} = 1) \\ &= P(x_i = 1, x_{i-1} = 0 | B_{x_i} = 1, B_{x_{i-1}} = 1) \\ &= \frac{n(x_i = 1, x_{i-1} = 0, B_{x_{i-1}} = 1)}{n(\cdot, B_{x_i} = 1, B_{x_{i-1}} = 1)} \end{aligned} \tag{7}$$

$$\begin{aligned} \alpha_{10}(x) &= P(x_i = 0, x_{i-1} = 1 | F_{x_i} = 1, F_{x_{i-1}} = 1) \\ &= P(x_i = 0, x_{i-1} = 1 | F_{x_i} = 1) \\ &= P(x_i = 0, x_{i-1} = 1 | F_{x_i} = 1, B_{x_i} = 1) \\ &= P(x_i = 0, x_{i-1} = 1 | B_{x_i} = 1) \\ &= \frac{n(x_i = 0, x_{i-1} = 1, B_{x_i} = 1)}{n(\cdot, B_{x_i} = 1)} \end{aligned} \tag{8}$$

$$\begin{aligned} \alpha_{11}(x) &= P(x_i = 1, x_{i-1} = 1 | F_{x_i} = 1, F_{x_{i-1}} = 1) \\ &= P(x_i = 1, x_{i-1} = 1) \\ &= \frac{n(x_i = 1, x_{i-1} = 1)}{n(\cdot)} \end{aligned} \tag{9}$$

The $\cdot$ appeared in the denumerators of the above formulae takes x_i and x_{i-1} positions and represents both can take all possible combinations. For instance,

$$\begin{aligned} n(\cdot, B_{x_i} = 1) = n(x_i = 0, x_{i-1} = 0, B_{x_i} = 1) + n(x_i = 0, x_{i-1} = 1, B_{x_i} = 1) + \\ n(x_i = 1, x_{i-1} = 0, B_{x_i} = 1) + n(x_i = 1, x_{i-1} = 1, B_{x_i} = 1) \end{aligned}$$

The above shows that by inspecting two consecutive observations obtained on a receiver and its sibling brothers, we can estimate the transition probability of the link connected to the receiver instead of using any iterative procedure to search for it. When the transition probabilities of all leaf links are obtained by the above method, the dimensions of the solution space is halved for a binary tree. We then can either use the traditional EM algorithm to search for the parameters of the other links or move one level up as we propose in the follows.

3.2 Internal Link

For an internal link, x, there are two simple methods to estimate its transition probability. The first one simply ignores the impact of its children's transition probabilities on its transition probability, treats link x and the subtree rooted at node x as a virtual link and use the transition probability of the virtual link as the transition probability of link x. While, the second one considers the impact of the transition probabilities of its children on its transition probability.

Link x plus the subtree rooted at it can be considered as a virtual link, denoted as V_x, and the transition probabilities of the virtual link can be estimated from the observations of $R(x)$ and $R(B_x)$, where B_x is a set of nodes that are sibling brothers of x and $R(B_x)$ consists of those receivers attached to subtree i, $i \in B$. As leaf links, the observations of V_x is strongly related to that of B_x. Let V_{x_i} denote the observation of V_x for probe i sent to it, which is defined as:

$$V_{x_i} = \begin{cases} 1, \exists i,\ i \in R(X),\ i = 1 \\ 0, \forall i,\ i \in R(X),\ i = 0 \end{cases} \tag{10}$$

Similarly, let B_{x_i} denote the observation of B_x:

$$B_{x_i} = \begin{cases} 1, \exists i,\ i \in R(B_x),\ i = 1 \\ 0, \forall i,\ i \in R(B_x),\ i = 0 \end{cases} \tag{11}$$

Then, applying the same principle used in last section, we have

$$\begin{aligned} \alpha_{00}(V_x) &= P(V_{x_i} = 0, V_{x_{i-1}} = 0 | F_{x_i} = 1, F_{x_{i-1}} = 1) \\ &= P(V_{x_i} = 0, V_{x_{i-1}} = 0 | F_{x_i} = 1, F_{x_{i-1}} = 1, B_{x_i} = 1, B_{x_{i-1}} = 1) \\ &= P(V_{x_i} = 0, V_{x_{i-1}} = 0 | B_{x_i} = 1, B_{x_{i-1}} = 1) \\ &= \frac{n(V_{x_i} = 0, V_{x_{i-1}} = 0, B_{x_i} = 1, B_{x_{i-1}} = 1)}{n(\cdot, B_{x_i} = 1, B_{x_{i-1}} = 1)} \end{aligned} \tag{12}$$

$$\begin{aligned} \alpha_{01}(V_x) &= P(V_{x_i} = 1, V_{x_{i-1}} = 0 | F_{x_i} = 1, F_{x_{i-1}} = 1) \\ &= P(V_{x_i} = 1, V_{x_{i-1}} = 0 | F_{x_i} = 1, F_{x_{i-1}} = 1, B_{V_{x_i}} = 1, B_{V_{x_{i-1}}} = 1) \\ &= P(V_{x_i} = 1, V_{x_{i-1}} = 0 | B_{V_{x_i}} = 1, B_{V_{x_{i-1}}} = 1) \\ &= \frac{n(V_{x_i} = 1, V_{x_{i-1}} = 0, B_{V_{x_{i-1}}} = 1)}{n(\cdot, B_{V_{x_i}} = 1, B_{V_{x_{i-1}}} = 1)} \end{aligned} \tag{13}$$

$$\begin{aligned}\alpha_{10}(V_x) &= P(V_{x_i}=0, V_{x_{i-1}}=1 | F_{x_i}=1, F_{x_{i-1}}=1)\\ &= P(V_{x_i}=0, V_{x_{i-1}}=1 | F_{x_i}=1)\\ &= P(V_{x_i}=0, V_{x_{i-1}}=1 | F_{x_i}=1, B_{V_{x_i}}=1)\\ &= P(V_{x_i}=0, V_{x_{i-1}}=1 | B_{V_{x_i}}=1)\\ &= \frac{n(V_{x_i}=0, V_{x_{i-1}}=1, B_{V_{x_i}}=1)}{n(\cdot, B_{V_{x_i}}=1)}\end{aligned} \tag{14}$$

$$\begin{aligned}\alpha_{11}(V_x) &= P(V_{x_i}=1, V_{x_{i-1}}=1 | F_{x_i}=1, F_{x_{i-1}}=1)\\ &= P(V_{x_i}=1, V_{x_{i-1}}=1)\\ &= \frac{n(V_{x_i}=1, V_{x_{i-1}}=1)}{n(\cdot)}\end{aligned} \tag{15}$$

We can use the transition probabilities of V_x as the transition probabilities of link x, if the consecutive simultaneous losses on the links embedded in subtree x is impossible.

Note that the transition probabilities of link x are not equal to that of V_x since some identical observations may be created by different state transitions. For instance, two consecutive losses on link x create the same observations at the receivers attached to subtree x as the two simultaneous consecutive losses on the subtree rooted at node x; i.e., all links that with node x as parent simultaneously lost the two probes. The bottom up method in this situation is incapable to tell the difference between these two. However, from the probability point of view, the probability of simultaneous losses is low, and consecutive simultaneous losses is lower. Therefore, using $\alpha_{00}(V_x)$ for $\alpha_{00}(x)$ is a good estimate. Using $\alpha_{01}(V_x)$ for $\alpha_{01}(x)$ we need to consider:

1. the initial state 0 is created by a loss on link x or by a simultaneous loss occurred on subtree x;
2. the destination state 1 is observable or not.

These two can compensate each other well because the gain received from ignoring the simultaneous loss in 1) is reduced by the loss of discounting the other simultaneous loss at destination state in 2); i.e. the second probe passed link x but lost in subtree x. Similarly for $\alpha_{10}(V_x)$.

$\alpha_{00}(V_x)$ this estimation can be very accurate because since the former has the consecutive simultaneous losses in a subtree cannot be distinguished from the consecutive losses at its parent link, we need to remove their impact from estimation. For the first one, once the transition probabilities of all its children, which can be a set of leaf links, or a set of subtrees, or a combination of the previous two, have been estimated, the transition probability of the subtree rooted at node X can be estimated by the similar method.

4 Related Work

Network tomography has a number of components for loss, delay, and bandwidth, respectively. Each component has its unique name to distinguish itself from

others. Loss tomography, as named, aims to find loss rates of links. It depends on sending probes to the receivers attached to the end-nodes and apply the correlation observed by the receivers to identify the loss rates of those links that form a multicast tree [7], [8], [9], [10] [3], [4]. Two methods are widely used to create correlated observations, i.e., multicast probes and unicast probes. A multicast-based method, as named, sends probes from a source to all receivers along a multicast tree that covers the interested network on a specific basis, e.g. periodically or exponentially. The observations of the receivers that share the same parent or ancestor have strong correlation because of the intrinsic nature of multicast, which creates the foundation to determine the parameters of related links. On the other hand, the unicast-based approach targets those networks that do not support multicasting. To have correlated observations, the unicast approach groups a number of probes together and send them to different receivers in a short period. If the period is small enough, there is a very low probability that a traffic surge could interrupt the packet group, which makes the group works as a multicast sent to multiple receivers. Any difference observed by the receivers is due to the different segments on their paths. Then, similar inference techniques as those used in multicast-based methods are applied to identify loss rates on the shared path and not shared paths.

Cáceres *et al.* are the pioneer to use the multicast-based approach to create correlation and subsequently find loss rates [4], [11] , [12]. They assumed a Bernoulli loss model for a link, and derived a high order polynomial to describe the relation between a node and its children. By solving the polynomial which normally requires an iterative procedure, the loss rates embedded in the polynomial can be identified.

Harfoush *et al.* and Coates *et al.* independently proposed the use of the unicast-based approach to discover link-level characteristics [13], [14]. Their simulations confirm the feasibility of this method. Coates and Nowak also suggested to use EM algorithm to estimate the correlation between packet pairs for loss rates.

A common feature of those approaches is that they all use the iterative approximating approach to find a feasible solution, and the computation time increases exponentially as th e number of hidden nodes/links, which makes those methods unscalable. However, we in this paper show the complexity can be under control when the problem can be factorized and the uniqueness of multicast is given special attention.

5 Conclusion

Network tomography as an emerging technology depends on statistical inference to identify hidden information from incomplete observations. Instead of taking an iterative approximating procedure to identify hidden information because such a procedure can take considerable amount of time and has the risk of trapping into a local optimum, we in this paper present a bottom up method to estimate the transition probability of a link that takes a step by step approach, from

bottom up. The advantage of this approach relies on its simplicity, efficiency and consistency. More, the method can be executed in a parallel and distributed manner.

References

1. W. Zhu and Z. Geng. A bottom up inference of loss rate. *Computer Communications*, 28, 2005.
2. G. Liang and B. Yu. Maximum pseudo likelihood estimation in network tomography. *IEEE trans. on Signal Processing*, 51(8), 2003.
3. M. Coates, A. Hero, R. Nowak, and B. Yu. Internet tomography. *IEEE Signal Processing Magazine*, 19(3), 2002.
4. R. Cáceres, N.G. Duffield, J. Horowitz, and D. Towsley. Multicast-based inference of network-internal loss characteristics. *IEEE Trans. on Information Theory*, 45, 1999.
5. W. Zhu. Using Bayesian Networks on Network Tomography. *Computer Communications, Elsevier Science, B.V.*, 26(2), 2003.
6. P. Barford, A. Bestavros, A. Bradley, and M. Crovella. Changes in web client access patterns characteristics and caching implications. Technical report, Boston University, 1998.
7. Felix: Independent monitoring for network survivality. Technical report, ftp://ftp.bellcore.com/pub/mwg/felix/index.html.
8. Ipma: Internet performance measurement and analysis. Technical report, http://www.merit.edu/ipma.
9. J. Mahdavi, V. Paxson, A. Adams, and M. Mathis. Creating a scalable architecture for internet measurement. In *INET'98*.
10. Surveyor. Technical report, http://io.advanced.org/surveyor.
11. R. Cáceres, N.G. Duffield, S.B. Moon, and D. Towsley. Inference of Internal Loss Rates in the MBone . In *IEEE/ISOC Global Internet'99*, 1999.
12. R. Cáceres, N.G. Duffield, S.B. Moon, and D. Towsley. Inferring link-level performance from end-to-end multicast measurements. Technical report, University of Massachusetts, 1999.
13. K. Harfoush, A. Bestavros, and J. Byers. Robust identification of shared losses using end-to-end unicast probes. In *Technical Report BUCS-2000-013*, Boston University, 2000.
14. M. Coates and R. Nowak. Unicast network tomography using EM algorthms. Technical Report TR-0004, Rice University, September 2000.

An Extended $GI^X/M/1/N$ Queueing Model for Evaluating the Performance of AQM Algorithms with Aggregate Traffic

Wang Hao[1,2] and Yan Wei[1]

[1] Network Laboratory, Department of Computer Science, Peking University, Beijing 100871, China
{wanghao, yanwei}@net.pku.edu.cn
[2] Software College, Jiangxi Normal University, Nanchang, JiangXi 330027, China
wanghao@jxnu.edu.cn

Abstract. TCP/AQM models focus on responsive long-lived TCP flows and are often used as a guideline to design new AQM algorithms. Nevertheless, the Internet traffic is aggregate. Besides responsive long-lived TCP flows, there are 70%-80% unresponsive short-lived TCP flows and UDP flows which are ignored by these TCP/AQM models. In this paper, we first extend the $GI^X/M/1/N$ queueing model with batch arrivals by thinning of input flows and then use probability distributions of aggregate traffic as input to this extended model to evaluate and compare the performance of four classical AQM algorithms: TD, RED, GRED and Adaptive RED.

1 Introduction

The traditional technique for managing the router's queue employs Tail Drop (TD) queue management. Due to TD's well-known drawbacks, Active Queue Management (AQM) has been recommended by IETF as a method for congestion avoidance [1]. RED [9] is the first AQM algorithm proposed for deployment in TCP/IP networks. In addition to RED, a lot of AQM algorithms such as GRED [2], and Adaptive RED [3] have been proposed to improve the performance of the original RED.

Related Work. Because of the importance of AQM algorithms in congestion control, many studies have been published on modeling the interaction between TCP and AQM algorithms (denoted by TCP/AQM models) [17,18], which focus on responsive long-lived TCP flows and are often used as a guideline to design new AQM algorithms by control theory [19]. Nevertheless, the Internet traffic is aggregate. Besides responsive long-lived TCP flows, there are 70%-80% unresponsive short-lived (i.e. Web-like) TCP flows and UDP flows which are ignored by these TCP/AQM models. Moreover, UDP is preferred by voice and video applications, more and more voice and video traffic will appear in the Internet, which will have prominent impact on Internet stability, as Simon has noted [20]. *However, TCP/AQM models do not take into account the effect of aggregate traffic on AQM algorithms and are not adequate to evaluate and compare the performance of AQM algorithms with aggregate traffic.*

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 395 – 404, 2005.

Bonald [8] developed a simple analytic model based on M/M/1/N queueing model with aggregate traffic to evaluate and compare the performance of TD and RED. However, the empirical study has shown that the Internet traffic is not Poisson traffic and the interarrivals of packets are random variables with Pareto distributions or heavy-tailed Weibull distributions [14,15,16]. Moreover, the empirical study has shown that the percent of packets back-to-back with the next increases with the number of active connections [16]. The arrivals of packets back-to-back with the next can be treated as batches in queueing system's view. Hence, a more sophisticated model for evaluating and comparing the performance of AQM algorithms should be based on a GI^X/M/1/N queueing model with batch arrivals [7].

Contribution of This Paper. Keeping above facts in mind, we have sufficient reason to employ Weibull or Pareto distribution as packet interarrivals of a queueing model based on GI^X/M/1/N with batch arrivals to evaluate and compare the performance of AQM algorithms with aggregate traffic. In our opinion, an AQM algorithm mainly confronts with aggregate traffic, not particularly with long-lived TCP flows. The queueing model is aiming at the scenario that a router will not only accept packets from long-lived TCP flows but also accept packets from short-lived TCP flows and UDP flows, i.e. aggregate traffic. The original GI^X/M/1/N queueing model with batch arrivals [7] accepts every packet that arrives at the router and hence cannot directly be used as a model for evaluating the performance of AQM algorithms. In this paper, we first extend the GI^X/M/1/N queueing model with batch arrivals *by thinning of input flows*. Based on the extended queueing model, a series of important metrics to assess the performance of AQM algorithms are suggested and definitely expressed as formulas of probabilities of queue length. Due to the arrival process with Pareto distributed interarrival is a popular model of self-similar processes [11], we evaluate and compared the performance of four classical AQM algorithms (TD, RED, GRED and Adaptive RED) by numerical analysis using Pareto interarrivals as input to the extended model. The main results we get are consistent with those produced from simulations or experiments [4,6,12].

The rest of the paper is organized as follows. In section 2, we deduce the extended queueing model. In section 3, the metrics to evaluate and compare the performance of AQM algorithms are present in terms of probabilities of queue length. Four classical AQM algorithms: TD, RED, GRED and Adaptive RED are assessed in section 4 using the extended queueing model. Section 5 gives the conclusions.

2 The Extended Queueing Model for Evaluating the Performance of AQM Algorithms

In order to characterize the essential properties of AQM algorithms and employ the tools of queueing model to assess the performance of AQM algorithms, it is necessary to formally define what an AQM algorithm is. After that, the relation between the interarrivals of packets before dropping and after dropping is established. Finally, the GI^X/M/1/N queueing model with batch arrivals is extended by thinning of input flows to evaluate and compare the performance of AQM algorithms.

2.1 A Formal Definition of AQM Algorithms

Similar to Bonald [8], we suppose that the packet drop probabilities depend on the instantaneous queue size rather than on the average queue size. Moreover, for batch arrival of packets, we assume that AQM algorithms use the same drop probability on all packets in the same batch. We abbreviate distribution function of random variables to **df** and probability density function of random variables to **pdf** below.

Definition 1: Assume that the buffer queue of a router can accommodate N packets (including the one being served), partition the buffer queue into m segments using a sequence of m+1 positive integers L_0, L_1, L_2, …, L_m satisfying (See Fig.1 and Fig.2) $0=L_0<L_1<L_2<\cdots<L_m=N$.

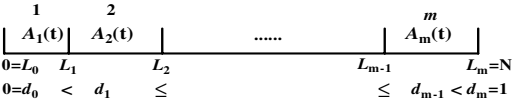


Fig. 1. The partition of the buffer queue

These segments are numbered 1 to m in sequence. Each segment corresponds to a drop probability d_i (i=0,1,…,m) satisfying (see Fig. 1)

$$0=d_0\le d_1\le d_2\le\ldots\le d_{m-1}<d_m=1$$

Suppose that the interarrival times of packets arriving at the router are independent and identically distributed random variables (**i.i.d. r.v.'s**) with df $A_0(t)$ and that there are currently k packets in the buffer queue. If $L_0\le k<L_1$, an AQM algorithm evenly drops the arriving packets with probability d_0=0. In this case, the df of interarrival times that packets arrive at the buffer queue is $A_1(t)=A_0(t)$. If $L_1\le k<L_2$, an AQM algorithm evenly drops the arriving packets with probability d_1. In this case, the df of interarrival times that packets arrive at the buffer queue is $A_2(t)$. And finally, if $L_{m-1}\le k<L_m$, an AQM algorithm evenly drops the arriving packets with probability d_{m-1}. In this case, the df of interarrival times that packets arrive at the buffer queue is $A_m(t)$. If $L_m\le k$, an AQM algorithm drops the arriving packets with probability d_m =1 (see Fig.2).

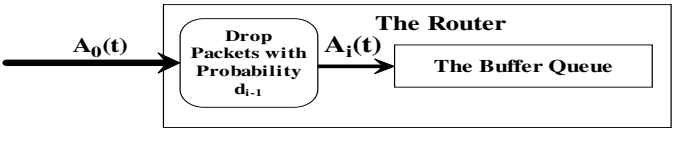


Fig. 2. The relationship between interarrival times

In this paper, the mean of df $A_i(t)$ is denoted by a_i where (i=0,1,2…m).

$$a_i = \int_0^\infty x\,\mathrm{d}A_i(x)\,. \tag{1}$$

and the pdf of $A_i(t)$ is denoted by $a_i(t)$.

2.2 Thinning of Input Flows

Because of dropping of arriving packets, the interarrival times that packets arrive at the buffer queue become longer, which is called *thinning of input flows* [13]. By thinning of input flows, we can express $A_1(t)$, $A_2(t)$, $A_3(t)$,…, $A_m(t)$ through $A_0(t)$. In the case of d_0=0, we know $A_1(t) = A_0(t)$ without any doubt. In the case of $d_i \neq 0$ (i=1,2,3,…,m-1), we have the following theorem.

Theorem 1: if $A_0(t)$ is a subexponential distribution function (a subset of heavy-tailed distribution function, such as Pareto, heavy-tailed Weibull) [11], then $A_i(t)$ (i=2,3,…,m) can be estimated by

$$A_i(t) = 1 - \mathrm{E}[v_i][1 - A_0(t)] \quad . \tag{2}$$

where $t \geq \tau$ and τ is determined by

$$\int_{\tau}^{\infty} a_0(t)dt = \frac{1}{\mathrm{E}[v_i]} \quad . \tag{3}$$

Due to the Remark 2 above, discrete r.v. v_i is evenly distributed in the set $\{1, 2, ..., \lceil 1/r_{i-1} \rceil\}$, where $r_{i-1} = 1 - d_{i-1}$ and

$$\mathrm{E}[v_i] \approx 1/2 + 1/(2r_{i-1}) \quad . \tag{4}$$

The proof of Theorem 1 is given in Appendix.

2.3 The Extended Queueing Model for Evaluating the Performance of AQM Algorithms Based on GI^X/M/1/N

In this section, we will extend the queueing model GI^X/M/1/N [7] with batch arrivals by means of thinning of input flows. We assume that packets arrive in batches of random size X with $\Pr(X = i) = g_i$ ($i \geq 1$) and mean $\bar{g}$. $A_0(t)$, $A_1(t)$, $A_2(t)$, $A_3(t)$,…, $A_m(t)$ are defined as before (see Fig.2). The mean a_i of $A_i(t)$ is defined by Eq.(1). The service time is exponentially distributed with mean $1/\mu$. The maximum number of packets allowed in the router at any time is N, where one packet is being served and the others ($\leq N - 1$) are waiting in the buffer queue. The interarrival times, the batch sizes and service times are mutually independent. Since the buffer is finite, if a batch arriving at the buffer queue does not find enough space in the buffer, then the vacant spaces are filled up and the remaining packets for which there is no space are rejected. The offered load (or traffic intensity) of the queueing system is $\rho = \frac{\bar{g}}{a_0\mu}$.

In order to evaluate the performance of AQM algorithms, we need some metrics (see section 3). These metrics all can be obtained through the probabilities q_k's and p_k's of queue length of the extended queueing model (see below).

Let q_k (k=0,1,…,N) denote the steady state probabilities of k packets in the router at prearrival epochs and p_k (k=0,1,…,N) denote the steady state probabilities of k packets in the router at an arbitrary moment. Based on the above assumptions, we have proved the following theorem.

Theorem 2. If the probabilities q_k's are known, then the probabilities p_k's can be obtained by the following equations (k=0,1,…,N - 1)

$$p_{k+1} = \frac{1}{\mu}\sum_{i=0}^{k} p_i(0) \sum_{j=k-i+1}^{\infty} g_j, 0 \le k \le N-1 \text{ and } p_0 = 1 - \sum_{k=1}^{N} p_k \ .$$

where

$$p_k(0) = \frac{q_k}{a_1}, 0 \le k \le L_1 - 1 \ ; \ p_k(0) = \frac{q_k}{a_2}, L_1 \le k \le L_2 - 1 \ ; \ \ldots \ p_k(0) = \frac{q_k}{a_m}, L_{m-1} \le k \le N \ ;$$

A more detailed description of the queueing system and the proof of Theorem 2 can be found in Ref. [21].

We will obtain q_k's using the embedded Markov chain technique as in Ref. [7]. For the limited space, the detailed discussion is omitted. Please refer to Ref. [7] for more information. The q_k ($0 \le k \le N$) can be determined by solving the system of linear equations

$$q_j = \sum_{i=0}^{N} q_i h_{i,j}, 0 \le j \le N \ . \tag{5}$$

$$\sum_{j=0}^{N} q_j = 1 \ . \tag{6}$$

where $h_{i,j}$'s are the one-step transition probabilities of the embedded Markov chain given by

$$h_{i,j} = \begin{cases} \sum_{k=j-i}^{N-i} \beta_{i+k-j}^{u(i)} g_k + \beta_{N-j}^{u(i)} \sum_{k=N-i+1}^{\infty} g_k, j > i \ge 0, j \ge 1 \\ \sum_{k=1}^{N-i} \beta_{i+k-j}^{u(i)} g_k + \beta_{N-j}^{u(i)} \sum_{k=N-i+1}^{\infty} g_k, 1 \le j \le i \end{cases} . \tag{7}$$

$$h_{i,0} = 1 - \sum_{j=1}^{N} h_{i,j} \ . \tag{8}$$

where $\beta_r^{u(i)}$ is given by

$$\beta_r^{u(i)} = \int_0^{\infty} \frac{e^{-\mu t}(\mu t)^r}{r!} dA_{u(i)}(t), r \ge 0 \ . \tag{9}$$

$u(i)$ is an indicator function used to map the current queue length i to the segment number.

$$u(i) = \begin{cases} 1, \ 0 \le i < L_1 \\ 2, \ L_1 \le i < L_2 \\ \vdots \\ m, \ L_{m-1} \le i \le N \end{cases} . \tag{10}$$

3 The Metrics to Assess the Performance of AQM Algorithms

In order to assess the performance of different AQM algorithms, performance-evaluation metrics and their corresponding formulas in terms of p_k's and q_k's are given below, where p_k's and q_k's are the same as in section 2. The following metrics: packet loss rate, the mean and variance of queue length, the distribution of the number of consecutive packet losses and its mean and variance are employed to assess the performance of AQM algorithms.

The Metrics Concerning Packet Loss Rate. We use the arbitrary packet loss rate as the metric to measure the packet loss rate of TD (see Ref. [7])

$$\text{TD's packet loss rate}=\sum_{k=0}^{N} q_k \sum_{j=N-k}^{\infty} g_j^- \quad . \tag{11}$$

where g_j^- denotes the probability of j packets ahead of an arbitrary packet within the batch and is given by $g_j^- = \dfrac{\sum_{i=j+1}^{\infty} g_i}{\overline{g}}, j \geq 0$.

Other AQM's packet loss rate is given by Bonald [8]

$$\text{Other AQM's packet loss rate}= \sum_{i=L_1+1}^{N} q_i d_{u(i)} \quad . \tag{12}$$

The Metrics Concerning Queue Length. Let r.v. ζ denote the queue length at an arbitrary moment, then we obtain

$$\overline{\zeta} \triangleq \mathrm{E}[\zeta] = \sum_{k=1}^{N} (k-1) p_k \quad . \tag{13}$$

$$\mathrm{Var}[\zeta]=\sum_{k=1}^{N} (k-1-\overline{\zeta})^2 p_k \quad . \tag{14}$$

The Metrics Concerning the Number of Consecutive Packet Losses. The probability distribution of consecutive packet losses is a critical metric for AQM algorithms as it indicates whether the global synchronization of TCP flows may occur [6,8,12]. Since AQM spreads out packet drops, it is expected to avoid the global synchronization that TD suffers [9]. In the following, we first infer TD's probability distribution of consecutive packet losses and then the other AQM's.

Tail Drop. Because the interarrival times of packets constitute a renewal process, let $V(x)$ denote the distribution of the excess life of the renewal process. According to the renewal theorem [10], we know

$$V(x) = \frac{1}{a_0} \int_0^x (1 - A_0(t)) dt \quad . \tag{15}$$

Let p denote the probability that a batch arrives within a service completion. We get

$$p = \int_0^{\infty} V(x) d(1 - e^{-\mu x}) = \int_0^{\infty} \mu V(x) e^{-\mu x} dx \quad . \tag{16}$$

Let r.v. ξ denote queue length at prearrival epochs. Let r.v. η denote the number of lost packets when a batch arrives. Let r.v. γ_{TD} denote the number of consecutive batch arrivals for TD algorithm, we obtain

$$\Pr(\eta = k) = \sum_{i=0}^{N} \Pr(\xi = i) \Pr(\eta = k \mid \xi = i) = \sum_{i=0}^{N} q_i g_{N+k-i} \quad . \tag{17}$$

Because r.v. η and r.v. γ_{TD} are mutually independent, the probability that there occur at least n times consecutive batch arrivals and each arrival loses k packets is

$$\Pr(\eta = k, \gamma_{TD} \geq n) = \Pr(\eta = k) p^{n-1}; \forall n \geq 1 \quad . \tag{18}$$

So we obtain

$$\mathrm{E}[\eta\gamma_{TD}] = \mathrm{E}[\eta]\mathrm{E}[\gamma_{TD}] = \frac{\mathrm{E}[\eta]}{1-p} \quad . \tag{19}$$

$$\mathrm{Var}[\eta\gamma_{TD}] = \frac{\mathrm{Var}[\eta] + p\mathrm{E}[\eta^2]}{(1-p)^2} \quad . \tag{20}$$

Other AQMs. Let r.v. γ_{AQM} denote the number of consecutive packet losses for any other AQM algorithms in the condition that a drop has occurred. we obtain

$$\Pr(\gamma_{AQM} \geq n) = \frac{\sum_{i=L_1+1}^{N-1} q_i d_{u(i)}{}^n}{\sum_{i=L_1+1}^{N-1} q_i d_{u(i)}}; \forall n \geq 1 \quad . \tag{21}$$

$$\mathrm{E}(\gamma_{AQM}) = \sum_{n=1}^{\infty} \Pr(\gamma_{AQM} \geq n) = \frac{\sum_{i=L_1+1}^{N-1} q_i \frac{d_{u(i)}}{1-d_{u(i)}}}{\sum_{i=L_1+1}^{N-1} q_i d_{u(i)}} \quad . \tag{22}$$

$$\mathrm{Var}(\gamma_{AQM}) = \mathrm{E}[\gamma_{AQM}^2] - (\mathrm{E}[\gamma_{AQM}])^2 = \frac{\sum_{i=L_1+1}^{N-1} q_i \frac{(1+d_{u(i)})d_{u(i)}}{(1-d_{u(i)})^2}}{\sum_{i=L_1+1}^{N-1} q_i d_{u(i)}} - (\mathrm{E}[\gamma_{AQM}])^2 \quad . \tag{23}$$

4 Numerical Analysis of AQM Algorithms

In this section, we show the numerical results of performance metrics presented in section 3 for TD, RED, GRED and Adaptive RED (abbr. to ARED).

The distributions of r.v. X for different offered loads are derived using linear regression according to the data given by Ref. [16]. For the limited space, only *part of numerical results* for N=30 are present.

Let LRate_{TD}, LRate_{RED}, LRate_{GRED} and LRate_{ARED} denote the packet loss rate for TD, RED, GRED and ARED respectively. We obtain the following results

i) If offered load<1 then $LRate_{TD}<LRate_{RED}<LRate_{GRED}<LRate_{ARED}$
ii) If offered load≥1 then $LRate_{RED}<LRate_{GRED}<LRate_{ARED}<LRate_{TD}$

But the difference between them is so small (see Fig.3) that we can not distinguish them. The above facts show that AQM mechanisms can not significantly decrease the packet loss rate comparing with TD when congestion occurs. This result accords with Ref.[4,12].

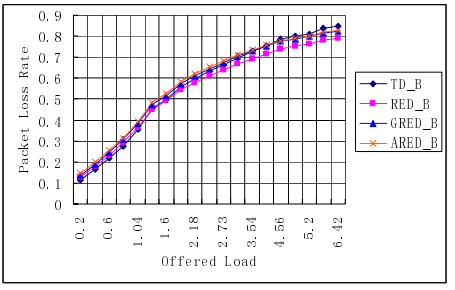


Fig. 3. The packet lost rate

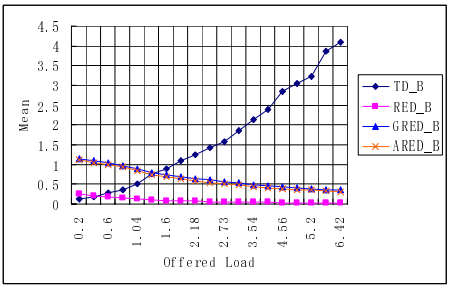


Fig. 4. The average number of consecutive packet losses

Fig.4 displays the mean of consecutive packet losses. Fig.4 shows that in the case of lower offered load, TD may not cause the global synchronization of TCP flows but RED, GRED and ARED may; in the case of higher offered load , TD may cause the global synchronization of TCP flows but RED, GRED and ARED may not. The above facts extend the results given by Bonald [8] and Iannaccone [12].

5 Conclusions

Traffic in the Internet consists of responsive long-lived TCP flows, short-lived web-like TCP flows and UDP flows. However, TCP/AQM models focus on responsive long-lived TCP flows and ignore unresponsive short-lived web-like flows and UDP flows. So, TCP/AQM model is not adequate to evaluate the performance of AQM algorithms. In this paper, we propose a novel approach to access the performance of AQM algorithms based on an extended $GI^X/M/1/N$ queueing model with batch arrivals. We use the probability distributions of aggregate traffic as input to this extended model to evaluate and compare the performance of four classical AQM algorithms: TD, RED, GRED and Adaptive RED by numerical analysis.

References

1. B. Braden, et al. Recommendations on Queue Management and Congestion Avoidance in the Internet, RFC 2309, Apr. 1998.
2. S. Floyd, Recommendations on using the gentle variant of RED, *http://www.aciri.org/floyd/red/ gentle.html*, Mar. 2000.

3. S. Floyd, Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management, *http://www.icir.org/floyd/papers.html,* Aug. 2001.
4. Tuan Anh Trinh et al, A Comprehensive Performance Analysis of Random Early Detection Mechanism, *Telecommunication System*, Vol25, pp9-31, 2004.
5. Chengyu Zhu, A Comparison of Active Queue Management Algorithms Using the OPNET Modeler, *IEEE Communication Magazine*, Jun. 2002.
6. Christof Brandauer, Comparison of Tail Drop and Active Queue Management Performance for bulk-data and Web-like Internet Traffic, *Proc. 6th IEEE Symposium on Computers and Communications*, pp. 3-5, Jul. 2001.
7. P. Vijaya Laxmi and U. C. Gupta, Analysis of finite-buffer multi-server queues with group arrivals: GI^X/M/c/N, *Queueing System*, 36(1/3): 125-140, 2000.
8. T. Bonald, Analytic evaluation of RED performance, *INFOCOM 2000*.
9. S. Floyd and V. Jacobson, Random Early Detection Gateways for Congestion Avoidance, *IEEE/ACM Transactions on Networking*, Vol.1, Issue 4, Aug. 1993.
10. Samuel Karlin and Howard M. Taylor, A First Course In Stochastic Processes, pp. 192-197, *Academic Press* ,1975.
11. Karl Sigman, Appendix: A Primer on Heavy-tailed Distributions, *Queueing Systems.* Vol. 33 pp. 261-275, 1999.
12. G. Iannaccone, M. May, and C. Diot, Aggregate Traffic Performance with Active Queue Management and Drop from Tail, *SigComm 2001*.
13. Vladimir V. Kalashnikov, Mathematical Methods in Queueing Theory, *Kluwer Academic Publishers*, 1994.
14. A. Tudjarov, D. Temkov, and T. Janevski, et al. Empirical modeling of Internet traffic at middle-level burstiness. In *Proc. 12th IEEE Mediterranean*, IEEE, 2004. 535-538
15. Jin Cao, William S. Cleveland, Dong Lin, and Don X. Sun, On the Nonstationarity of Internet Traffic, ACM SIGMETRICS 2001.
16. J. Cao, et al, The Effect of Statistical Multiplexing on the Long-Range Dependence of Internet Packet Traffic, *Bell Labs Technical Report*, 2002, http://cm.bell-labs.com/ cm/ms/departments/sia/InternetTraffic/webpapers.html.
17. V. Misra, W. Gong, and D. Towsley. Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED. In *proc. SIGCOMM'2000*, ACM, 2000, 30(4): 151-160
18. P. Tinnakornsrisuphap and J. La Richard. Characteriza-tion of Queue Fluctuations in Probabilistic AQM Mechanisms. In *Proc. SIGMETRIC'2004*, ACM, 2004. 283-294
19. C. V. Hollot, V. Misra, and D. Towsley, et al. A Control Theoretic Analysis of RED. In *Proc. INFOCOM'2001*, IEEE, 2001. 1510-1519
20. L. Simon. Keynote Speech by SIGCOMM Award Winner Simon Lam. http://www.acm. org/ sigs/sig comm/sigcomm 2004/conf_program.html, 2004
21. Wang Hao. A Queueing System GI^X/M/1/N with Batch Arrivals and Randomly Dropping Packets Mechanism. Accepted by *Mathematics in Practice and Theory*.

Appendix: The Proof of Theorem 1

For i=2,3,…,m, by definition 1, the fact that an AQM algorithm evenly drops the arriving packets with probability d_{i-1} is equivalent to accepting the arriving packets with probability $r_{i-1} = 1 - d_{i-1}$. Assume packets arrive at random instants τ_0, τ_1, τ_2 ,… , τ_n, … ; where $\tau_0<\tau_1<\tau_2<\ldots<\tau_n<\ldots$, and τ_0=0. The interval between two consecutive packets $e_k=\tau_k-\tau_{k-1}$(k=1, 2, … , n , …) is i.i.d. r.v.'s with df A_0(t)(see Fig.5).

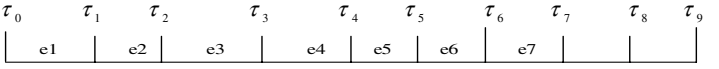


Fig. 5. The interval between two consecutive packets

After dropping of arriving packets, the interval between two consecutive packets becomes $e_1 + e_2 + \cdots + e_{\nu_i}$, where discrete r.v. ν_i is evenly distributed in the set $\{1, 2, ..., \lceil 1/r_{i-1} \rceil\}$ due to Floyd9, $r_{i-1} = 1 - d_{i-1}$ and $\mathrm{E}[\nu_i] \approx 1/2 + 1/(2r_{i-1})$.

That is to say, after dropping of arriving packets, the buffer queue accepts a packet every other $e_1 + e_2 + \cdots + e_{\nu_i}$. If the df A_0(t) of e_k is subexponential df, according to the properties of subexponential df 11 , we know that

$$\Pr(e_1 + e_2 + \cdots + e_{\nu_i} > t) \sim \mathrm{E}[\nu_i] \Pr(e_1 > t)$$

Hence, we can *estimate* df $A_i(t)$ of r.v. $e_1 + e_2 + \cdots + e_{\nu_i}$ to be $A_i(t) = 1 - \mathrm{E}[\nu_i][1 - A_0(t)]$. Because the interval between two consecutive packets become longer, we must demand $t \geq \tau$, where τ is determined by $\int_{\tau}^{\infty} dA_i(t) = \int_{\tau}^{\infty} a_i(t)dt = \int_{\tau}^{\infty} \mathrm{E}[\nu_i] a_0(t)dt = 1$.

Fair and Smooth Scheduling for Virtual Output Queuing Switches Achieving 100% Throughput

Min Song[1], Sachin Shetty[1], and Wu Li[2]

[1] Electrical and Computer Engineering Department, Old Dominion University,
231 Kaufmann Hall, Norfolk, VA 23508
{msong,sshetty}@odu.edu
[2] Multidisciplinary Optimization Branch, NASA Langley Research Center,
Mail Stop 159, Hampton, VA 23681
w.li@larc.nasa.gov

Abstract. Cell scheduling has received extensive attention. Most recent studies, however, focus on achieving a 100% switch throughput under the uniform arrivals. As the demand for quality of service increases, two important goals are to provide predictive cell latency and to reduce the output burstiness. This paper presents a new scheduling algorithm, Worst-case Iterative Longest Port First (W*i*LPF), which improves the performance of the well-known scheduler Iterative Longest Port First (*i*LPF) such that both cell latency and output burstiness are well controlled. Simulation results are provided to verify how W*i*LPF outperforms *i*LPF.

1 Introduction

Due to the high price of output queuing switches and the low throughput of input queuing switches, modern switches mostly deploy virtual output queuing (VOQ) [3][4][6] technique. In VOQ-based technique, each input organizes the buffer to N logical queues that are each associated with an output (an $N \times N$ switch is assumed in this paper); queue $q_{i,j}$ contains cells arriving at input i and destined to output j. Therefore, at each time slot, a scheduler is needed to choose a maximum of N cells from the total N^2 logical queues to forward to N outputs. The objectives of cell schedulers for VOQ-based switches are to provide 1) a 100% switch throughput, 2) fair service to each queue/flow, and 3) smooth scheduling and therefore a reduced output burstiness. Meanwhile, the schedulers must be fast and simple to implement. We assume that data are transferred across the switch fabric in fixed sized cells and that time is slotted. A speedup of one is assumed for the switch fabric.

The rest of the paper is organized as follows. Section 2 introduces the cell scheduling techniques. Section 3 presents the new cell scheduling algorithm. The simulation analysis is given at Section 4. Section 5 provides the conclusions.

2 Cell Scheduling Techniques

For VOQ-based switches, the cell scheduling problem can be viewed as a bipartite graph-matching problem [7][8]. In essence, the scheduling algorithm needs to resolve

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 405 – 413, 2005.

both *output contention* and *input contention*. Conceptually, two scheduling levels exist: the port level and the queue level. The scheduling process can be performed in either a distributed or a centralized way. In distributed scheduling, the matching decision is made independently at each port. A handshake protocol between inputs and outputs is deployed to perform the following three operations [4]: *REQUEST*—each unmatched input broadcasts its requests to the outputs that it has cells to go; *GRANT*—each unmatched output selects one request independently and issues a grant to the corresponding input; and *ACCEPT*—each input independently chooses one grant to accept.

In centralized scheduling, a unique scheduler collects information from the entire switch and makes the scheduling decision. The information collected includes queue occupancy, the waiting time of head-of-line (HOL) cells, or the cell arrival rates. An example of a centralized scheduling process is the Iterative Longest Port First (*i*LPF) with running time complexity $O(N^2)$ [3]. It includes two steps: *SORT*—all inputs and outputs are sorted based on their port occupancies and their requests are reordered according to their port occupancies, and *MATCH*—for each output and input from largest to smallest, if a request is made and both input and output are unmatched, then the scheduler will match them. The port occupancy is calculated as follows:

$$R_i(n) = \sum_{j=1}^{N} l_{i,j}(n),\ C_j(n) = \sum_{i=1}^{N} l_{i,j}(n) \tag{1}$$

where $l_{i,j}(n)$ denotes the occupancy of queue $q_{i,j}$ at cell time n. The notations $R_i(n)$ and $C_j(n)$ are the input port occupancy and output port occupancy at cell time n, respectively. The *i*LPF algorithm is a well-known scheduling algorithm that achieves 100% throughput and is stable for all admissible independent arrival processes. It is also simple to implement in hardware.

Unfortunately, most scheduling algorithms are designed solely to achieve a 100% switch throughput. They only bound the expected values, such as queue length or waiting time, by using Lyapunov stability analysis. However, bounding the expected values is not sufficient to provide predictable latency to individual connection. For example, the *i*LPF algorithm performs well for uniform traffic, but not as well for non-uniform traffic and for switches working in an asymmetric mode. It causes the latency for some queues to be unpredictable.

Another issue addressed in this paper is the output burstiness. In packet switched networks, traffic patterns become increasingly irregular as packets are multiplexed and buffered at the intermediate nodes [1][2]. Schedulers should try to smooth the traffic as much as possible. Smooth scheduling helps networks accommodate more traffic, reduce the traffic burstiness, and provide a tight end-to-end delay bound in high-speed networks. Results from our preliminary study have been presented in [9].

3 The New Scheduling Algorithm

We consider a switch that works under an asymmetric mode, i.e., the traffic distribution is non-uniform. As an example, Figure 1 shows a 2×2 VOQ-based switch, in which the arrival rates for queues are distributed as $\lambda_{1,1} = 0.89$, $\lambda_{1,2} = 0.1$, $\lambda_{2,1} = 0.1$, and $\lambda_{2,2} = 0.1$. The scenario in Figure 1 happens when the output link 1 is close to *hot-spot* servers.

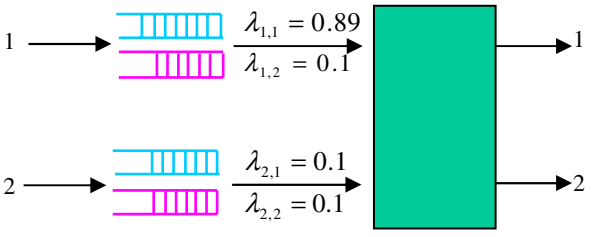


Fig. 1. A 2×2 VOQ-based switch works in asymmetric traffic mode

After the warm-up running τ slots, the following two inequities follow

$$l_{1,1}(\tau) > l_{1,2}(\tau) \text{ and } l_{2,1}(\tau) > l_{2,2}(\tau)$$

Then according to Equation (1), the following two port inequities follow

$$C_1(\tau) > C_2(\tau) \text{ and } R_1(\tau) > R_2(\tau)$$

Thus, according to *i*LPF, queues $q_{1,1}$ and $q_{2,2}$ continue receiving service until one of the two port inequities does not follow. Until then, queues $q_{1,2}$ and $q_{2,1}$ cannot receive any service. Thus, cells in queues $q_{1,2}$ and $q_{2,1}$, henceforth called *tagged queues*, experience significant delay. The reasons for this phenomenon are *1*) the *link-dependency*, i.e., whether a queue gets service or not depends on not only the traffic of its own link, but also the traffic from other links; and *2*) the *longest port preference*, i.e., *i*LPF gives preference to the queue with the longest port occupancy. There is no mechanism to guarantee a minimum amount of service to an individual queue. Consequently, tagged queues experience unexpected delays, and queues tend to receive batch service that increases the output burstiness. To alleviate these problems, we have designed a Worst-case Controller (Fig. 2) that monitors the queue occupancy and feeds back to the centralized scheduler to dynamically adapt the scheduling decision.

$$w_{i,j} = \left\lceil 1 / \lambda_{i,j} \times \max\left(\sum_i \lambda_{i,j}, \sum_j \lambda_{i,j} \right) \right\rceil$$

$$\Delta l_{i,j}(n) = l_{i,j}(n) - l_{i,j}(n-1)$$

if $[(\Delta l_{i,j}(n) = 1)$ *or* $((\Delta l_{i,j}(n) = 0) \wedge (l_{i,j}(n-1) = l_{i,j}(n-1)^+))]$

then $\Lambda l_{i,j} = \Lambda l_{i,j} + 1$

else $\Lambda l_{i,j} = 0$

if $(\Lambda l_{i,j} \geq w_{i,j})$

then mark queue $q_{i,j}$ as a worst-case queue, $\Lambda l_{i,j} = 0$

Fig. 2. Worst-case Controller (WC) [1]

[1] $l_{i,j}(n)$ and $l_{i,j}(n^+)$ represent the queue $q_{i,j}$ occupancy at the beginning and the end of cell slot n, respectively.

In particular, if WC finds that a nonempty queue $q_{i,j}$ has not received service, hence called a *worst-case queue*, for more than $w_{i,j}$ cell times, and both input i and output j are not yet matched, then the WC will match them. We call this process *worst-case matching*. If a conflict occurs among the worst-case queues, the one with the longest port occupancy gets service. Thus, a worst-case queue may need to wait, at maximum, for $2(N-2)$ slots to get service. This deterministic bound in head-of-line-cell waiting time effectively guarantees that each queue (and thus its constituent flows) receives its reserved service share and that the service time to each queue spreads as evenly as possible.

We call this property *fair and smooth scheduling*. The combination of WC and *i*LPF is called Worst-case Iterative Longest Port First (W*i*LPF) (Fig. 3), where the WC is embedded at the end of the *SORT* step of *i*LPF algorithm. This process ensures that the worst-case matching has a higher priority than the normal matching. Similar to *i*LPF, the two steps in W*i*LPF can be pipelined to keep the running time complexity of $O(N^2)$. It should be noticed that the WC effectively functions as a traffic shaper or rate controller [5].

Step I: Sorting & Worst-case Matching

1. Sort inputs and outputs based on their port occupancies
2. Reorder requests according to their input and output port occupancies
3. Run WC for each output and input from largest to smallest
   if (queue $q_{i,j}$ is a worst-case queue) and (both input and output unmatched)
   then match them

Step II: iLPF Matching

1. *for* each output from largest to smallest
2. *for* each input from largest to smallest
3. *if* (there is a request) and (both input and output unmatched)
   then match them

Fig. 3. The W*i*LPF algorithm

4 Simulation Analysis

To validate the W*i*LPF algorithm, we conducted the simulation using a discrete-event simulator written for the purpose[2]. Simulations were run by using a 3×3 VOQ-based switch. The arrival rates for the three queues at links 1 and 2 are fixed as 0.79, 0.1, and 0.1, respectively. For link 3, the arrival rates for the first two queues are both fixed as 0.1; the arrival rate for the third queue is a variable from 0.1 to 0.7. All simulation runs have been fixed at one million cell times. Both Bernoulli traffic and Bursty traffic ($E[B] = 16$) are used. Fig. 4 shows the Markov transition diagram for

[2] The simulator was designed based on the simulator used in paper [3].

the cell arrival process at link 2, where states 1, 2, and 3 represent arriving cells for $q_{2,1}$, $q_{2,2}$, and $q_{2,3}$, respectively; $p_1 = \frac{\lambda_{2,1}}{\lambda_{2,1} + \lambda_{2,2} + \lambda_{2,3}}$, $p_2 = \frac{\lambda_{2,2}}{\lambda_{2,1} + \lambda_{2,2} + \lambda_{2,3}}$, and $p_3 = \frac{\lambda_{2,3}}{\lambda_{2,1} + \lambda_{2,2} + \lambda_{2,3}}$

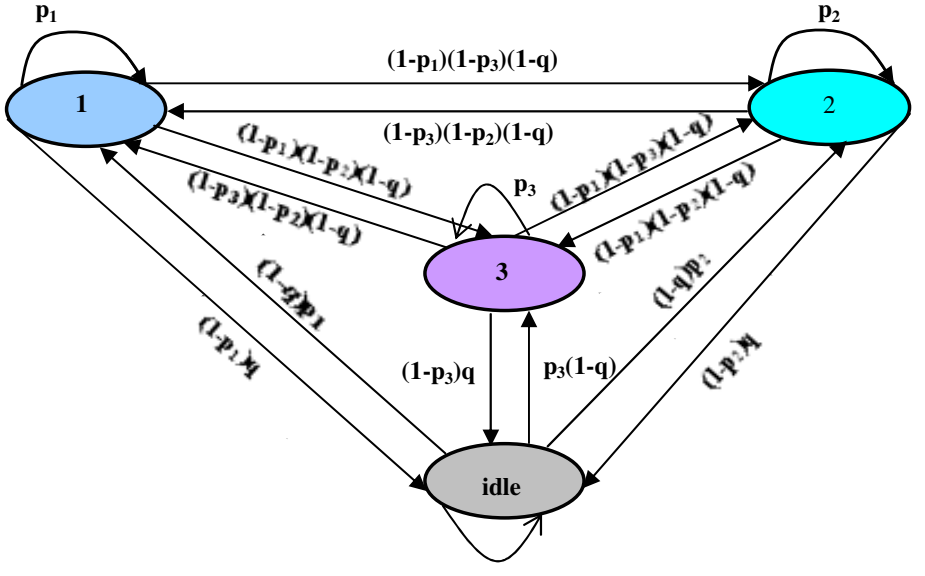


Fig. 4. Markov chain transition diagram for the cell arrival process at link 2

Tables 1 and 2 provide the average cell latency for each queue as a function of the utilization of link 3 under *i*LPF and W*i*LPF algorithms for Bernoulli traffic. Although the latencies of queues $q_{1,1}$, $q_{2,2}$, $q_{2,3}$, and $q_{3,3}$ in W*i*LPF are increased at a maximum 15 cell times, the latencies for all other queues in W*i*LPF are decreased with a maximum 42 cell times. All queue latencies are upper bounded by the expected waiting time of an *M*/*D*/1 queue, in which $d = \lambda / 2\mu(\mu - \lambda)$.

Tables 3 and 4 provide the average cell latencies for Bursty traffic. The average cell latencies for queues $q_{1,2}$, $q_{1,3}$, $q_{2,1}$, and $q_{3,1}$ in W*i*LPF are reduced for maximum six cell times and for queues $q_{1,1}$, $q_{2,2}$, $q_{2,3}$, and $q_{3,3}$ the latencies are increased for maximum four cell times.

The most significant performance improvement in W*i*LPF can be seen in the HOL cells holding time as shown in Tables 5 and 6.

Table 1. Average cell latency (in cell times) for each queue as a function of the utilization of link 3 (data are in the format of *i*LPF/W*i*LPF)

Queues	0.3	0.4	0.5	0.6
$q_{1,1}$	8.4/20.2	7.6/19	8.6/20.8	8.2/20.5
$q_{1,2}$	20.8/21.3	21.4/20.6	24.6/22	27.2/22.8
$q_{1,3}$	40.9/29.1	39.5/28.5	41/29.9	40.3/29.2
$q_{2,1}$	54/51.9	51.6/46.9	61.2/52.2	63.5/50.6
$q_{2,2}$	6.6/3.5	6/3.4	6.1/3.6	5.6/3.5
$q_{2,3}$	5.1/2.8	5.1/2.9	5.4/2.9	5.6/3.1
$q_{3,1}$	103/60.7	93.5/53.8	102.5/61	96.8/57.6
$q_{3,2}$	5.3/5.2	5.5/5.2	5.3/5.1	5.8/5
$q_{3,3}$	0.7/0.3	0.8/0.4	0.9/0.5	1.2/0.6

Table 2. Average cell latency (in cell times) for each queue as a function of the utilization of link 3 (data are in the format of *i*LPF/W*i*LPF)

Queues	0.7	0.8	0.9
$q_{1,1}$	8.6/20.9	8.75/20.8	10.7/25.3
$q_{1,2}$	30/23.1	35.8/23.6	48.6/27.2
$q_{1,3}$	40/28.3	41.2/27.7	42.9/27.9
$q_{2,1}$	69.6/50.6	77.4/52.8	101/58.9
$q_{2,2}$	5.4/3.5	5.2/3.5	5.4/3.6
$q_{2,3}$	5.9/3.3	6.7/3.8	9.9/6.1
$q_{3,1}$	95.6/57	93.1/55.9	98.9/60.2
$q_{3,2}$	6.2/5.1	7.1/5.2	8.9/5.7
$q_{3,3}$	1.6/0.8	2.6/1.3	5.5/3.4

Table 3. Average cell latency (in cell times) for each queue as a function of the utilization of link 3 (data are in the format of *i*LPF/W*i*LPF)

Queues	0.3	0.4	0.5	0.6
$q_{1,1}$	5.1/7.3	5.3/7.8	5.7/8.5	6.3/9.7
$q_{1,2}$	12.6/10.4	13.9/11.4	15/12	17.1/13.9
$q_{1,3}$	8.8/8.6	9.6/9.6	10.5/10.8	12.5/12.7
$q_{2,1}$	26.6/20.2	23.9/18.8	23.9/17.9	24/18.2
$q_{2,2}$	5.5/6.8	5.5/7	5.5/7.4	5.8/7.9
$q_{2,3}$	4.8/5.1	6.5/6.5	8.4/7.8	11.5/9.3
$q_{3,1}$	23.6/20.7	22.7/18.9	22.3/18.3	23.2/17.8
$q_{3,2}$	7.6/7.6	8.9/8.1	10.4/8.5	12.3/10.4
$q_{3,3}$	3.9/5.8	3.9/5.1	4.3/5	4.7/6.5

Table 4. Average cell latency (in cell times) for each queue as a function of the utilization of link 3 (data are in the format of *i*LPF/W*i*LPF)

Queues	0.7	0.8	0.9
$q_{1,1}$	7.1/11.1	8.1/13	10.3/15
$q_{1,2}$	20/15.2	22.5/17	28/19
$q_{1,3}$	14.7/14.3	17.7/16	24.8/21
$q_{2,1}$	24/18	24.2/19.3	25/20
$q_{2,2}$	6.4/9	7.2/10.4	9.4/16
$q_{2,3}$	15.2/12.1	18.9/14.8	27.5/20.2
$q_{3,1}$	24.9/18.2	26/18	29.2/21.4
$q_{3,2}$	14.7/12.4	17.6/15	22.8/20.4
$q_{3,3}$	5.5/7.7	6.5/9.4	9.2/15.9

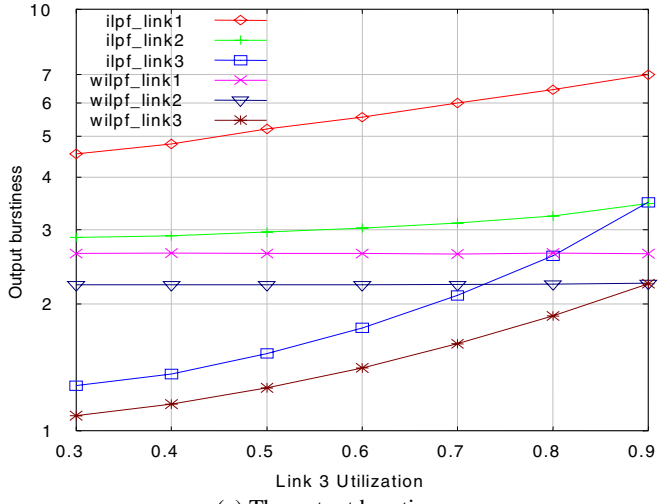
Table 5. Maximum HOL cells holding time for each queue as a function of the utilization of link 3 (data are in the format of *i*LPF/W*i*LPF)

Queues	0.3	0.4	0.5	0.6
$q_{1,1}$	39/3	31/3	46/3	34/3
$q_{1,2}$	159/10	145/10	179/10	165/10
$q_{1,3}$	199/10	180/10	187/10	230/10
$q_{2,1}$	287/11	275/11	301/11	307/11
$q_{2,2}$	26/4	27/4	35/ 4	25/4
$q_{2,3}$	59/11	67/11	85/11	81/11
$q_{3,1}$	154/10	168/10	169/10	160/10
$q_{3,2}$	106/11	93/11	113/11	92/11
$q_{3,3}$	15/4	16/4	20/4	22/ 4

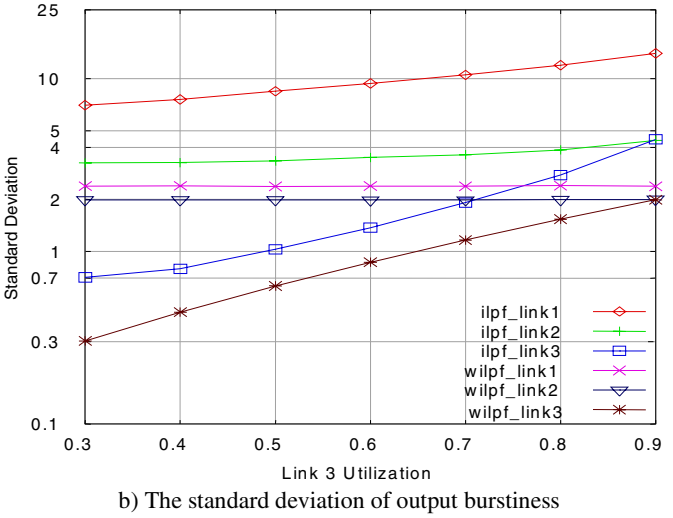
Table 6. Maximum HOL cells holding time for each queue as a function of the utilization of link 3 (data are in the format of *i*LPF/W*i*LPF)

Queues	0.7	0.8	0.9
$q_{1,1}$	44/3	38/3	40/3
$q_{1,2}$	157/10	222/10	217/10
$q_{1,3}$	176/10	196/10	188/10
$q_{2,1}$	293/11	280/11	314/11
$q_{2,2}$	35/4	38/4	32/4
$q_{2,3}$	55/11	72/11	95/11
$q_{3,1}$	213/10	221/10	304/10
$q_{3,2}$	83/11	83/11	109/11
$q_{3,3}$	21/4	23/4	31/4

Because the W*i*LPF algorithm spreads the service time to queues evenly, both the output burstiness and its standard deviation (Fig. 5) are exceedingly reduced.



(a) The output burstiness



b) The standard deviation of output burstiness

Fig. 5. The output burstiness (a) and standard deviation (b) as the function of link 3 utilization

5 Conclusions

To achieve deterministic cell latency, smooth output traffic, and a high switch throughput, we have designed a new cell scheduling algorithm, W*i*LPF, for VOQ-based switches. W*i*LPF has two components: a worst-case controller and a central-

ized scheduler. The worst-case controller monitors the queue behavior and feeds back to the centralized scheduler. The worst-case controller, which is unique to W*i*LPF, can be easily embedded into the centralized scheduler without increasing the overall running time complexity of $O(N^2)$. Analysis and simulation results suggest that W*i*LPF reduces the overall cell latency and significantly smoothes the output traffic, and keeps the same switch throughput and same running complexity as of *i*LPF. Similar to *i*LPF, the two steps in W*i*LPF can be pipelined to reduce the running time. This means that the matching algorithm operates on weights that are one slot out of date. However, it is still stable for all admissible independent arrival processes.

References

[1] A. Raha, S. Kamat, X.H. Jia, and W. Zhao, "Using Traffic Regulation to Meet End-to-End Deadlines in ATM Networks," *IEEE Trans. on Computers*, Vol. 48, Sept. 1999, pp. 917–935.

[2] L. Georgiadis, R. Guérin, V. Peris, and K.N. Sivarajan, "Efficient Network QoS Provisioning Based on per Node Traffic Shaping," *IEEE/ACM Trans. on Networking*, Vol. 4, Aug. 1996, pp. 482–501.

[3] A. Mekkittikul and N. McKeown, "A Practical Scheduling Algorithm to Achieve 100% Throughput in Input-Queued Switches," *Proc. of the IEEE INFOCOM*, Vol. 2, April 1998, pp. 792–799.

[4] T. Anderson, S. Owicki, J. Saxe, and C. Thacker, "High Speed Switch Scheduling for Local Area Networks," *ACM Trans. on Computer Systems*, Vol. 11, Nov. 1993, pp. 319–352.

[5] D. Stiliadis, and A. Varma, "Rate-Proportional Servers: A Design Methodology for Fair Queuing Algorithms," *IEEE Trans. on Networking*, Vol. 6, April 1998, pp. 164–174.

[6] N. McKeown, B. Prabhakar, and M. Zhu, "Matching Output Queuing with Combined Input and Output Queuing," *IEEE Journal on Selected Area in Comm.*, Vol. 17, June 1999, pp.1030–1038.

[7] A. C. Kam, and K.Y. Siu, "Linear-Complexity Algorithms for QoS Support in Input-Queued Switches with No Speedup," *IEEE Journal on Selected Area in Comm.*, Vol. 17, June 1999, pp. 1040–1056.

[8] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% Throughput in an Input-Queued Switch", *IEEE Trans. on Commu.*, Vol. 47, No. 8, August 1999, pp. 1260-1267.

[9] M. Song, and M. Alam, "Two Scheduling Algorithms for Input-queued Switches Guaranteeing Voice QoS," *Proc. of IEEE Global Telecommunications Conference*'01, Texas, November 25-29, 2001, Vol. 1, pp. 92–96.

Detour Path Optimization Algorithm Based on Traffic Duration Time in MPLS Network

Ilhyung Jung[1], Hwa Jong Kim[2], and Jun Kyun Choi[1]

[1] Information and Communications University (ICU),
P.O. Box 77, Yusong, Daejon, Korea
{dragon, jkchoi}@icu.ac.kr
[2] Kangwon National University,
192-1, Hyoja2-Dong, Chunchon, Kangwon-Do, Korea
hjkim@kangwon.ac.kr

Abstract. For QoS control, traffic engineering (TE) of large IP backbone networks becomes a critical issue. However, provisioning network resource efficiently through TE is very difficult for ISPs because the traffic volume usually fluctuates widely over time. The congestion especially from short-lived traffic is difficult to handle due to its bursty arrival process. In the paper, we propose a detour routing schemes for short-lived traffic when congestion occurs. Our study shows that additional hops in detour paths should be carefully restricted to avoid network resource waste under heavy load. The proposed algorithm has less packet loss probability and less resource waste because we restricted the hop count by one or two.

1 Introduction

Lots of researches have been done on the Quality of Service (QoS) to support a predefined performance contract between a service provider and end user. For QoS control, traffic engineering (TE) of large IP backbone networks becomes a critical issue in recent years. However, provisioning network resource efficiently through TE is very difficult for ISPs because the traffic volume usually fluctuates widely over time. Recent studies show that only 20% of the flows have more than 10 packets but these flows carry 85% of the total traffic [1], [4], [8]. A long-lived traffic has a less bursty arrival process, while a short-lived traffic has more bursty arrival process. A hybrid routing algorithm was proposed to reduce the overhead of routing complexity using these properties [1].

In this paper, we proposed an optimized detour path selection algorithm based on the flow duration in an environment where hybrid routing algorithm works. We also applied various routing policies for short-lived traffic and a long-lived traffic to reduce resource burden [8].

The structure of the paper is as follows. In Section 2, related works are reviewed and analyzed. Section 3 explains the methodologies of routing algorithms based on traffic duration time. In Section 4, we show simulation results under two types of networks. Conclusion and a summary are in presented in Section 5.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 414 – 422, 2005.

2 Related Works

2.1 QoS Routing and Its Restrictions in MPLS Network

For each new flow, network operator should assign a path to meet the flow's QoS requirements such as end-to-end delay or bandwidth guarantees [9]. Basically, QoS routing protocols must distribute topology information quickly and they must react to changes and minimize control traffic overhead. QoS routing also suffers from various problems such as diverse QoS specifications, dynamically changing network states in the mixture of the best-effort traffic [1], [4], [9], [6]. This makes the QoS routing complicated.

Moreover, the Internet traffic between particular points is unpredictable and fluctuates widely over time [8]. It is noted that most internet flows are short-lived, and Link-State Update (LSU) propagation time and route computation time is relatively long to handle short-lived traffic [1]. A pure MPLS solution is probably too costly from the signaling point of view because the MPLS network also consists mainly of short-lived traffic.

2.2 Hybrid Routing Scheme

Two different schemes were proposed to allow the efficient utilization of MPLS for inter-domain traffics well as the number of signaling operations [7], [13]. The first scheme is to aggregate traffic from several network prefixes inside a single LSP, and the second one is to utilize MPLS for high bandwidth flows and normal IP routing for low bandwidth flows. By introducing aggregation long-lived traffics, it is shown that performance is improved and reduced overhead via simulations [1], [4], [7], [12].

Hybrid routing algorithm, which is one of the second scheme, classifies arriving packets into flows and applies a trigger (e.g., arrival of some number of packets within a certain time interval) to detect long-lived flows [1]. Then, the router dynamically establishes a shortcut connection that carries the remaining packets of the flow. The hybrid routing was introduced in [4], [5], [7].

2.3 Detour Path Routing

In the proposed detour path routing, which is one of the alternative path routing (APR), when the set up trial on the primary path fails, the call can be tried on alternative path in a connection oriented network. Simple updating of network status information may increase scalability, reduce routing information accuracy and thus increase connection rejection rate. Detour path routing is a technique that can help to compensate for the routing inaccuracy and improve routing performance [4].

Detour paths will use more resources than primary paths. Therefore, under heavy traffic load, much use of detour may result in congestion especially for long-lived traffic [3], [9], [4]. Our Algorithm uses detour path only for the short-lived traffic to reduce the negative effect of the APR.

3 Detour Path Optimization Algorithm for Short-Lived Traffic in MPLS Network

This section describes the details of the proposed algorithm in load balancing routing. By default, router forward arriving packets onto the path selected by a traditional link-state routing protocol. When the accumulated size or duration of the flow exceeds a threshold (in terms of bytes, packets or seconds), the router would select a dynamic route for the remaining packets in the flow depending on the bandwidth provisioning rule [1]. A variety of load-sensitive routing algorithms can be used in path selection for long-lived flows to achieve high resource utilization and avoid congestion problems. From the insights of previous studies of load-sensitive routing, a dynamic algorithm favors short paths in order to avoid consuming extra resources in the network [6]. So, we should choose the widest-shortest path for long-lived traffic because long routes make it difficult to select a feasible path for subsequent long-lived flows. When a link is overloaded, the router distributes some of the traffic over less-busy links by automatically "bumping" some packets in "non-optimal" directions. Packets routed in a non-optimal direction take more than the minimum required number of hops to reach their destination, however the network throughput is increased because congestion is reduced.

When an LSR receives a packet in MPLS network, the LSR usually attempts to minimize network congestion by routing the packet to the output port with the fewest packets queued for transmission. The number of packets queued at LSRs output ports indirectly convey information about the current load. But this routing scheme requires a number of signaling operations, link-state update messages and route computation. Therefore, simple, fast and robust algorithm is essential for bursty and unpredictable short-lived traffic while load-sensitive routing is used for long-lived traffic.

Fig. 1 show how the proposed detour path scheme works. Each LSR checks that the link which is forwarding the packet is congested or not for short-lived traffic. If the link is congested, we mark the congestion link infeasible and re-try the route selection. If this procedure returns "success," we check how many additional hops the path consumes. If the new route does not exceed the Max-Hopcount that is our restriction, additional hop counts, based on the original path, the new path takes the congestion test again. This procedure is important to prevent a looping problem or a resequence problem. If the new-extracted path has fewer hops than the Max-Hopcount, we re-try to forward the traffic after the re-selected path is checked. If the node can not find the re-extracted path, we mark that node as an infeasible node. Then we return the traffic to the previous node that is called crank-back node.

The suggested algorithm for short-lived traffic works on a simple distributed routing scheme. The advantage of distributed routing for the short-lived traffic is that the routers need not keep persistent state for paths or participate in a path set-up procedure before data transfer begins because path set-up procedure is relatively long in delivering short-lived traffic.

Since there are multiple routing paths for packets flowing between a given source-destination pair, our algorithm may cause packets to get out of sequence and may require resequencing buffers at the destination. The required buffer size in our algorithm and the length of the time-out interval can be computed using statistics of

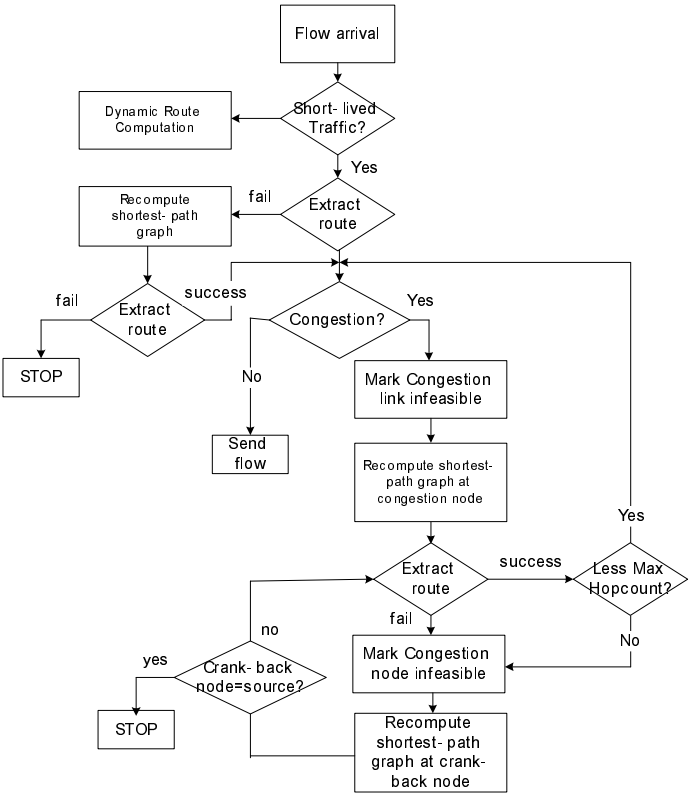


Fig. 1. Flow handling procedure of detour path selection algorithm

the source-to-destination delay distribution [10]. The required resequencing buffer size can be small because the proposed algorithm yields a small variance in the source-to-destination delay.

In the proposed algorithm, we restrict the number of additional hops in detour routing because the long routes make it difficult to select feasible paths for subsequent flows [9]. Detour routing usually consumes excess bandwidth that would otherwise be available to other traffic. This is generally called the "domino" effect. Restricting additional hops minimizes resource wastes and usage time of the resources consumed by alternate routing.

4 Simulation Result

This section evaluates our algorithm via simulation experiments. After a brief description of the simulation environment, we compare the proposed scheme to the traditional static routing algorithm at a hybrid scheme under various traffic loads. We show that, in contrast to shortest-path routing, our algorithm has lower packet loss probability and consumes minimum network resources.

4.1 Simulation Model

The simulation tool, Routesim [1] allows us to simulate the dynamics of load-sensitive routing, static shortest-path routing, and the effects of out-of-date link-state information. We adapt the detour path routing algorithm in packet-switching level not in call-setup level. Previous studies shows that choosing a detour path at the blocking node, that is called as Progressive Control (PC), tends to forward the traffic faster than Source Control(SC) [4]. Thus, supporting PC with or without crank-back can reduce more packet loss probability.

Implementing the algorithm has two major parts. First part is to implement a function which returns multiple detour paths from a blocking node to the destination. Second is to implement route finding trials at crank-bank node. When a link is over-utilized (for example 60% of total capacity), the link state is marked as Congestion.

4.2 Simulation Assumptions and Performance Measure Parameters

In simulations, we denote the short-lived and long-lived traffic classes as N_{short} and N_{long}, respectively. The link capacity c_s is allocated for N_{short}, and c_l is allocated for N_{long}. For simplicity, flows are modeled as to consume a fixed bandwidth for their duration. In choosing a traffic model, we must balance the need for accuracy in representing Internet traffic flows with practical models that are amenable to simulation of large networks. The assumptions in the simulation are as follows:

Flow Durations. In order to accurately model the heavy-tailed nature of flow durations, the flow duration in Routesim is modeled with an empirical distribution drawn from a packet trace from the AT&T World-Net ISP network [1]. The flow durations were heavy-tailed, which means that there were lots of flows with small durations and a few calls with long durations.

Flow Arrivals. We assumed flow arrivals to be a uniform traffic matrix specification with Poisson flow inter-arrival distribution. The value of λ is chosen to vary the offered network load, ρ (ρ varies from 0.6 to 0.9 in most of our experiments). This assumption slightly overstates the performance of the traditional dynamic routing scheme which would normally have to deal with more bursty arrivals of short-lived flows.

Flow Bandwidth. Flow bandwidth is uniformly distributed with a 200% spread of the mean bandwidth value $\overline{b}$. The value of $\overline{b}$ is chosen to be about 1-5% (mainly 1.5 %) of the average link capacity. Bandwidth for long-lived traffic is assigned to be 85% while 15% for short-lived traffic.

Network Topology. In order to study the effects of different topologies, we used two topologies: the Random and the MCI topology. Their degrees of connectivity are quiet different, such as 7.0 and 3.37. The random graphs were generated using Waxman's model. The two topologies we described in Table 1 were widely used in other routing studies [1], [4], [9]. The 'Avg. path length' in Table 1 represents the mean distance (in the number of hops) between nodes, averaged across all source-

destination pairs. Each node in the topology represents a core switch which handles traffic for one or more sources, and also carries transit traffic to and from other switches or routers.

Congestion Situation. When N_{short} reaches its capacity c_s, we call this as congestion in short-lived traffic. If the buffer thresholds are too small, the algorithm will overreact to normal state and too many packets will be bumped to longer routing paths.

Table 1. Topologies Used for Simulations

Topology	Nodes	Links	Degrees	Subnet	Net
Random	50	350	7.0	4	2.19
MCI	19	64	3.37	4	2.34

We simulate the algorithm under dual mode: applying a widest-shortest algorithm for long-lived traffic and shortest path first algorithm for short-lived traffic. We use the distance cost when selecting the path as "hop". To study the effect of additional hops, we vary additional hops from 1, 2 and infinite when using the algorithm. Paths are setup by on-demand routing policy for long-lived traffic while distributed routing for short-lived traffic. We evaluate the packet loss probability under various packet arrival rates in order to see the effect of additional hops for short-lived traffic. Average path length is checked to evaluate the proposed algorithm under heavy-traffic load. We also compare the alternate routing at the blocking node with the crank-back node. Their average path length and delay time are simulated.

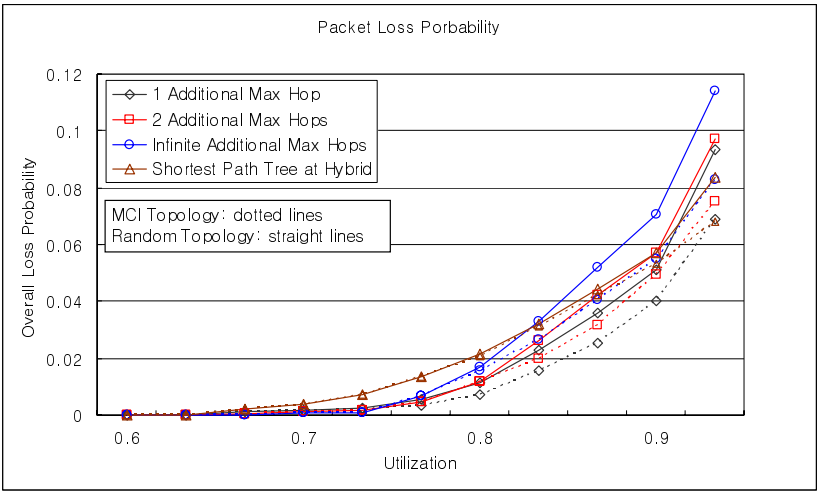
4.3 Simulation Results

First, we study the performance of the proposed algorithm. The two networks with different connectivity were simulated with and without alternate routing. We considered only detour routing at the congested node for this experiment. The simulation results are shown in Fig 2. We found from Fig 2(a) that a single additional hop in the algorithm leads to a significant improvement in the loss probability. Adding one more hop in our algorithm slightly reduces the loss probability when it is working in the range of 0.7 and 0.8 utilization. But the situation changes as the arrival rate exceeds 0.8 utilization. The packet loss probability becomes higher with two additional hops compared to that with a single alternate hop. Adding more alternate hops further degrades the packet loss probability without performance gain under any loading region.

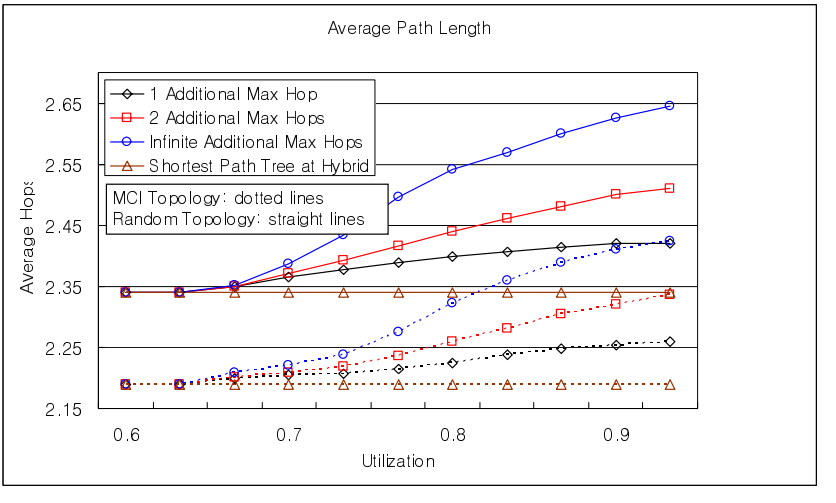
We used the average path length (in hops) of the chosen paths as the measure of the resource usage of the paths. From Fig 2(b), as the network load increases, the number of average hops increases. The simulation results shows that the alternate path tends to be longer than the primary path for a heavy packet arrival rate, which means the detour routing requires more network resources than the primary routing. The limit of additional hops must be carefully chosen to achieve satisfactory network performance. From the experiment, we find that less than two additional hops are sufficient to achieve the benefit of detour routing without significantly increasing

packet transfer delay in the range of 0.6 to 0.85 utilization. This result presents Braess' paradox that is an important consideration for the analysis of any network system that has detour routes [11]. Braess' paradox says that detour routing, if not properly controlled, can cause a reduction in overall performance.

In order to study the effect of different degrees of connectivity, we evaluate random graph with 7 degrees under the same traffic load. Dotted lines in Fig 2 show that



(a) Overall Packet Loss Probability



(b) Average Path Length

Fig. 2. Impact of detour routing under MCI and Random

richly connected topology significantly has low packet loss probability. Additional resources which detour routing consumes, increases with the number of alternate hops allowed. But the bad effect of detour routing under various packet arrival rates is less than the one at MCI. From this experiment we find that, in Random topology, less than two additional hops are sufficient to achieve the benefit of alternate routing without significantly increasing packet transfer delay. The network with rich connectivity makes it possible to quickly disperse packets away from congested parts of the network with the proposed algorithm for short-lived traffic in distributed routing. So, we note the connectivity of network when we apply the detour routing algorithm as well as note link utilization.

We also compared the algorithm with the crank-back scheme. We performed these experiments under the MCI topology and allowed only one additional hop. The packet loss probability of alternate routing at crank-back node has almost the same result with the alternate routing in at the congestion-node. However, detour routing in our algorithm with the crank-back node yields longer paths than the proposed algorithm without crank-back.

5 Conclusions

In this paper, we introduced an efficient routing scheme that handle shortest-path first algorithm combined with detour routing in hybrid routing algorithm. Our study shows that additional hops in detour paths should be carefully restricted to avoid network resource waste under heavy load. The proposed algorithm has less packet loss probability and less resource waste because we restricted the resource by one or two. The network with rich connectivity has significantly less packet loss probability even though resource waste is the same as the network with poor connectivity. Finally, detour routing at the congestion node and the crank-back node shows almost same packet loss probability while detour routing at the crank-back node consume more additional hops.

Acknowledgments

This research was supported by the MIC(Ministry of Information and Communication), Korea, under the BcN ITRC(Broadband Convergence Network Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment).

References

[1] A, Shaikh, Jennifer Rexford, Kang G. Shin,: 'Evaluating the Impact of Stale Link State on Quality-of-Service Routing' Transactions of IEEE/ACM, Volume 9, Apr 2001, pp 162-176

[2] Keping Long, Zhongshan Zhang, Shiduan Cheng,: 'Load Balancing Algorithms in MPLS Engineering'. High Performance Switching and Routing, 2001 IEEE Workshop 2001, pp 175-179

[3] KRUPP, S 'Stabilization of alternate routing networks' Proceedings of IEEE International conference on Communications, ICC'82, Philadelphia, PA, June 1982, 3I.2.1-3I.2.5
[4] M. Sivabalan and H. T. Mouftah, 'Design of link-state alternative path routing protocols for connection-oriented networks' Proceedings of IEE, Vol. 146, No. 3, June 1999, pp 163-170
[5] V. Srinivasan, G. Varghese, S. Suri, and M. Waldvogel, 'Fast scalable algorithms for level four switching' in Proceedings of ACM SIGCOMM, September 1998, pp 191-202
[6] S. Chen and K. Nahrstedt, 'An overview of quality of service routing for next-generation high-speed networks: Problems and solutions' IEEE Network Magazine, November/December 1998, pp 64-79
[7] David Lloyd, Donal O'Mahony 'Smart IP Switching: A Hybrid System for Fast IP-based Network Backbones' IEEE, Jun 1998, pp 500-506
[8] A Feldmann, J. Rexford, and R. Caceres, 'Efficient policies for carrying Web traffic over flow-switched networks' IEEE/ACM Transactions on Networking, December 1998, pp 673-685
[9] Q. Ma and P. Steenkiste, 'On path selection for traffic with bandwidth guarantees' in Proceedings of IEEE International Conference on Network Protocols, Atlanta, GA, October 1997, pp 191-202
[10] Mark J. Karol and Salman Shaikh "A Simple Adaptive Routing Scheme for ShuffleNet Multihop Lightwave Networks" IEEE GLOBECOM 88, December 1988, pp 1640-1647
[11] N. G. Bean, F. P. Kelly, P.G. Taylor "Braess's Paradox in a Loss Network" Journal of Applied Probability, 1997, pp 155-159
[12] Steve Uhlig, Olivier Bonaventure "On the cost of using MPLS for interdomain traffic" Proceedings of the First COST 263 International Workshop on Quality of Furture Internet Services, 2000, pp 141-152
[13] I.F. Akyildiz, T. Anjali, L. Chen, J.C. de Oliveira, C. Scoglio, A. Sciuto, J.A. Smith, G. Uhl "A new traffic engineering manager for DiffServ/MPLS networks: design and implementation on an IP QoS Testbed" Computer Communications, 26(4), Mar 2003, pp 388-403

HAWK: Halting Anomalies with Weighted Choking to Rescue Well-Behaved TCP Sessions from Shrew DDoS Attacks*

Yu-Kwong Kwok, Rohit Tripathi, Yu Chen, and Kai Hwang

University of Southern California, Los Angeles, CA 90089, USA

Abstract. High availability in network services is crucial for effective large-scale distributed computing. While distributed denial-of-service (DDoS) attacks through massive packet flooding have baffled researchers for years, a new type of even more detrimental attack—shrew attacks (periodic intensive packet bursts with low average rate)—has recently been identified. Shrew attacks can significantly degrade well-behaved TCP sessions, repel potential new connections, and are very difficult to detect, not to mention defend against, due to its low average rate.

We propose a new stateful adaptive queue management technique called HAWK (Halting Anomaly with Weighted choKing) which works by judiciously identifying malicious shrew packet flows using a small flow table and dropping such packets decisively to halt the attack such that well-behaved TCP sessions can re-gain their bandwidth shares. Our NS-2 based extensive performance results indicate that HAWK is highly agile.

1 Introduction

Various kinds of malicious attacks have hindered the development of effective wide-area distributed computing. The most notable type of attack is the so-called Distributed Denial-of-Service (DDoS) attack [7], which works by overwhelming the systems with bogus or defective traffic that undermines the systems' ability to function normally. DDoS attacks aims at consuming resources (CPU cycles, system memory or network bandwidth) by flooding bogus traffic at sites so as to deny services to the actual user and prevent legitimate transactions from completing [1]. The TCP, UDP, and ICMP flooding attacks fall in this category.

Unfortunately, while finding effective solutions to combat DDoS attacks has baffled researchers for years, an even more detrimental type of network-based attack has recently been identified [2]. This special class of attack is referred to as low-rate TCP-targeted DDoS attack or shrew attack [2] that denies bandwidth resources to

* Manuscript accepted for presentation at ICCNMC 2005 in August 2005. This research was supported by an NSF ITR Research Grant under contract number ACI-0325409. Corresponding Author: Kai Hwang, Email: kaihwang@usc.edu, Tel: 213-740-4470, Fax: 213-740-4418.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 423 – 432, 2005.

legitimate TCP flows in a *stealthy* manner. Indeed, unlike traditional DDoS attacks, which are easy to detect by observing that the victim site is totally unable to respond, a shrew attack is very difficult to detect [2] because the adverse effects on well-behaved network connections are not easily observable. Commercial Web sites would then suffer from stealthy losses of potential new connections (hence, new transactions).

The key idea behind a shrew attack is to exploit TCP's *Retransmission Time-Out* (RTO) mechanism to synchronize *intelligent* (i.e., carefully articulated) low average rate bursts. Thus, a shrew attack can also be referred to as *degradation-of-service* or *pulsing* attack, as opposed to the well-known denial-of-service attack. Unlike a regular zombie that paralyzes a system by flooding it with a steady stream of attack traffic, the pulsing zombie attacks its target with irregular small bursts of attack traffic from multiple sources over an extended period of time (see Figure 1). As such, pulsing zombie attacks are more difficult for routers or counter-DDoS mechanisms to detect and trace. The reason is that unlike flooding DDoS attacks, they are slow and gradual, and thus they do not immediately appear as malicious.

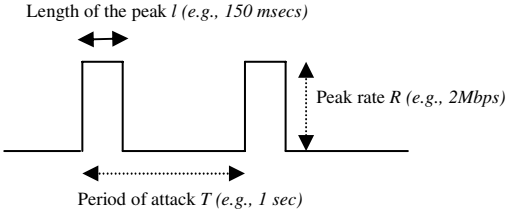


Fig. 1. An illustration of the shrew attack stream with a square waveform

As indicated by Kuzmanovic and Knightly [2], it is very difficult to detect such a shrew attack. The main challenge lies in separating a bogus traffic flow from a "flash crowd" [5] without maintaining complicated and expensive per flow state information at the routers. In this paper, we meet this challenge by proposing a novel effective detection technique, called HAWK [1] (*Halting Anomaly with Weighted choKing*), which is an active queue management method based on only *partial state*. Specifically, HAWK works by judiciously monitoring the packet flows with the help of a small flow table. Traffic statistics are accumulated in the flow table using a technique similar to the CHOKe algorithm [3].

A packet flow will be marked as malicious if its traffic statistics in the flow table indicate that the flow is far too bursty over an extended period of time (e.g., 5 secs) with very high rate bursts appearing in short time-spans (e.g., 100 msecs). Once a flow is identified to be malicious, HAWK will drop its packets decisively in order to help the well-behaved sessions to re-gain their entitled bandwidth shares. Furthermore, the HAWK algorithm is able to defend the targeted victim against both

[1] Hawks are natural enemies of shrews.

single source and distributed shrew attacks while maintaining low overhead in terms of processing and memory resources. Our NS-2 based simulation results indicate that the HAWK algorithm is highly effective.

The rest of the paper is organized as follows. In the next section, we first describe the key characteristics of service-degrading network attacks, and then we introduce our HAWK algorithm for traffic burst characterization and corresponding flow classification. Simulation setting and experimental environment details are given in Section 3. In the same section, we present the NS-2 experimental results and provide our interpretations. Finally, we give some concluding remarks in Section 4.

2 The Proposed HAWK Algorithm

We propose to use our HAWK algorithm at the bottleneck link router for characterizing bursts of the attack stream and classifying them into legitimate or illegitimate sources. HAWK maintains a very small state information data structure—the flow table—in order to track down the shrew flows. The flow table only keeps the traffic statistics of potentially malicious flows and confirmed malicious flows, and thus, normally, occupies very little storage space. The maintenance of the flow table is further elaborated below and the storage space requirements are discussed in detail in Section 3.

The router maintains a single queue for all the incoming flows and the average queue size computation is done using exponential moving average as in RED [6] and CHOKe [3]. But unlike from these previous approaches, our decision-making algorithm involves flow table comparisons and statistical computations that are used to characterize and classify flows into different threat level categories. Whenever a new packet arrives at the queue, if the average queue size is less than the minimum threshold of the buffer (Min_{Th}) the packet is admitted into the queue.

Furthermore, HAWK checks each incoming packet against the existing entries in the flow table, and if there is a match the corresponding flow table statistics are updated. In the "random matching" process, the following checking actions are carried out. If the average queue size is greater than the maximum threshold (Max_{Th}) the incoming packet is dropped after checking and updating the flow table statistics. For the intermediate case when the average queue size value is between the minimum (Min_{Th}) and maximum (Max_{Th}) thresholds, we use a mechanism similar to CHOKe [3] by admitting the packet with a probability p which depends on the average queue size. For instance, if the queue size is over the maximum threshold (Max_{Th}), the packet is dropped with a probability 1. Similarly, if the queue size is below the minimum threshold (Min_{Th}), the packet is dropped with a probability 0. In the intermediate case, additional checking of the flow table and statistics computations are performed for flow classifications.

In their modeling and simulations, Kuzmanovic *et al.* show the relationship between the throughput of a TCP flow and the denial-of-service inter-burst period. Our NS-2 simulations modeled single flow TCP and single flow DDoS stream interaction and the modeled flow [2]. The inter-burst periods of one second and lower

are most fatal to the TCP throughput. The destructive impact reduces as the inter-burst period is increased.

Furthermore, it is found that for the most severe impact without being identified by existing routing architectures, these shrew bursts should occur in a small time window of 100-250 milliseconds. As such, if we take into account the periodicity of bursts with respect to two separate time windows, one having a smaller time scale of 100-250 milliseconds and the other having a larger time scale of 1-5 seconds, we can classify the attacks into different threat levels.

On initially identifying a high burst rate flow over a short time scale, if it is found that the average queue size is larger than the Min_{Th}, we perform the following checking. For each new incoming packet, we randomly pick a packet currently in the queue. If the two packets are found to be from the same flow, then we proceed to update the flow table statistics in order to see if the flow is to be considered as a malicious shrew flow. Otherwise, the packet is admitted without further action.

Once the flow is identified as a high rate stream on a short time scale, we correlate these identified bursty flows over a longer time scale using our HAWK algorithm with the help of the small flow table. Thus, at most of the time, the resource requirement of the flow table is of the order of the number of potential attack sources. A cumulative burst value is maintained along with the packet entry times for each of the identified flows. The cumulative burst value and the associated time provide an insight into the burstiness of the flows.

A large cumulative burst for an extended period of time indicates a potential malicious source. For shorter time scales we use a window size as 200 milliseconds. The rationale behind using this value is that for burst lengths longer than this, in order to maintain the same low average rate the DDoS stream would have to keep its peak rate low, thus decreasing the severity of the attack.

Cumulative Burst gives an insight into the average bursts from a given flow over a series of larger time frames. ***Traffic Burst Rate*** above the threshold values over consecutive one second window is logged. If this trend is found to follow in more than or equal to three blocks within the last five seconds ($\mathbf{C_{thresh}}$), the flow is confirmed as a malicious shrew flow and is blocked. We choose the value of three blocks in five seconds time scale to target the most severe effects of the DDoS streams. Also this provides some leniency to normally bursty flow which may send large but intermittent bursts. But since these natural bursts normally cannot extend symmetrically on a larger time scale of five seconds, we can be sure that our chosen time scale would be unbiased towards these naturally bursty flows. Finally, it should be noted that a time period of five seconds is the *shortest time* to confirm a successful detection. We call this five-second time window as *HAWK Window Size*.

Furthermore, if some legitimate flow shows this behavior, it is good to block such a flow so as to provide fairness to other legitimate flows. Since the pre-filtering at the router input still maintains statistics for these flows, they can be removed from the flow table if they react to routers' congestion indication and do not send large bursts for the next two time windows of one second each. This is again a configurable system parameter.

Periods of more than two seconds are not very severe. Thus, we choose the value of two seconds to balance the tradeoff between an optimal flow table size in presence of normally bursty flows and detecting malicious bursts having higher periods. As such, our algorithm sets a flow as malicious if it detects three or more than three bursts over the threshold within a longer spanning window of five seconds.

Traffic Burst Threshold value is chosen based on the link capacity of the routers' output link. It was identified that any burst lower than one third of the link capacity is not severe enough to produce desired DDoS effect on the legitimate TCP flows. So, in performance study, we set the value of $\boldsymbol{BF_{TH}}$ as one third of the bottleneck link capacity. The attacker can gather information about the bottleneck link capacity using some of the probing schemes in existence [4].

For distributed shrew attacks, instead of Source Address, we maintain Source Subnet that provides the cumulative traffic coming from the infected subnet to the destination victim. The calculation of packet dropping probability $\boldsymbol{p}$ when the average queue size exceeds the minimum threshold is done as in RED [6] and CHOKe [3], i.e., based on the exponential weighted moving averages of the average queue size. Typical values of suitable RED parameters for a gateway router with limited load would be: $\boldsymbol{Min_{Th}}$ ranging between 30 to 100 packets, $\boldsymbol{Max_{Th}}$ set to $3\boldsymbol{Min_{Th}}$ and $\boldsymbol{w_q} = 0.002$ [6].

The proposed HAWK algorithm can characterize the low-rate TCP-targeted attack using the flow table driven packet dropping technique, as formalized in the flowchart shown in Figure 2. Upon any packet dropped, the legitimate TCP flows will follow the standard protocol semantics and will cut down their rates in accordance with the end-system congestion avoidance mechanisms of TCP. Thus, the values of $\boldsymbol{C_{burst}}$ and $\boldsymbol{BF_{rate}}$ will always remain much lower than the threshold values for these parameters.

Whenever the average queue size is higher than the minimum threshold, a comparison of incoming packet with the already queued packets will result in a success with a high probability if the attack burst was sent during that time frame. The flow table statistics are incremented and the corresponding counter value and the burst parameters for that time frame would progress towards the threshold ultimately, resulting in confirmation of the flow as malicious.

3 NS-2 Simulation Results

In this section, we first describe the simulation set up for evaluating our algorithm in detecting and penalizing attacking hosts and providing fairness to the legitimate users. We use NS-2 to carry out our simulations and we compare the results of our proposed algorithm with those of two well-known active queue management (AQM) algorithms: Drop Tail and CHOKe [3]. As mentioned earlier, response time (i.e., the time duration from the attack launching instant to the attack detected instant) is a very important measure as it determines the duration of damage to a victim site. Thus, we would also examine the response time performance of our algorithm in identifying and blocking the malicious shrew hosts along with the false positives generated using our scheme.

Our simulations consist of a variety of network configurations and traffic patterns simulating both single source as well as multi-source attacks coming from a single LAN and/or distributed LANs. For simulating attacks from different LANs, we use different delay values on the links. The collected statistics are used to plot normalized throughput against attack inter-burst period. The normalized throughput value provides the metric for evaluating the efficiency of our algorithm.

The malicious hosts are modeled as UDP sources sending traffic flows in periodic intervals. The machine GR is the last hop gateway router interfacing to the victim machine connected to the outside network through an AS cloud. We perform statistics

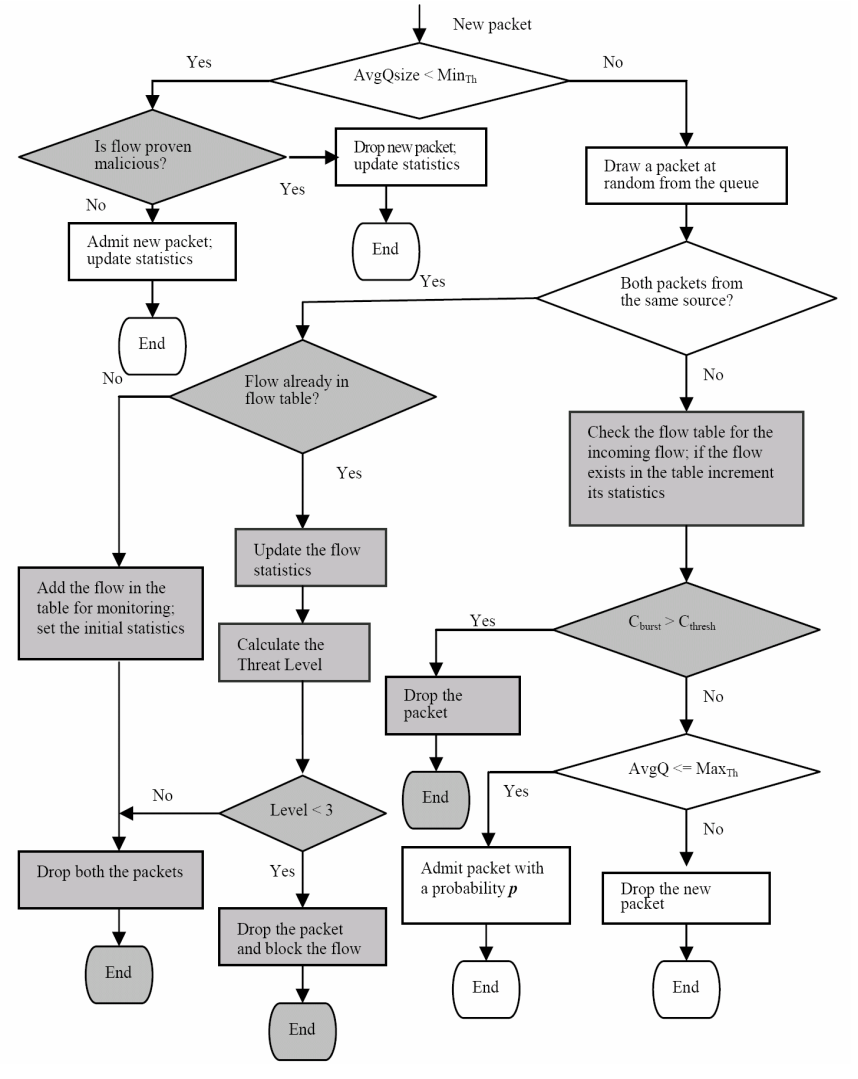


Fig. 2. The proposed HAWK algorithm

collection and computations on the last hop router GR. For a distributed attack environment the only key parameter that we would like to focus on is the different delays that a flow gets in reaching the victim's end. We achieve this by providing different link delays to each of the malicious hosts.

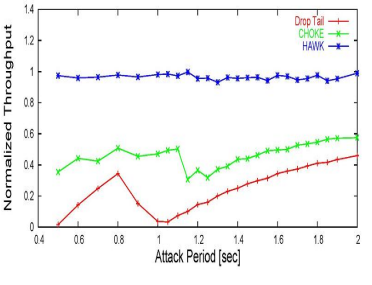
We first use *Normalized Throughput* as the comparison metric, and it is defined as follows:

***Normalized Throughput* =**
Average throughput achieved by the TCP flow (or aggregate) with DDoS stream / Throughput achieved without DDoS stream

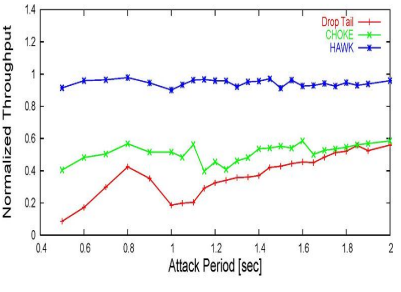
The value of the normalized throughput gives us an indication of the severity of the damage done by the attack stream. The lower the normalized throughput is, the greater the damage. Unless otherwise specified we use the output link capacity of the last hop router as 2 Mbps, link delay as 120 milliseconds. The shrew attack stream is simulated to generate a square wave having a peak rate of 2 Mbps and a burst length of 150 milliseconds to target TCP flows with average RTT of 150 milliseconds and lower.

Since all the TCP variants are equally vulnerable to the shrew DDoS stream of 50 milliseconds or higher [2], for experimental purpose we use TCP-SACK. Our simulation uses a shorter time scale window of 200 milliseconds and a larger window of five seconds with internal one second blocks. Traffic Burst Threshold value is taken as one third of the bottleneck link capacity.

We first consider the single source scenario. The simulation results of the throughput achieved, under different queuing schemes, by the legitimate TCP flows with different number of shrew DDoS streams are shown in Figure 3. The *x*-axis indicates the period of the burst and the *y*-axis indicates the normalized throughput where value of one indicates the theoretical throughput without the attack stream. It can be clearly seen that under Drop Tail the throughput of the legitimate flow almost reaches zero for period values of one second and lower.



(a) one shrew flow and one well-behaved TCP flow



(b) one shrew flow and five well-behaved TCP flows

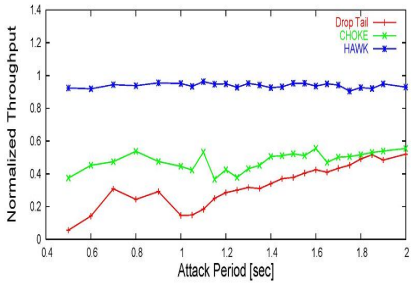
Fig. 3. Performance comparison among Drop Tail, CHOKe, and HAWK in terms of normalized throughput

Further increase in the time period of the attack stream increases the throughput of the legitimate flow but it is still far below the actual attainable throughput of one. The results for the CHOKe queuing as shown in Figure 3 indicate a slight improvement in TCP performance but it is clear that CHOKe algorithm cannot achieve the desired goal of providing fair share of the link capacity to the legitimate TCP flows.

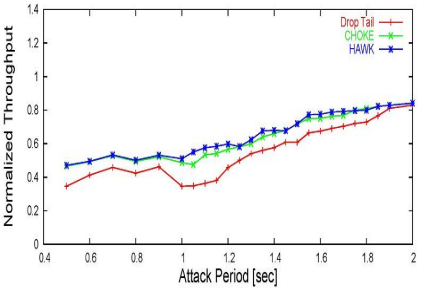
With our HAWK algorithm, we can see that the gain in the TCP throughput is significant throughout the two seconds attack period that we consider in this study. This is due to the fact that for identifying and correlating burst streams we have used three or more blocks of captured bursts within our larger time scale.

Next we consider the multiple-source scenario. The experiment is repeated with five legitimate flows and two DDoS streams so as to find out the impact of attack streams if the attacks are launched from multiple collaborating sites. This kind of scenario is one of the most common cases of distributed denial-of-service attack, where a malicious user compromises a large number of hosts called zombies to launch a coordinated attack with lower peak rate which means that for two DDoS shrews each source sends traffic at half the rate determined in the previous experiment. We would consider two different scenarios here. The link capacity and burst period are kept the same as above in both cases and the effect is seen on five TCP flows.

Firstly, let us consider the case where these zombies are on the same subnet so that all have the same packet delay towards the victim. As shown in the Figure 4(a), the average throughput is almost similar to the previous experiment. Similar to the case of one legitimate flow, the trend shows that the DDoS attack stream has much worse impact for attack periods of one second and lower because of the minimum RTO value of one second for TCP and the best throughput is again given by HAWK. Here a modified adaptive filtering scheme is used where traffic coming from the same subnet is considered together to generate statistics.



(a) attacks on the same subnet



(b) attacks on multiple subnets

Fig. 4. Effect of distributed shrew attacks from the same or different subnets (five well-behaved TCP flows)

Secondly, let us consider the case when these zombies are on different subnets and trying to collaborate for launching a shrew attack on the victim. Being on different subnets, these zombies would have different packet delays towards the victim. This signifies a more realistic scenario if this kind of shrew attack is to be launched from distributed zombies across the globe.

Four zombies are used, each sending at one fourth the peak rate. The link delays from the four zombies till the GR are chosen as 100, 120, 140, 160 milliseconds. As shown in Figure 4(b), the impact of the attack is reduced in this case. This is due to the fact that now the short attack stream from each malicious source reaches at the bottleneck router RV at different times and the router serves legitimate TCP flows more frequently. But the normalized throughput is still less than the ideal value of one.

The result suggests that the different queuing mechanisms CHOKe and HAWK are unable to produce any significant improvement over Drop Tail scheme. This indicates that for lower attack periods, the effect of shrew attack is more prominent. Though it can logically be assumed that with more number of zombies spread out and each sending at a very small fraction of the bottleneck bandwidth, the legitimate TCP flow aggregate would get fair share of the bandwidth.

4 Conclusions and Future Work

In this paper, we have proposed an adaptive packet-filtering scheme to address the open problem of TCP targeted shrew degradation-of-service attacks. Simulation results demonstrate that our algorithm, called HAWK, outperforms other router assisted queuing mechanisms for combating this special class of network based attacks. Our algorithm is easy to implement and requires a very small storage space that is most of the time only of the order of the number of potential malicious hosts.

Our major on-going work is the implementation of our scheme on the DETER [8] testbed so that we can test the efficacy of the HAWK algorithm in a real environment. Another important research avenue is to extend our scheme to a distributed environment, where multiple routers can interact to identify these attacks even earlier and under wider range of traffic patterns and topologies.

References

1. CERT/CC and FedCIRC, “Advisory CA-2000-01 Denial-of-Service Developments,” *Carnegie Mellon Software Eng. Institute*, Jan. 2000.
2. A. Kuzmanovic and E. W. Knightly, “Low-Rate TCP-Targeted Denial of Service Attacks—The Shrew vs. the Mice and Elephants,” Proceedings of ACM SIGCOMM 2003, Aug. 2003.
3. R. Pan, B. Prabhakar, and K. Psounis, “CHOKe: A Stateless Active Queue Management Scheme for Approximating Fair Bandwidth Allocation,” INFOCOM 2000, vol. 2, pp. 942–951, Apr. 2000.
4. M. Jain and C. Dovrolis, “End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput,” Proceedings of ACM SIGCOMM ’02, Aug. 2002.
5. J. Jung, B. Krishnamurthy, and M. Rabinovich, “Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites,” Proceedings of 11th Word Wide Web Conference.
6. S. Floyd and V. Jacobson, “Random Early Detection Gateways for Congestion Avoidance,” IEEE/ACM Transactions on Networking, vol. 1, no. 4, pp. 397–413, Aug. 1993.

7. S. M. Specht and R. B. Lee, "Distributed Denial of Service: Taxonomies of Attacks, Tools, and Countermeasures," Proceedings of the 17th Int'l Conf. Parallel and Distributed Comp. Systems, pp. 536–543, Sept. 2004.
8. DETER and EMIST Projects, "Cyber Defense Technology: Networking and Evaluation," Comm. ACM, pp. 58–61, Mar. 2004. Also from DETER Website: http://www.isi.edu/deter/docs/acmpaper.pdf

Biographical Sketches of Authors

Yu-Kwong Kwok is an Associate Professor in the Department of Electrical and Electronic Engineering at HKU. Dr. Kwok is currently on leave from HKU and is a Visiting Associate Professor at the University of Southern California. His research interests include Grid computing, mobile computing, wireless communications, network protocols, and distributed computing algorithms. He is a Senior Member of the IEEE. Dr. Kwok is a recipient of the 2003-2004 Outstanding Young Researcher Award given by HKU. He can be reached at ykwok@hku.hk.

Rohit Tripathi received his B.S. degree in Electronics Engineering from Institute of Technology—B.H.U., India in 2000. He received the M.S. degree in Computer Engineering at USC in 2005. He has worked as a Software Engineer at Hughes Software Systems, India, for three years. At Hughes he focused on developing software for IP routing and network based services including Multicasting, VoIP and Network Management. His present research interests are in the area of network security. He can be reached at rohittri@usc.edu.

Yu Chen received his B.S and M.S in Electrical Engineering from Chongqing University, China in 1994 and 1997, respectively. He is presently pursuing the Ph.D. degree in Electrical Engineering at the University of Southern California. His research interest includes Internet security, automated intrusion detection and response systems, and distributed security infrastructure for Grid Computing environment. He can be reached at cheny@usc.edu.

Kai Hwang is a Professor and Director of Internet and Grid Computing Laboratory at the University of Southern California. He received the Ph.D. from the University of California, Berkeley. An IEEE Fellow, he specializes in computer architecture, parallel processing, Internet and wireless security, and distributed computing systems. He has authored or coauthored 7 scientific books and 180 journal/conference papers in these areas. Hwang is the founding Editor-in-Chief of the Journal of Parallel and Distributed Computing. Currently, he is also an Associate Editor of the IEEE Transactions on Parallel and Distributed Systems. He has performed advisory and consulting work for IBM Fishkill, Intel SSD, MIT Lincoln Lab., ETL in Japan, and GMD in Germany. Presently, he leads the NSF-supported ITR GridSec project at USC. The GridSec group develops security-binding techniques for trusted job scheduling in Grid computing, distributed IDS and pushback of DDoS attacks, fuzzy-logic trust models and selfish Grid Computing models, and self-defense software systems for protecting network-centric computing resources. Professor Hwang can be reached at kaihwang@usc.edu or through the URL: http://GridSec.usc.edu/Hwang.html.

Improved Thumbprint and Its Application for Intrusion Detection

Jianhua Yang and Shou-Hsuan Stephen Huang

Department of Computer Science, University of Houston,
4800 Calhoun Rd. Houston, TX, 77204, USA
{jhyang, shuang}@cs.uh.edu

Abstract. This paper proposes RTT-thumbprint to traceback intruders, and to detect stepping-stone intrusion; RTT-thumbprint is a sequence of timestamp pairs of a send packet and its corresponding echoed packets. Each pair of timestamps represents a round trip time (RTT) of a packet. Besides the advantages of efficiency, secrecy, and robustness, RTT-thumbprint has the ability to defeat intruder's random delay and chaff manipulation. An exhaustive and a heuristic algorithm are proposed to correlate RTT-thumbprints. The results showed that the heuristic algorithm performs as good as the exhaustive one but is more efficient

1 Introduction

Most intruders usually chain many computers to hide their identities before launching their attacks. One way to catch such intruders is to trace them back along the connection chains; this technology is called connection traceback, one of the technologies for intrusion detection. Many practical methods were proposed to traceback intruders after 1990. Some of the representatives are Distributed Intrusion Detection System (DIDS) [1], Caller Identification System (CIS) [2], Caller ID [3], Thumbprint [3], time-based [4] approach, deviation-based [5] approach, and Temporal Thumbprint [6] (T-thumbprint).

DIDS is a method to trace all the users in the network and collect audit trails for each user; the audit information collected are sent to a DIDS server to be analyzed. The important problem with this approach is it can not be applied to a large network because its centralized DIDS server is a bottleneck of the whole system. CIS works by asking each host along a connection chain to provide the information that recorded the previous login situation with the goal of being applied to a large network. However, it suffers from incurring additional network load, and leaking private information of each host involved. Caller ID is a method to largely reduce network load, but it suffers from failing to break back, performing the tracing only while an intruder is still active, and running the risk of accidentally damaging the intermediate hosts. Thumbprint is a method to traceback intruders by comparing the thumbprints of different connections, where thumbprint is the summary of a certain section of a connection's content. Content-based thumbprint is very useful in tracing intruders, but it cannot be applied to

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 433 – 442, 2005.

encrypted sessions. Time-based approach can be applied to encrypted sessions and can be used to traceback intruders by comparing distinctive timing characteristics of interactive connections. Deviation-based method is to traceback intruders by comparing the deviations between two connections; similar to time-based method, this method is available on detecting encrypted sessions. But these two methods suffer from being vulnerable to intruder's manipulation, and high false positive rate. T-thumbprint, which is defined as a sequence of interval between timestamps of two continuous send packets in a connection, can be used to traceback intruders with the advantages of efficiency, secrecy, and robustness. But it does not provide a full solution in defeating intruders' random delay and chaff manipulation [6].

This paper proposes a new time-based thumbprint, Round Trip Time- (RTT-) thumbprint to characterize packets in a connection session, as well as two algorithms (exhaustive and heuristic) to correlate RTT-thumbprints. Instead of using timestamps of send packets or contents in a connection, RTT-thumbprint uses a sequence of timestamp pairs between each send packet and its corresponding echoed packets to characterize a roundtrip packet pair. The experiment results and analysis showed that RTT-thumbprint can handle intruders' random delay and chaff manipulation better than other methods.

The rest of this paper is arranged as follows. In Section 2, we define some preliminaries used in this paper. Section 3 discusses the definition of RTT-thumbprint, its correlating algorithm, and its ability to defeat intruders' manipulation. Finally, in Section 4, conclusions and future work are presented.

2 Preliminaries

We began stating our assumptions: (1) the research object is limited to an interactive connection session made by telnet, rlogin, rsh, ssh or similar tools; (2) we assume that thumbprints are collected at approximately the same time interval; (3) we assume that any users, when connect to a host, may need to pause to read, think, or respond to the previous operations, and the gaps between two continuous operations caused by human interaction are measured in seconds. These gaps are considerably larger than a typical round trip time of a network; (4) a user can only delay each packet sent or received; any superfluous packets inserted into a connection will be removed shortly without reaching the destination.

Suppose a user logs in from Host *1*, and eventually connects to Host *n,* which is the destination, through Host *2*, … , and Host *n-1,* as shown in Figure 1. We formally give the definitions of the following terms and notations.

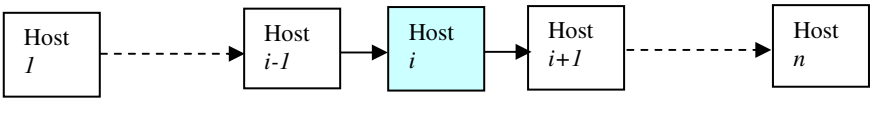


Fig. 1. A connection chain

Connection: When a user from a host logs into another host, we call this a connection session between the two hosts.

Outgoing and incoming connection: For any host, if a connection directly comes from another host, then this connection is defined as an incoming connection. If a connection comes from this host and directly connects to another host, then it is defined as an outgoing connection.

We need to define some terminologies for timestamps. Given two sequences T: $\{(t_{11}, t_{12}), (t_{21}, t_{22}), \ldots, (t_{n1}, t_{n2})\}$, and U: $\{(u_{11}, u_{12}), (u_{21}, u_{22}), \ldots, (u_{m1}, u_{m2})\}$ with length n, and m respectively, we assume that the conditions $0<t_{i1}<t_{i2}$, $0<u_{j1}<u_{j2}$, where $i=1,\ldots, n$, and $j=1,\ldots, m$, are satisfied for sequences T and U. We define element-inclusion, and sequence-inclusion in the following.

Element-inclusion: For any timestamp pairs $t = (t_{i1}, t_{i2})$, and $u = (u_{j1}, u_{j2})$ if the condition $0< t_{i1}< u_{j1} < u_{j2}< t_{i2}$ is satisfied, then we say u is included in t, we denote it as $u \subset t$.

Sequence-inclusion: For two sequences U and T of length n, we say that sequence U is included in sequence T, denoted $U \subset T$, if $U[i] \subset T[i]$, where i=1,…, n.

Sequence-inclusion definition only gives us the result that one sequence is completely included in another sequence. However, most times, when we correlate sequences, which come from thumbprint, what we need to handle is part of a sequence is included in another sequence. Under this situation, the sequence-inclusion problem becomes to compute a longest inclusion subsequence.

Longest Inclusion Subsequence (LIS): Given the above two sequences T and U(of length m and n), if (1) there is a subsequence U′ of U and a subsequence T′ of T, such that U′ is included in T′ and (2) there are no other U′ and T′ with a longer length that is included in T, we define U′ as the longest (common) inclusion subsequence of U and T. The problem of computing a longest subsequence from one sequence that is included in another sequence is the longest inclusion subsequence problem.

The length of the longest inclusion subsequence can then be used to measure the similarity of two sequences of timestamp pairs. The similarity ratio (SR) of two sequences is defined as $p/\min(m, n)$, where p is the length of a longest inclusion subsequence.

3 RTT-Thumbprint and Its Correlating Algorithm

3.1 Motivation

The design of TCP/IP protocol makes it difficult to reliably traceback to original intruders if they obscure their identities by logging through a chain of multiple hosts. However, once a chain is established between an intruder and a victim, it was shown [7] that every packet sent from a intruder is going to be decrypted and then encrypted in each host in between and forwarded to the victim at the end of a chain; a corresponded packet is going to be echoed from the victim and propagate to the

intruder side in a similar way. If we monitor the outgoing connection of each host in between, we can observe the send packet from this host and one echo packet from the adjacent host downstream. This fact prompted us to consider that these send and echo packet pairs as a unique characteristic to identify a connection. Thus we can use each sequence of packet RTT timestamps to characterize an encrypted connection.

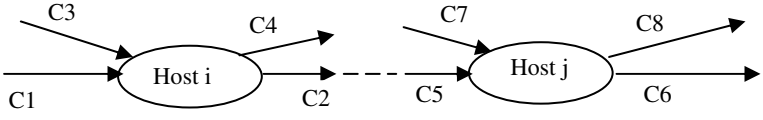


Fig. 2. Illustrate the basic idea of RTT-thumbprint

We assume there are two compromised hosts: Host *i,* and Host *j* that belong to one chain <C1, C2, …, C5, C6>, as shown in Figure 2. Host *i* has two incoming connections: C1, C3, and two outgoing connections: C2, C4; Host *j* has two incoming connections: C5, C7, and two outgoing connections: C6, C8. If we monitor all the send and echo packets of each outgoing connection of Hosts *i* and *j* continuously, and match them, we should get a sequence like $\{(S_1, E_1), (S_2, E_2), \ldots, (S_n, E_n)\}$, where each S_i represents one send packet, and E_i represents one echo packet and matches with S_i [7]. If we only take the first pair for each sequence and put the timestamps information into each pair, then we get the information for each outgoing connection, C2: (t_{2s1}, t_{2e1}), C4: (t_{4s1}, t_{4e1}), C6: (t_{6s1}, t_{6e1}), C8: (t_{8s1}, t_{8e1}). If connection C2 and connection C6 are in a same chain, the relations $0< t_{2s1} < t_{6s1}$, and $t_{2e1} > t_{6e1} > 0$ must be true. If two connections are not in the same chain, such as C4, and C8, the above relations are not likely to be true. However, if we check several consecutive pairs, the probability that the relations held for two connections that are not in a same chain should be very low, otherwise, this probability should be very high. In other words, we can get a gap between a probability that two connections are in a same chain and a probability that two connections are not in a same chain. If this gap is higher than a predefined threshold, we can safely consider the two thumbprints belong to a same connection chain, otherwise, they are not. We use the timestamps of consecutive matched pairs in one connection as our thumbprint, which can be used to identify a connection uniquely and to traceback intruders.

3.2 RTT-Thumbprint

Given a sequence of 'Send' and its matched 'Echo' packet pairs $\{(S_1, E_1), (S_2, E_2),\ldots, (S_n, E_n)\}$ from Host *1* to Host *2*, let $\{(t_{s1}, t_{e1}), (t_{s2}, t_{e2}), \ldots, (t_{sn}, t_{en})\}$ be the corresponding timestamps of $\{(S_1, E_1), (S_2, E_2),\ldots, (S_n, E_n)\}$. We define a RTT temporal thumbprint (RTT-thumbprint) of a connection to be a sequence $\{(t_{s1}, t_{e1}), (t_{s2}, t_{e2}), \ldots, (t_{sn}, t_{en})\}$. Each element represents a timestamps pair of matched send and echo packets. In the following sections, for convenience, we usually use array T[1, 2, …, n] to represent a RTT-thumbprint.

The length of a RTT-thumbprint (n) is vital in tracing intruders; how to select the size depends on the network. For local network, n can be relatively small because of less network fluctuation, such as 64. But for wide area network, n should be larger because of serious network fluctuation. A thumbprint for incoming connection is called incoming RTT-thumbprint, denoted as iRTT-thumbprint, and a thumbprint for outgoing connection is called outgoing RTT-thumbprint, denoted as oRTT-thumbprint. For convenience, we usually use RTT-thumbprint to represent both incoming and outgoing thumbprints.

The first step of creating a RTT thumbprint is to match up send packets with echo packets; the algorithm to do so can be found in [7]. The issue about RTT-thumbprint is how to determine if two RTT-thumbprints match. For example, in some cases, we need to determine if iT-thumbprint is included in oT-thumbprint of a same host, or to determine if oT-thumbprint of one host is included in oT-thumbprint of another host so as to decide if the two hosts are in a same connection chain. We use SR to determine if two RTT-thumbprints are similar. We assume two RTT-thumbprints are $T = \{(t_{s1}, t_{e1}), (t_{s2}, t_{e2}), \ldots, (t_{sn}, t_{en})\}$ and $U = \{(u_{s1}, u_{e1}), (u_{s2}, u_{e2}), \ldots, (u_{sn}, u_{en})\}$, respectively, and can determine what is the longest inclusion sequence of T and U. The problem of the element-inclusion relation incurs a high false positive in some cases. To avoid this problem and to be aware of a fact that is if Host i and Host j are in a same chain, the time gap that a packet propagates from Host i to Host j is supposed to approximately equal to the time gap that the corresponded echo packet propagates from Host j to Host i, in practical, for any given ε between 0 and 1, we usually use the following relation to determine element-inclusion upon two corresponding pairs: (t_{si}, t_{ei}), and (u_{sj}, u_{ej}).

$$\Delta_1 > 0, \Delta_2 > 0, \text{ and } \frac{\left\|\Delta_1\right| - \left|\Delta_2\right\|}{\left|\Delta_1\right| + \left|\Delta_2\right|} < \varepsilon, \tag{1}$$

where $\Delta_1 = u_{sj} - t_{si}$, $\Delta_2 = t_{ei} - u_{ej}$ and suppose there is no clock skew between the two hosts. This is to avoid every small RTT (u_{sj}, u_{ej}) to fall into (t_{si}, t_{ei}). We will match only those that fall somewhere in the middle of the other interval.

3.3 Issues on RTT-Thumbprint Collection and Correlation

We shall address issues related to collecting and correlating RTT-thumbprints. When we collect a RTT-thumbprint in a host, the most important and difficult issue is to find the echoed packet for each packet sent; in other words, it is to match each send packet with each echoed packet. There are several reasons to make packet matching difficult [8, 9, 10, 11, 12].

First, any lost packets during transmission are retransmitted either automatically by the sending client having not received an acknowledgement or on request of the receiving server. Retransmission of the same packet continues until either an acknowledgement is received or until the connection timeout expires. This is going to affect to match a send and an echo packet because we are faced with one echo packet that could match with two or more send packets. Second, cumulative

acknowledgements may take place; this mechanism benefits reducing network traffic, but complicates the packet matching. Third, the size of the transmit window is not necessary to be one, so several packets can be allowed to send continuously without receiving any Ack packet. Several Send-Ack-Echo overlaps each other makes packet matching difficult. Finally, the packets only communicating between two adjacent hosts, such as Ignore packet, Keep-alive and Key re-exchange packet, will also make packet matching complex, as well as complicating RTT-thumbprint correlating because these kinds of packets do not propagate to the victim side. In summary, there is no one-to-one mapping between send packets and echo packets which makes the match difficult.

Another reason that affects RTT-thumbprint correlation is clock skew. We collect each RTT-thumbprint based on the local host clock, which may be different from other host clock. We can not directly compare two RTT-thumbprints with each coming from a different host by using relation (1) because there may be clock skew, about which we are not sure how much it is as it is changing all the time. So if we want to correlate two RTT-thumbprints correctly, we need to determine or estimate the clock skew between two hosts first. We are going to discuss how to estimate clock skew in the following section.

3.4 RTT-Thumbprint Correlating Algorithm

Given two RTT-thumbprints T = {(t_{s1}, t_{e1}), (t_{s2}, t_{e2}), …, (t_{sn}, t_{en})}, and U = {(u_{s1}, u_{e1}), (u_{s2}, u_{e2}), …, (u_{sn}, u_{en})}, supposing there is no clock skew problem, the exhaustive solution to correlating them is to check if U is included in T by comparing with each element of T using relation (1) until all the elements in U are checked. The problem with this solution is that it takes a long time to get the correlating results. Considering a packet propagation scenario, if T and U are in a same chain, the first element of U is supposed to be included in the first element of T, and so on for others. Even in non-ideal situation, it is not necessary to compare each element in U with all the elements in T. We propose a heuristic algorithm to correlate RTT-thumbprints; with this algorithm, instead of comparing with all the elements in T, we only check N elements (N is a predefined number, in our experiment we select N = 4) from the current position of T.

Clock skew is a very important factor for correlating RTT-thumbprints. We propose a method to estimate clock skew between two hosts approximately: Assuming that T and U are RTT-thumbprints collected at Host *i* and Host *j* respectively, and U[j] is included in T[i], We use the fact that the timestamps of a send packet in Host *i* plus the time this packet propagates from Host *i* to Host *j* should approximately equal to the timestamp of the send packet collected in Host *j*. However, due to the clock skew, there will be some difference. We can estimate the packet propagation time and use that to estimate the clock skew for this pair (i,j).

$$cs = (t_{si} + p_t) - u_{sj} \tag{2}$$

Where $p_t = \dfrac{\Delta_t - \Delta_u}{2}$, $\Delta_t = t_{ei} - t_{si}, \Delta_u = u_{ej-}u_{sj}$, and *cs*: clock skew.

3.4.1 Exhaustive Algorithm

For each of the estimated clock skew, the exhaustive algorithm computes the longest inclusion sequence of two thumbprints.

```
program Exhaustive algorithm
  function main(){
  float sr = 0;
  for (i=0; i<=NumT; i++)
     for (j=0; j<=NumU; j++){
        Δt = tei-tsi; Δu=uej-usj; pt=(Δt-Δu)/2;
        cs = u_sj-(t_si + p_t);
        if (compute_SR(cs)>sr) sr = compute_SR(cs);
     }
  return (sr);
  }
  function Compute_SR(cs)
  CurrPT = CurrPU = 0;
  while (there are more elements in U and T){
     lbT=CurrPT; lbU=CurrPU; ubT=NumT; ubU=NumU;
     Match=flase; i=lbU;
     while (i<=ubU && !Match){
        j=lbT;
        while (j<=ubT && !Match){
           Δ1=(u_sj-cs)-t_si; Δ2=t_ei-(u_ej-cs);
           ra1=abs(abs(Δ1)-abs(Δ2))/(abs(Δ1)+abs(Δ2));
           if (Δ1>0 && Δ2>0 && ra1<ε){
              counter++; CurrPT=j+1; CurrPU=i+1;
              Match=true;
           };
           j++;
        }
        i++;
     }
     if (Match) {currPT=j; currPU=i;}
     else {currPT++; currPU++;};
    }
    return SR=counter/min(NumT , NumU);
  }
end.
```

In the above algorithm, U and T are two sequences corresponding two RTT-thumbprints; 'counter' represents the largest number of elements in U that are included in T; NumT, and NumU are the lengths of T, and U; lbT, and lbU are the lower bound of T, and U; ubT, and ubU are the upper bound of T, and U.

The premise that we can use equation (2) to estimate the clock skew is that we assume two corresponded elements T[i] and U[j] are generated by a same packet. Since we don't know this, we have to compute all such pairs and find the best solution. The exhaustive method to estimate clock skew is to traverse all the elements in T and U; that means we are going to get *nm* values to approximate clock skew. We try each value to compute SR, and the largest one of SRs will be the SR with LIS between T and U. Thus we have an $O(m^2n^2)$ algorithm.

3.4.2 Heuristic Algorithm

The exhaustive algorithm can give us best solution for correlating RTT-thumbprint, but with penalty of inefficiency. In most cases it is not necessary to compute all the pairs in T and U to estimate the clock skew. If the two thumbprints are taken at roughly the same time (this can be guaranteed by assumption 2) , most probably that U[i] will correspond to one of the element between T[i] and T[i+N], where N is a predefined number. The heuristic algorithm is similar to exhaustive one but the outer loop in the main algorithm is limited to N iteration, as well as the loop in the function *Compute_SR (cs)*. We will not show this algorithm here because of the limited space. The experiment results showed that the heuristic algorithm has as good performance as the exhaustive one but is more efficient.

We estimate how many basic operations with the heuristic algorithm. There are *mN* estimated clock skews, for each one, we correlate T and U with O(mN) basic operations. Therefore, we have $O(m^2N^2)$ basic operations altogether. The ratio between the heuristic computation and the exhaustive computation is $O(\frac{m^2N^2}{m^2n^2}) = O(\left(\frac{N}{n}\right)^2)$. If we select N=4, and n=64, the computation time of the heuristic algorithm is only 0.39% of that of the exhaustive algorithm. One experiment result showed that the heuristic algorithm cost less than one second while exhaustive one cost hundreds of second when correlating two thumbprints each with length of 64.

3.5 Experiment Results and Analysis

The test environment is set up in the Computer Science Department Lab, at the University of Houston. Our program TT (Thumbprint Traceback) uses Libpcap [13] to do traceback simulation. There are two hosts acl08 and acl09 in local campus domain 'cs.uh.edu' under our control, and there are other hosts Mex (in Mexico), Epic (in California), and Bayou (on the campus) not under our control, but we do have regular accounts to login those machines. We did our experiment conservatively so as to make our results more reliable. We established four connection chains Host *i* (i=1 to 4)→ Acl09 → Mex → Acl08 → Epic → Bayou by using SSH along the same path and did the experiment at the same location and the same time, with the same contents as input. This increased the probability that the connection from a chain is included in another connection from a different chain. This is the worst case that RTT-thumbprint can handle.

Table 1 shows one of the RTT-thumbprint correlating results with the first line for the heuristic algorithm and the second line for the exhaustive algorithm. The value showed on the table is SR, which means how many elements in U (collected in acl08) are included in T (collected in acl09). It is clear that two thumbprints of the same connection chain have very high similarity while those not in a same chain have very low similarity. The results also show us that the heuristic algorithm can be as good as the exhaustive one in SR, but the heuristic is much more efficient and can be used in real-time traceback.

3.6 Avoiding Intruder's Manipulation

RTT-thumbprint is more robust than other methods used to traceback intruders in defeating intruder's manipulation. Most probably, intruders who are aware of the risks of being traced try to evade the traceback by modifying their connections. To defeat the RTT-thumbprint, they may randomly delay the outgoing packets or randomly inject some packets into the connection and so that the outgoing and incoming connections appear unrelated.

Table 1. RTT-thumbprint exhaustive and heuristic correlating results between two hosts on the Internet

Connections at ACL09	Connections at ACL08			
	C0(%)	C1(%)	C2(%)	C3(%)
C0(%)	100.00 100.00	1.26 0.00	0.57 0.57	0.52 0.52
C1(%)	-	92.63 94.66	0.57 0.57	0.00 0.00
C2(%)	-	-	87.42 91.25	0.62 0.62
C3(%)	-	-	-	89.00 93.71

Random delay manipulation cannot affect RTT-thumbprint to traceback intruders. The assumption (4) stated that each packet can only be delayed, rather than accelerated. Suppose all the packets in Host *j* are delayed by an intruder, u_{sj}, and u_{ej} are going to be bigger. No matter how large u_{sj} is, it is always larger than t_{si}, and therefore relation (1) holds all the time for send packets. Echoed packet E_i in Host *i* is always late than the echo packet E_j in Host *j*, so t_{ei} is always bigger than u_{ej} and therefore relation (1) holds whatever how an intruder delays the packets in Host *j*. RTT-thumbprint can defeat random delay manipulation completely.

Chaff manipulation cannot affect RTT-thumbprint to traceback intruders too much. The assumption (4) stated that the chaff will be removed shortly without reaching the destination. We assume an intruder inserts some packets in Host *j*, and then removes them in Host *j+k* (here *k* is any integer that guarantee Host *i+k* is not the destination host). Because all the packets inserted in Host *j* do not reach the destination, we can not capture the matched pair for the chaff either in Host *i* or Host *j*. Obviously, chaff will not affect using RTT-thumbprint to traceback intruders. The only effect is to make downstream and upstream propagation time unsymmetrical, and then affect RTT-thumbprint correlating. But we can handle this problem by adjusting the ε. Therefore, RTT-thumbprint can avoid chaff manipulation.

4 Conclusions and Future Work

We have shown how RTT-thumbprints can be used to characterize a connection session, and discussed how to correlate RTT-thumbprints to traceback intruders.

Besides the advantages that other kinds of thumbprint have, RTT-thumbprints also have the advantage of avoiding random delay and chaff manipulation. Two algorithms have been proposed to correlate RTT-thumbprint: exhaustive and heuristic. The experiment results showed that the heuristic algorithm can get almost the same performance as the exhaustive one but is more efficient.

The results we obtained in this paper are under the four assumptions. In the future, we would like to relax the assumptions, or remove some of them. Such as if the chaff reach the destination host, we will study its effect to RTT-thumbprint collecting, and correlating. We would also like to study the effect of network distance on the correlation of RTT-thumbprints, and to build a real traceback system by using RTT-thumbprints.

References

1. S. Snapp (ed.): DIDS (Distributed Intrusion Detection System)-Motivation, Architecture, and Early Prototype. In Proceedings of 14th National Computer Security Conference, October (1991) 167-176.
2. H. Jung (ed.): Caller Identification System in the Internet Environment. Proceedings of 4th USENIX Security Symposium, (1993) 17-32.
3. Stuart Staniford-Chen , L. Todd Heberlein: Holding Intruders Accountable on the Internet. Proceedings of the 1995 IEEE Symposium on Security and Privacy, Oakland, CA, May (1995) 39-49.
4. Yin Zhang and Vern Paxson: Detecting stepping-stone. Proceedings of the 9th USENIX Security Symposium, Denver, CO, (2000) 67-81.
5. K. Yoda, H. Etoh: Finding Connection Chain for Tracing Intruders. In Proceedings of the 6th European Symposium on Research in Computer Security (LNCS 1985), Toulouse, France, October (2000) 31-42.
6. Jianhua Yang, Shou-Hsuan Stephen Huang: Correlating Temporal Thumbprints for Tracing Intruders. To appear in Proceedings of 3rd International Conference on Computing, Communications and Control Technologies (CCCT'05), Austin, TX, July (2005).
7. Jianhua Yang, Shou-Hsuan Stephen Huang: Matching TCP Packets and Its Application to the Detection of Long Connection Chains, Proceedings (IEEE) of 19th International Conference on Advanced Information Networking and Applications (AINA'05), Taipei, Taiwan, China, March (2005) 1005-1010.
8. T. Ylonen: SSH—Secure Login Connections Over the Internet. In 6th USENIX Security Symposium, San Jose, CA, USA, (1996) 37-42.
9. University of Southern California: Transmission Control Protocol. RFC 793, Sep. (1981).
10. Martin P. Clark, "Data Networks, IP and the Internet Protocols, Design and Operation", Wiley, New York, 2003.
11. T. Ylonen, "SSH Transport Layer Protocol", draft IETF document, http://www.ietf.org /internet-drafts/ draft-ietf-secsh-transport-18.txt, accessed June 2004.
12. T. Ylonen, "SSH Protocol Architecture", draft IETF document, http://www.ietf.org /internet-drafts/draft-ietf-secsh-architecture-16.txt, accessed June 2004.
13. Lawrence Berkeley National Laboratory (LBNL): The Packet Capture library. ftp: //ftp.ee.lbl.gov/libpcap.tar.z, accessed March 2004.

Performance Enhancement of Wireless Cipher Communication

Jinkeun Hong[1] and Kihong Kim[2]

[1] Division of Information & Communication, Cheonan University,
115 Anseo-dong, Cheonan-si, Chungnam, 330-704, Korea
jkhong@cheonan.ac.kr
[2] Graduate School of Information Security, Korea University,
1, 5-Ka, Anam-dong, Sungbuk-ku, Seoul, 136-701, Korea
hong0612@hanmir.com

Abstract. An interleaving algorithm is applied to reduce the loss of ciphered information when a cipher system transmits over a wireless security environment. As such, a new scheme for deciding the interleaving depth over a wireless mobile environment is described. Simulations confirm that the proposed dynamic allocation algorithm (DAA) with a non-fixed interleaving depth produced a better performance over a fading channel than a static allocation algorithm (SAA) with a fixed interleaving depth.

1 Introduction

Aviation industries are undergoing a major paradigm shift in the introduction of new network technologies [1-3]. The Eurocontrol (Europe) is also investigating the feasibility of coordination between ADS-B civil network and tactical network for various aviation needs of the ubiquitous environment and the various ways of migrating aviation authority backbone infrastructure. Tactical information Link16 is a NATO term for a message standard that includes an anti-jam, secure data and voice system with standard waveforms and messages used for exchanging tactical information between different military platforms, thereby providing a common communications network to a large community of airborne, surface, and even subsurface or space elements [4-8].

In previous studies about tactical networks, applying layering principles for legacy system Link16 presented by Warren [4], and White B. [5] presented layered communication architecture for the global grid, while Donal B. F. [6] introduced digital messaging on the Comanche helicopter, the area of tactical data links, air traffic management, and software programmable radios has been researched by B. E. White [7]. As the coordination concept of ADS-B civil network and tactical networks becomes more widespread, the necessity of security for these networks is of increasing importance [8-10].

However, in order to solve security issues in secure tactical networks, the efficiency and transmission performance of security services must be taken into account. From the point of view of aeronautical environmental characteristics, research on optimizing the security considerations of tactical network services, such as low bandwidth, limited consumed power energy and memory processing capacity, and

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 443 – 451, 2005.

cryptography restrictions is important issue. A cipher system using a link-by-link encryption technique is generally used for security. Except for error propagation, the security level is reflected by the period, Common Immunity, and linear complexity and since these properties are easy to implement in terms of hardware and do not create any communication channel delays, a stream cipher system is usually applied to radio communications. However, when enciphered data is transmitted on a radio channel, poor communication channel environments, multi-path fading, and interference result in a burst of errors at the decipher output. The fading received at the mobile unit is caused by multi-path reflections of the transmitted encrypted information by local scatters, such as forests, buildings, and other human-built structures, or natural obstacles such as forests surrounding a mobile unit [11-13].

Interleaving is one practical solution for combating burst errors, where a poor encryption communication channel resulting from a burst of errors can be enhanced using an interleaving scheme, and the transmission performance over a wireless channel and radio communication channel has already been evaluated when using an interleaving method in [14-17]. About the area of interleaving research, X. Gui, et al. [14] proposed a novel chip interleaving in DS SS system, and the subject of multiple access over fading multi-path channels employing chip interleaving code division direct sequence spread spectrum has researched by Y. N. Link, et al. [15], the research of required interleaving depth in Rayleigh fading channels has been proposed by King I. C., et al. [16]. And also, in terms of transmission performance, the performance considerations for secure tactical networks, such as mobility, bandwidth, and BER, are very important.

This paper presents a cipher system for security in Link 16, plus an effective interleaving scheme is applied to the ciphered information to enhance the transmission performance over a fading channel. As such, a frame of ciphered information is lost if the synchronization pattern and session key for the frame are lost. Therefore, applying an interleaving method to reduce the frame loss and thereby enhance the transmission performance would seem to be an effective option that can be evaluated using the non fixed efficient interleaving scheme.

In a cipher system, the synchronization pattern indicates the start and end-point of a frame. If a synchronization pattern is lost due to burst errors, the synchronization pattern can no longer be detected, and since the frame is lost, the encrypted communication channel will fail. Therefore, reducing loss of the SP means reducing loss of the cipher frame. Furthermore, if error bits exist beyond the correcting capability of the transmitting session key stream, decrypting the ciphered information also becomes impossible. As such, this study examines two types of interleaving using a fixed interleaving depth algorithm and non fixed interleaving depth algorithm, where an SAA has a fixed interleaving depth, while a DAA does not. Interleaving methods also include block interleaving, helical interleaving, random interleaving, and extended random interleaving.

Section 2 reviews the nature of a fading channel and provides statistical expressions for burst error sequences, then section 3 outlines the cipher system with synchronization information. Thereafter, interleaving scheme based on a variable depth of interleaving using a non fixed interleaving depth allocation algorithm is explained and simulation results presented in section 4. Finally, section 5 summarizes the results of this study.

2 Characteristics of Wireless Mobile Environment

Wireless fading channel modeling is used to perform a statistical analysis based on defining the relational functions, such as the probability density function (PDF), cumulative probability distribution (CPD), level crossing rate (LCR), average duration of fades (ADF), and bit error rate (BER).

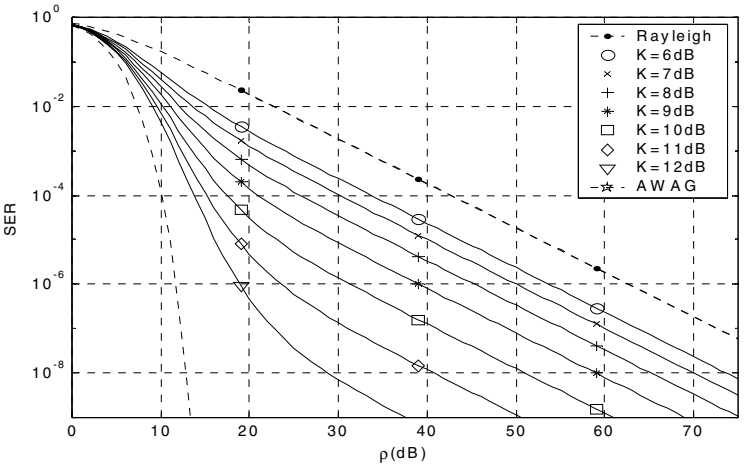


Fig. 1. SER(symbol error rate) of S/N in Rician fading channel

The mean burst length is derived from the defined relational functions and experiments are used to consider the interleaving depth based on the mean burst length. The Rician pdf represents a direct wave plus reflected waves:

$$P_R(\gamma)=\frac{K+1}{\gamma_0}e^{[-K-\frac{\gamma(K+1)}{\gamma_0}]}*I_0[2\sqrt{\frac{\gamma K(K+1)}{\gamma_0}}] \tag{1}$$

where, a is the amplitude of the direct wave, r is the envelope of the fading signal, and $I_0(\cdot)$ is a modified Bessel function of zero order. Therefore, the Rician pdf can be expressed as follows:

$$P(\gamma \le L)=\int_{\gamma=0}^{L} P_R(\gamma)d\gamma \tag{2}$$

In the above equation $\gamma(=\rho=r^2)$ is the C/N ratio and K is the power ratio of the direct wave and reflected waves, and $\gamma_0(=E(r^2)=a^2+2\sigma^2)$ is the power ratio of the received carrier signal and noise. The equation of CPD (F(L)) for Rician fading is used as follows:

$$BER(\rho,K)=\frac{1+K}{2(\rho+1+K)}\exp(\frac{-K\rho}{\rho+1+K}) \tag{3}$$

In a Rician fading channel, the symbol error rate (SER) is applied using differential phase shift keying (DPSK), as in Eq. (4).

$$SER(\rho, K) = 1 - (1 - BER)^5 \tag{4}$$

The crossings of the positive slopes are counted at level *L*. The total number of crossings *N* over a *T* second length of data divided by *T* seconds then becomes the level crossing rate:

$$n(L) = \frac{N}{T} \tag{5}$$

As such, the level crossing rate of a typical fading signal can be calculated. The average duration of fades is defined as the sum of *N* fades at level *L* divided by *N*:

$$t(L) = \frac{\sum_{i=1}^{N} \tau_i}{N} \tag{6}$$

where, τ_i is the individual fade.

$$F(L) = P(\gamma \le L) = \frac{\sum_{i=1}^{N} \tau_i}{T} \tag{7}$$

Now, the product of Eq. (5) and (6) becomes the CPD, and can be derived LCR, ADF [11-13].

Table 1. Mean burst length for variation of SER at K=0, f=969MHz, v=24Km/h

Factor \ Mean power deviation	-25dB	-10dB	0dB	5dB
n_0	53	53	53	53
n_L	0.066	0.284	0.33	0.07
LCR	3.49	15.05	17.49	3.71
ADF	0.00086	0.00658	0.0343	0.2533
CPD	0.003	0.099	0.6	0.94
t_0	0.0189	0.0189	0.0189	0.0189
t_L	0.0455	0.3486	1.8182	18.8
Transmission rate 19.2Kbps	16	128	664	6416
28.8Kbps	25	190	990	10234
57.6Kbps	50	360	1980	20468
115.2kbps	100	720	3960	40936

3 Wireless Cipher System

This paper presents a secure cipher system. Plus, an interleaving scheme is also applied to the ciphered information to enhance the transmission performance over a fading channel. The proposed cipher system consists of a keystream generator, synchronization pattern generator, and session key generator. For the stream cipher

system, the keystream generator generates a number of random sequences of an approximately unlimited period using a seed number of outside keys. Meanwhile, the synchronization pattern is composed of a synchronization pattern generator and synchronization pattern detector. To provide robust encrypted communication, the transmitter and receiver are both synchronized using a synchronization pattern. If the received synchronization pattern is detected normally, the error-corrected coded session key bitstream is received and the ciphered data is deciphered. Plus, a session keystream generator with a nonlinear keystream generator is applied to generate the transmitting session keystream (ECCSK). The initial seed of the keystream generator is composed of the private key using the cipher/decipher and transmitting session key. The Reed-Solomon RS(31,15) is applied to correct the error in the session keystream and ciphered data stream.

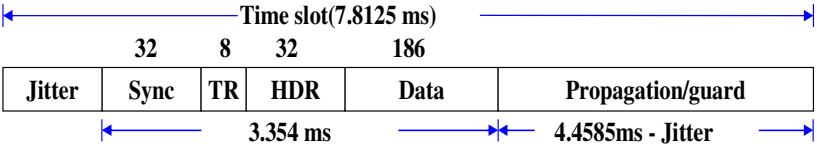


Fig. 2. TDMA time slot architecture of Link 16 tactical data link

The keystream generator was designed considering the security level [18,19], i.e. the linear complexity, randomness, common immunity, period, and composition of a nonlinear function. As such, the security level of a nonlinear keystream generator with a full adder, carry, and feedback memory was satisfied. In the Link16 tactical data link system, the synchronization part is a pattern of DP symbol packets that allows the receipt of JUs to synchronize with the transmission. The pattern is changed from time slot to time slot, and within a time slot the pattern differs between nets. A fixed pattern of four DP symbol packets is used for time refinement (TR). The header part is a word that provides information on the message transmitted in a time slot, then the data is the message transmitted in the time slot. The propagation/guard interval is the time period that allows for the propagation of the signal to the maximum range and time required for the JUs to prepare for the transmissions in the next time slot.

4 Performance of DAA and Experimental Results

When ciphered information is transmitted over a Rician fading channel in which the received signal level is time variant, some of the ciphered information is lost due to burst errors, resulting a loss of the synchronization pattern and error in the session key in a period of synchronization.

To reduce this loss of ciphered information, an interleaving scheme is applied to the ciphered information to enhance the transmission performance over a radio channel. Interleaving is an effective way of randomizing burst errors, plus, burst errors can not be corrected without the application of interleaving and de-interleaving. The function of the received power (n_L) at K=0 can be expressed as follows:

$$\begin{bmatrix} n_{L_0} \\ \dots \\ n_{L_{n-1}} \end{bmatrix} = \begin{bmatrix} L_0 e^{-L_0} \\ \dots \\ L_{n-1} e^{-L_{n-1}} \end{bmatrix} \tag{8}$$

where, K=0. The local mean *m(x)* is where each value corresponds to the average field strength at each local point. The length 2*w* (=40λ) is considered to be the proper length to use fading channel K=0.

$$m(x) = \frac{\sum_{y=0}^{n-1} n_L(y)}{w} \tag{9}$$

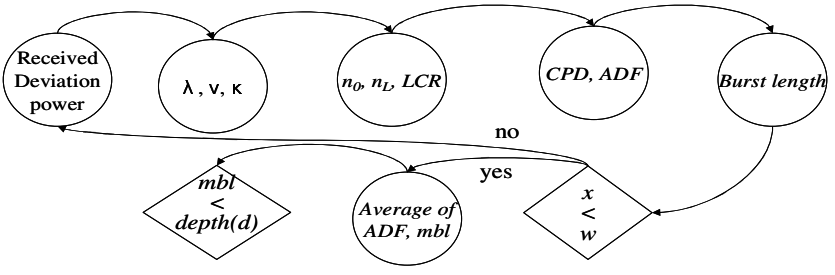


Fig. 3. DAA interleaving flow

However, in practice, *w* within the 20λ to 40λ range is acceptable. If the mobile unit moves slowly, the rate of fluctuation also moves slowly and the length w is increased. For instance, at 969MHz, the wavelength is 0.35m. Assuming that the speed of the mobile unit is 24Km/h, the rate of fluctuation of the signal reception at 10-dB below the received average power of the fading signal is 15 nulls per second. The decision on the interleaving depth can be determined based on the received local mean power *m(x)* per 80 fades, as obtained from Eq. (9).

Where *w* is the time interval for the local mean power. The level crossing rate (*n(L)*) is determined based on the nomalized factor n_0 and value of n_L, while the average duration of fades (*t(L)*) is determined based on the values of $1/n_0$ and CPD/n_L. The relation between the level crossing rate (*n(L)*) and local mean *m(x)*={m_0, m_1, m_2, ..., m_{n-1}} can be expressed as follows {(n_0*m_0), (n_0*m_1),, …, (n_0*m_{n-1})}, where n_0 is 2.5·v/λ. The ADF, *t(L)* can be expressed as follows:

$$\begin{bmatrix} t(L_0) \\ \dots \\ t(L_{n-1}) \end{bmatrix} = \begin{bmatrix} \frac{F(L_0)}{n_0} \\ \dots \\ \frac{F(L_{n-1})}{n_0} \end{bmatrix} \tag{10}$$

Therefore, the relationship between the mean burst length (*mbl*), the transmission rate (*B*), and the average duration of fades (*t(L)*) can be expressed as follows:

$$\begin{bmatrix} mbl_{\,0} \\ \dots \\ mbl_{\,n-1} \end{bmatrix} = \begin{bmatrix} B * t(L_0) \\ \dots \\ B * t(L_{n-1}) \end{bmatrix} \tag{11}$$

Let $\{k_n\}$, n=0, 1, 2, ..., be a constant process with a finite set of states $\{k_0, k_1, ..., k_{n-1}\}$. In deriving the equation, the required condition under which the FEC scheme can still correct all errors is as follows:

$$\begin{bmatrix} k_0 \\ \dots \\ k_{n-1} \end{bmatrix} = \begin{bmatrix} mbl_{\,0} \\ \dots \\ mbl_{\,n-1} \end{bmatrix} * \begin{bmatrix} d_0 \\ \dots \\ d_{n-1} \end{bmatrix} \tag{12}$$

These interleaving schemes were evaluated in a simulation environment where the radio channel was a Rician fading channel, the transmission rate was 28.8Kbps, the frame size was 28.8Kbits, the communication access time was 60minutes, the BER was $10^{-2} \sim 10^{-6}$, the moving velocity was 24Km/h, and the carrier frequency applied was 969MHz.

Table 2. Comparison of delayed time relative to depth of SAA

Transmission rate	depth=26	depth=52	depth=104	depth=208
28.8Kbps	1sec	2sec	4sec	8sec
57.6Kbps	0.5sec	1sec	2sec	4sec
115.2Kbps	0.25sec	0.5sec	1sec	2sec

If the interleaving depth applied is smaller than the required depth, the resulting performance will be even worse than without interleaving. However, it is difficult to adapt the depth of interleaving in a variational fading channel, plus, the required depth should be sufficient to handle the resulting errors in the SAA. Therefore, to adapt the depth of interleaving in the variational fading channel, the flexible DAA method was applied. The resulting performance of the DAA and SAA is shown in Tables 3 and 4, respectively.

When the depth of the SAA was 208, as shown in Table 3, the error bits of the de-ciphered data were degraded 54.5% at a SAA depth of 26, 36.3% at a SAA depth of 52, 6.6% at a SAA depth of 104. When the depth of the DAA was 26, 52, 104, 208, as shown in Table 4, the performance of the DAA block interleaving was better than that of the others. When the iteration of transmission was set at 48, the delay time is given in Table 4. With regard to the corrected symbol rate and delay time, the performance of the DAA was better than that of the SAA.

Table 3. Comparison of error bits relative to depth of SAA (SER : $10^{-2} \sim 10^{-6}$)

SER	depth=26	depth=52	depth=104	depth=208
10-2	5.72E+06	4.90E+06	3.34E+06	3.12E+06
10-3	5.75E+05	4.93E+05	3.34E+05	3.12E+05
10-4	5.73E+04	4.92E+04	3.35E+04	3.12E+04
10-5	5.72E+03	4.93E+03	3.34E+03	3.12E+03
10-6	5.73E+02	4.93E+02	3.34E+02	3.12E+02

The corrected symbol rate in the DAA applied is higher than that of the other types (depth=26, 52, 104). At a SAA depth of 26, the corrected symbol rate was corrected 6.5%, 19.4% at a SAA depth of 52, 45.4% at a SAA depth of 104. Meanwhile, Table 4 presents a comparison of DAA and SAA with 48iterations. When the delayed time when using DAA was about 210sec, however, the delayed time by the SAA depth of 26 was about 48sec, the SAA depth of 52 was 96sec, the SAA depth of 104 was 192sec, the SAA depth of 208 was 384sec. Therefore, when increasing the depth, the corrected symbol rate and delayed time were enhanced. With regard to the delayed time and corrected symbol rate, the performance of the proposed method was superior to that of SAA when applied to allow the delayed time of DAA.

Table 4. Comparison of DAA and SAA with 48 iterations (SER : 10^{-2})

Depth	Corrected symbol rate	Delayed time
DAA	100%	210sec
SAA = 26	6.5%	48sec
SAA = 52	19.4%	96sec
SAA = 104	45.4%	192sec
SAA = 208	100%	384sec

Consequently, the results of the transmission performance when using the DAA and SAA confirmed that the performance of the proposed DAA method was better for the case of signal recovery in an erasure channel.

5 Conclusions

This paper examines a cipher system for security in Link16, plus an interleaving scheme is applied to the ciphered information to enhance the transmission performance over a fading channel. As such, a frame of ciphered information is lost if the synchronization pattern and session key for the frame are lost. Therefore, applying an interleaving method to reduce the frame loss and thereby enhance the transmission performance would seem to be an effective option that can be evaluated using the non fixed interleaving depth scheme. A cipher system was proposed using an effective interleaving scheme for the interleaving depth to enhance the transmission performance of the ciphered information.

Experimental results showed that the BER performance of the proposed efficient interleaving scheme was higher than that of the fixed interleaving depth scheme.

References

1. Mulkerin, T.: Free flight is in the future: large-scale controller pilot data link communication emulation testbed, Aerospace and electronic systems magazine, IEEE, 2003.
2. Oishi, R. T.: Future applications and the Aeronautical telecommunication network, IEEE Aerospace conference, 2001.

3. EUROCONTROL: Feasibility study for civil aviation data link for ADS-B based on MIDS/LINK 16, TRS/157/02, 2000.
4. Warren, J. W.: Applying Layering Principles to Legacy Systems: Link16 as a Case Study, IEEE MILCOM, 2001.
5. White, B. E.: Layered Communication Architecture for the Global Grid, IEEE, MILCOM, 2001.
6. Donald B. F.: Digital messaging on the Comanche helicopter, DASC, 2000.
7. B. E. White: Tactical data links, air traffic management, and software programmable radios, DASC, 1999.
8. H. J. Beker and F. C. Piper, Cipher Systems: The Protection of Communications, Northwood Books, London, 1982.
9. Bruce. S.: Applied Cryptography 2nd ed.: Protocols, Algorithm, and Source code in C, John Willy & Son, New York, 1996.
10. Rainer A. R.: Analysis and Design of Stream Ciphers, Springer-Verlag, Berlin, 1986.
11. W. C. Y. Lee: Mobile Cellular Telecommunications: Analog and Digital Systems, 2nd ed., McGraw-Hill, Singapore, 1996.
12. William C. Y. Lee: Mobile Communications Engineering, McGraw-Hill, New York, 1982.
13. William C. Y. Lee: Mobile Communications Design Fundamentals, John Willey & Sons, New York, 1993.
14. X. Gui and T. S. Ng.: A novel chip interleaving DS SS system, IEEE Trans. Vehicle Technology, vol.49, no.1, pp.21-27, 2000.
15. Y. N. Link and D. W. Lin: Multiple access over fading multi-path channels employing chip interleaving code division direct sequence spread spectrum, IEICE Trans. on Communication, vol.E86-B, no.1, pp.114-121, 2003.
16. King I. C. and Justin C-I C.: Required Interleaving Depth in Rayleigh Fading Channels, Globelcom, vol. 2, pp.1417-1421, 1996.
17. S. H. L., et al.: Effective Interleaving Method in Wireless ATM Networks, ICT, vol.3, pp. 1091-1096, 1997.
18. M. Kimberley: Comparison of two statistical tests for key-stream sequences, Electronics Letters, vol. 23, no.8, pp. 365-366, 1987.
19. Mulkerin, T.: Free flight is in the future: large-scale controller pilot data link communication testbed, Aerospace and electronic systems magazine, IEEE, 2003.

SAS: A Scalar Anonymous Communication System

Hongyun Xu[1,2], Xinwen Fu[3], Ye Zhu[3], Riccardo Bettati[3],
Jianer Chen[3], and Wei Zhao[3]

[1] Hunan University, Changsha, Hunan, 410082, PRC
[2] Central South University, Changsha, Hunan, 410083, PRC
xhongyun@yahoo.com
[3] Texas A&M University, College Station, TX 77840, USA
{xinwenfu, y0z2537, bettati, chen, zhao}@cs.tamu.edu

Abstract. Anonymity technologies have gained more and more attention for communication privacy. In general, users obtain anonymity at a certain cost in an anonymous communication system, which uses rerouting to increase the system's robustness. However, a long rerouting path incurs large overhead and decreases the quality of service (QoS). In this paper, we propose the Scalar Anonymity System (SAS) in order to provide a tradeoff between anonymity and cost for different users with different requirements. In SAS, by selecting the level of anonymity, a user obtains the corresponding anonymity and QoS and also sustains the corresponding load of traffic rerouting for other users. Our theoretical analysis and simulation experiments verify the effectiveness of SAS.

1 Introduction

In this paper, we propose a scalar anonymity system, which provides users a tradeoff between the degree of anonymity, and the cost.

Concerns about privacy and security have gained more and more attention in conjunction with the rapid growth and public acceptance of the Internet as a means of communication and information dissemination. Anonymity has become necessary and legitimate in many scenarios, including anonymous email, web browsing and e-Voting. In these scenarios, encryption alone cannot achieve the anonymity required by participants [1,2,3].

Since Chaum [7] proposed mix networks, researchers have developed various anonymity systems for different applications: message-based, high-latency systems for applications such as anonymous email [13,16] and flow-based, low-latency systems [4, 8, 9] for latency sensitive applications such as anonymous Web browsing or other TCP applications. In this paper, we focus on flow-based mix networks.

In a flow-based mix network, in addition to properly encrypting packets, we need to reroute them through a series of Mixes to achieve strong anonymity in the presence of compromised nodes and global timing attacks. While additional Mixes on the route of a path inherently increase anonymity, long rerouting paths have a negative effect on the QoS, such as end-to-end latency, or TCP throughput. Moreover, most anonymity systems utilize overlay protocols, further increasing the overhead since

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 452 – 461, 2005.

two nodes may span a number of routers. Thus, the performance of TCP applications in anonymity systems is very sensitive to the length of the path in terms of anonymity system nodes.

In this paper we propose the Scalar Anonymity System (SAS) to provide a tradeoff between anonymity and QoS. Our contributions are summarized as follows:

1. SAS provides scalar anonymity for users. That is, based on the QoS requirements of their applications, users can tune to the required level of anonymity.
2. SAS provides scalar forwarding load for users. The forwarding load of a node is defined as this node's number of appearances on all rerouting paths [4, 5]. Users who need a higher level of anonymity experience higher forwarding load.
3. We compare the effectiveness of predecessor attacks [6] against SAS with those of comparable systems. Our simulations show that SAS is highly resilient to predecessor attacks.

The remainder of this paper is organized as follows: We introduce the related work in Section 2 and network model in Section 3. In Section 4, we propose SAS and discuss its performance. We evaluate SAS in Section 5 and conclude the paper in Section 6.

2 Related Work

For anonymous email applications, Chaum [7] proposed to use relay servers, i.e. Mixes, rerouting messages, which are encrypted by public keys of the Mixes. An encrypted message is analogous to an onion constructed by the sender, who sends the onion to the first Mix. By using its private key, the first Mix peels off the first layer, which is encrypted using the public key of the first Mix's public key. After receiving the second Mix's address, the first Mix sends the peeled onion. This process continues in this recursive way. The core part of the onion is the receiver's address and the real message, which is then sent to the receiver by the last Mix. Chaum also proposed the return address and digital pseudonyms for users to communicate with each other in an anonymous way.

Helsingius [12] implemented the first Internet anonymous remailer, which is a single application node that just replaces the original email's source address with the remailer's address. It has no reply function and is subject to all the attacks mentioned below. Eric Hughes and Hal Finney [23] built the cypherpunk remailer, a real distributed mix network with reply functions that uses PGP to encrypt and decrypt messages. The system is subjected to a global passive attack and replay attack to its reply mechanism. Gülcü and Tsudik [13] developed a relatively full-fledged anonymous email system, Babel. Their reply technique does not need the sender to remember the secret seed to decrypt the reply message, but it is subject to replay attack. They studied the threat from the trickle attack, a powerful active attack. Another defect of Babel is that a mix itself can differentiate the forwarding and reply messages. Cottrell [14]developed Mixmaster, which counters a global passive attack by using message padding and also counters trickle and flood attacks [13, 15] by using a pool batching strategy. Mixmaster does not have a reply function. Danezis, Dingledine and Mathewson [16] developed Mixminion. Although Mixminon still has

many problems, its design considers a relatively complete set of attacks that researchers have found [15, 17-21]. The authors suggest a list of research topics for future study.

Low-latency anonymous communication can be further divided into systems using core mix networks and peer-to-peer networks. In a system using a core mix network, users connect to a pool of Mixes, which provides anonymous communication, and users select a forwarding path through this core network to the receiver. Onion routing [8], Freedom [10], and Tor [9] belong to this category. In a system using a peer-to-peer network, every node in the network is a Mix, but can also be a sender and a receiver. Obviously, a peer-to-peer mix network can be very large and may provide better anonymity in the case when many participants use the anonymity service and enough traffic is generated around the network. Crowds [4], Tarzan [11] and P^5 [22] belong to this category.

3 Network Model

In this section, we introduce the network model used in our study. Low-latency anonymous communication can use either core networks or peer-to-peer networks. In a system using a *core network*, users connect to a pool of special nodes providing anonymity service and select a forwarding path through this core network to the receiver. *Onion Routing* [8], *Tor* [9], and many others belong to this category. In a *peer-to-peer anonymity network*, every node contributes to the anonymity service, but it can also be a sender, a receiver, or both. Obviously, peer-to-peer mix networks scale well and provide better anonymity if the number of participants is large. *Crowds* [4], *Tarzan* [11], and many others belong to this category.

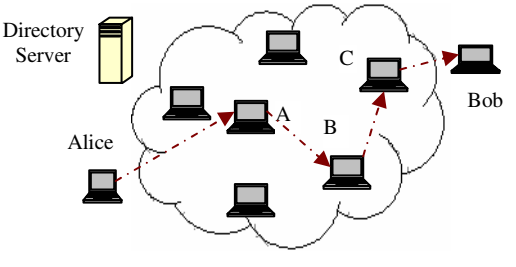


Fig. 1. Network Model

Figure 1 illustrates the network model. Here the Directory Server provides the information of nodes participating in the anonymity network[1]. A sender Alice chooses nodes from the Directory Server and forms a path to the receiver Bob. For example, in Figure 1, Alice chooses relay nodes A, B and C and the path is Alice→A→B →C→Bob, where nodes A, B and C relay Alice's messages to Bob. Obviously, if the

[1] The directory server's function should be similar to the Directory Server used in Tor or Blender in Crowds.

Directory Server only stores a group of specified nodes, the whole anonymity system is core-network based and if the Directory Server stores every participating node, the whole anonymity system is peer-to-peer- network based.

The scalar anonymity system introduced in this paper is a general one and can use either core networks or peer-to-peer networks.

4 Scalar Anonymity System

In this section, we first introduce the framework of the Scalar Anonymity System (SAS), and then analyze its performance.

4.1 Framework of SAS

Rerouting-based anonymity systems typically rely on onion-like encryption against attacks by packet correlation or compromised mix nodes. In this paper, we will not provide elaboration. Instead, we will focus on the forwarding policy.

In this paper, we assume that the mix network adopts a forwarding policy as in Crowds [4], in which a probability guides if a packet is sent to the receiver or next hop. In order to provide a tradeoff between anonymity and QoS for different applications, each node selects a forwarding probability level. Assume that SAS provides m ($m \geq 1$) levels of anonymity corresponding to m different forwarding probabilities, denoted as the forwarding probability set $P_f=\{p_{f1},\ldots, p_{fk},\ldots, p_{fm}\}$. Without loss of generality we assume $p_{f1} < \ldots < p_{fk} < \ldots < p_{fm}$. Here we assume that a bigger forwarding probability provides stronger anonymity, and this assumption holds in our context although it may not hold in other cases [24].

In order to provide a tradeoff between anonymity and forwarding load, every node will be assigned a forwarding load level according to its anonymity level. Assume that SAS maintains $m(m \geq 1)$ levels of forwarding load, denoted as the forwarding load level set $L=\{l_1, \ldots, l_k, \ldots l_m\}$. Each node existing in the directory server will be assigned a weight equal to a forwarding load level from L. Assume that the numbers of nodes with forwarding load levels l_1 *to* l_m are $n_1, \ldots, n_k, \ldots, n_m$ and $n= n_1+\ldots+n_k+\ldots+n_m$, where n is the total number of nodes in the system.

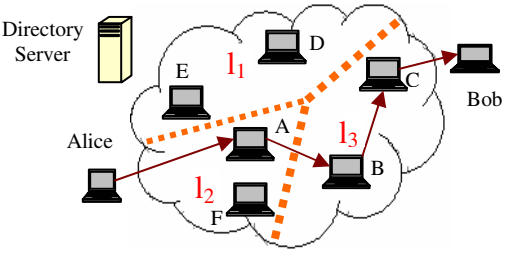


Fig. 2. Scalar Anonymity Systems

Figure 2 illustrates the weight assignment strategy. We have three forwarding load levels l_1, l_2, l_3. Nodes E and D have forwarding load level l_1, nodes A and F have forwarding load level l_2, and nodes B and C have forwarding load level l_3. Alice and Bob can also be assigned the forwarding load level if SAS is a peer-to-peer network based system.

The protocol works as follows.

1. Alice selects a forwarding probability p_{fk}, which is assigned as the forwarding probability for her packets. The directory server assigns to this node, a forwarding load level according to its forwarding probability level, which will be used as the weight of her node.
2. Alice constructs the path using a non-fair coin. When Alice flips the coin, the head appears with a probability p_{fk}. Whenever Alice has a packet to send, she performs the following steps:
   a. She flips the coin. If it turns up with tail facing, she sends the packet to Bob, otherwise, she selects a node to which she forwards the packet. In SAS, a node with weight l_k is chosen with the probability of $l_k / \sum_{i=1}^{m} l_i \cdot n_i$. Thus, the bigger the weight of a node, the higher probability that the corresponding node is chosen as Alice's next forwarding node.
   b. Say Node A in Figure 2 is chosen as Alice's next forwarding node. Node A flips the same coin again and decides if it should forward the packet. This process repeats until one node like Node C in Figure 2 decides to forward the packet to Bob.
3. Once the path is chosen, Alice encrypts the packet in an onion-like way as in Tor and sends the packet out to her first forwarding node.

We claim that the above protocol can achieve anonymity, QoS, and forwarding load in a scalar fashion. Users can choose different forwarding probabilities, and so select a desired level of anonymity. On the other hand, users control the latency (and therefore the QoS) of flows through the forwarding parameter. Furthermore, users can experience different forwarding load by selecting different forwarding probabilities. In the following paragraphs, we verify our claims.

4.2 Anonymity Analysis of SAS

We study SAS's anonymity in terms of the predecessor attack [4,6]. We will show that users with a higher forwarding probability tend to be more resilient to the predecessor attack.

Overview of the Predecessor Attack. This attack is based on the assumption that a sender might choose to remain in contact with a receiver for an extended period of time. In that case, the session between the sender and receiver is subject to a number of resets, (i.e., Web browsing). The interval between two subsequent resets is called a round. A sender will construct a rerouting path in every round. We assume that the attacker compromises some nodes to assist him collecting useful information. In every round, if compromised nodes appear on the rerouting path, the attacker logs the direct predecessor of the first compromised node. After a certain number of rounds,

the attacker analyzes what he has logged and determines the sender, which is selected based on the biggest number of appearances. Please refer to [4,6] for details.

SAS' Resistance to Predecessor Attacks. Recall in SAS, we assume that the numbers of nodes are $n_1,\ldots,n_k,\ldots,n_m$ corresponding to forwarding load level $l_1,..,\ l_k,\ \ldots,\ l_m$, and the total number of nodes is n. Assume that there are c_k compromised nodes in forwarding load level l_k, this leads to the following theorem.

Theorem 1. In SAS, the upper bound of rounds, R_k, that an attacker needs to determine the sender of forwarding probability p_{fk}, is given in **(1)**.

$$R_k = \frac{n'}{c'}\log(n') + \frac{n'}{c' - n'\sigma'}\log(n') \tag{1}$$

where

$$n' = \sum_{i=1}^{m} l_i n_i \quad , \tag{2}$$

$$c' = \sum_{i=1}^{m} l_i c_i \quad , \tag{3}$$

and σ' is given in (4),

$$\sigma' = \frac{p_{fk} c'}{n'^2 - n' p_{fk}(n' - c')} \ . \tag{4}$$

We have the following observations from Theorem 1: R_k, is an increasing function of the forwarding probability p_{fk}. To verify this, we substitute (4) into **(1)**, and get (5) which clearly shows the relationship between R_k and p_{fk}.

$$R_k = \frac{n'}{c'}\log(n')\left(1 + \frac{1}{1 - \frac{n'}{\frac{n'^2}{p_{fk}} - n'(n' - c')}}\right) \tag{5}$$

Thus, an anonymity system can achieve scalar anonymity by letting users choose the forwarding probability p_{fk}.

4.3 Quality of Service in SAS

In this subsection, we will discuss the properties of QoS provided by SAS. Without loss of generality, QoS is measured by the delay which is measured by the length of the rerouting path. We will show that users are able to effectively trade off anonymity against QoS in SAS.

Lemma 1. The expected path length, $E(PathLength_k)$, of a node with forwarding probability p_{fk} can be calculated in (6).

$$E(PathLength_k) = \frac{1}{1 - p_{fk}} \tag{6}$$

From Lemma 1, we can see that, the expected path length is an increasing function of p_{fk}. Thus, an anonymity system like SAS can achieve scalar QoS by letting users choose the forwarding probability p_{fk}, i.e. users can tune the level of anonymity based on the QoS requirements of their applications.

4.4 Forwarding Load in SAS

Recall in SAS, we assume that the numbers of nodes are $n_1,\ldots,n_k,\ldots,n_m$ corresponding to forwarding load levels $l_1,..,\ l_k,\ \ldots,\ l_m$. Assume that there are $P_1,\ P_2,\ldots,P_k,\ldots,\ P_m$ rerouting paths corresponding to the forwarding probability levels $p_{f1},\ p_{f2},\ \ldots,\ p_{fk},\ \ldots,\ p_{fm}$. We have the following theorem.

Theorem 2. In SAS, the expected load, $E(Load_k)$, of a node with forwarding load level l_k is given in (7).

$$E(Load_k) = l_k \frac{\sum_{k'=1}^{m} \frac{P_{k'}}{(1 - p_{fk'})}}{n'}, \tag{7}$$

Where, n' is defined in (2), and $P_{k'}$ is the number of rerouting paths initiated by nodes with forwarding probability $p_{fk'}$.

From Theorem 2, we can see that the expected load of a node with forwarding load level l_k, is an increasing function of l_k. Thus, an anonymity system like SAS can achieve scalar forwarding load.

5 Evaluation

In this section, we will evaluate SAS from three aspects: anonymity, QoS, and forwarding load. In our simulation we choose forwarding load levels l_1=1, l_2=2, l_3=3, and forwarding probability levels p_{f1}=0.5, p_{f2}=0.7, p_{f3}=0.9. The numbers of nodes corresponding to three forwarding load levels are n_1=300, n_2=300, n_3=300 and the numbers of rerouting paths corresponding to three forwarding probability levels are P_1=300, P_2=300, P_3=300. We assume that 1% of the nodes are compromised.

5.1 Anonymity in SAS

Figure 3 shows the simulation results of predecessor attacks determining a sender with different forwarding probability levels in SAS. Each data point is based on the average of thirty runs. The figure shows that if an attacker hopes to find the sender with probability 0.5, they need to observe about 54 rounds in order to find a sender with forwarding probability 0.5, need to observe about 160 rounds in order to find a sender with forwarding probability 0.7 and need to observe 220 rounds in order to find a sender with forwarding probability 0.9. That is to say, the bigger the forwarding probability is, the more rounds for the attacker need to observe.

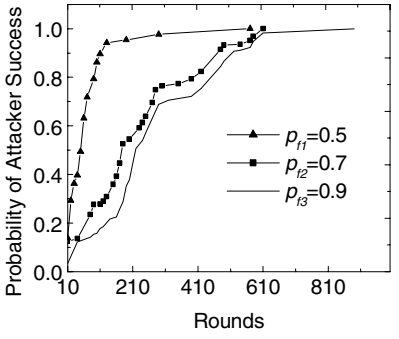


Fig. 3. Simulations of the predece-ssor attack against nodes with different forwarding probability levels in SAS

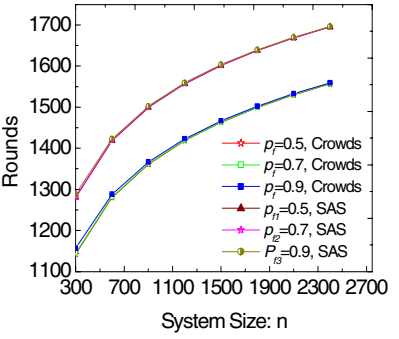


Fig. 4. Comparisons of rounds for predecessor attacks in SAS and in Crowds

Figure 4 shows the comparison results of the round upper bound for an attacker to determine a sender in SAS and in Crowds with high probability. The figure shows that an attacker requires many more rounds to find the sender with high probability in SAS than in Crowds when the compromised nodes are distributed uniformly. For example, in a Crowds system with 600 nodes and a forwarding probability of 0.9, an attacker only needs to observe 1288.312 rounds to determine the sender with high probability when the rate of compromised nodes is 1%. However, In a SAS with the same network, an attacker needs to observe 1422.928 rounds to determine the sender, whose forwarding probability is 0.9 when the rate of compromised nodes is 1%.

5.2 QoS and Forwarding Load in SAS

In SAS, we achieve the tradeoff between QoS and anonymity by controlling the path length and the tradeoff between forwarding load and anonymity by using the forwarding load level.

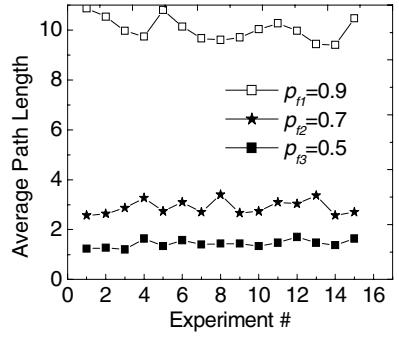


Fig. 5. Simulation of path length for nodes with different forwarding proba- bility levels in SAS

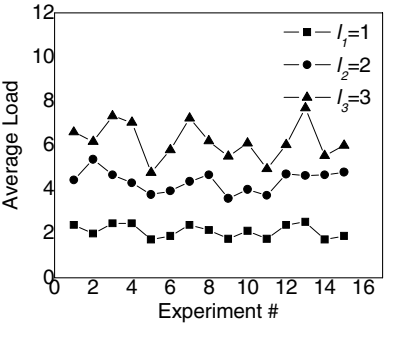


Fig. 6. Simulations of load for nodes with different forwarding load levels in SAS

Figure 5 shows the average path length under different forwarding probabilities. The horizontal axis is the index of our experiments. We carried out 15 groups of simulation. The figure shows that, the bigger the forwarding probability, the longer the rerouting path, hence the longer the delay. Figure 6 shows the simulation results of forwarding load for nodes with different forwarding load levels. Each data point is based on the average forwarding load of all nodes with the same forwarding load levels. The figure shows that, the bigger the forwarding load level, the heavier the forwarding load.

6 Conclusion

In this paper, we propose a scalar anonymous communication system. The scalar anonymity is achieved with layered encryption, and multi-hop routing. In SAS, we achieve a tradeoff among QoS, forwarding load, and anonymity by providing different forwarding probability levels for different nodes, and by assigning a forwarding load level for every node .We demonstrated the effectiveness of SAS by theoretical analyses and simulations. Moreover, we calculated the rounds that an attacker deploying predecessor attack needs in order to determine a sender with high probability in SAS, and compared SAS with Crowds under the same attack. We find that SAS is more resilient to this attack than Crowds when the compromised nodes are uniformly distributed.

References

1. D. X. Song, D. Wagner, and X. Tian: Timing analysis of keystrokes and timing attacks on ssh. In Proceedings of 10th USENIX Security Symposium, 2001
2. Q. Sun, D. R. Simon, Y. Wang, W. Russell, V. N. Padmanabhan, and L. Qiu: Statistical identification of encrypted web browsing traffic. In Proceedings of IEEE Symposium on Security and Privacy, 2002
3. X. Fu, B. Graham, D. Xuan, R. Bettati, and W. Zhao: Empirical and theoretical evaluation of active probing attacks and their countermeasures. In Proceedings of 6th Information Hiding Workshop (IH), 2004
4. M. .K. Reiter and A.D. Rubin: Crowds: Anonymity for Web Transaction. ACM Transaction on formation and System Security, 1(1):66-92, 1998
5. Hongfei Sui, Jianer Chen, Songqiao Chen, Jianxin Wang. Payload analysis of anonymous communication system with host-based rerouting mechanism. Proceeding of the Eighth IEEE International Symposium on Computer and Communication, 2003.
6. Matthew Wright, Micah Adler, Brian Neil Levine, and Clay Shields: An Analysis of the Degradation of Anonymous Protocols. In the Proceedings of the Network and Distributed Security Symposium - NDSS '02, February 2002
7. Chaum D.: Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM, 1981, 24(2), 84-90
8. P. Syverson, D. Goldschlag, and M. Reed: Anonymous Connections and Onion Routing. Proceedings of the IEEE Symposium on security and privacy, Oakland, CA, IEEE CS Press, pp.44-54, May 1997

9. Roger Dingledine, Nick Mathewson, and Paul Syverson: Tor: The Second-Generation Onion Router. In the Proceedings of the 13th USENIX Security Symposium, August 2004
10. Philippe Boucher, Adam Shostack,and Ian Goldberg: Freedom Systems 2.0 Architecture. Zero Knowledge Systems, Inc. White Paper, December 2000
11. Michael J. Freedman, Robert Morris: Tarzan: A Peer-to-Peer Anonymizing Network Layer. In Proceedings of the 9th ACM Conference on Computer and Communications Security(CCS'02), Washington , DC, November 2002
12. Johan Helsingius: Press release: John Helsingius closes his internet remailer. http://www.penet.fi/press-english.html, 1996
13. Ceki Gülcü and Gene Tsudik: Mixing E-mail with Babel. In proceedings of the Network and Distributed Security Symposium- NDSS'96, pages 2-16. IEEE, February 1996
14. Ulf Moller and Lance Cottrell: Mixmaster Protocol – Version 2. http://www.eskiomo. com/~rowdenw/crypt /mix/draft-moeller-mixmaster2-protocol- 00.txt, January 2000
15. A. Serjantov, R. Dingledine, and P. Syverson: From a trickle to a flood: active attacks on several mix types. Citeseer.nj.nec.com/serjantov02from.html, 2002
16. George Danezis, Roger Dingledine, and Nick Mathewson: Mixminion: Design of a Type Ⅲ Anonymous Remailer Protocol. In Proceedings of the 2003 IEEE Symposium on Security and Privacy, May 2003
17. Adam Back, Ulf Moller, and Anton Stiglic: Traffic analysis attacks and trade-offs in anonymity providing systems. In Ira S. Moskowitz, editor, Proceedings of Information Hiding Workshop(IH 2001), pages 245-257. Springer-Verlag, LNCS 2137, April 2001
18. Oliver Bethold and Heinrich Langos: Dummy traffic against long term intersection attacks. In Roger Dingledine and Paul Syverson, editors, Proceedings of Privacy Enhancing Technologies workshop(PET 2002). Springer-Verlag, LNCS 2482, April 2002
19. Oliver Berthold, Andreas Pfitzmann, and Ronny Standtke: The disadvantages of free MIX routes and how to overcome them. In H.Federrath, editor, Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability, pages 30-45. Springer- Verlag, LNCS 2009, July 2000
20. M.Mitomo and K.Kurosawa: Attack for Flash MIX. In Proceedings of ASIACRYPT 2000. Springer-Verlag, LNcs 1976, 2000
21. J. Raymond: Traffic analysis: Protocols, attacks, design issues and open problems. In H.Federrath, editor, Designing Privacy Enhancing Technologies: Proceedings of Internationsl Workshop on Design Issues in Anonymity and Unobservability, volume 2009 of LNCS, pages 10-29. Springer- Verlag, 2001
22. Rob Sherwood, Robby Bhattacharjee, and Aravind Srinivasan: P^5: A protocol for scalable anonymous communication. In Proceedings of the 2002 IEEE Symposium on Security and Privacy, May 2002
23. Parekh, S.: Prospects for remailers – where is anonymity heading on the internet. http://www.firstmonday.dk/issues2/remailers(1996)
24. Y. Guan, Xinwen Fu, R. Bettati and Wei Zhao, An Optimal Strategy for Anonymous Communication Protocols. In Proceedings of IEEE International Conference on Distribtued Computing Systems, 2002

Two New Fast Methods for Simultaneous Scalar Multiplication in Elliptic Curve Cryptosystems

Runhua Shi and Jiaxing Cheng

Key Laboratory of Intelligent Computing & Signal Processing, Anhui University, Ministry of Education, Hefei, Anhui, 230039, PR China
hfsrh@sina.com, cjx@ahu.edu.cn

Abstract. This paper defines a signed factorial expansion of an integer, and proposes two new methods for simultaneous scalar multiplication based on the expansion when multiple bases are fixed in advance. Where the First method can be parallelized easily, and the Second method only requires a half of elliptic points stored of the First method, relatively. In addition, it greatly improves up the implementation speed of simultaneous scalar multiplication when using the vector key in the methods. The theoretical analyses and the implementation results show that our methods are faster than the current fast methods.

1 Introduction

Elliptic curve cryptography (ECC) was proposed in 1985 independently by Victor Miller [1] and Neal Koblitz [2]. Because elliptic curve cryptosystems can provide a higher level of security with smaller key sizes, they have become the focus of intensive research in various fields and several standardizing bodies, such as the IEEE, ANIS etc, have already issued standards for elliptic curve cryptosystems. The basic operation of ECC systems is kP, where k is an integer and P is an elliptic curve point. This operation is called scalar multiplication or point multiplication, which dominates the execution time of elliptic curve cryptographic schemes. Some fast methods [3-10] for scalar multiplication have been proposed. However, some elliptic curve cryptosystems such as signature generation of ECDSA signature scheme require the operation which to compute the point $kP + lQ$ (called multi-scalar multiplication) from elliptic points P, Q and integers k, l. By Shamir's trick [11], the method which is peculiar to multi-scalar multiplication is to compute $kP + lQ$ simultaneously without separation of scalar multiplication kP and lQ. That is, $kP + lQ$ can be efficiently computed by a simultaneous multiple scalar multiplication, which is called simultaneous scalar multiplication [4,5,11-16].

In the article, we review some efficient methods that have been introduced for simultaneous scalar multiplication when multiple bases are fixed, and then propose two new methods based on the signed factorial expansion. The theory analyses and the implementation results show that the new methods are faster than the current methods.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 462 – 470, 2005.

2 Simultaneous Scalar Multiplication

There are some efficient methods for simultaneous scalar multiplication when multiple bases are not known in advance, such as the Shamir method (for simultaneous scalar multiplication) [11], the fast Shamir method, the Joint NAF method [15] and the Joint Sparse Form method [16] etc. But, in this paper, we mainly discuss other methods when multiple bases are fixed. A simultaneous scalar multiplication method can be separated two stages [12], that is, in the precomputation stage, we compute elliptic points which are required at the evaluation stage, and store them in a table. In the evaluation stage, we compute the multi-scalar multiplied point using the table that is prepared at the precomputation stage.

2.1 Simultaneous Multi-scalar Multiplication Method

When multiple bases are fixed, a precomputed reference table can accelerate computational speed. While the original Simultaneous Multiple Exponentiation (SME) method with a precomputed reference table was applied to multiple modular exponentiation [6], the SME method can be modified to support multiple scalar multiplication over an elliptic curve as follows [13]:

Algorithm 1. Simultaneous Multi-Scalar Multiplication Method

Input: $k = (k_{n-1}k_{n-2}\cdots k_0)_2$, $l = (l_{n-1}l_{n-2}\cdots l_0)_2$, P, Q.
Output: $kP + lQ$.
//the precomputation stage:
For i from 0 to 3 do
  $M[i] \leftarrow \sum_{j=0}^{1} i_j G_j$, where $i = (i_1, i_0)_2$, $G_1 = P$, $G_0 = Q$.
//the evaluation stage:
$R \leftarrow O$; // the point at infinity defined by O
For i from $n-1$ down to 0 do {
  $R \leftarrow 2R$;
  $R \leftarrow R + M[(k_i, l_i)_2]$
}
Return (R).

The precomputation stage needs 3 elliptic points stored, and the evaluation stage requires $n-1$ point doublings and $3(n-1)/4$ point additions on average [13].

2.2 Simultaneous 2^w-ary Method

The simultaneous 2^w-ary method is improved to compute multi-scalar multiplication from the 2^w-ary method, which is proposed to compute single scalar multiplication. In the precomputation stage of the simultaneous 2^w-ary method, it computes elliptic points $t_1P + t_2Q$ for any $t_1, t_2 \in [0, 2^w - 1]$, from elliptic points P, Q and a window

width w, which needs $2^{2w}-1$ elliptic points stored. In the evaluation stage, it computes the multi-scalar multiplied point $kP+lQ$ using the table prepared at the precomputation stage, which requires $n-1$ point doublings and $(n-1)(1-2^{-2w})/w$ point additions on average [12]. The algorithm is given as follows:

Algorithm 2. Simultaneous 2^w-ary Method

Input: a window width w, $k=(k_{n-1}k_{n-2}\cdots k_0)_2$, $l=(l_{n-1}l_{n-2}\cdots l_0)_2$, P,Q.
Output: $kP+lQ$.
//the precomputation stage:
t_1P+t_2Q, where $(t_1,t_2)\in\{0,...,2^w-1\}^2$.
//the evaluation stage:
$R\leftarrow O$;
For i from $\lfloor (n-1)/w \rfloor w$ down to 0 step w do {
  For $j=1$ to w do $R\leftarrow 2R$;
  If $((k[i+w-1...i],l[i+w-1...i])\neq(0,0))$ then
    $R\leftarrow R+(k[i+w-1...i]P+l[i+w-1...i]Q)$, where each $k[i+w-1...i]$ and $l[i+w-1...i]$ is an integer of a bit string of length w.
}
Return (R).

2.3 Simultaneous Sliding Window Method

The Simultaneous Sliding Window method is introduced at [12,14]. We assume that $w\geq 2$ for the window width w in Simultaneous Sliding Window method. The precomputation stage needs $2^{2w}-2^{2(w-1)}$ elliptic points stored, and the evaluation stage requires $n-1$ point doublings and $(n-1)/(w+1/(2^2-1))$ point additions on average [12]. This method is described in Algorithm 3.

Algorithm 3. Simultaneous Sliding Window Method

Input: a window width w, $k=(k_{n-1}k_{n-2}\cdots k_0)_2$, $l=(l_{n-1}l_{n-2}\cdots l_0)_2$, P,Q.
Output: $kP+lQ$.
//the precomputation stage:
t_1P+t_2Q, where $(t_1,t_2)\in\{0,...,2^w-1\}^2$, and at least one of the t_i is odd.
//the evaluation stage:
$R\leftarrow O$;
$i\leftarrow n-1$;
While $i\geq 0$ do
  If $k[i]=0$ and $l[i]=0$ then { $R\leftarrow 2R$; $i\leftarrow i-1$ }
  Else {

$i_{new} \leftarrow \max\{i-w,-1\}$;
$I \leftarrow i_{new}+1$;
While ($k[I]=0$ and $l[I]=0$) do $I \leftarrow I+1$;
$s_1 \leftarrow k[i...I]$; $s_2 \leftarrow l[i...I]$;
While $i \geq I$ do
{ $R \leftarrow 2R$; $i \leftarrow i-1$;}
$R \leftarrow R+s_1P+s_2Q$;
While $i > i_{new}$ do
{ $R \leftarrow 2R$; $i \leftarrow i-1$ }
}
Return (R) .

3 Proposed Methods

Firstly we introduce two following theorems, which will be used in the proposed methods.

Theorem 1: k is an integer. If $0 \leq k < n!$, then k can be represented as:

$$k = \sum_{i=1}^{n-1} k_i (i!) \ .$$

Where $0 \leq k_i \leq i$. It is called the factorial expansion of k .

Proof. Let $\omega_0 = k$, and $k_1 = \omega_0 - 2\times\lfloor \omega_0/2 \rfloor$, $\omega_1 = \lfloor \omega_0/2 \rfloor$;
($\lfloor x \rfloor$: The largest integer less than or equal to x .)
$k_2 = \omega_1 - 3\times\lfloor \omega_1/3 \rfloor$, $\omega_2 = \lfloor \omega_1/3 \rfloor$; … …
$k_{n-1} = \omega_{n-2} - n\times\lfloor \omega_{n-2}/n \rfloor$, $\omega_{n-1} = \lfloor \omega_{n-2}/n \rfloor$. Then,

$$\begin{aligned}\sum_{i=1}^{n-1} k_i (i!) &= (\omega_0 - 2\times\lfloor \omega_0/2 \rfloor)\cdot 1! + (\omega_1 - 3\times\lfloor \omega_1/3 \rfloor)\cdot 2! + (\omega_2 - 4\times\lfloor \omega_2/4 \rfloor)\cdot 3! + \cdots + \\ &\quad (\omega_{n-3} - (n-1)\times\lfloor \omega_{n-3}/(n-1) \rfloor)\cdot (n-2)! + (\omega_{n-2} - n\times\lfloor \omega_{n-2}/n \rfloor)\cdot (n-1)! \\ &= \omega_0 - n!\times\lfloor \omega_{n-2}/n \rfloor .\end{aligned}$$

Where $\lfloor \omega_{n-2}/n \rfloor = \lfloor \lfloor \omega_{n-3} \rfloor / n(n-1) \rfloor \leq \lfloor \omega_{n-3}/n(n-1) \rfloor \leq \cdots \leq \lfloor \omega_0/n! \rfloor = 0$. So,
$k = \sum_{i=1}^{n-1} k_i (i!)$.

Theorem 2: k is an integer. If $0 \leq k < n!$, then k can be expanded as:

$$k = \sum_{i=1}^{n-1} k_i^{'} (i!) \ .$$

Where $\left|k_i^{'}\right| \leq \lfloor (i+1)/2 \rfloor$. It is called the signed factorial expansion of k .

Note that, given $(i+1)\cdot(i!) = (i+1)!$, Theorem 2 can be easily proved from the factorial expansion of k . For example, $5188 = 0\cdot 1! - 1\cdot 2! + 1\cdot 3! + 1\cdot 4! + 1\cdot 5! + 0\cdot 6! + 1\cdot 7!$, which is the signed factorial expansion of 5188. That is, 5188 can be represented as the vector $(0,-1,1,1,1,0,1)$ over the base $(1!,2!,3!,4!,5!,6!,7!)$. Based on the signed

factor expansion, we present two new methods for simultaneous scalar multiplication as follows:

3.1 The First Method

According to Theorem 2, the integers of k and l are represented respectively as $k=\sum_{i=1}^{n-1}k_i^{'}(i!)$ and $l=\sum_{i=1}^{n-1}l_i^{'}(i!)$, where $|k_i^{'}|\le\lfloor(i+1)/2\rfloor$ and $|l_i^{'}|\le\lfloor(i+1)/2\rfloor$. Then $kP+lQ$ can be computed as follows:

$$
\begin{aligned}
kP+lQ &= (\sum_{i=1}^{n-1}k_i^{'}(i!))P+(\sum_{i=1}^{n-1}l_i^{'}(i!))Q \\
&= k_1^{'}\cdot P+k_2^{'}\cdot(2!\cdot P)+\cdots+k_{n-2}^{'}\cdot((n-2)!\cdot P)+k_{n-1}^{'}\cdot((n-1)!\cdot P) \\
&\quad + l_1^{'}\cdot Q+l_2^{'}\cdot(2!\cdot Q)+\cdots+l_{n-2}^{'}\cdot((n-2)!\cdot Q)+l_{n-1}^{'}\cdot((n-1)!\cdot Q) \\
&= \sum_{i=1}^{2n-2}(k_i\cdot P_i)=\sum_{d=1}^{h}d(\sum_{i:|k_i|=d}a_i\cdot P_i)=\sum_{d=1}^{h}d\cdot Q_d \\
&= Q_h+(Q_h+Q_{h-1})+\ldots+(Q_h+Q_{h-1}+\ldots+Q_1)\ .
\end{aligned}
\tag{1}
$$

Where $k_i=\begin{cases}k_i^{'} & i\le n-1\\ l_{i-n+1}^{'} & i>n-1\end{cases}$, $P_i=\begin{cases}i!\cdot P & i\le n-1\\ (i-n+1)!\cdot Q & i>n-1\end{cases}$, $a_i=\begin{cases}1 & if\quad k_i>0\\ -1 & if\quad k_i<0\end{cases}$, $h=\max\{|k_1|,|k_2|,\ldots|k_{2n-2}|\}\le\lfloor n/2\rfloor$. The First method includes two stages: the precomputation stage and the evaluation stage, which is described in Algorithm 4. In the precomputation stage, we compute elliptic points P_i ($1\le i\le 2n-2$), and store them in a table; in the evaluation stage, we use the same strategy as the BGMW method [9] to compute $\sum_{d=1}^{h}d\cdot Q_d$ with the precomputation table.

In Algorithm 4, the precomputation stage requires at most $2n-2$ elliptic curve points stored. The evaluation stage requires $(2(n-1)\cdot\eta-1)+(h-1)$ point additions on average, where $\eta=2\cdot(2/3+4/5+\ldots+(2\lfloor n/2\rfloor)/(2\lfloor n/2\rfloor+1))/(n-1)$, which is the probability of the non-zero of k_i ($1\le i\le 2(n-1)$).

Algorithm 4. The First Method

Input: P,Q,k,l, where $k=\sum_{i=1}^{n-1}k_i^{'}(i!)$, $|k_i^{'}|\le\lfloor(i+1)/2\rfloor$,
and $l=\sum_{i=1}^{n-1}l_i^{'}(i!)$, $|l_i^{'}|\le\lfloor(i+1)/2\rfloor$.

Output: $kP+lQ$.

//The precomputation stage:

$P_1\leftarrow P$;

$P_n\leftarrow Q$;

For $i=2$ to $2(n-1)$ do

If $i \le n-1$ then $P_i \leftarrow i \cdot P_i$

Else $P_i \leftarrow (i-n+1) \cdot P_i$.

//The evaluation stage:

$A \leftarrow O$; $B \leftarrow O$;

For d=h down to 1 do {

For i=0 to 2(n-1) do {

If $k_i = d$ then $B \leftarrow B + P_i$;

If $k_i = -d$ then $B \leftarrow B - P_i$ }

$A \leftarrow A + B$;

}

Return (A) .

By formula (1), the First method can be parallelized. Suppose that we have h processors. Then, each processor can calculate its Q_d separately. The time needed to calculate Q_d depends on the number of k_i 's equal to d . In addition, computing $\sum_{d=1}^{h} d \cdot Q_d$ can still be computed in parallel, which takes $O(\log h)$ point additions to calculate the final result.

3.2 The Second Method

The Second method especially fits to compute $k_1P_1 + k_2P_2 + ... + k_sP_s$, where s is fairly larger (for example, $s = 10$). The larger s is, the larger the ratio of speed-up for the Second method is. Similarly, since any integer k can be represented as $k = \sum_{i=1}^{n-1} k_i^{'}(i!)$ where $\left|k_i^{'}\right| \le \lfloor (i+1)/2 \rfloor$, $k_1P_1 + k_2P_2 + ... + k_sP_s$ can be computed as follows:

$$
\begin{aligned}
k_1P_1 + k_2P_2 + ... + k_sP_s &= \sum_{j=1}^{s} \sum_{i=1}^{n-1} k_{j,i}^{'}(i!)P_j \\
&= \sum_{i=1}^{n-1} (\sum_{j=1}^{s} k_{j,i}^{'}(i!)P_j) = \sum_{i=1}^{n-1} i!(\sum_{j=1}^{s} k_{j,i}^{'}P_j) \\
&= 1(2(\cdots n - 2(n-1((k_{1,n-1}^{'}P_1 + ... + k_{s,n-1}^{'}P_s) + (k_{1,n-2}^{'}P_1 + ... + k_{s,n-2}^{'}P_s)) \cdots \\
&\quad (k_{1,2}^{'}P_1 + ... + k_{s,2}^{'}P_s)) + (k_{1,1}^{'}P_1 + ... + k_{s,1}^{'}P_s)) \ .
\end{aligned}
\tag{2}
$$

Where $k_j = \sum_{i=1}^{n-1} k_{j,i}^{'}(i!)$, $\left|k_{j,i}^{'}\right| \le \lfloor (i+1)/2 \rfloor$. The Second method also includes two stages: the precomputation stage and the evaluation stage. In the precomputation stage, it computes elliptic points $k_{j,i}^{'}P_j$ ($1 \le j \le s, 1 \le i \le n-1$), and store them in a table; in the evaluation stage, it computes iterated point addition and point multiplication with the precomputation table by formula (2), where computing point

multiplications of $(n-1)$ small integers uses the binary method. The Second method sees Algorithm 5.

In Algorithm 5, the precomputation stage requires $s\lfloor n/2 \rfloor - s$ elliptic points stored. The evaluation stage requires $D(n)$ point doublings and $A(n)+s\eta(n-1)-1$ point additions on average, where $D(n)$, $A(n)$ respectively denote the total number of point doublings, point additions which it requires in order to compute point multiplications of the group small integers 1,2,...,$n-1$ (For example, for the 160-bit integer, $n=41$, $D(n)=145$, $A(n)=62$).

Algorithm 5. The Second Method

Input: $P_1, P_2, ..., P_s, k_1, k_2, ..., k_s$, where $k_j = \sum_{i=1}^{n-1} k'_{j,i}(i!)$, $\left|k'_{j,i}\right| \le \lfloor (i+1)/2 \rfloor$, $j \in [1,s]$.

Output: $k_1 \cdot P_1 + k_2 \cdot P_2 + \cdots + k_s \cdot P_s$.

//the precomputation stage:

Compute tP_i for $1 < t \le \lfloor n/2 \rfloor, 1 \le i \le s$.

//the evaluation stage:

$Q \leftarrow O$;

For *i* from *n*-1 down to 1 do {

For *j* from 1 to *s* do If $k'_{j,i} \neq 0$ then $Q \leftarrow Q + (k'_{j,i} P_j)$;

$Q \leftarrow iQ$ } //using the binary method

Return (Q).

4 Comparison

Given in Table 1 is a comparison of our methods with other efficient methods for simultaneous scalar multiplication when multiple bases are fixed. Note that the number of bases is 10 for the Second method, but 2 for other methods in Table 1.

Table 1. The rough costs of multi-scalar multiplication (160-bits)

Method ($kP + lQ$)	Window width *w*	Elliptic point stored	Elliptic operations (Av.)	
			Doubling	Addition
Simultaneous Multi-Scalar Multiplication	-	1	159	119.25
Simultaneous 2^w -ary	4	253	159	39.84
Simultaneous Sliding Window	4	192	159	36.70
First	-	78	0	91.98
Second ($k_1P_1 + k_2P_2 + ... + k_{10}P_{10}$)	-	190	145	430.9

By Table 1, it is clear that our methods are faster than other methods, relatively. In addition, the First method is faster than the Second method and can be computed in parallel, but the Second method only requires a half of elliptic points stored of the First method, relatively.

5 Implementation

In this section we give average timings of our methods and the Simultaneous 2^w-ary method, for our C++ implementation on an IBM PC (1.67GHz\256MB). We use elliptic curve over field of $F_{2^{163}}$ recommended by NIST [5].

While implementing our methods, we propose the idea which key is represented by a vector. It is one-to-one between the integer k and the vector $(k_1^{'}, k_2^{'}, ..., k_{n-1}^{'})$ by $k = \sum_{i=1}^{n-1} k_i^{'} \cdot i!$ and $\left|k_i^{'}\right| \le \lfloor (i+1)/2 \rfloor$. So, we may take the vector $(k_1^{'}, k_2^{'}, ..., k_{n-1}^{'})$ as the vector key instead of the integer key k, which can improve up the computational speed because it saves the timings of computing the coefficients $k_i^{'}$ $(1 \le i < n)$. That is, we require not to compute the coefficients $k_i^{'}$ in the signed factorial expansion of k, but to directly make the vector $(k_1^{'}, k_2^{'}, ..., k_{n-1}^{'})$ at random.

Table 2. The costs of multi-scalar multiplication

Method	Window width w	Elliptic points stored	Time (Av., ms)
Simultaneous 2^w-ary $(kP + lQ)$	4	253	23.78
First $(kP + lQ)$	-	78	10.43
Second $(k_1P_1 + k_2P_2 + ... + k_{10}P_{10})$	-	190	61.63

The implementation results are given in Table2. It is obvious that the First method for multi-scalar multiplication is fastest in Table 2. For example, it speeds up 128% compared with the Simultaneous 2^w-ary method and needs fewer stored points. Furthermore, the First method can be computed in parallel, thus it could lead to efficient implementations of elliptic curves cryptosystems in multiprocessor architectures. In addition, the Second method only requires a half of elliptic points stored of the First methods, relatively. So it very fits to compute simultaneous scalar multiplication when the number of multi-base is larger.

6 Conclusion

A large integer can be expanded into a signed factorial expansion, and is represented as a group of signed small integers. Correspondingly, multi-scalar multiplication can

be simultaneously obtained from the single scalar multiplications of multi-group signed small integers. We introduce two new methods for simultaneous scalar multiplication based on the signed factorial expansion where the First method can be parallelized easily, and the Second method especially suits to the computation when the number of multi-base is larger (e.g., Multi-party protocols). Then we implement our methods using the vector key. The theoretical analyses and the implementation results show that our methods are faster than the current methods when multiple bases are fixed in advance.

References

1. V.Miller: Uses of elliptic curves in cryptography. Advances in Cryptology - CRYPTO '85, LNCS, vol.218. Springer-Verlag, Berlin, (1986) 417-426
2. N.Koblitz: Elliptic curve cryptosystems. Math. Comp., vol.48 (1987) 203-209
3. Julio López and Ricardo Dahab: An Overview of Elliptic Curve Cryptography. Relatório Té-cnico IC-00-10. Institute of Computing, State University of Campinas, Braizl (2000)
4. M.Brown, D.Hankerson, J.Lopez and A.Menezes: Software implementation of the NIST elliptic curves over prime fields. Topics in Cryptology—CT-RST 2001, LNCS, vol.2020 (2001) 250-265
5. D.Hankerson, J.López, and A.Menezes: Software Implementation of Elliptic Curve Cryptography Over Binary Fields. Cryptographic Hardware and Embedded systems-CHES 2000, LNCS, vol.1965 (2000) 3-24
6. Alfred J.Menezes, Paul C van Oorschot and Scott A. Wanstone: Handbook of Applied Cryptography. CRC Press (1996)
7. F.Morain and J.Olivos: Speeding up the computations on an elliptic curve using addition-subtraction chains. Informatique théorique et applications, vol.24 (1990) 531-544
8. J.Solinas: Efficient arithmetic on Koblitz curves. Designs, Codes and Cryptography, vol.19 (2000) 195-249
9. E.F.Brickell, D.M.Gordon, K.S.McCurley, and D.B.Wilson: Fast exponentiation with precomputation. Advances in Cryptology-Proceedings of EURO-CRYPT'92, LNCS, vol.658. Springer-Verlag, Berlin (1993) 200-207
10. C.H.Lim and P.J.Lee: More flexible exponentiation with precomputation. Advances in Cryptology-Crypto'94, LNCS, vol.839. Springer-Verlag, Berlin (1994) 95-107
11. ElGamal,T: A public-key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory, vol.31 (1985) 469-472
12. Bodo Möller: Algorithms for multi-exponentiation. In S.Vaudenay, A.M. Youssed(Eds): Selected Area in Cryptography-SAC 2001. LNCS, vol.2259. Spinger-Verlag, Berlin (2001) 165-180
13. Kunio Kobayashi, Hikaru Morita, and Mitsuari Hakuta: Multiple Scalar-Multiplication Algorithm over Elliptic Curve. IEICE TRANS. INF. &SYST., VOL.E84-D, NO.2 (2001) 271-276
14. Yen,S.-M., Laih,C.-S., and Lenstra,A.: Multi-exponentiation. IEE Proceedings-Computers and Digital Techiques, vol.141 (1994) 325-326
15. Roberto M.Avanzi: On Multi-Exponentiation In Cryptography. Available at http://www.arehcc.com (2002)
16. J.A.Solinas: Low-Weight Binary Representations for Pairs of Integers. Technical Report CORR 2001-41, CACR. Available at http://www.acr.math.uwaterloo.ca/techreports/ 2001/ corr2001-41.ps (2001)

Network-Based Anomaly Detection Using an Elman Network*

En Cheng, Hai Jin, Zongfen Han, and Jianhua Sun

Cluster and Grid Computing Lab,
Huazhong University of Science and Technology, Wuhan 430074, China
hjin@hust.edu.cn

Abstract. An intrusion detection model based on Elman network is proposed to detect anomalies in network traffic. The model applies an Elman network for anomaly detection in order to provide the detector with an internal memory and therefore necessary dynamic characteristics. Unlike the existing applications of Artificial Neural Networks to detect intrusion that extract a set of attributes from only the packet headers but discard the packet payload, the present model adopts the concept of clustering the payload to alleviate information loss by retaining part of the information related to the packet payload. The model has been applied to DARPA IDS Evaluation dataset and the results demonstrate that with the two unique features, the model can identify not only intra-packet anomalies, but also inter-packet sequence anomalies.

1 Introduction

Intrusion detection has become a critical technology to ensure the security of network systems on which companies and government agencies have an ever increasing dependence. Intrusion detection schemes can be classified into two categories: misuse and anomaly detection. Misuse detection (also known as signature detection) compares a user's activities with the known behaviors of attackers attempting to penetrate a system [1][2], but has the obvious disadvantage of incapability of detecting new unknown attacks. Anomaly detection attempts to detect intrusions by finding significant departures from normal behavior [3][4], and thus is capable of detecting new attacks. However, one clear drawback of anomaly detection is its inability to identify the specific type of the current attack.

A recent study conducted by U.S. Defense Advanced Research Projects Agency (DARPA) highlights the strengths and weaknesses of current researching approaches to intrusion detection. This study also indicates that it is imperative to introduce a new paradigm for intrusion detection, which can provide reasonable levels of detection against novel attacks and even variations of known attacks. Therefore, an alternative solution should be implemented for an anomaly

* This work is supported by National Natural Science Foundation of China under grants No.90412010.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 471–480, 2005.

detection system to model what is normal instead what is anomalous in order to recognize future unseen, but similar behavior by generalizing from previously observed behavior.

Our investigation in this paper focuses on applying an Elman network to generate models for detecting anomalies in network traffic. To overcome the common limitations of the currently available models adopting *Artificial Neural Networks* (ANNs), i.e., discarding the payload and retaining only the information in the packet header, the clustering information of the payload is applied in our model. This is because though the models built with features extracting from only the packet header can detect DoS and probe attacks on the TCP/IP stack, but can not detect attacks with exploit code transmitted to a public server in the application payload. In addition, time representation is believed to be an important element when analyzing network traffic considering that DoS attacks take place using a number of successive packets that are targeted towards a host in a finite time span [5]. Unlike previous applications of ANNs to intrusion detection which lack the necessary dynamic characteristics, our study adopts an Elman network for anomaly detection in order to provide the detector with an internal memory which has the same function as the time window used in perceptrons. With two unique features, an Elman network based anomaly detector is expected to identify not only intra-packet anomalies, but also inter-packet sequence anomalies.

The rest of this paper is organized as follows. In section 2, we discuss some research background. In section 3, related work in the field is reviewed. In section 4, we present our network-based anomaly detection model using an Elman network. In section 5, we evaluate our intrusion detection model using DARPA IDS Evaluation dataset. Section 6 ends with a conclusion and some discussions.

2 Research Background

With the potential to resolve a number of problems encountered by other approaches to intrusion detection, ANNs has been identified as a very promising technique of intrusion detection systems. The first advantage in the detection of instances of anomaly is the flexibility that the network can provide [6]. ANNs is capable of analyzing the data from the network, even if the data is incomplete or distorted. Similarly, ANNs can conduct an analysis with non-linear data. These characteristics are important in a networked environment where the received information is subject to a random failure of the system. Therefore, ANNs has been proposed as an alternative to the statistical analysis component of anomaly detection systems [8][9]. However, existing applications of ANNs to intrusion detection extract a set of attributes just from packet headers but discard packet payload, thus leading to an unacceptable information loss: most attacks, in fact, are detectable with the consideration of the payload of a packet.

There are two main types of ANNs: feed-forward and recurrent neural networks. The former allows flow of information from the input layer to the output layer in one direction only and is called *Feed-forward Neural Networks* (FNNs).

The latter allows information to loop back to the same processing element and named as *Recurrent Neural Networks* (RNNs). In purely feed-forward network topologies, the output produced by any input is independent of prior inputs. While this characteristic is appropriate for tasks which require processing of independent inputs, it is not desirable when the inputs are sequential elements of a stream of data. On the other hand, networks with a dynamic memory or recurrent networks are more suitable for representing a dynamic system, which has a dynamic mapping between its output and input. Therefore, RNNs has attracted the attention of researchers in the field of dynamic system classification. Among the various networks proposed in literature, the Elman network shows unique advantages because of its internal time representation: the hidden units can map both an external input and also the previous internal state (by means of the *context* units) to some desired output, developing internal representations of the temporal properties of the sequential input. This favorable feature makes the Elman network very suitable to deal with intra-packet and inter-packet correlation.

3 Related Work

ANNs has been shown to be capable of identifying TCP/IP network events. There are some different researches on the application of ANNs for intrusion detection. Lee and Heinbuch [10] use hierarchical back-propagation neural network to detect SYN flooding and port scanning intrusions. In HyperView [7], a sample of the system's normal traffic is fed to an ANN, which subsequently learns to recognize certain features of normal traffic. A system developed by Rhodes *et al.* [11] uses multiple *Self-Organizing Maps* (SOMs) for intrusion detection. They use a collection of more specialized maps to process network traffic for each layered protocol separately. SOMs have also been used as anomaly intrusion detectors [12]. A SOM is used to cluster and then graphically display the network data for the user to determine which clusters contained attacks. Cannady develops a network-based neural network detection system in which packet-level network data is retrieved from a database and then classified according to 9 features extracted from packet [13]. Our method is different from [13] in that we use an Elman network for anomaly detection in order to provide the detector with an internal memory which has the same function as the time window used in perceptrons. Moreover, our method allows to generalize input further than Cannady's method and to identify both intra-packet and inter-packet sequence anomalies.

4 Anomaly Detection Based on Elman Network

4.1 Clustering the Payload

It has been found that the computational complexity of unsupervised learning algorithms scales up steeply with the size of the considered data. Some researches

using ANNs for intrusion detection solve this problem by retaining the information only in the packet header and discarding the payload but lead to an unacceptable information loss. In order to overcome this drawback, we adopt the approach of clustering the payload, which allows to retain part of the information related to the packet payload. Owing to most network traffic belonging to a small number of regular used services and protocols, an unsupervised clustering algorithm is capable to classify payloads to a relatively small number of classes. This classification of payloads can be added as one of the features analyzed by the Elman network to the information decoded from the packet header.

An interesting characteristic of many clustering algorithms is the built-in capability to group the objects, classically defined as follows:

Def. 1. An algorithm by which objects are grouped in classes, so that intra-class similarity is maximized and inter-class similarity is minimized [14].

Results of the study performed to assess the performance of some clustering algorithms on the KDD 1999 Cup intrusion detection dataset show that for a given attack category (DoS and Probes), K-means clustering algorithm demonstrates superior detection performance compared to others [15]. Using the K-means clustering algorithm to classify the payload of the packets, different clusters are specified and generated for each output class. All output classes are numbered and the serial number of the class represents one feature called *clus_payload*.

4.2 Building the Elman Network

Previous traffic studies of TCP/IP have examined the statistics of the aggregated packet arrival process on local area networks [16], at border routers, and inside a wide-area backbone. These studies have shown that packet interarrival times are not Poisson, but rather follow a packet-train model. A packet train consists of packets going in both directions. These studies also indicate that successive packets have a tendency to belong to the same train and the time

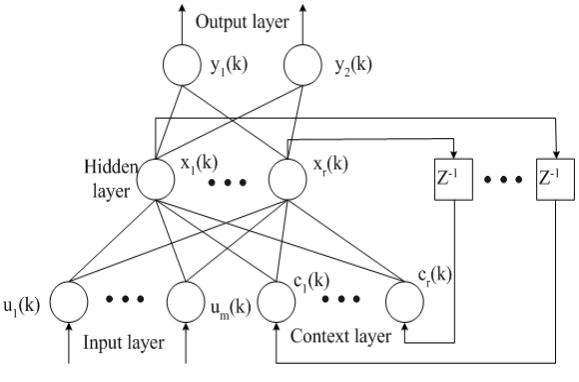


Fig. 1. Structure of the Elman network

intervals between successive packets form a time series. Among the various architectures of ANNs, the Elman network, which was first proposed for learning and representing temporal structure in linguistics, is suitable for the application to time-series prediction. The basic structure of the Elman network is illustrated in Fig.1. It comprises four layers, namely, input layer, hidden layer, output layer, and context layer. There are adjustable weights connecting each two neighboring layers. Before training, the number of nodes of each layers should be decided. For input node, the number is determined by the feature vector of the packet, which is described in detail in data acquisition and preprocessing section; for output node, the number is depended on the representation of the detecting result; for hidden and context layer, the number of nodes is an adjustable parameter, and the optimal number is acquired through simulations.

The inputs of network are $u(k) \in R^m, c(k) \in R^r, x(k) \in R^r$, then the outputs in each layer can be given by

$$x_j(k) = f(\sum_{i=1}^{m} v_{ij}(k) * u_i(k) + \sum_{i=1}^{r} w_{ij}(k) * c_i(k)) \tag{1}$$

$$y_j(k) = g(\sum_{i=1}^{r} h_{ij}(k) * x_i(k)) \tag{2}$$

$$c_i(k) = x_i(k-1) \tag{3}$$

where,

v_{ij} : The weight connects input node to hidden node.
w_{ij} : The weight connects context node to hidden node.
h_{ij} : The weight connects hidden node to output node.
$f(.)$: The nonlinear output function of hidden layer.
$g(.)$: The nonlinear output function of output layer.

For each unit in the hidden layer an additional unit called *context* unit is added and there are recurrent connections from the hidden units back to the *context* units. The weights of the recurrent connections are fixed and the forward weights get trained in the same way as feed-forward networks. After calculating the outputs of the hidden units, the current values get copied into the corresponding *context* units through a *unit delay*. These values are used in the next time step. This recurrence gives the network dynamical properties, which make it possible for the network to have internal memory, which has the same function as the time window used in perceptrons, i.e. to give the neural network a perception of the past. By keeping a trace of past events into its internal memory, the network have long term memory, coded in the connections, which stores the law of the network's traffic, and short term memory, coded in the activations of the neurons, which stores current information about the TCP/IP packet. Therefore, an Elman network based anomaly detector has the conspicuous capability to identify not only intra-packet anomalies but also inter-packet sequence anomalies.

4.3 Training the Elman Network

Considering a number of ANNs topologies and training algorithms available, the choice of an appropriate pair (*architecture, training*) is intimately dependent on the purpose and can be decisive for its success. Several training algorithms have been proposed to adjust the weight values in ANNs. Examples of these methods are the *Dynamic Back-Propagation* algorithm (DBP), the *Real Time Recurrent Learning* algorithm (RTRL), and the *Back-Propagation Through Time* (BPTT). In order to develop an efficient strategy for real-time anomaly detection, we focus on the training speed when make a choice among the various training algorithms. In [17], the Truncated-BPTT (T-BPTT) algorithm showed the advantage of fast training, therefore, we propose the combination of an Elman network with a Truncated-BPTT (T-BPTT) algorithm to develop an efficiently working real-time anomaly detector.

5 Experiments and Results

5.1 Data Acquisition and Preprocessing

As noted before, we need a large amount of experimental data, in particular dumps of common network traffic, to feed into the Elman network, in the format described by the "libpcap" libraries. There is one source of dataset which makes the full payload available for inspection: the dataset created by the Lincoln Laboratory at MIT, also known as "DARPA IDS Evaluation dataset". When packet data is summarized into the connection records, each record contains a set of features for general network traffic analysis. For each TCP/IP connection, 41 various quantitative and qualitative features are extracted. The importance of correctly choosing features for machine learning problem has been widely discussed in literature, and our approach is not an exception.

Using a set of benchmark data from the KDD (Knowledge Discovery and Data Mining) competition designed by DARPA, previous studies demonstrated that efficient classifiers could be built using ANNs that used only the (13 of 41) most significant features of the data and delivered only-slightly-lower detection accuracy in the binary attack/normal classification [18]. Choosing more features beyond the (13 of 41) most significant ones would increase computational requirements that are critical for real-time processing and affects the convergence speed and stability of the Elman network. Therefore, the feature vector for an Elman network includes the clustering result of the payload and the (13 of 41) most significant features. Table 1 below shows 14 different features used in an Elman network.

The Elman network based anomaly detection model is trained on the DARPA dataset using week 1 (5 days, attack free) and week 3 (7 days, attack free), then evaluated on the dataset using weeks 4 and 5. Beyond data acquisition, three stages of preprocessing have implemented to make input data suitable for an Elman network. In the first stage, 14 different features of each packet are extracted. Among those features, the acquisition of *clus_payload* through K-means cluster algorithm is independent of other features. In order to attain the

Table 1. List of features (Type C is character, while N is numeric)

#	Feature name	Description	Type
1	*dur*	Length of the connection	N
2	*pro*	Type of the protocol, e.g. tcp, udp, etc	C
3	*service*	Destination port	C
4	*src_bytes*	# of data bytes from source to destination	N
5	*dst_bytes*	# of data bytes from destination to source	N
6	*urgent*	# of urgent packets	N
7	*count*	# of connections to the same host as the current one during past 2 seconds	N
8	*srv_count*	# of connections to the same service as the current one during past 2 seconds	N
9	*dst_host_count*	# of connections to the same host as the current one during past 100 connections	N
10	*dst_host_srv_count*	# of connections to the same service as the current one during past 100 connections	N
11	*same_srv_rate*	% of connections to the same service	N
12	*dst_host_same_srv_rate*	% of connections to the same host during past 100 connections	N
13	*dst_host_same_srv_port_rate*	% of connections to the same service during past 100 connections	N
14	*clus_payload*	the clustering result of the payload	N

optimal classification, the clustering simulations have run over 2, 8, 16, 32, 64, 128, 256, 512, and 1024 clusters for the training data. Clusters are trained until the average squared error difference between two epochs is less than 1%. Manual inspection reveals that the simulation having 256 clusters is more respondent to our expectations than others. In the second stage, two features, namely *pro* and *service*, are converted into a numeric representation, for the other twelve components are already in a numerical format that could be input into an Elman Network. For representing a non-attack event, two additional elements, namely *1.0* and *0.0*, are added to each record. During training these elements are used as the target output of the Elman network for each record. The third phase of preprocessing involves the normalization of the feature vector which consists of 14 different numeric features. Due to the large variations of these numeric representations, each vector has been initially normalized so that its components are in the range of [-1, 1]. Through the normalization, the feature vector is more suitable for the Elman network applications.

The standard normalization given by

$$nv[i] = \frac{v[i]}{\sqrt{\sum_{k=1}^{n} v[k]^2}} \tag{4}$$

is used for this purpose. Here $nv[i]$ is the normalized value of feature i, $v[i]$ is the feature value of i, and n is the number of features in a vector.

5.2 Experimental Results

In this work, an anomaly detection model based on the Elman network is implemented and applied for anomaly detection against the DARPA IDS Evaluation dataset. When it comes to the performance of an intrusion detection system, it is necessary to take into account both the detection ability and the false positive rate. To fully demonstrate the advantages of the Elman network over the *Feed-forward Neural Network* (FNN), we implement a FNN as well and apply it for anomaly detection against the same dataset as the Elman network. Meanwhile, they are trained in the same method with 13-input nodes (only the features extracted from the packet header) and 14-input nodes (including *clus_payload*) to evaluate the significance of the payload. A measure of the overall effectiveness of a given intrusion detection system can be provided by the ROC curve. A ROC curve is a parametric curve that is generated by varying the threshold of the intrusive measure which is a tunable parameter. The curve is a plot of the likelihood that an intrusion is detected against the likelihood that a non-intrusion is misclassified (i.e., a *false positive*). We use ROC curves to compare intrusion detection ability of the anomaly detection system to false positives. The ROC curve allows the end user of an intrusion detection system to assess the trade-off between detection ability and false alarm rate in order to properly tune the system for acceptable tolerances.

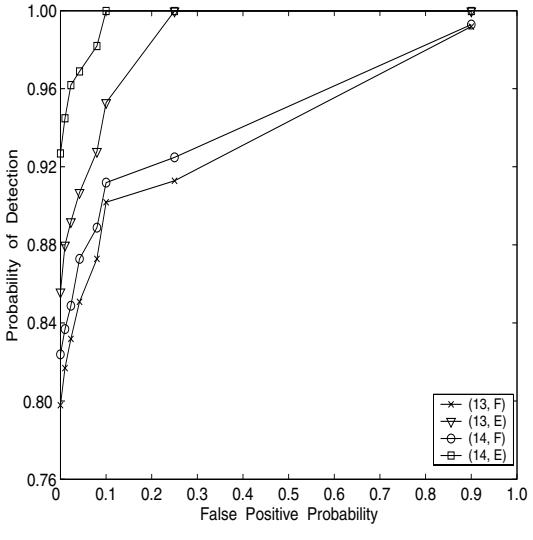


Fig. 2. Performance of the Elman network and the FNN expressed as ROC curves against the DARPA evaluation data.The horizontal axis represents the percentage of false positives while the vertical axis represents the percentage of correct detections for different operating thresholds of the technique. 13 and 14 represent the number of input nodes, F means a FNN, while E means an Elman network. The Elman network with 14 input nodes performs the best overall

Before training, the number of hidden nodes should be decided. However, the optimal numbers for hidden nodes of the Elman network and the FNN are unknown before training. Therefore, the Elman network and the FNN are trained with 15, 25, 30, 35, 40, 45, 50, 55, and 60 hidden nodes to avoid poor performance. Each neural network is trained until the total error made during an epoch stops decreasing, or 3,000 epochs have been reached. During training period, $4 * 9$ networks are trained for anomaly detection, and the network that classified data most accurately is selected. The detector is evaluated on the DARPA dataset using week 4 and 5, which contains 201 instances of 58 different attacks. Different networks produce different ROC curves. The performance of the Elman network in comparison to the FNN is shown in Fig.2.

As illustrated in Fig. 2, with the number of input nodes of 13 and 14, the FNN is able to detect 79.8%, 82.4% of all intrusions with no false positives, while the Elman networks are able to detect 85.6%, 92.7% — a very significant improvement over the FNN. Furthermore, the Elman network with 14 input nodes is able to detect as much as 96.2% of all intrusions with a false positive rate of 2.3%. To achieve detection ability better than 96.2%, one need to accept an increase in false positives. The ROC curve for the Elman network with 14-input nodes is the left-most curve that quickly reaches 100% detection. The results indicate that anomaly detection model based on the Elman network performs better than detection model based on the FNNs.

6 Conclusions

Research and development of intrusion detection systems have been ongoing since the early 1980's. Current intrusion detection systems lack the ability to generalize from previously observed attacks to detect even slight variations of known attacks. The motivations of using an Elman network for anomaly detection were presented. Most neural network architectures must be retrained if the system is capable of improving its analysis in response to changes in the input patterns, (e.g., "new" events are recognized with a consistent probability of being an attack until the network is retrained to improve the recognition of these events). Adaptive resonance theory offers an increased level of adaptability for neural networks, and this approach is being investigated for possible use in an intrusion detection system. The preliminary results from our experiment present a positive indication of the potential offered by the Elman network, and our future work will involve the refinement of this approach and the development of a full-scale demonstration system.

References

1. W. Lee, S. Stolfo, and P. K. Chan, "Learning patterns from unix process execution traces for intrusion detection", *Proc. of AAAI'97 Workshop on AI Methods in Fraud and Risk Management*, 1997.

2. G. Vigna and R. A. Kemmerer, "Netstat: A network-based intrusion detection approach", *Proc. of the 1998 Annual Computer Security Applications Conference (ACSAC'98)*, Los Alamitos, CA, IEEE Computer Society, Dec. 1998, pp.25-34.
3. P. A. Porras and P. G. Neumann, "Emerald: Event monitoring enabling responses to anomalous live disturbances", *Proc. of the 20th National Information Systems Security Conference*, Oct. 1997, pp.353-365.
4. A. K. Ghosh, J. Wanken, and F. Charron, "Detecting anomalous and unknown intrusions against programs", *Proc. of the 1998 Annual Computer Security Applications Conference (ACSAC'98)*, Dec. 1998.
5. K. Labib and R. Vemuri, "TNSOM: A real-time network-based intrusion detection system using self-organizing maps", *Technical report*, Dept. of Applied Science, University of California, Davis, 2002.
6. T. F. Lunt, "IDES: an intelligent system for detecting intruders", *Proc. of the Symposium: Computer Security, Threat and Countermeasures*, Rome, Italy, Nov. 1990.
7. Z. Pan and S. Chen, "Hybrid Neural Network and C4.5 for Misuse Detection", *Proc. of the 2nd International Conference on Machine Learning and Cybernetics*, Xi'an, Nov. 2003.
8. H. Debar and B. Dorizzi, "An Application of a Recurrent Network to an Intrusion Detection System", *Proc. of DARPA Information Survivability Conference and Exposition*, 2000, Vol.2, pp.12-26.
9. J. Ryan, M. Lin, and R. Miikkulainen, "Intrusion Detection with Neural Networks", *Proc. of the 1997 Conference on Advances in neural information processing systems*, Menlo Park, CA, pp.72-79.
10. S. C. Lee and D. V. Heinbuch, "Training a Neural-Network Based Intrusion Detector to Recognize Novel Attacks", *Information Assurance and Security*, 2000, pp.40-46.
11. B. Rhodes, J. Mahaffey, and J. Cannady, "Multiple Self-Organizing Maps for Intrusion Detection", *Proc. of the 2000 National Information Systems Security Conference*, Baltimore, 2000.
12. L. Girardin and D. Brodbeck, "A Visual Approach for Monitoring Logs", *Proc. of the 12th System Administration Conference (LISA'98)*, Boston, MA, Dec. 1998, pp.299-308.
13. J. Cannady, "Artificial Neural Networks for Misuse Detection", *Proc. of the 1998 National Information Systems Security Conference (NISSC'98)*, Arlington, VA, Oct. 1998, pp.443-456.
14. J. Han and M. Kamber, *Data Mining: concepts and techniques*, Morgan-Kauffman, 2000.
15. M. R. Sabhnani and G. Serpen, "Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset within Misuse Detection Context", *Proc. of International Conference on Machine Learning: Models, Technologies and Applications*, Las Vegas, 2003, pp.209-215.
16. R. Gusella, "A Measurement Study of Diskless Workstation Traffic on an Ethernet", *IEEE Transactions on Communications*, Sep. 1990.
17. P. Werbos, "Backpropagation through time, what it does and how do it", *Proc. of the IEEE*, Vol.78, No.10, 1990, pp.1550-1560.
18. S. Mukkamala, G. Janoski, and A. Sung, "Intrusion detection: using neural networks and support vector machines", *Proc. of the 2002 International Joint Conference on Neural Networks (IJCNN'02)*, 2002, pp.770-789.

On Mitigating Network Partitioning in Peer-to-Peer Massively Multiplayer Games

Yuan He[1,2], Yi Zhang[1,2], and Jiang Guo[3]

[1] Laboratory for Internet Software Technologies, Institute of Software,
The Chinese Academy of Sciences, Beijing 100080, China
[2] Graduate School of the Chinese Academy of Sciences, Beijing 100039, China
{heyuan, zhangyi}@itechs.iscas.ac.cn
[3] Department of Electrical and Computer Engineering,
University of Toronto, Ontario, Canada
jguo@eecg.toronto.edu

Abstract. Recently, peer-to-peer infrastructure has been proposed to support massively multiplayer games in the literature. However, when underlying network partitions due to network outages, the game world will partition into several parallel game worlds and it is difficult and costly to merge them when the network partitions disappear. Existing approaches resort to a centralized server to arbitrate. Aiming at mitigating the effects brought by network partitions, we propose a fully distributed algorithm based on state-stack matching. Our theoretical analysis and numerical results show that our approach can resolve the merging issue at the least loss of game states with high probability.

1 Introduction

Massively multi-player games (MMGs) have a long history following a centralized infrastructure. Recently, researchers are proposed to use Peer-to-Peer overlays to support massively multi-player games (MMGs) [1]. In such a peer-to-peer gaming infrastructure, except that a central server is required to keep player account information, all other game states are stored in a distributed way all over all peers participating in the game.

In most MMGs, the player assumes the role of a character in a virtual world. A typical multiplayer game world is made up of immutable terrain, characters controlled by players (PCs), mutable objects such as food, tools, weapons, and non-player characters. Different game states have different access patterns and consistency requirements.

Existing P2P approach [1] proposes to store object states in a distributed way. Furthermore, copies of object states are replicated in order to increase availability with the presence of node failures or network outages. However, such a P2P approach leads to an undesired problem: in case of network outages, the underlying network partitions; the system can continue to allow shared states access, but with no communication between partitions, the original game world splits into two or more parallel worlds, likely with loss of consistency. Things go worse with potential paradoxes when the network condition recovers and partitions are about to merge.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 481 – 490, 2005.

As a rule in a peer-to-peer massively multiplayer game, consistency is most important: we can not imagine that a game full of paradoxes has the possibility to attract players. We start from existing P2P approach [1] and focus our attention on mitigating the adverse effect caused by network partitioning or the consistency issue. Simply put, in this paper, we explore two questions:

1) Can we keep the consistency of split game worlds?

2) When merging partitioned game worlds, can we solve conflicts with least game states loss?

The remainder of this paper is organized as follows. In Section 2 we briefly introduce the existing approach using P2P overlay support MMGs. In Section 3 we present our proposed merging algorithms based on state-stack matching and theoretical analysis. Section 4 shows experimental results using numerical method. Section 5 concludes this paper.

2 P2P Support for MMGs

Some methods have been worked out [1], [2], [3] and work well under the assumption of low failure frequency and graceful network behavior. However, performance becomes poor in the face of network partitions as introduced in Section 1. Without effective communication or coordination during the course, paradoxes probably can't be avoided.

In order to ensure availability and consistency in the face of network partitions and merge, we propose a distributed strategy of dynamic state management. It is based on the coordinator-based mechanism proposed in [1], which will be briefly introduced as follows.

2.1 The Coordinator-Based Mechanism

Building the whole system on top of the classical Pastry peer-to-peer infrastructure, we can group players and objects by regions according to their game situations (Fig. 1), and distribute the regions onto different peers by mapping them to the Pastry key space. For example, players in a same game scene have closest relationship. These peers, together with all those objects in this scene, form their common region. Each region is assigned an ID, computed by hashing the region's textual name using a collision resistant hash function. [2], [4]

Each region or object is assigned a coordinator, to which all updates information are sent. The coordinator both resolves conflicting updates, and is a repository for the current value of the corresponding object. In a comparatively common design, the coordinator not only coordinates all shared objects in the region, but also serves as the root of the multicast tree, as well as the distribution server for the region map. [5] Although mapping all synchronization and update information to a single node simplifies the system design, it might incur a high network load on this coordinator if the game complexity increases. However, the load can be distributed by assigning a different ID for each type of object in the region, thus mapping them onto different peers.

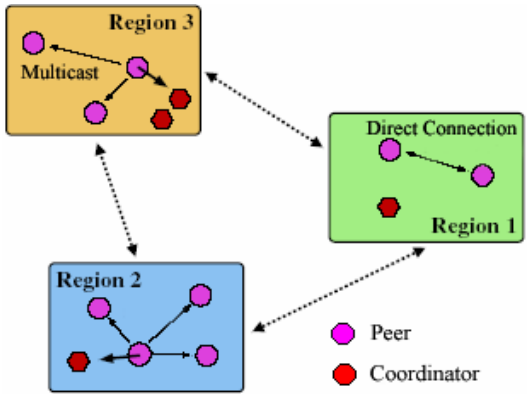


Fig. 1. Regions, peers, and coordinators

Successful updates are multicast in periodic message forms in the scope of current region to keep each player's local copy fresh. To keep consistency among players, timely message delivery is necessary, which will be discussed in part B. A primary-backup mechanism [1] has been applied to tolerate fail-stop failures of the network. Node Failures are detected using regular game events, without any additional network traffic. The coordinator's state is replicated once a failure is detected. The algorithm tries to keep at least one replica up under all circumstances, to prevent losses. As our objective is not replication algorithm, but the strategy to deal with network partitions and merge, we just give a simple introduction of replication algorithm here, and don't make any more comments on this issue in the following sections.

3 Mitigating Network Partitioning

When the scenario of network partitions appears, the original game world is partitioned into several parallel worlds. Then we try to force each parallel game world to be independent normal world. When the partitions again merge, states possessed by different parts are combined in a smart way, potential paradoxes are removed and then valid game state still can be attained.

3.1 Independent Game Worlds Rebuilding on Network Partitions

In our distributed strategy, we embed in a periodic message mechanism to ensure consistency and availability. Messages between coordinators and common peers are sorted to three classes:

Message a: Request for current local state on peers, sent to all common peers by the coordinator.
Message b: Combined updated state of whole region, produced by the coordinator after it gets each peer's latest local state (message c), and then multicast to all peers.
Message c: Peer's current Local state, sent by a common peer to the coordinator.

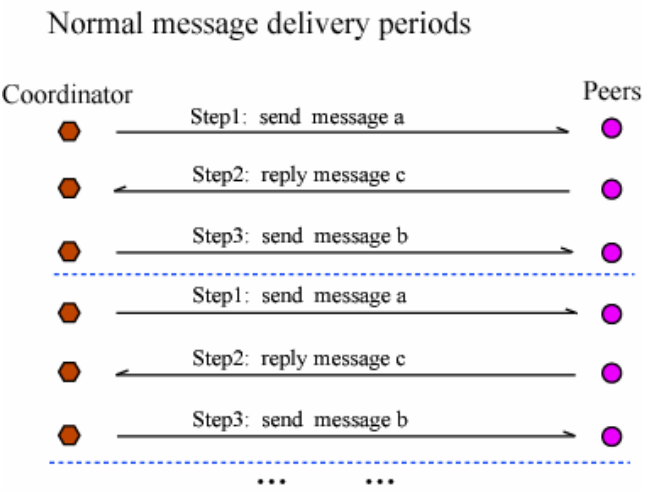


Fig. 2. Periodic messages delivery

Generally, the coordinator first multicasts message a to all peers in the region. Receiving message a, each peer send its current local state copy in format of message c to the coordinator as response. After getting all messages c, the coordinator removes the conflicts and produces an updated region state. Then the coordinator multicasts message b to all peers with the content of region state update. This is called a normal message delivery period, shown in Fig. 2. We can set the coordinator's request frequency at a proper fixed value according to the network conditions and average node abilities so as to meet the demands of a peer-to-peer game.

Now we continue to discuss the system's behavior when network partitions happen. Without loss of generality, we just discuss the scenario with only a single coordinator in each region and assume that an original region is just partitioned into two parts. As shown in Fig. 3, region O is partitioned into A and B. Peers in region A and B lose the communication with peers in the other region.

We suppose node P is the coordinator. Immediately after partitions, peers in region B can not receive message a from P. Therefore, P can not receive response from peers in region B either. At this moment, P is able to ascertain happening of partitions. It can also make sure which peers are still connective, and which peers are not.

Then P should adjust the state possessed by it: First, it marks those connections and information related to the peers without response invalid, which means those peers are not in the same region as P now. Second, P generates a new state update only with support of the messages from connective peers. Because A and B are new regions, they may be assigned a new region ID computed by hashing. As a result, A and B will both have new coordinators of their own (P_a, P_b). Suppose P remain in A, then P should send its newly produced region state update to P_a and leave the job to P_a to multicasts message a inside region A when the next message delivery period starts.

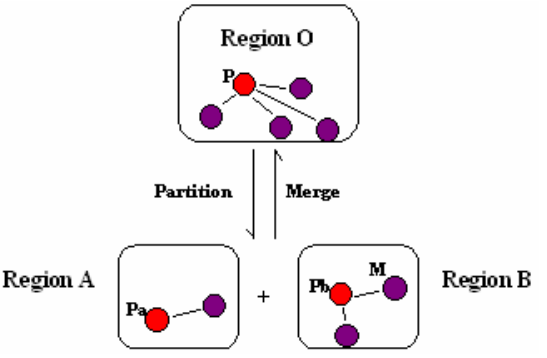


Fig. 3. Region partitions

On the aspect of other peers, peers in region B (such as M) can not receive any messages from their former coordinator P. After this state keeps over the length of a message delivery period, each peer is able to ascertain happening of partitions. Connections between these peers and P_b will be set up. P_b will take the job as the new coordinator of region B and immediately starts a new message delivery period.

In this way, region A and B with new coordinators are properly rebuilt after network partitions. Because invalid peer states and information have been removed during rebuilding, the new coordinators (P_a, P_b) only keep the necessary state of their own region and no conflicts exist now. Independent game worlds come into being and the game successfully continues.

3.2 Regions Merge Mechanism

Through the mechanism in part B, parallel game worlds are rebuilt after network partitions and peers in each region behave without conflicts. Yet, these regions are about a same unique game scene in fact. Each takes itself as the only representative of this scene in the whole game world. Accesses and changes to an object perhaps take place concurrently, so their independent behavior will probably cause paradoxes when they try to merge again, which threaten consistency.

In [1], a central server blessed mechanism is adopted to keep consistency. But it assumes that partitions still keep connections with a central account server. If network partitions thoroughly, including outages to the central server, coordination among regions can't be ensured without the central server and paradoxes after merge can't be avoided yet.

In our distributed strategy, the central server is no longer needed. Coordinators are strengthened to keep consistency and reduce the conflict probability, designed as follows: Each coordinator not only possesses the latest region state, but also keeps a state-stack, which stores a certain number of recent region states, from stack bottom to its top in time order. (Fig. 4) Once the coordinator produces a new region state update, it pushes the state into the state-stack while it multicasts the state to other peers. Considering the game's complexity, the low conflict probability and the average peer capability, we may set the state-stack to a reasonable size N.

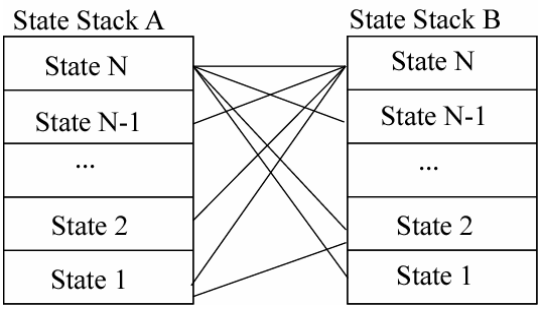


Fig. 4. State-stacks and matches of states

On the inverted condition of Fig. 3, let's focus on the scenario when regions merge. Without loss of generality, we assume each region's coordinator has already kept full state-stack storing slots of states. When merge attempt starts, two coordinators of region and compare their stored states in pairs. Generally each state is compared to the states in the other stack. A pair of states without any conflicts is called good match. At most N2 times of compares will be executed and we'll eventually get set of good matches. One match will be selected and combined into reasonable state of the merged region O. This match of states without any conflicts between each other ensures that there isn't any paradox in region O.

Later states match before combination equals to less loss of state information after merge. For example, we have two matches, (9, 8), (6, 9). Match (9, 8) will be selected, because the latest state sequence number is (10, 10), and match (9, 8) loses only periods of states in all, comparatively less than periods lost by match (6, 9). Unless the good matches-set is empty, at least 1 match will be found. Successful combined state is then sent to the new assigned coordinator of region O. Subsequently, normal message delivery period in region will start. Merging region A, to region succeeds then.

Even if the good matches set is empty at first, things keep unchanged for while. When next message delivery period begins, new state will be pushed into each stack. Then we go on to compare the new state with states stored in stack before. If any good matches are found, regions can be merged immediately. Otherwise, we may still keep on this circulatory process to seek good matches. Through the relevant experiments in Section 4, we can see that fairly high probability of successful merges can be obtained. And at the same time, the most states information can be reserved.

3.3 Theoretical Analysis of Network Partitions Process

3.3.1 Time Complexity

A complete process of resolving network partitions is composed of two segments:

$T = T_1 + T_2$

T: whole process time;

T_1: time for coordinator to ascertain network partitions;

T_2: a message delivery period for the coordinator and peers to set up connections and rebuild new region state.

We assume the time of a normal message delivery period is T_0.

T_2 will never start until the coordinator ascertains the partitions in T_1. In our distributed strategy, the only rule for the coordinator to judge partitions is that it has not received any response message from some certain peers after the time for all peers to send back messages c passes, as is indicated by Fig. 2. Then T_1 equals to the interval between the network partitions and the receipt of messages c in one period. Considering the unpredictable network faults might appear any time in the period at a completely same possibility, we can draw a conclusion as follows:

$T_1 = 0.5 * T_0$ averagely.

On the other hand, the length of T_2 obviously equals to T_0.

$T= T_1 + T_2 = 0.5 * T_0 + T_0 = 1.5* T_0$ (1)

T_0 is assigned a proper fixed value according to the network conditions and average node abilities to meet the demands of a P2P game. From (1) we can see that, once the coordinator's request frequency is set, the complete process time of resolving network partitions is fixed, no matter how many peers exist in the network or how they partition. Usually T_0 is only a very short interval (at most 1 second level), so our strategy gives an effective solution against network partitions in P2P games.

3.3.2 Overhead

During the process of partitions, messages are sent between peers and coordinators for game worlds rebuilding. To simplify the following computation, let's assume messages a, b, and c have same length and M messages should be delivered for resolving problems during an X-peers region partitions. Now we compute proportion M/X, the average individual overhead.

Without loss of generality, a region is also assumed to be partitioned into two parts with X_1 and X_2 peers each. $X = X_1+ X_2$. From discussion in the previous parts, we've seen that all messages to be delivered during network partitions are the messages sent in a whole normal message delivery period. One more special message should be counted in, which is the state-copy sent from the former coordinator to the new coordinator in one region (such as P and Pa in Fig.3). Then,

$M = 3*(X_1-1) + 3*(X_2-1) + 1= 3(X_1+ X_2) -5 = 3X-5$

$M/X = 3-5/X$ (if $X->\infty$, $M/X=3$)

In the real peer-to-peer networks, number of peers in a region is usually very large. We draw a conclusion that the average overhead equals to a light level of 3 messages delivery per peer during the whole process of partitions.

4 Simulations

We develop simulation program according to the region merge mechanism proposed above. In practical P2P game, it has big states set with large number of different region states. However, not all states conflict with each other. We can group them by classes under the rule that conflicts don't appear inside class and exist between any two classes inevitably. If two states are in the same class, they will form good match and then can be combined. As for two regions which need to merge, if at least one good match exists in the state-stacks of their respective coordinators, merge will

certainly succeed. In order to simulate the stochastic distribution of the game states, states in all slots are generated stochastically. 1000 times of merge tests will be performed and the probability of successful merges can be eventually obtained.

Related parameters, *N*: stack size, *K*: number of state classes, *P*: probability of successful merges.

4.1 Experiment 1

We set *K*=1000, use increasing values for *N*, and check *P*'s variety, as shown in Table 1 and Fig. 5.

Table 1 and Fig. 5 come up with the following conclusion: For certain P2P game with fixed number of states, larger state-stacks on coordinators lead to higher probability of successful merges. In our design, *K*=1000, *N*=50 effectively achieve high probability of successful merges over 91%.

Table 1. Probability as *K*=1000

N	*K*	*P*
5	1000	0.025
10	1000	0.095
15	1000	0.203
20	1000	0.328
25	1000	0.465
30	1000	0.595
40	1000	0.797
50	1000	0.919

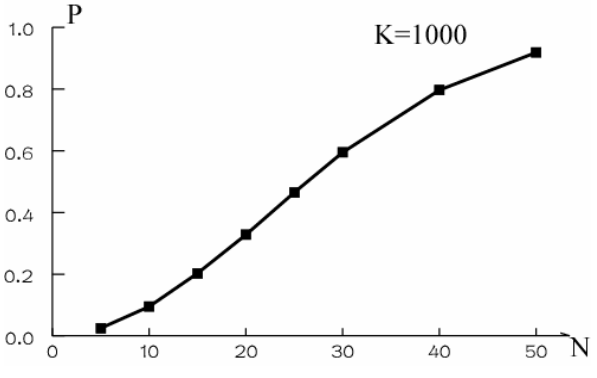


Fig. 5. Probability variety as *K*=1000

4.2 Experiment 2

What size of state-stack is necessary for coordinator if we want high success probability? Satisfactory to everybody, farther results have been found through times of experiments.

Table 2 shows only part of our experimental results. From these results, we can see that when N^2/K equals to a constant, P is relatively localized around a fixed value too. The detailed theoretic analysis is too complicated and we may just simply consider that N^2/K entirely determines the value of P.

Based on the data listed, rules may be formed as follows: If we want probability P no less than about 90%, N^2/K should equal to at least 1. That's to say, $N = \sqrt{K}$; If we want higher probability no less than about 98%, N^2/K should equal to at least 4. That's to say, $N = 2\sqrt{K}$.

Table 2. Probability as K=100, 400, 900, 1600

N	K	N^2/K	P
10	100	1	0.632
15	100	2.25	0.895
20	100	4	0.981
20	400	1	0.633
30	400	2.25	0.895
40	400	4	0.982
30	900	1	0.633
45	900	2.25	0.896
60	900	4	0.982
40	1600	1	0.633
60	1600	2.25	0.896
80	1600	4	0.983

As N is the size of a coordinator's state-stack. From the discussions above, we can draw a conclusion: For a certain degree of success probability, space complexity of the region merge mechanism is $O(\sqrt{K})$.

On the other hand, each slot in the state-stacks usually describes its corresponding game-state with a marked bit-map structure, the variation of each bit representing one entity in the game scenes. Accordingly the memory cost of each slot is about 1KB on normal occasions and not more than 10KB at most.

Therefore, for the duty of keep essential state-stack in the region merge mechanism, the memory cost of each coordinator is: C= 10KB*N = $20\sqrt{K}$ KB at most. Even if the number of state classes K increases fairly large, e.g. K=1*10^6, C still keeps at a relatively economic level of 20 MB, which may be easily provided by all the participant PCs.

5 Conclusion

Network faults such as network partitions and merge have deep negative impact on availability and consistency. On top of Pastry infrastructure and the coordinator-based mechanism, our distributed strategy makes outstanding contributions on resolving network partitions and merge.

Embedded in periodic message mechanism, independent parallel game worlds are rebuilt after network partitions in an effective way without any additional spending. By introducing the state-stack structure to the coordinator-based mechanism, the game system in our design gets the ability in the face of regions merge. Under the rule of choosing latest state matches for combination, the most game-states can be reserved after merge. Proved by the analysis and simulations results in the end, our strategy runs with good time and space efficiency.

Comparing with other P2P game systems such as the central server bless mechanism introduced in [1], systems with our distributed strategy has the ability in the face of thorough network partitions and merge. However, more work should be done for better states replication algorithms. Security in the particular process of network partitions and merge is another field we are going to research in the future.

Acknowledgment

This work is supported by the National Natural Science Foundation of China under Grant No.60373053; One-Hundred-People Program of Chinese Academy of Sciences under Grant No.BCK35873.

References

1. B. Knutsson, H. Lu, W. Xu, and B. Hopkins: Peer-to-Peer Support for Massively Multiplayer Games. In IEEE Infocom (March, 2004).
2. A. Rowstron and P. Druschel: Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. In Proc. of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001) (November, 2001).
3. E. Cronin, B. Filstrup, and A. Kurc: A Distributed Multiplayer Game Server System. In University of Michigan technical report (May 2001).
4. E. Sit and R. Morris: Security Considerations for Peer-to-Peer Distributed Hash Tables. In First International Workshop on Peer-to-Peer Systems (IPTPS '02) (March, 2002).
5. H. Lu, B. Knutsson, M. Delap, J. Fiore, and B. Wu: The Design of Synchronization Mechanisms for Peer-to-Peer Massively Multiplayer Games. In Penn CIS Tech Report (2004).

P2P-Based Software Engineering Management

Lina Zhao[1], Yin Zhang, Sanyuan Zhang, and Xiuzi Ye

College of Computer Science /State Key Lab of CAD&CG,
Zhejiang University, 310027, Hangzhou, P.R. China
[1]renazhao78@hotmail.com, [1]zhaoln@sindyware.com

Abstract. With the rapid development of the network technologies, software development is becoming more and more complicated. Traditional software engineering management methods based on Client/Server structure have not been very competent for large-scale software development. This paper proposes a software engineering management method based on P2P structure. By exploring the characteristics of P2P, our system takes "Peer Group" as the unit to build up services including source code management, task assignment and software visualization. Our framework overcomes the server's "bottleneck" existed in the C/S structure. Although it still has a server in the network, the server only provides indexing function and does not store the document. In this way, the loads of the server are transferred to the network boundary. Our method takes full advantages of computation resources, and can increase the system's performance. We will also show in the paper the performance increases by our method in software development for practical applications.

1 Introduction

Nowadays, network is becoming larger and larger in scale. Computing mode has already developed gradually from the traditional host/terminal, customer/server mode to multi-layer client/server and browser/server mode. However, all of these techniques have to rely on the server. With the development of the hardware technology, the computing power each client can carry is increasing dramatically. Although the computing ability of each client is less or even much less than a high performance server, the clustering of many clients can provide considerably great computing ability and in some cases may even exceed any high performance server. The P2P [1] [2] (i.e., peer to peer) mode as a computing mode developed in the early 1990's has found its wide use recently. The ability in P2P mode in handling largely distributed tasks satisfies the need of network development, and provides us with a way of thinking and solving problems.

The procedure of software development is actually the process of perfecting all kinds of documents in computers. Documents and source codes are all saved in form of files. Current software engineering management systems all have servers to complete the necessary tasks [8]. However, for large-scale software development, the difficulties to manage huge amount of files increase considerably. Currently the tools used for source code management, such as Win CVS, VSS etc, are all use the server

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 491 – 500, 2005.

and follow traditional management methods. With the gradually expansion of software in scales, these software tools will not be suitable for complicated software development. In addition, developers may be behind different firewalls, and may use VPN method to find out the center server. This is inconvenient for developers and system safety. Therefore, we propose a new solution to manage the entire process of software development by using the P2P technique. Peers belong to a certain group that provides the services. For a specific document, the creation of each new version will just send an abstract description to the catalog server. The actual file will not need to send to the server. At the same time, in order to let developers visualize the software procedure and understand the progress in real time, we implement part of the visualization technologies developed in [9]-[14] in our system. We also introduce in the paper the concept of "role" and "interested document" to help peers to provide other peers the services with the local documents.

2 Overview of Our Approach

In the process of software development, requirement changes frequently, since it is too hard for the users to develop all the requirements in the very beginning. In most cases, developers spend much of their time and energy in coding in hectic, and not many people have the time to modify the coding related system models and charts created by tools used before. This may lead to inconsistency between the previously created documents, charts and the codes developed at the time, and can bring lots of trouble in re-engineering and system maintenance. Of course, having a good development habit may resolve this consistency problem. However, it is impossible to demand everybody to have high development standard [8]. This requires that software development environments and management tools must be easy enough for the developers to use. Moreover, C/S based system structure could bring the server with huge amount of loads, and hence the whole project at high risk. For example, a breakdown of the server can have devastated effects on the whole project development progress, and can lead to uncountable losses.

Therefore, in this paper we propose a P2P based technique to manage the entire software engineering. In our approach, each peer will only run the services belonging to its own group, and perform the services according to the machine loads and respond ability. From the source code management perspective, the creation of every new version of the document will lead to the insertion of a shared item in the local service data by the local peer who chooses another peer in the same group to upload the file without sending the document out to the server. Each group is responsible for the management of its corresponding subsystem, and the whole system is managed by the "catalog server".

The proposed software engineering management system will do the majority of the work in the software engineering process. It not only makes the development management convenient, but enables each peer to make the full use of its computational power. Thus the high performance of the whole system will make the 3D visualization of the software possible. The topology of the system is shown in Fig. 1. The whole system only needs one "catalog server" which can be instead of by "Super Peer" who holds favorable computation capability. Its main task is to complete the

following two missions. One is to act as a group information depository such as group configuration and services. When a group is set up, a copy of the group information will be sent out to the server. Another mission is to perform the document control to the indexing function. The “catalog server” does not need to store the physical document, but to maintain some "abstract" information of the document. The peers are distributed equally in the network, and they belong to the different logic collections because they are in different groups. The peers located in the same group will provide the same services to the outside.

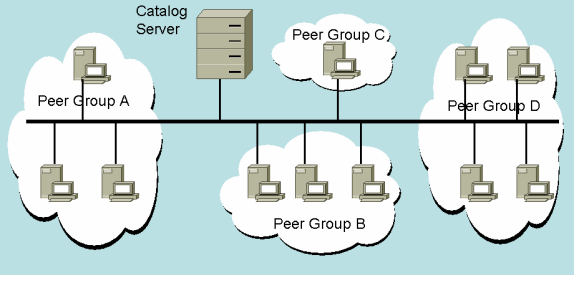


Fig. 1. The topology of the entire system

3 P2P Based Multi-layer Management Model

The system is based on the JXTA platform, and is implemented by Microsoft C#. The whole system can be divided into 3 layers, namely the basic platform layer, the service layer and the application layer.

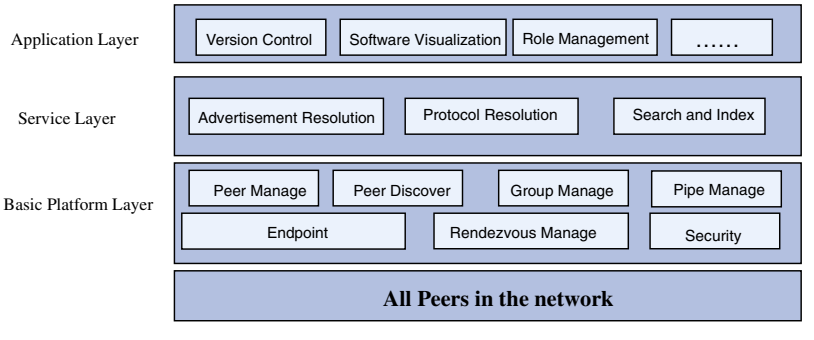


Fig. 2. The P2P architecture for software engineering management

3.1 The Basic Platform Layer

The traditional C/S structure has an obvious advantage, namely its data concentration which makes it easy to manage. However, when adopting the P2P technique in our

software management system, because the resources are relatively dispersed, the difficulty of management will increase. At any time a peer may join in the network or leave. This needs a protocol to make a peer's occurrence felt by others. For the sake of easy control, our system doesn't provide peer registration on-line. Project managers assign the peer with an ID and Group in advance. When the P2P system starts, every peer in the network will look for the group that it belongs to by the unique "Peer ID". After passing the legitimacy verification, the peer will stay in the P2P network. At the time the peer leaves, it sends a message to rendezvous to change its own state. The "catalog server" maintains a state table for all peers, according to which the peer management runs.

3.2 The Service Layer

The system communication is based on messages and advertisements. The service layer mainly analyzes various commands coming from the application layer by parsing messages and advertisements in form of XML, manipulates and invokes the particular service in the basic framework layer. This service layer implements the advertisement resolution, and protocol resolution using JXTA protocol. It acts as a bridge between the application layer and the basic framework layer.

3.3 The Application Layer

Based on the P2P network, by using services the service layer provides, the application layer implements the following functions: version control in software engineering code management, software visualization in development process, role management (which is unique to our software engineering management system), and so on. Owing to the needs of the communication and management, most applications of this layer carry out by web service.

4 Key Technologies

4.1 Management Technology Based on Peer Group

In a team of large-scale software development there are many development groups, and members in each group cooperate to complete parts of the functions. In a software development environment, members in the same group usually have similar interests and close relationships. Usually, the services and documents or data they are using are very much alike. Therefore, a peer group is created for the project subgroup. Unlike in JXTA where peer group is a virtual guard, in our system, we assign this virtual guard with an actual project group. This can make the group more concrete, and the service and data more pertinence. When a peer registers on the network, the group service will be a concrete entity that all members in the group can visit. The peer will accept legitimacy verification. Wherever the service and the data are, the group service will always be presented as a single control point with uniform behavior. Then legitimate member will obtain the group service code and data. An on-line advertisement will be broadcasted finally.

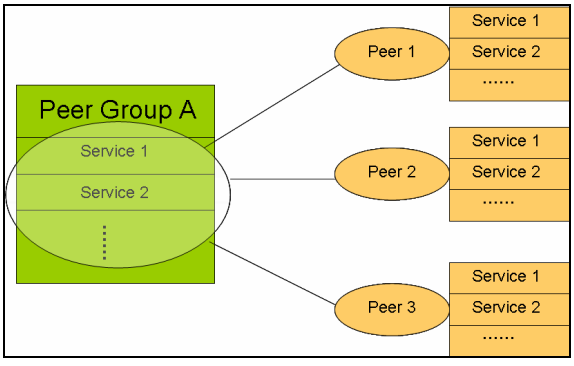


Fig. 3. The group services exist in each peer of the same group

It is shown in Fig. 3 that a peer group is in fact a combination point of service. Peers in a group are abstractly combined together, and form a logical union. Peers belong to the same group own the unique group ID, provide the identical service and manage the similar data. When a new peer joins in the network, it will send out an advertisement to look for its group. The peers in the same group will send a response after receiving the request. The original peer chooses the peer that has the quickest response to log in. Because the services each peer provides in the same group are identical, regardless of where the peer logs in, the services it uses are all the same.

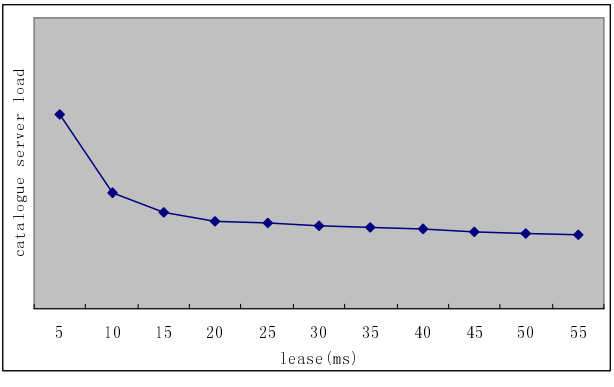


Fig. 4. A performance test

Because of the restrictions on the network bandwidth and the complexity of the P2P structure, it is possible to lose or delay the response. Sometimes there may not be any other peers in the same group on-line. Therefore if the original peer waits for the response indefinitely, performance decline may well be deduced. Aiming at this kind of circumstances, a “lease” technique is adopted. A “lease” is a group property, and all peers in the same group use the same “lease”. It specifies a time slice. Whenever a

peer sends out a request, the clock will start to count down. Once the time decreases to zero, this request will fail, and the request will be sent out again. However at this time point, the request is directly transferred to the “catalog server”. Since the registration information and the core services of each group are already copied to the "catalog server", when the request was sent out the first time, the "catalog server" is ready for the group information. So long as the request is obtained again, its contents will be quickly sent out. Therefore, the peer's request definitely will be responded in a certain time. However, if the "lease" is not set properly, low performance of the whole system may occur. This will increase the load of the "catalog server". A performance test is shown in Fig. 4. With the time increase of the “lease”, the burden of the server can ease gradually. However when the “lease” is longer than 25 ms, the load of the server will not have an obvious change.

4.2 P2P Technology in Source Code Management

Source code management is an important part of software engineering. Traditional mode needs every client to connect the server to acquire the specified version of a document or to hand over newly created documents. However with the expansion of the software in scale, this kind of mode is no longer suitable. We found that management methods based on the distribute structure can ease the load of the server. Because of the high similarity between documents inside the same group, P2P can be adopted.

In the traditional structure, the servers usually are high performance machines. For the sake of safety, managers backup the contents of the documents periodically. However, in the P2P structure, the circumstance is different. The load and contents are all assigned to the boundary of the network. This can have potential stability problems. For example, a failure of a certain peer or its temporarily left the network may result in a seek invalidation. To solve this kind of problems, to ensure the system safety and stability, and to reduce the difficulty of P2P system management, concepts of “interested document” and “role” are employed in the source code management. A few peers may act as the same type of role for a document to complete the functions the role takes. The management of document is achieved by roles. On the contrary roles are allotted to the peers based on its legal power to the document.

Definition1: Interested document
If a peer is the author of a document or the agent that the author authorizes the right to, the document will be the peer’s interested document.

Definition2: Role
If a document is an "interested document" of a certain peer, then according to different operation legal power that the peer has, some roles will be assigned to the peer. One peer can play several roles to an interested document, and in the same way one role can own several peers. Same role peers of the same document have the highest privilege to cooperate.

When a new document or a new version is checked in by a peer, an abstract will be sent to the "catalog sever". It specifies the IP address of the peer, document name, directory, version number, length, modification date, modification reason and so on. Then according to the role priority and on-line circumstance the peer will choose

another peer to deposit the duplicate. If there are no other peers in the same group online, the duplicate will be sent to the "catalog server". Because the "catalog server" also acts as a peer, and the only particularity is that it can belong to any group, and behave as a substitute for any group. It is found out that under the worst circumstance, each group only has one peer. Then the duplicate will be sent to the "catalog server". In this case, the P2P structure becomes the traditional C/S structure.

When a peer checks out a certain document, it sends out request to the "catalog server". Then the "catalog server" will return the "abstract" and the list of peers who own the document. After receiving the information, the peer will try to accept the file block from the peers in the list. There are at least two peers who own the complete document in the network. With the increase of software development in scale, the number of peers who holds the document will go up, so the speed of obtaining a document will increase. In the same manner, an additional version number will be sent to the "catalog server" if a specified version is checked out.

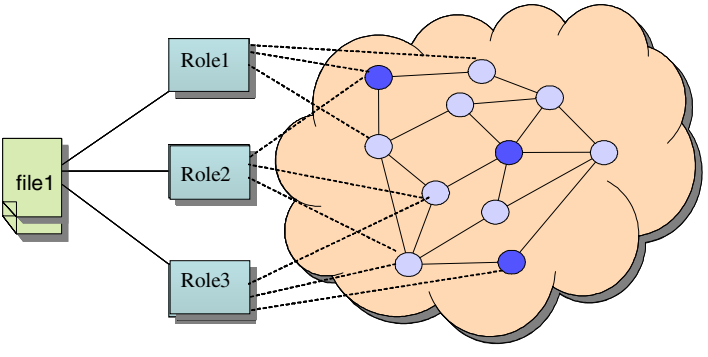


Fig. 5. The relationship between files and roles

4.3 Software Visualization

Based on P2P structure, the communications among peers are much more than document transfer and version control. We can take advantage of the computing ability of each peer to perform the system visualization. For a document, when it is checked in, a time point will be recorded. When the software is visualized, in order to make the peers cooperate, tasks are assigned based on roles and the time information is used for synchronization. Thus a few peers are combined to complete a common task. This can increase the system performance considerably and make large-scale software visualization possible.

With the development of technology, the third generation software development method will gradually not be able to satisfy the network need. It lacks initiative and intelligence. A new generation software development mode, namely agent based mode is becoming more and more popular, especially with the emergence of grid technology.

Reference [14] introduces a system on graph-based visualization, which extracts information from CVS and displays it using a temporal graph visualizer. The

information can be used to show the evolution of a legacy program. Our system applies agent technique to the representation and visualization method in [14] (where color of node and edge are used to represent information). As each peer only needs to be responsible for its own interested document, it only performs its own interested document visualization. If it is required to look into other documents' visualization, it only needs to contract the responsible peer to obtain it, who will transfer the information as a document. The agents on each peer are responsible for analyzing the output data of local peer, creating the drawing missions and broadcasting the advertisement.

```
<TaskHelpQuery>
    <FileID>Interested document ID</FileID>
    <RoleID></RoleID>
</TaskHelpQuery>
```

A request advertisement format is shown above. The advertisement is very brief. It contains the interested document ID and the role ID that the request peer owns. When other peers receive the advertisement, they will start a state check and performance valuation for its own. If they have enough CPU idle time and memory to complete the mission together, they then send out a response message. Otherwise, they will do nothing. The response message format is described as follows.

```
<TaskHelpResponse>
    <IsInterested>true/false</ IsInterested>
    <RoleID></RoleID>
</ TaskHelpResponse>
```

If the document that FileID points out is also the response peer's interested document, the IsInterested tag will be true, otherwise false. The response peer will return the IsInterested tag and its own RoleID together.

After the request peer receives the response message, it will do one of the followings.

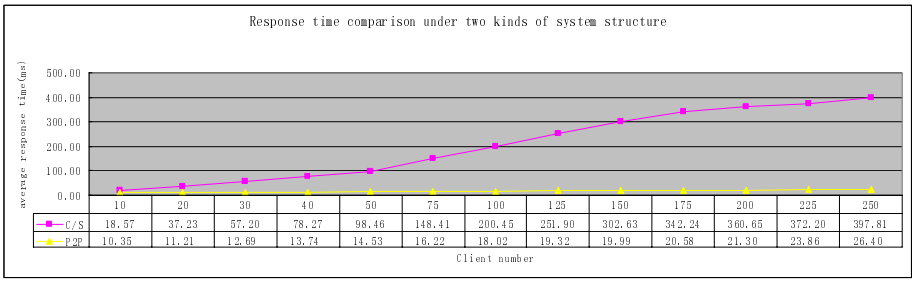
a. If the document is also an "interested document" to the response peer and two peers have the same RoleID, the request peer will send out an invoking message, which specifies the position of code and data.
b. If the document is an "interested document" to the response peer but the two peers don't have the same RoleID, the request peer will send out the require code and position of data.
c. If the document is neither an "interested document" to the response peer, nor the two peers have the same RoleID, the request peer needs to send out both the code and data (entitled Codat).

After the response peer receives the tasks, it will run the code and return the result to the request peer.

5 Performance Test

We developed the proposed software management system in this paper on Microsoft .Net platform and applied it in the actual development management process. The whole project contains about 250 members and every group has less than 10 people. Usually the active group number is about 50. Because of the diversity of the geographical positions, these groups are connected by Internet.

The P2P structure reduced the load of the server. However, the messages that were transferred for cooperation between peers need to frequently deliver in the network. This may result in performance degradation, and peers with full burden may have great burden. We tested the response time to a single mission. Fig. 6 is the result. The data in C/S structure is obtained by dividing all peers into "one group a peer". It can be found that, with the increase of the client number, the respond time in C/S structure has quickly increased. However the P2P structure did not experience the obvious performance degradation. The larger the client number is, the more obvious the performance differences the two structures have.



	10	20	30	40	50	75	100	125	150	175	200	225	250
C/S	18.57	37.23	57.20	78.27	98.46	148.41	200.45	251.90	302.63	342.24	360.65	372.20	397.81
P2P	10.35	11.21	12.69	13.74	14.53	16.22	18.02	19.32	19.99	20.58	21.30	23.86	26.40

Fig. 6. Comparison with the response time to a single mission in two structures

6 Conclusion and Future Work

There are a lot of uncertain factors in the P2P based software engineering system. Much time and resources need to be spent on system's normal operations. However, when it is applied to manage the software engineering, the management complexity will reduce considerably. Since peers in the same group are similar with occurrence in software develop process, the cost to support P2P network operation is very limited, and this shows the advantages of the P2P system.

The data volume within a distribute system such Grid, P2P is generally huge, and the relation between them are also complicated. Using software visualization technique, the meaning of these data can be presented very clearly. Visual software development can let the developers focus on the logic and the business processes. Therefore the room for visual software development in distribute system is pretty large. Visual software development tools can help the developers to complete their tasks very efficiently. 3D visual components and artificial intelligence can be used in software visualization for software engineering management. Moreover, software

engineering diagrams, such as GANT diagram, may be used to automatically assign and perform the tasks, and manage the peers. Combining P2P technique, agent technique and UML together with software visualization may provide considerably help to software engineering management.

Acknowledgements

The authors would like to thank the support from the China NSFs under Grant #60473106, #60273060 and #60333010, China Ministry of Education under Grant#20030335064, China Ministry of Science and Technology under Grant #2003AA4Z3120.

References

[1] Peer-to-Peer Working Group Homepage. http://www.peer-to-peerwg.org/index.html.
[2] Napster Homepage. http://www.napster.com .
[3] .NET My Services homepage (formerly code-named Hailstorm). http://www.microsoft.com/net/netmyservices.asp, http://www.microsoft.com/net/hailstorm.asp.
[4] Jxta Homepage. http://www.jxta.org .
[5] OpenP2P Homepage. http://www.openp2p.com.
[6] Web Service workshop. http://www.w3.org/2002/ws/.
[7] O'Reilly Network. http://onjava.com/.
[8] UML Software Engineering Organization. http://www.uml.org.cn/.
[9] Jorma S., Marja K., "Program Animation Based on the Roles of Variables", ACM Symposium on Software Visualization, San Diego, CA, 2003.
[10] Andrian M., Louis F., Jonathan I.M., "3D Representations for Software Visualization", ACM Symposium on Software Visualization, San Diego, CA, 2003.
[11] Cheng Z., Kenneth L.S., Thormas P.C., "Graph Visualization for the Analysis of the Structure and Dynamics of Extreme-Scale Supercomputers", ACM Symposium on Software Visualization, San Diego, CA, 2003.
[12] Niklas E., Philippas T., "Growing Squares: Animated Visualization of Causal Relations", ACM Symposium on Software Visualization, San Diego, CA, 2003.
[13] Qin W., Wei W., Rhodes B., Karel D., Bruno D.: Evolve, "An Open Extensible Software Visualization Framework", ACM Symposium on Software Visualization, San Diego, CA, 2003.
[14] Christian C., Stephen K., Jasvir N., "A System for Graph-Based Visualization of the Evolution of Software", ACM Symposium on Software Visualization, San Diego, CA, 2003.
[15] America Earthquake Grid. http://www.neesgrid.org/.
[16] WANG Q., Dai Y., Tian J., Zhao T., Li X.M., "An Infrastructure for Attribute Addressable P2P Network: Barnet", Journal of Software, 2003, vol 14.
[17] Shi W.Y., "Digital City Service Management Mode and Platform Design based on P2PSMS", Degree Paper, 2002, 6.
[18] Tian L.W, Yin C.W., "Researching and Implementing of Intelligent Professional Search for Virtual Enterprise", Chinese journal of computers, 2004, vol 3.

A Computational Reputation Model in P2P Networks Based on Trust and Distrust

Wei Lin[1,2], Yongtian Yang[1], and Shuqin Zhang[1]

[1] College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China
[2] School of Computer Information Science and Technology in Shenyang University of Technology, Shenyang 110023, China
{linwei, yangyongtian, zhangsq}@hrbeu.edu.cn

Abstract. Reputation mechanism can be used for choosing the peers suitable to collaborate with in P2P networks, so reputation and trust management has attracted many researchers' attentions. In this paper, a reputation management model for P2P systems is presented, in which trust and distrust are integrated, and multiple levels are differentiated on each aspect. The Confidence Index (*CI*) is introduced in the model for representing the confidence in evaluation results. Then a peer selection algorithm is designed for P2P collaborations. Simulation evaluation with different settings shows it effectively isolates malicious peers, and enhances the performances of the networks.

1 Introduction

In recent years, Peer-to-Peer (P2P) has gained widespread attentions. Such architectures and systems are characterized by direct access between peer computers, rather than through a centralized server. Different from traditional C/S architectures with the functionalities of clients and servers strictly separated, P2P architecture support each node to make similar contributions. For example, currently popular P2P file sharing applications on the Internet allows users to contribute, search, and obtain file.

In P2P networks reputation systems are widely used to propagate trust information and establish trust between peers [1],[2],[4],[6],[7],[10],[12]. Through the reputation system a peer can evaluate the peer it deals with after a transaction, and the accumulation of such evaluations makes up a reputation for the involved peers, moreover peers can exchange the reputation information of others so that a peer is able to know a stranger. Most of the current trust and reputation systems use positive or negative ratings information solely to build the reputations of others. But little work gives a mathematical approach to the evaluation of distrust, while distrust is at least as important as trust [9]. It is not proper only to use trust to express reputation, when trust value is low the new coming and bad-behavioring peers cannot be differentiated.

This paper rectifies the situation. In the proposed model, trust and distrust are treated equally, which evaluate positive and negative experiences with others, respectively. So the reputations of peers in networks involve two aspects: trust and distrust, by which peers can make more elaborate reputation based transaction policy. In addition, Confidence Index (*CI*) is introduced for expressing the confidence in the evalua-

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 501 – 508, 2005.

tion results. Simulation shows the reputation model can effectively isolate the malicious peers and enhance performances of the networks.

2 Trust and Distrust Evaluation

Based on the performances in a transaction, a peer can give the target peer a trust/distrust rating $<l_t, l_d>$, where l_t and l_d are trust and distrust rating, respectively. From the transaction records, the evaluation result for the target peer is represented as $<T(t, d), c>$, where t and d are the trust and distrust value, respectively, and c is the Confidence Index for $T(t, d)$.

2.1 Trust Degree

Trust represents how well collaborator behaviors. Based on each transaction one peer can give its collaborators a trust level, such as linguistic label "*high*" or "*low*". So the comprehensive trust value can be achieved from the transactions history. For calculating the trust value each trust level corresponds to a positive numeric value. The higher the trust level is, the larger the corresponding numeric value is. The trust value can be calculated as following:

$$tv = \sum \sigma^{i-1} \cdot n_i \ , \ 0 < \sigma < 1 . \tag{1}$$

Where σ is the decaying factor, which means the more recent experiences will take up a more important role in the computed trust value. n_i is the numerical value related to the ith trust rating from the most recent. In respect that peers may change their behaviors over time, and the earlier ratings may have little impact on the calculation result, it is desirable to consider more recent ratings, that is, to consider only the most recent k ratings, and discard those previous ones.

In practical use, the trust degree can be deduced as following:

$$t = \frac{tv}{tv_{\max}} \ . \tag{2}$$

where tv_{max} is the maximum trust value as possible, for example, if σ is 0.5, and k is 4, the value for highest trust level is 3, the value of tv_{max} is $3+0.5*3+0.5^2*3+0.5^3*3=5.625$.

2.2 Distrust Degree

Similar to trust, distrust represents how the collaborator maliciously behaves. For a transaction, one peer can give a distrust level, and the different levels represent different degrees the malicious behavior can harm. In the calculation of distrust value, each distrust level relates to a positive numeric value, of course, if the transaction is trusted completely, the corresponding numeric value should be 0. The distrust value can be calculated as following:

$$dv = \sum \rho^{i-1} m_i \ , \ 0 < \rho < 1 . \tag{3}$$

Where ρ, similar to σ in formula (1), is decay factor. m_i is the numeric value corresponding the ith distrust rating with the most recent first. The distrust degree can be calculated as following:

$$d = \frac{dv}{dv_{\max}} \ . \tag{4}$$

where dv_{max} is the maximum distrust value as possible and its calculation is similar to tv_{max}.

2.3 Confidence Index

Considering the factors of freshness, amount of experiences, Confidence Index (*CI*) is introduced to express the confidence level in evaluation results from these ratings. It is determined by the following factors: aging of ratings, number of ratings etc. The *CI* c for the reputation value of target peer q given by peer p is computed as following:

$$c_p^q = \mu \cdot \lambda^{t_{now} - t_{av}} \ , \ 0 < \lambda < 1 \ . \tag{5}$$

Where $\mu = n_{total}/n_{max}$, n_{total} is the number of the ratings considered, and n_{max} is the maximum number to be considered for a peer, and the upper limit for μ is 1. t_{av} is the average time of these ratings considered, and t_{now} is current time. λ is a longevity factor.

3 Reputation Computing

In P2P networks, when a peer p wants to have a transaction with target q, but has no sufficient information for q locally, p will send a reputation query to the networks. Upon receiving the query peer returns recommendation $<T, c>$ based on local transaction records with q. By combining these recommendations the reputation of q is achieved.

3.1 Weighted Aggregating Recommendations

Considering that different peers may provide diverse recommendations for the same peer, a weighted approach is adopted to compute the target peer's comprehensive reputation. In computing the reputation, one keeps the weights assigned to others in its Weight Table (*WT*). The queried reputation value of the target peer q is computed as following:

$$R^q = \sum_{r \in P} \frac{w_r \cdot c_r^q}{\sum_{i \in P} w_i \cdot c_i} \cdot T_r^q \ . \tag{6}$$

where $<T_r^q, c_r^q>$ is the recommendation for peer q reported by recommender r; and w_r denotes the weight for it. P denotes the set of peers who have given recommendations in this time.

From *CI*s in the received recommendations, one computes the *CI* c^q for the queried reputation value R^q of the target peer q as following:

$$c^q = \sum_{p \in P} \frac{w_r \cdot c_r^q}{S} \quad . \tag{7}$$

Where S is the sum of weights of the peers in P, so the queried reputation of the target peer q can be represented as $<R^q, c^q>$.

3.2 Adjustment of Weights

The weights for peers can be tuned up or down through detecting the accuracies of their reports: weights assigned to the peers who give more accurate ratings should be increased, and weights assigned to those who give deceptive ones should be decreased 3. In this way, recommenders will have different impacts on the comprehensive reputations.

After a transaction, by comparing each recommendation for the target peer with the local evaluation the accuracy of the recommendation can be assessed. Precisely, the accuracy A of the recommendation $<t_r, d_r, c_r>$ can be defined with 1 representing '*accurate*' and 0 representing '*not accurate*'. Formally define $v_r = t_r - d_r$, $v_l = t - d$, where $<t, d>$ is the updated local rating. If v_r and v_l are of the same sign and $|v_r - v_l| < \theta$, then A=1. Otherwise A=0. That is,

$$A = \begin{cases} 1 & v_r \cdot v_l \geq 0 \ \ and \ \ |v_r - v_l| < \theta \\ 0 & otherwise \end{cases} \tag{8}$$

where the threshold θ can be tuned for different measures against accuracy.

So the weight for corresponding recommender will be updated based on the accuracy of its recommendation. Specifically, with formula (9) and formula (10) one peer can increase and decrease the weight, respectively:

$$w' = \begin{cases} w_u & w + \Delta_1 > w_u \\ w + \Delta_1 & otherwise \end{cases} \tag{9}$$

$$w' = \begin{cases} 0 & w - \Delta_2 < 0 \\ w - \Delta_2 & otherwise \end{cases} \tag{10}$$

where w is old weight, and w' is the updated one. Δ_2 should be bigger than Δ_1 for punishment. Furthermore, to prevent a few peers from dominating the overall computation results, an upper limit w_u can be set for weights. For a newcomer, a small weight such as Δ_1 can be assigned initially to guarantee the chance to adopt its opinion.

4 Peer Selection

In the process of selecting provider, the following scheme, similar to that in [1], is adopted: For a group of providers G responding the service query, the *CI* for provider

$\boldsymbol{p}$ is $\boldsymbol{c_p}$, and the $\boldsymbol{G_o}$ denotes the respondents in $\boldsymbol{G}$ with $\boldsymbol{c} \geq \boldsymbol{c_T}$. The cardinalities of these sets can be denoted as $\boldsymbol{n}(\boldsymbol{G}) = |\boldsymbol{G}|$. If the $\boldsymbol{n}(\boldsymbol{G_o})$ is large enough, that is if $\boldsymbol{n}(\boldsymbol{G_o}) \geq \boldsymbol{n_c}$, the collaborator can be selected from peers in $\boldsymbol{G_o}$.

If $\boldsymbol{n}(\boldsymbol{G_o}) < \boldsymbol{n_c}$, a set of $\boldsymbol{n_c} - \boldsymbol{n}(\boldsymbol{G_o})$ random peers in $\boldsymbol{G}-\boldsymbol{G_o}$, denoted by $\boldsymbol{G_q}$, are selected to be queried for their reputations. Based on the queried reputation and the local evaluation the synthetic value for each peer in $\boldsymbol{G_q}$ can be computed as following:

$$T_{total} = \frac{c_L \cdot T + c_R \cdot R}{c_L + c_R} \quad . \tag{11}$$

where $\langle \boldsymbol{T}, \boldsymbol{c_L} \rangle$ is the local evaluation. $\langle \boldsymbol{R}, \boldsymbol{c_R} \rangle$ is the queried reputation. So the collaborator can be selected from $\boldsymbol{G_o}+\boldsymbol{G_q}$. In this way, one peer can avoid to a certain extent collaborating with the peer that has given it bad performances, even if it gives the rest of the network good ones.

In reputation based peer selections, the *min-distrust-max-trust* strategy is used. For a group of candidates, the one with minimum distrust degree is given priority over maximum trust degree.

5 Experiments

In this section, the performances of the proposed model are examined, and a Gnutella-like P2P file sharing application with the reputation model is implemented.

5.1 Settings

In the simulation systems each peer is linked to a certain number of neighbors, and a file query message issued by a peer is propagated over these links for a certain number of hops specified by the TTL. In simulation run, periodically these peers can randomly select a file unavailable locally, and sending the file query message to networks via neighbors. Upon receiving the file query, peer will return the file from local file storage, so the requester can select a source from these respondents for downloading. In experiments, each peer can give a certain QoS, which refers peers' different capabilities in uploading files. Moreover, there are three kinds of malicious peers providing inauthentic files hurting security with various degrees. The common simulation networks parameters in experiments are shown as Table 1:

Table 1. Simulation Parameters

Parameter Description	Value
number of peers in networks	1000
number of neighbors of each peer	3
number of distinct file versions	1000
number of files hold by each peer initially	10
ratio of malicious peers to all	20%
ratios of peers with four QoS to all	20%, 30%, 30%, 20%
ratios of three sort malicious peers	50%, 30%, 20%

In our implemented model, the trust levels and distrust levels are defined as Table 2 and Table 3, based on four QoS and on the types of malicious providers.

Table 2. Trust Levels

Trust Level (TL)	Number	Trust Level (TL)	Number
T1	1	T3	3
T2	2	T4	4

Table 3. Distrust Levels

Distrust Level (DTL)	Number	Distrust Level (DTL)	Number
DT1	1	DT3	3
DT2	2		

After a download, the trust level can be given for the source based on the QoS provided, and the distrust level based on the provider. Other parameters of the model in the simulation are set as Table 4:

Table 4. Model Parameters

Parameter Description	Value
decaying factor σ, ρ in formula (1), (2)	0.8
longevity factor λ for calculating *CI* in formula (5)	0.95
maximum number of the considered most recent ratings	4
threshold θ in formula (8)	0.1
increment Δ_1, decrement Δ_2 in formula (9), (10), respectively	1, 2
upper limit w_u for weights	8
threshold *CI* to be needed for query in peer selection	0.1
threshold number of peers considered in peer selections	4

5.2 Results

In experiments, we are particularly interested in the malicious or non-malicious transactions (downloads) versus all downloads: If the reputation mechanism reflects peers' actual behavior, the chance that malicious peers are chosen as download source should be minimized, and the chance that good peers are chosen be increased. One non-malicious transaction is defined as the one in trust evaluation for which distrust level is 0, and the malicious transaction the reverse. The best download is defined as the non-malicious one with the highest trust level, i.e. 4. To review the model's performances under scenarios with different attackers, the two parameters are defined:

Φ: The ratio of malicious downloads to every 1000 downloads
Ψ: The ratio of best downloads to every 1000 downloads

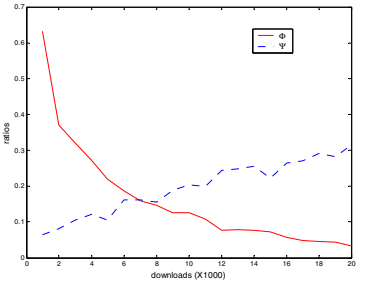


Fig. 1. Naïve attack, native attackers always provide bad performances when selected as the source, and give dishonest recommendations upon receiving reputation queries

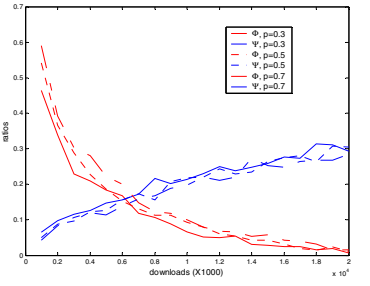


Fig. 2. Sneak attack, Sneak attackers act like a reliable peer at most of the time and try to execute malicious transaction with others with a certain probability **p**

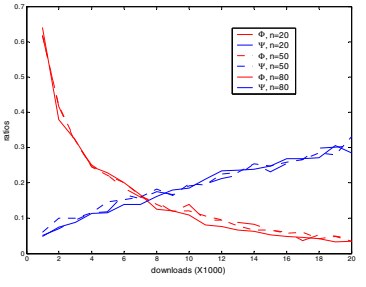


Fig. 3. Sybil attack, Sybil attackers initiate many identities in networks. After **n** malicious transactions one identity is used, the identity is replaced with another one. Since frequently discarding old identity, the transactions of one identity are sparse in the networks, so the reputation information is weak

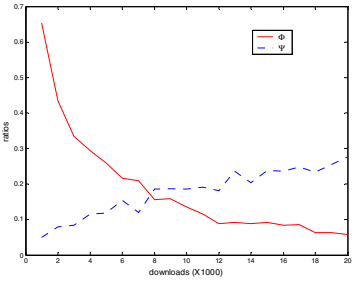


Fig. 4. Clique attack, clique attackers form a malicious collective who give positive opinions to their colluders when they receive a relevant trust queries

Results of simulations under different attack scenarios are shown in Figures 1–4, in which all **Φ** curves descend, and all **Ψ** curves ascend, remarkably. Therefore, the results indicate that the proposed model runs well in withstanding typical attacks, isolating malicious peers, and improving quality of downloads in P2P networks. Then the robustness and effectiveness of the reputation model is proved.

6 Conclusions and Future Work

In this paper, a reputation computation model is proposed for P2P networks, in which trust and distrust are considered. The comprehensive reputations can be computed by using weighted approach based on recommendations from others, furthermore, by adjusting weights for recommenders according to the accuracies of them, deceptive

recommenders can be easily detected, and their impacts are suppressed. In the scheme, the weighted algorithm ensures the robustness to the malicious recommenders. Subsequently, the reputation based P2P collaboration mechanism is designed. The file sharing networks simulations evaluate the effectiveness of the model under different scenarios.

But the model proposed in the paper only uses a simple scheme to adjust weights, and other parameters in the model may also be adjusted adaptively. In future, more complex schemes have to be designed. For example, heuristic methods can be applied to improve the performances of the model.

References

1. A. A. Selcuk, E. Uzun, and M. R. Pariente. A Reputation-Based Trust Management System for P2P Networks. CCGRID2004: 4th IEEE/ACM International Symposium on Cluster Computing and the Grid, 2004.
2. Aberer, K. and Despotovic, Z. Managing Trust in a Peer-2-Peer Information System. Proceedings of the Tenth International Conference on Information and Knowledge Management (ACM CIKM'01), 310-317, 2001.
3. B. Yu and M. P. Singh. Detecting Deception in Reputation Management. Proc. of 2nd Intl. Joint Conf. on Autonomous Agents and Multi-Agent Systems, 73-80, 2003.
4. E. Damiani, D. C. di Vimercati, S. Paraboschi, P. Samarati, and F. Violante. Reputation-based approach for choosing reliable resources in peerto-peer networks. In Proc. of the 9th ACM Conference on Computer and Communications Security. 2002.
5. F. Azzedin and M. Maheswaran. Trust Brokering System and Its Application to Resource Management in Public-Resource Grids, 2004 International Parallel and Distributed Processing Symposium (IPDPS 2004), 2004
6. L. Xiong and L. Liu. A reputation-based trust model for peer-to-peer ecommerce communities. In IEEE Conference on E-Commerce (CEC'03). 2003.
7. P. Dewan and P. Dasgupta. PRIDE: Peer-to-Peer Reputation Infrastructure for Decentralized Environments. The 13th Intl. World Wide Web Conference, 2004.
8. P. Resnick, R. Zeckhauser, E. Friedman, and K. Kuwabara. Reputation Systems. Communications of the ACM, 43(12): 45–48, 2000.
9. R Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins, Propagation of Trust and Distrust. In Proc. International WWW Conference, New York, USA. 2004.
10. S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The EigenTrust Algorithm for Reputation Management in P2P Networks. In Proc. of the Twelfth International World Wide Web Conference, 2003.
11. S. Saroiu, P. K. Gummadi, and S. D. Gribble. A Measurement Study of Peer-to-Peer File Sharing Systems. In Proc. of Multimedia Computing and Networking 2002 (MMCN '02), San Jose, CA, USA, 2002.
12. W. Dou , H.M. Wang, Y. Jia, P. Zou. A Recommendation-Based Peer-to-Peer Trust Model. Journal of Software, Vol.15, No.4, 2004 (in Chinese).

Web Services Peer-to-Peer Discovery Service for Automated Web Service Composition*

Jianqiang Hu, Changguo Guo, Huaimin Wang, and Peng Zou

School of Computer Science, National University of Defense Technology,
410073 Changsha, China
jqhucn@hotmail.com

Abstract. Current Web service discovery methods are based on centralized approaches where Web services are described with service interface functions but not process-related information. It cannot guarantee compatibility of Web service composition, nor can it makes Web services easy to complete a deadlock-free and bounded process transaction. Furthermore, centralized approaches to service discovery suffer from problems such as high operational and maintenance cost, single point of failure, and scalability. Therefore, we propose a structured peer-to-peer framework for Web service discovery in which Web services are located based on service functionality and process behavior. It guarantees semantic compatibility of Web service composition, and achieves the automated composition at the level of executable processes.

1 Introduction

Web services and related technologies promise to facilitate efficient execution B2B e-ecommerce by integrating business applications across networks like the Internet. In particular, the process-based composition of Web services is gaining a considerable momentum as an approach for the effective integration of distributed, heterogeneous, and autonomous applications. In this approach, each *component service* performs an encapsulated function ranging from a simple request-reply to a full business process [1]; multiple component services are federated into *composite services* whose business logic is expressed as a process model (e.g. BPEL4WS [2], OWL-s [3]). It is critical to search an appropriate component service to compose Web services and complete loosely coupled business processes which require dynamic and flexible binding of services.

Based on functionality description (e.g. WSDL) without process-related information, UDDI [4] supports searching of name-value pairs, which cannot guarantee the compatibility of Web service composition and yields a deadlock-free and bounded business transaction. Meantime, UDDI has a centralized architecture suffering from problems such as high operational and maintenance cost, single point of failure, etc. Fortunately, Peer-to-Peer, as a complete distributed computing model, could supply a scalable, flexible and robust scheme for the Web service discovery. There exist most

* This work was supported by National 863 High Technology Plan of China under the grant No.2003AA115410, No.2003AA115210, No.2003AA111020, No. 2004AA112020.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 509 – 518, 2005.

of approaches, e.g. Speed-R system [5] and Hypercube ontology-based P2P system [6], to mainly focus on the scalability. Meantime, these methods locate Web services based on their functionality but not the business aspects of the services, and cannot guarantee compatibility of Web service composition based on process model. To our knowledge, a service for searching an appropriate *component service* based on process description language does not yet exist in the decentralized infrastructure.

In this paper, a service discovery approach is presented that allows searching and finding Web services by a comparison of behaviors of business processes for compatibility. In order to avoid adhering to specific some description language, a model of **ADFSA** (Annotated Deterministic Finite State Automata) can be introduced. Furthermore, the system can be executed on top of the structured Peer-to-Peer overlay network to enhance the scalability.

The rest of the paper is organized as follows: In Section 2 we introduce our Web service model based on annotated deterministic finite state automata. Section 3 describes the proposed Web service discovery technique. The related work is summarized in Section 4. Section 5 concludes and outlines the future work.

2 A Model for Web Services

Web services are typically described at three major levels: Messages, abstract processes, and execution processes [7]. (1)Message descriptions such as WSDL describe the syntax and structure of messages; (2) Abstract processes describe the sequences in which messages may be exchanged; (3) Execution process description extends abstract process description with information necessary to execute a business process. There are several proposals for specifying abstract processes regardless of concrete implementation, including WSCL, OWL-s and the abstract part of BPEL. A successful process includes states realizing the interaction with partners represented by exchanging messages. When searching for a potential partner, it is necessary that the exchanged message sequences of the process are compatible.

2.1 Compatibility

The exemplary scenario used for further discussion is two services composition within a virtual enterprise. Fig.1 depicts two business processes involving trading services: Customer service *C* and Ticket service *T*. Nodes represent the states of a business process; edges represent state transitions, which are labeled with messages denoted as *from~recipient#messagename*, where *from* is the message sender, *recipient* is the message recipient and *messagename* is the name of the message. A process involves two kinds of messages: mandatory message and optional message. Consequently, we annotate "$\wedge$" as mandatory transition of message, and "$\vee$" as genuine alternatives of message in the following examples [8].

Fig.1(a) shows the Ticket business process, where it starts a ticket order *C~T#OrderTicket* message, followed by a VISA payment *C~T#PayVISA* message and a delivery message *T~C#Delivery* to reach the end state. The Customer process depicted in Fig.1(b) starts the process with a ticket order *C~T#OrderTicket*, then it insists on delivery *T~C#Delivery* before payment by VISA *C~T#PayVISA* ***or*** by cash *C~T#PayCash*.

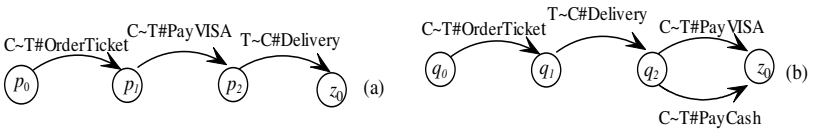


Fig. 1. (a) Ticket Message Sequence; (b) Customer Message Sequence with optional message C~T#PayVISA ⋁ C~T#PayCash

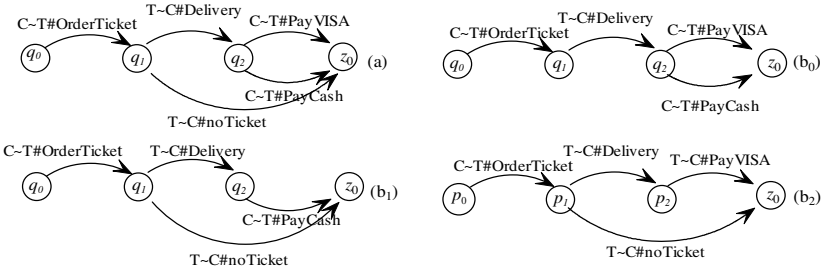


Fig. 2. (a) Ticket Message Sequence insisting on *T~C#Delivery* ⋀ *T~C#noTicket* and *C~T#PayVISA* ⋁ *C~T#PayCash*; (b_0) Customer Message Sequence requiring *C~T#PayVISA* ⋁ *C~T#PayCash*; (b_1), (b_2) Customer Message Sequence insisting on *T~C#Delivery* ⋀ *T~C#noTicket*

Fig.2(a) shows the Ticket business process, where it starts a ticket order *C~T#OrderTicket* message, followed by *T~C#Delivery* message and either a VISA payment *C~T#PayVISA* ***or*** a cash payment *C~T#PayCash* (optional message) to reach the end state. If the tickets have been sold out, it ***must*** reject the ticket order by using a no ticket message *T~C#noTicket*. Fig.2 (b_0) shows a Customer business process. As it cannot support the required *T~C#noTicket* message, Ticket business process (a) and Customer business process (b_0) cannot achieve the successful business interaction. Conversely, the business processes in Fig.2(b_1) (b_2) can handle the *T~C#Delivery* ⋀ *T~C#noTicket* messages (mandatory message), while they support only one payment message. However, the Ticket process (a) and the customer processes (b_1) (or (b_2)) are compatible because they may successfully interact.

In summary, the two examples in Fig.1 and 2 illustrate that message sequence and mandatory choices need to be taken into account to determine whether a service is compatible with another service or not from a successful business interact viewpoint. Generally, the compatibility of Web services is divided into syntactical compatibility and semantic compatibility as follows:

Definition 1 (Syntactical compatibility). Let A and B be Web services, A and B are syntactically compatible, *iff* each common interface is an output interface of one service and an input interface of the other service.

Ticket service has two input interfaces (TicketOrder, PayVISA) and one output interface (Delivery); Customer service has two output interfaces (TicketOrder, Pay-

VISA) and one input interface (Delivery). They are syntactically compatible with completely matching interfaces. Fig.1 shows two Web services match at level of individual messages without considering message sequences. However, they require an opposite order of message sequence and cannot successfully interact. Ticket service is in state p_1 waiting for *PayVISA* message, and Customer service is in state q_1 waiting for *Delivery* message at the same time. Therefore, both services have syntactically compatible interfaces but the resulting process leads to a deadlock.

Definition 2 (Semantic compatibility). Let A and B be two syntactically compatible Web services, A and B are semantically compatible, *iff*: (1) For each reachable message sequence state (starting at $[q_0]$), the final state $[z_0]$ is reachable; (2) For each reachable message sequence state q such that $q \geqq [z_0]$ holds $q=[z_0]$.

For more information, e.g. the precise definition of reachable state sees [9]. Fig.2(a) and (b_0) are syntactically compatible but not semantically compatible because Customer service cannot handle mandatory choices. In order to guarantee a successful business interaction, we take into account message sequences rather than individual messages. Two Web services (See Fig.2(a), (b_1)) are semantically compatible and share the same message sequence including mandatory choices. They solve an internal conflict and achieve the goal of automating the composition at the level of executable processes, i.e. the automated composition at the semantic level.

2.2 Modeling

Finite State Automata has a clear semantic for the automated composition of Web services (e.g., Customer service C and Ticket service T in section 2.1). There exist approaches and standards, e.g. BPEL4WS and OWL-s, to define the behavior of business processes and interactions among them. However, they require a much higher computational complexity compared to Finite State Automata. What's more, these proposed approaches do not exceed the expressive capability of Finite State Automata. Formally, deterministic finite state automata can be represented as follows:

Definition 3 (Deterministic Finite State Automata (DFSA)). A deterministic finite state automaton x is represented as a tuple $DFSA_x=<Q, \sum, f, q_0, Z>$ where: (1) Q is a finite set of states; (2) $\sum \subseteq R \times R \times M$ is a finite set of messages in M sent by a sender in R to a receiver in R; (3) f is a function, where: $Q \times \sum \mapsto Q$. It maps a (state, message) pair to a state; (4) $q_0 \in S$ is a start state; (5) $Z \subseteq S$ is a set of final states.

DFSA model cannot completely fulfill the following requirement: messages sent by a party a particular state ***must*** be supported by the corresponding received party (See Fig.2 (a), (b_1)). This is because the sender has the choice to select a particular message to be sent, while the receiver must be able to handle all possible choices of the sender. Therefore, **DFSA** model cannot distinguish between mandatory and optional messages. In order to avoid this advantage, an annotated set L is introduced into ADFSA model.

Definition 4 (Annotated Deterministic Finite State Automata). An annotated finite state automaton x is represented as a tuple $ADFSA_x=< Q, \sum, f, q_0, Z, L>$ where: (1) Q is a finite set of states; (2) $\sum \subseteq R \times R \times M$ is a finite set of messages in M sent by a sender in R to a receiver in R; (3) f: $Q \times \sum \mapsto Q$ maps a (state, message) pair to a state; (4) $q_0 \in Q$ is a start state; (5) $Z \subseteq Q$ is a set of final states; (6) L is an annotated set. It represents relation of states by using logic terms $\{ \vee , \wedge \}$.

ADFSA can convey not only message sequences but also mandatory choices. Moreover, it constitutes a reasonable foundation for the purpose of automated composite service at the level of executable processes. For example, **ADFSA** model can easily describe Web services at the execution processes level in Fig2.(b_1) with the following tuple:

$<\{q_0, q_1, q_2, z_0\}, \{T\sim C\#OrderTicket, T\sim C\#Delivery \wedge T\sim C\#noTicket, C\sim T\#PayCash\}, f, q_0, z_0, \{ \vee , \wedge \}>$, where f is defined below: $f(q_0, T\sim C\#OrderTicket)=q_1$, $f(q_1, T\sim C\#Delivery \wedge T\sim C\#noTicket)=q_2$, $f(q_2, C\sim T\#PayCash)=z_0$.

2.3 Service Match

Service Match is a critical measure to determine whether service partner is compatible with request partner or not from a business process viewpoint. Generally, it is very difficult to match Web services based on process description (e.g. BPEL4WS, OWL-s) directly. Fortunately, a Finite State Automata can be mapped from process description languages [8]. Based on **ADFSA** model, we can define the following kinds of service match to check whether or not service partner fulfills request partner requirement in terms of business process compatibility.

Definition 5 (Exact Match). Exact Match of two service is an isomorphism from $ADFSA_x=<Q, \sum, f, q_0, Z, L>$ to $ADFSA_{x'}=<Q', \sum', f', q_0', Z', L'>$, where the isomorphism is function g: $Q \mapsto Q'$ such that g is a homomorphism and bijection. i.e. (1) A homomorphism form $ADFSA_x$ to $ADFSA_{x'}$ is a function g: $Q \mapsto Q'$ such that: $\forall q \in Q$ and $\forall m \in \sum$, $g(f(q, m))=f'(g(q), m)$; (2) A inverse function g^{-1} from $ADFSA_{x'}$ to $ADFSA_x$ exists. Formally, Exact($ADFSA_x$, $ADFSA_{x'}$).

According to this definition, isomorphic **ADFSA**s are structurally identical. Their states may have different names, but their state graph, if the nodes are relabeled, look exactly the same. Exact match is too accurate and restrictive in practice. Generally, Plugin match [10] is enough, i.e. finding an **ADFSA** that can be plugged into the place where the request **ADFSA** was raised. Exact match is a special case of plugin match.

Definition 6 (Plugin Match). Plugin Match of two service is a simulation from $ADFSA_x=<Q, \sum, f, q_0, Z, L>$ to $ADFSA_{x'}=<Q', \sum', f', q_0', Z', L'>$, if there is a function g: $Q' \mapsto Q$ such that $\forall q' \in Q'$ and $\forall m \in \sum$, $f(g(q'), m)=f'(q', m)$. Formally, Plugin($ADFSA_x$, $ADFSA_{x'}$).

We formally analyzed the definitions of exact match and plugin match of Web services and also found that it is difficult to define a function that can show whether two services are Exact match and Plugin match or not. This situation justifies the necessity of a simple approach that facilitates the searching compatible Web services and achieves the successful interaction.

Definition 7 (Reachable Path Finite Automaton). An execution of a Web service is defined as a message sequence of interactions. The successful execution path from the start state to an end state is called a Reachable Path Finite Automaton (RPFA).

A necessary condition to achieve compatible interactions is that they share at least a Reachable Path Finite Automaton. For example, the request **ADFSA** shown in Fig.2 (a) has two RPFAs:

<*OrderTicket*, *Delivery* ∧ *noTicket*, *PayVISA*>;
<*OrderTicket*, *Delivery* ∧ *noTicket*, *PayCash*>.

where "∧" annotates mandatory message. In order to interact successfully, a potential partner should plug match with the above **ADFSA**. For example, a valid interaction sequence should be <*OrderTicket*, *Delivery* ∧ *noTicket*, *PayCash*> in Fig.2(b_1) or <*OrerTicket*, *Delivery* ∧ *noTicket*, *PayVISA*> in Fig.2(b_2). Meantime, they gain separately an automated composition at the level of executable processes.

3 A Peer-to-Peer Web Service Discovery Model

In this section, we describe our model which provides Web services discovery on top of the Chord peer-to-peer system [11, 12]. We chose Chord as the underlying peer-to-peer layer because it is simple and still achieves very good resilience and proximity performance. Our discovery model can be described as a search system which allows its users to publish and query for Web services via finite automata representations.

3.1 Web Services Publishing and Querying

Based on the good characteristics of Chord system, we can avoid problems (e.g. high maintenance cost, single point of failure, and poor scalability) of centralized approach. Given the process description of a Web service, it can be published and queried on top of Chord system.

1. Publishing Web Services
In order to enable the discovery of Web services according to process behavior, Web services are published via **ADFSA** that is mapped from process description. A possible method would be to hash the complete **ADFSA**. This solution is not efficient because the request partner may not know the complete finite automaton of potential partner. Meantime, request partner is focused on whether or not potential partner is compatible with itself. According to definition 7, we choose to hash RPFAs of Web services. Consequently, each Web service may be published several times. For example, Ticket service (see Fig.2(a)) will be published two times by hashing each of RPFAs. Each RPFA is used as the key and the peer responsible for the hash key stores information about complete annotated deterministic finite state automaton.

2. Querying Web Services
Web services can be searched by using RPFA as the key to route the query to the peer responsible for the RPFA. Each peer is composed of communication engine and local query engine. The communication engine is responsible for communication and collaboration with its neighbours in Chord system. Meanwhile, it receives and

responds query message. The local query engine receives queries from communication engine and queries the peer for matching A_2 with A_1, where A_1 is a RPFA included in the query message and A_2 is a set of RPFAs stored the peer. If $A_1 \subseteq A_2$, then A_2 can simulate A_1, i.e. **Plugin**(A_1, A_2), otherwise the system cannot find compatible services with request partner. Since both A_1 and A_2 are mapped from the process descriptions of Web services, queries enable the discovery of Web services according to their process behaviors.

3.2 Evaluation and Experimental Result

In order to evaluate the discovery efficient of our method, we have implemented a configurable tool based on LTSA (Labeled Transition System Analyzer) [13, 14], which can convert a BPEL description to **ADFSA**. We also constructed a simulator of Web service for publication and querying, which is executed on top of the Chord system. We search randomly actual process descriptions from Internet and then extract RPFAs from their **ADFSA**. Because the scalability and robustness are guaranteed by Chord characteristics, we are only focus on the latency and rate of query.

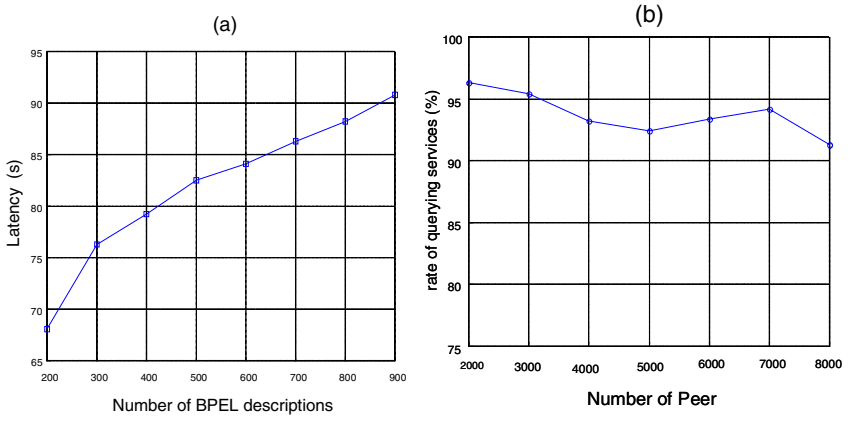


Fig. 3. (a) Effect of number of BPEL descriptions on latency; (b) Effect of number of peers on Rate of query

(1) Latency

Fig.3(a) shows the effect of number of BPEL descriptions on latency, where the peer number of our system is set to 3000 and thus the query can be routed via $\boldsymbol{O}(\log(3000))$ hops. It is obvious that the number of BPEL description has much influence on the latency in this figure. The minimum time is 68.3 seconds, when the number of BPEL descriptions is 200. When the number of BPEL descriptions is up to 900, the time is about 90.8 seconds. The latency increases rapidly with the number of BPEL descriptions, because the simulator takes more time transforming BPEL descriptions to **ADFSA** and extracting RPFAs.

(2) Rate of query
Rate of query means the successful rate of query that can be compatible with request partner. Fig.3(b) reveals the influence of the number of peers on the rate of query and displays a stable trend (between 91% and 97%). When the number of peer is 2000, the rate can be high as 96.3%. When the number of peer adds up to 8000, the rate is 91.2%. Therefore, the number of peers has little influence on the system.

In summary, experimental results show that our method is viable and effective.

4 Related Works

Centralized [4, 15] and peer-to-peer [5, 16] Web service discovery methods have been proposed in the literatures. Among these, UDDIe is present as an extension to UDDI in [15]. It extends UDDI and provides a more expressive way to describe and query services than UDDI, but does not avoid suffering from problems such as single point of failure and scalability. Literature [5] and [16] are similar to our method as they are built on top of structured P2P systems. Schlosser et al. use ontologies to publish and query Web services [6]. The main drawback of this method is the necessity of a common ontology used for annotating and querying services. Schmidt and Parashar describe each Web service by a set of keywords and then map the corresponding index to a DHT using Space Filling Curves [16]. All of the above methods locate Web services based on their functionality but not the business aspects of the services. Our work is different than these proposals as we consider both functionality and process behavior of the Web services during discovery, and guarantee semantic compatibility of Web service composition based on process model.

Our work is similar to the work presented in [8, 17, 18]. In [8], Web services are modeled as the annotated deterministic finite state automata, which help to match between different business processes. In [17], the process model of Web services is described with Petri Nets, which are used for simulation, verification composition. In [18], two services are regarded as compatible if every possible trace in one service has got a compatible one in the second one. This approach is similar to that used in our work; but unfortunately, the description how to do the compatibility check of the traces is not given. These proposals do not address the Web service discovery problem. In our work, we push some of the work required by these proposals to the discovery level by searching the Web services that are used to achieve Web services composition at the level of executable processes.

5 Conclusions and Future Work

A Web service is considered as the simple method invocation is sufficient for business-to-business and ecommerce settings. Current solutions for Web service discovery only consider the functionality of the Web services, but not their process behavior. In this paper, a structured peer-to-peer framework for Web service discovery in which Web services are located based on service functionality and process behavior. We

represent the process behavior of Web service with annotated finite state automaton for publishing and querying Web service on top of a peer-to-peer system. In addition, it guarantees semantic compatibility of Web service composition at the level of executable processes.

Issues not covered in this paper that are planned as future enhancements are: (1) transforming OWL-s and Petri Net into annotated deterministic finite state automata; (2) improving the algorithms on transforming BPEL into **ADFSA**.

References

1. Andreas Wombacher, Peter Fankhauser, Erich J. Neuhold: Transforming BPEL into Annotated Deterministic Finite State Automata for Service Discovery. In *Proceedings of International Conference on Web services*, pages 316-323, California, USA, 2004.
2. T. Andrews, F. Curbera, H. Dolakia, J. Goland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic, and S. Weeravarana: Business Process Execution Language for Web Services, 2003.
3. The OWL Services Coalition. OWL-S: Semantic Markup for Web Services. Technical White paper (OWL-S version 1.0), 2003.
4. UDDI. Uddi technical white paper. Technical report, http://www.uddi.org/
5. K. Sivashanmugam, K. Verma, R. Mulye, Z. Zhong: Speed-R: Semantic P2P Environment for diverse Web Service Registries.
6. M. Schlosser, M. Sintek, S. Decker, and W. Nejdl: A Scalable and Ontology-based P2P infrastructure for semantic Web services. In *Proceedings of the Second International Conference on Peer-to-Peer Computing*, pages 104–111, 2002.
7. P. Traverso, M. Pistore: Automated Composition of Semantic Web Services into Executable Process. In *Proceedings of Eighth IEEE International Symposium on Wearable Computers*, in Arlington, VA, October 2004.
8. A. Wombacher, P. Fankhauser, B. Mahleko, and E. Neuhold: Matchmaking for Business Processes Based on Choreographies. In *Proceedings of International Conference on e-Technology, e-Commerce and e-Service*, Taipei, Taiwan, March 2004.
9. Axel Martens: On Usability of Web services. In *Proceeding of Fourth International Conference on Web Information Systems Engineering Workshops*, Roma, Italy, December 13, 2003.
10. Xiang Go, Jian Yang and Mike.P.Papazoglou: The Capability Matching of Web Service. In *Proceedings of International Symposium on Multimedia Software Engineering*, California, USA, December 2002.
11. I. Stocia, R. Morries, D. Karger, M.F. Kaashoek, and H. Valakrishnan: Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. In *Proceedings of ACM SIGCOMM*, pages 149-160, 2001.
12. F. Emekci, O. D. Sahin, D. Agrawal, A. El Abbadi: A Peer-to-peer Framework for Web Service Discovery with Ranking. In *Proceedings of IEEE international conference on Web services*, California, USA, July 2004.
13. LTSA. http://www.doc.ic.ac.uk/ltsa/bpel4ws
14. Andreas Wombacher, Bendick Mahleko: IPSI-PF: A Business Process Matching Engine. In *Proceedings of Conference on Electronic Commerce*, California, USA, July 2004.
15. uddie. http://www.wesc.ac.uk/projects/uddie/, 2003.
16. C. Schmidt and M. Parashar: A Peer-to-peer Approach to Web Service Discovery. World Wide Web, 7(2):211–229, 2004.

17. Z. Cheng, M. P. Singh, and M. A. Vouk: Verifying Constraints on Web Service Compositions. In *Real World Semantic Web Applications*, 2002.
18. C. Molina-Jimenez, S. Shrivastava, E. Solaiman, and J. Warne: Contract Representation for Run-time Monitoring and Enforcement. In *Proceedings of International Conference on Electronic Commence*, pages 103-110, 2003.

Efficient Mining of Cross-Transaction Web Usage Patterns in Large Database*

Jian Chen, Liangyi Ou, Jian Yin, and Jin Huang

Department of Computer Science, Zhongshan University, Guangzhou, China
ellachen@gmail.com

Abstract. Web Usage Mining is the application of data mining techniques to large Web log databases in order to extract usage patterns. A cross-transaction association rule describes the association relationships among different user transactions in Web logs. In this paper, a Linear time intra-transaction frequent itemsets mining algorithm and the closure property of frequent itemsets are used to mining cross-transaction association rules from web log databases. We give the related preliminaries and present an efficient algorithm for efficient mining frequent cross-transaction closed pageviews sets in large Web log database. An extensive performance study shows that our algorithm can mining cross-transaction web usage patterns from large database efficiently.

1 Introduction

With the rapid development of e-Commerce and its increased popularity ease-use tools, the world is becoming more and more a global marketplace. But the amazing number of news, advertisements and other information of products in e-Commerce sites makes us feel it is necessary to find some new technologies that can dramatically reduce the useless information and help us sift through all the available information to find which is most valuable to us. Web usage mining is the process of applying data mining techniques to the discovery of usage patterns from Web data. Web usage mining techniques, which rely on offline pattern discovery from user transactions, can capture more fine-grained information of users' browsing behavior. One interesting information type is the Web association pattern, which describes the potential associations between the items or pages in the same user transaction. However, there is an important form of association rule, which is useful but could not be discovered by traditional association rule mining algorithm. Let us take Web user transactions database in an e-Commerce website as an example. By specifying the value of *minsupp*

* This work is supported by the National Natural Science Foundation of China (60205007), Natural Science Foundation of Guangdong Province (031558, 04300462), Research Foundation of National Science and Technology Plan Project (2004BA721A02), Research Foundation of Science and Technology Plan Project in Guangdong Province (2003C50118) and Research Foundation of Science and Technology Plan Project in Guangzhou City (2002Z3-E0017).

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 519–528, 2005.

(minimum support) and *minconf* (minimum confidence), traditional association rule mining may find the rules like:

- R_1: *80.6% of users who bought product A also bought product B. [A⇒B: (20%,80.6%)].*

where 80.6% is the confidence level of the rule and 20% is the support level of the rule indicating how frequent the rule holds.

While R_1 reflects some relationship among the pages in the same user transaction, its scope of prediction is limited; and people may be more interested in the following type of rules:

- R_2: *If the users bought product A, then at 72.8% of probability,* ***the next fourth day*** *he will bought product B. [A(1)⇒B(4): (20%,72.8%)].*

There is a fundamental different between R_1 and R_2. The classical association rule like R_1 expresses the associations among items within one user transaction (in the same day). We call them *intra-transaction association rules.* Rule R_2 represents some association relationship among the field values from different transaction records (in the different day). We call them *cross-transaction association rules* or *inter-transaction association rules* . The major advantage of cross-transactional association rules is that besides description they can also facilitate prediction, providing the user with explicit dimensional. It is often useful to know when to expect something to happen with accuracy (e.g. "four days later" or "5 miles father") instead of a fuzzy temporal window (e.g. "some day within 1 week") or a sequence (e.g. A and B will happen after C).

In order to make inter-transactional association rule mining truly practical and computationally tractable, many researchers have developed different methods for discovering inter-transaction association rules. *EH-Apriori* (Extended Hash Apriori) [1], an Apriori-like algorithm was presented by extending the notion of intra-transactional association rules to the multidimensional space. The authors also propose the use of templates and concept hierarchies as a means to reduce the large number of the produced rules. Anthony K.H. Tung [2] pointed out inter-transaction pattern is different from sequential pattern [3] because the latter treats the transactions of each customer along time as one transaction, which is essentially an intra-transaction association rule. The authors also proposed an algorithm named *FITI* (First Intra Then Inter) to discover frequent inter-transaction itemsets. FITI makes use of the property "A itemset whose extended form is a frequent inter-transaction itemset must be a frequent intra-transaction itemset", to enhance its efficiency in discovering frequent inter-transaction itemsets. A template-guided constraint-based inter-transactional association mining approach was described in [4,5]. The study the applicability of inter-transactional association rules to weather prediction in multi-station meteorological data is studied in [6].

In this paper, we will utilize the closure property of frequent itemsets to mining cross-transaction association rule aiming at discovering Web usage patterns hiding in Web logs. The rest of the paper is organized as follows: In Section 2

we firstly define the notion of cross-transaction frequent closed itemsets in domain knowledge. Section 3 gives the details about our framework for mining web cross-transaction association patterns. Experimental results are described in Section 4 along with the performance of our algorithm in a real world dataset. Finally, we summarize our research work and draw conclusions in Section 5.

2 Preliminaries

A complete statement of Web user transactions database includes a set of n pageviews $P = \{p_1, p_2, \ldots, p_n\}$ and a set of m user transactions $T = \{t_1, t_2, \ldots, t_m\}$ where each $t_i \in T$ (with a unique identifier TID) is a subset of P. *Pageviews* are semantically meaningful entities to which mining tasks are applied (such as pages or items on the Web site) and *User Transactions* is semantically meaningful groupings of pageviews in each user session. Conceptually, we view each transaction t as an l-length sequence of ordered pairs:

$$t =< (p_1^t, w(p_1^t)), (p_2^t, w(p_2^t)), \ldots, (p_l^t, w(p_l^t)) >$$

where each $p_i^t = p_j$ for some $j \in 1, \ldots, n$, and $w(p_i^t)$ is the weight associated with pageviews p_i^t in the transaction t. The weights can be determined in a number of ways. In this paper, since our focus is on association rule mining, we only use binary weights to represent existence or non-existence of pageviews access in the user transactions. Thus, a transaction can be viewed as a set of pageviews $s_t = \{p_i^t | 1 \leq i \leq l \wedge w(p_i^t) = 1\}$. Association rules capture the relationships among pageviews based on the navigational patterns of users.

For a pageviews set X, we denote its corresponding tidset as $TidSet(X)$, i.e., the set of all TIDs that contain X as a subset, $TidSet(X) = \bigcap_{x \in X} TidSet(x)$. For a TIDs set Y, we denote its corresponding pageviews set as $ItemSet(Y)$, i.e., the set of pageviews common to all the TIDs in Y, $ItemSet(Y) = \bigcap_{y \in Y} ItemSet(y)$. The support of a pageviews set X, denoted by $\sigma(X)$, is the number of transaction in which it occurs as a subset, i.e., $\sigma(X) = |TidSet(X)|$. For a predefined threshold *minimum support* $\sigma_{\min}$, We call X is *frequent* if $\sigma(X) \geq \sigma_{\min}$. A frequent pageviews set X is called *closed* if there exists no proper superset $X' \supset X$ with $\sigma(X') = \sigma(X)$. Then, we define the closure of pageviews set X in T, denote by $clo(X)$, by $ItemSet(TidSet(X)) = \bigcap_{t \in TidSet(X)} t$. For every pair of pageviews set X and Y, the following 5 properties hold [7]:

1. If $X \subseteq Y$, then $clo(X) \subseteq clo(Y)$.
2. If $TidSet(X) = TidSet(Y)$, then $clo(X) = clo(Y)$.
3. $clo(clo(X)) = clo(X)$.
4. $clo(X)$ is the unique smallest closed pageviews set including X.
5. A pageviews set X is a closed set iff $clo(X) = X$.

2.1 Basic Concepts and Terminology

Definition 1. *A **sliding window W** in a transaction database T is a block of ω continuous intervals, which starting from interval d_1 such that T contains a*

transaction at interval d_1. *Here* ω *is called the* ***span*** *of window. Each interval* d_j *in* W *is called a* ***sub_window*** *of* W *denoted as* W_u, *where* $u = d_j - d_1$, $1 \leq u \leq \omega$.

The definition of sliding window breaks the barrier of transaction and extends the scope of association rules from traditional intra-transaction to cross-transaction. The target of mining is to find out the rules which *span* less than or equal to ω intervals. The contextual properties of *span* can be time, space, temperature, latitude, and so on.

Definition 2. *Given the above* P *and* W, ***extended pageview*** *is defined as* $p_i(u) \in W_u$, *where* $p_i \in P$, $1 \leq i \leq n$, $1 \leq u \leq \omega$. *Then the* ***extended pageviews set*** *can be expressed as:*

$$EP = \{p_1(1), \ldots, p_1(\omega), \ldots, p_n(1), \ldots, p_n(\omega)\}$$

Definition 3. *When sliding window* W *starts from* k_{th} *transaction, an* ***extended transaction*** *will be generated as:*

$$et_k = \{p_i(u) | p_i \in t_j \wedge w(p_i) = 1\}$$

where $1 \leq i \leq l$, $1 \leq u \leq \omega$, $1 \leq k \leq n - \omega$, $k \leq j \leq k + \omega$.

Definition 4. ***Extended User Transactions Database***

$$ED = \{et_k | 1 \leq k \leq n - \omega\}$$

Definition 5. *A pageviews set* $CCP \subseteq EP$ *is a* ***Cross-transaction Closed Pageviews set*** *if there exists no another pageviews set* $C' \subseteq EP$, *such that*

1. C' *is a proper superset of* CCP,
2. *Every user transaction containing* CCP *also contains* C',

Property 1. If CCP is a cross-transaction closed pageviews set, and then in any sub_window W_u over CCP, $C' = \{p_i | p_i(u) \in CCP, 1 \leq i \leq m\}$ is an intra-transaction closed pageviews set.

Proof. We will prove this property by contradiction. Given the above conditions, if there exists a sub_window Wu such that C' is not an intra-transaction closed pageviews set. From the definition of closed pageviews set, there exists another pageviews set L' which make the following statements true:"$C' \subseteq L'$" and "L' is a closed pageviews set" and "$\sigma(C') = \sigma(L')$". Let $z \in L' - C'$, then its extended form $z(u) \notin CCP$. But each user transaction which contains C' also contains z, so each extended transaction which contains CCP also contains $z(u)$. Let us construct $L = CCP + \{z(u)\}$, we have: "$CCP \subseteq L$" and "$\sigma(CCP) = \sigma(L)$". This conclusion contradicts to the fact that "CCP is a cross-transaction closed pageviews set.

Property 2. The support of a cross-transaction frequent pageviews set is equal to the support of its cross-transaction closure.

Definition 6. *Given the above ED and EP, a* ***Web cross-transaction association pattern*** *is an implication expression of the form* $X \Rightarrow Y$*, which satisfies:*

1. $X \subseteq EP$, $Y \subseteq EP$, $X \cap Y \subseteq \emptyset$,
2. $\exists item_i(1) \in X$, $1 \leq i \leq m$,
   $\exists item_j(u) \in Y$, $1 \leq j \leq m$, $1 \leq u \leq \omega$, $u \neq 1$,
3. $\alpha(X \Rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)} \geq minconf$

where $minconf$ *is a predefined minimum confidence.*

2.2 An Example

Now, we take the Fig. 1 as an example to illustrate the terms given above.

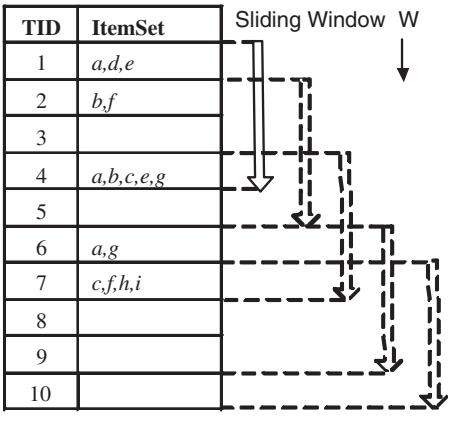


Fig. 1. The transactions database with sliding window

Then the extended user transactions database ED in Fig. 1 is as following:

$$ext\_T = \left\{ \begin{array}{l} \{a(1), d(1), e(1), b(2), f(2), a(4), b(4), c(4), e(4), g(4)\} \\ \{b(1), f(1), a(3), b(3), c(3), e(3), g(3)\} \\ \{a(1), b(1), c(1), e(1), g(1), a(3), e(3), g(3), c(4), f(4), h(4), i(4)\} \\ \{a(1), e(1), g(1), c(2), f(2), h(2), i(2)\} \\ \{c(1), f(1), h(1), i(1)\} \end{array} \right\}$$

The above user transactions database includes 5 useful transactions (we take the transaction which don't contain any items as null) and a sliding window W with $\omega = 4$. W must start from a transaction which contains at least one pageview. $a(1), b(3)$ and $h(4)$ are called as extended pageview in ED. And $ae(1) \Rightarrow c(4)$ is one of cross-transaction association rules in ED.

3 CFCPSM: Cross-Transaction Frequent Closed Pageviews Sets Miner

All association rules discovery process is composed of two main phases. The first one is finding all "*frequent*" itemsets with support higher than a user-specified *minsup* and the second one is generating all rules satisfying user-specified *minconf* from these frequent itemsets. The first phase is usually the bottleneck of the whole mining process. Moreover, in the cross-transaction association rule mining, because the boundary of transactions is broken, the number of potential "*frequent*" itemsets becomes extremely large. We provide a new efficient algorithm to mining cross-transaction frequent pageviews set by using its closure property, which can dramatically decrease the search space of algorithm. In Section 2, Property 1 shows that a pageviews set must be closed in intra-transaction if its extended form is closed in cross-transaction. It provides a different view of mining process. Instead of mining the cross-transaction patterns from extended user transactions database directly, we decompose the former phase into the following three steps.

3.1 Step 1: Finding All Intra-transaction Frequent Closed Pageviews Sets

According Property 1, we firstly discovering all intra-transaction frequent closed pageviews sets in Web log database. We use an efficient algorithm LCM ver2 [8] which is a backtracking algorithm for mining frequent closed pageview sets. LCM ver2 is a fast implementation for enumerating frequent closed itemsets, which is based on prefix preserving closure extension and the related parent-child relationship defined on frequent closed itemsets. The techniques like the frequency counting, occurrence deliver and hybrid of diffsets can reduce the practical computation time efficiently. The time complexity of LCM ver2 is theoretically bounded by a linear function in the number of frequent closed itemsets. It is a good choice for mining intra-transaction frequent closed pageviews sets.

Let *minsup* and *minconf* be 40% and 60% respectively. Then, the frequent closed pageviews sets ($FCPS$) and their respective support of the database ED in Fig. 1 will be: $ae = \{1, 4, 6\}(60\%), b = \{2, 4\}(40\%), c = \{4, 7\}(40\%), f = \{2, 7\}(40\%), aeg = \{4, 6\}(40\%)$. For the next step, all these intra-transaction frequent closed pageviews sets and their related $TidSets$ will be stored properly.

3.2 Step 2: Extending the TidSets of Pageviews Sets

With the continuous moving of the sliding window W, the old simple $TidSets$ can not reflect the current status information of intra-transaction frequent closed pageviews sets in cross-transaction. If some itemset of transaction n_1 appears in uth sub_window, in this time, sliding window W must start from transaction $n_1 - u + 1$. This situation is shown as Fig. 2:

Then, we will extended $TidSet(FCPS(u))$ to record the place of W starting from while $FCPS$ appears in sub_window W_u.

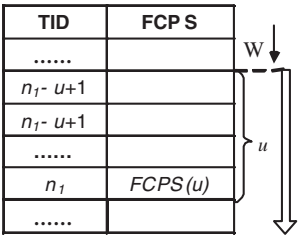


Fig. 2. W which start from Transaction $n_1 - u + 1$ takes $FCPS$ as $FCPS(u)$

Definition 7. *Given a frequent closed pageviews sets $FCPS$, its* ***extended transaction ID set*** *is:*

$$TidSet(FCPS(u)) = \{TID | FCPS(u) \in W_u \wedge W_u \in W \wedge W \ start \ from \ t_{TID}\}$$

Suppose $TidSet(FCPS) = \{n_1, n_2, \ldots, n_k\}$. When it appears in W_u, its extended $TidSet$ will be $TidSet(FCPS(u)) = \{n_1 - u + 1, n_2 - u + 1, \ldots, n_k - u + 1\}$. We take 5 efficient strategies to prune illegal TIDs from each new TidSet and remove the unreasonable TidSets:

1. If TID less than 1 or beyond the max transaction number n;
2. If TID make sliding window W starting from a null transaction;
3. If the length of the whole new $TidSet$ is shorter than $n \times minsupp$, then it cannot satisfy the support threshold;
4. Since we just have an interest in closed pageviews sets, if $FCPS_1 \subset FCPS_2$ and $TidSet(FCPS_1(u)) = TidSet(FCPS_2(u))$, then only $FCPS_2(u)$ and its $TidSet$ will be stored.
5. If $|TidSet(FCPS(u))|/n < minsupp$, then $|TidSet(FCPS(u+i))|/n < minsupp$ for any $i \geq 1$. We can stop the current window sliding which starts from $FCPS$.

3.3 Step 3: Mining All Cross-Transaction Frequent Closed Pageviews Sets

After Step 2, all TIDs in $TidSets$ have unique meanings. We obtain all extended form of intra-transaction pageviews sets and the extended user transactions database ED. We can accomplish the third step by combining several simple aggregate operations. Before doing that, we define the cross-transaction frequent closed pageviews sets as following.

Definition 8. *For each two extended intra-transaction frequent closed pageviews sets,*

$$TidSet(FCPS_1(1)) = \{m_1, m_2, \ldots, m_k\}$$

and

$$TidSet(FCPS_2(i)) = \{n_1, n_2, \ldots, n_l\} (2 \leq i \leq \omega)$$

if the total elements number of their TidSet intersection satisfies

$$|I| = |TidSet(FCPS_1(1)) \cap TidSet(FCPS_2(i))| \geq n \times minsupp$$

we call $CFPS = FCPS_1(1)FCPS_2(i)$ *is* frequent. *If CFPS also satisfies satisfies the Definition 5, it is a* closed *pageviews set.*

There are many methods to calculate the intersection of the $TidSets$ of any two $FCPS$ in different sub_window. The simplest way is calculating the intersection level-by-level while the window is sliding. But as the size of database and the maxspan increasing, the complexity of algorithm will grow geometrically. In order to avoid the bottleneck of performance, we take the closure property of frequent itemsets into account, trying to design a new algorithm to solve this problem fundamentally. We define the extended closure operation $clo(*)$ as follows:

$$\begin{aligned} & clo(FCPS_1(j_1)...FCPS_n(j_n)) \\ = & \, ItemSet(TidSet(FCPS_1(j_1)...FCPS_n(j_n))) \\ = & \, ItemSet(\bigcap_{k\in\{1,\ldots,n\}} TidSet(FCPS_k(j_k))) \end{aligned}$$

The extended closure operation $clo(*)$ satisfies Properties 1-5 as well. Hence we can construct a closed set prefix tree by closure relation. For intra-transaction frequent closed itemset $Y = \{FCPS_1(j_1)...FCPS_n(j_n)\}$ and an extended itemset $FCPS_{n+1}(j_{n+1})$, we calculate the closure of them:

$$S = clo(FCPS_1(j_1)...FCPS_n(j_n)FCPS_{n+1}(j_{n+1}))$$

We inherit the advantages of the data structure of LCM ver2. If the prefix of S, that is, $S_1(i_1)\ldots S_n(i_n)$ is equal to Y, then S is the child of Y, and it will be added to the set of cross-transaction frequent closed pageviews sets $CFCPS$. At the end of the algorithm, those patterns which are meaningless or unreasonable in real application, such as all extended items in $CFCPS$ occurring in the same sub_window, should be ignored. e.g. $a(1)b(1)c(1)$.

Thus, we get the final cross-transaction frequent closed pageviews sets $ae(1)f(2)$, $ae(1)c(4)$ and $b(1)aeg(3)$. Web association patterns $ae(1) \Rightarrow f(2)$, $ae(1) \Rightarrow c(4)$ and $b(1) \Rightarrow aeg(3)$ have 66.7%, 66.7% and 100% respectively.

4 Experimental Evaluation

All experiments were perform on Intel Pentium 4 2.6G, running Debian GNU/Linux, 512MB of main memory. All programs have been implemented in C++ Standard Template Library and use double float data type. We use the CTIdata dataset containing the preprocessed and filtered transantionized data for the main DePaul CTI Web server (http://www.cs.depaul.edu). The attributes of this dataset are shown as Table 1.

Table 1. The meanings of Parameters

Attributes	Meanings	Values
$\|T\|$	Total number of transactions	13794
$\|P\|$	Total number of pageviews	683
$average(\|t\|)$	Average length of transactions	5.04
$max(\|t\|)$	Max length of transactions	39

To show the effect of extending the notion of transactions, we vary the value of *span* and compare the number of patterns respectively. As can be seen from Fig. 3, since the notion of cross-transaction association breaks the boundaries of transactions and extend the scope of mining association rules from traditional single-dimensional, intra-transaction associations to multidimensional, cross-transaction associations, the number of potential pageviews and the number of rules will increase drastically.

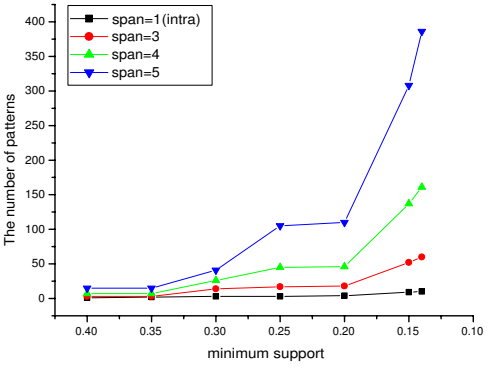


Fig. 3. Effect of increasing span

We will next investigate the how CFCPSM performs when the number of transactions in the database increases. We vary the number of transaction is dataset one from 2000 to 13794 with minimum support = 0.2. The result in Fig. 4 shows that the CPU time of CFCPSM increases linearly with the number of transactions in the database.

5 Conclusions

As the number of Web users grows, Web usage patterns which describe the hidden association information of users' browsing interest has attracted more and more attentions of researchers. Knowledge derived from the Web association patterns can be used for improving the organization of Web sites, efficient

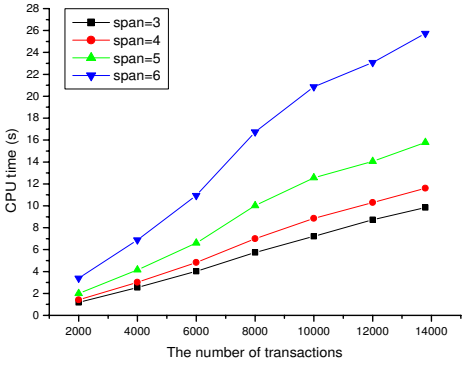


Fig. 4. Effect of increasing the number of transactions

personality and recommendation systems, and collecting business intelligence about the behavior of Web users, etc. In this paper, we provided a new different view of Web association patterns by extending the scope of it. The related definitions of properties were given and an efficient mining approach for this new form association rules was present in detailed.

References

1. H. Lu, L. Feng, and J. Han. Beyond intra-transactional association analysis: Mining multi-dimensional inter-transaction association rules. ACM Transactions on Information Systems, 18(4):423-454, 2000.
2. Anthony K.H. Tung, Hongjun Lu, Jiawei Han, and Ling Feng: Efficient Mining of Intertransaction Association Rules. IEEE Transactions on Knowledge and Data Engineering, 15(1): 43-56, 2003.
3. R. Agrawal and R. Srikant: Mining Sequential Patterns. In Proceedings of the 11th. International Conference on Data Engineering, pages 3-14, 1995.
4. L. Feng, H. Lu, J. Yu, and J. Han: Mining inter-transaction association rules with templates. In Proceedings of ACM CIKM, pages 225-233, 1999.
5. Ling Feng, Jeffrey Xu Yu, Hongjun Lu, Jiawei Han: A template model for multidimensional inter-transactional association rules. The International Journal on VLDB, 11(2): 153-175, 2002.
6. Ling Feng, Tharam S. Dillon, James Liu: Inter-transactional association rules for multi-dimensional contexts for prediction and their application to studying meteorological data. Data & Knowledge Engineering 37(1): 85-115, 2001.
7. N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In Proceedings of rhe 7th International Conference on Database Theory, pages 398-416, 1999.
8. Takiake Uno, Masashi Kiyomi, Hiroaki Arimura: LCM ver.2: Efficient Mining Algorithms for Frequent/Closed/Maximal Itemsets. In Proceedings of IEEE ICDM'04 Workshop FIMI'04.

Delay-Constrained Multicasting with Power-Control in Wireless Networks

Yuan Zhang and Bo Yang

School of Information Science and Engineering,
Jinan University, Jinan 250022, China
{yzhang, yangbo}@ujn.edu.cn

Abstract. We investigate a power-controlled transmission scheme for multicasting delay-constrained traffic in single-hop wireless networks. Particularly, we consider that packetized data arrives at the server destined for multiple clients within the transmission range. Each arrived packet needs to be transmitted to clients with a stringent delay constraint. We propose a novel formulation to capture the trade-off between transmission power and quality of service (measured by packets received within delay deadline) in this multicasting scenario. Using dynamic programming, the optimal transmission power can be obtained to provide a certain quality of service and minimize the total transmission energy. Through simulation, our proposed power-controlled multicasting scheme exhibits 20% energy savings over the standard constant SIR approach.

1 Introduction

A crucial issue in wireless networks is the trade-off between the "reach" of wireless transmission (namely the simultaneous reception by many nodes of a transmitted message) and the resulting interference by that transmission. We assume that the power level of a transmission can be chosen within a given range of values. Therefore, there is a trade-off between reaching more nodes in a single hop by using higher power (but at a higher interference cost) versus reaching fewer nodes in that single hop by using lower power (but at a lower interference cost).

[1]Another crucial issue is that of energy consumption because of the nonlinear attenuation properties of radio signals. On one hand, one wants to minimize the transmission energy consumption to prolong the battery life of mobile devices and reduce the interference effects on neighbor transmissions. On the other hand, to provide a certain quality of service (QoS) measured by packet loss rate or average delay in the erratic wireless channel, the transmission power is bounded below by the necessary signal interference ratio (SIR).

Multicasting enables data delivery to multiple recipients in a more efficient manner than unicasting and broadcasting. A packet is duplicated only when the delivery path toward the traffic destinations diverges at a node, thus helping to reduce unnecessary transmissions. Therefore, in wireless networks, where radio resources are scarce and most devices rely on limited energy supply, multicasting is a highly desirable feature.

[1] This work is supported by National 863 Program under Grant No. 2002AA4Z3240 and Key Special Science & Technology Foundation of Shandong Province.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 529 – 538, 2005.

We attempt to address the trade-off between transmission energy consumption and QoS in multicasting packetized data traffic over wireless links. To remove the burden of routing in multicasting, we only consider the single-hop case, where data traffic arrives at the server to be transmitted to several clients within one-hop transmission range. This scenario is realistic such as in wireless LAN where access point transmits videos to several media clients or in sensor networks where one individual sensor sends information to multiple neighbors to provide certain diversity.

The rest of this paper is organized as follows: Section 2 covers the related work in the literature. Section 3 presents the assumptions involved in our analysis. In section 4, we introduce a detailed system model and cast this power control problem onto a dynamic programming framework. Through simulation we evaluate the performance of our proposal and compare it with the constant SIR scheme in section 5. We also analyze the effects of some modeling parameters. Finally, section 6 summarizes our work and gives a brief statement of the future work.

2 Related Works

In the past, many studies have addressed the issue of power control in wireless environment. [1] analyzed the issues in power sensitive networks and presented a power-efficient architecture that introduces active link protection and noninvasive channel probing. [2] proposed a power-controlled multiple access scheme (PCMA) through exploring the trade-off among energy consumption, buffering cost, and delay. [3][4] further extended this PCMA algorithm in multiple-channel cases where the global power consumption is constrained. However, none of these have considered power-controlled transmission in multicasting.

The energy consumption in multicasting has also been addressed in the literature but mainly is on minimum energy routing. In these studies, establishing a minimum energy multicasting route is the main concern. [5] proposed a two-tier power controlled multicast routing algorithm for wireless ad hoc networks. With the same design philosophy [6] proposed a distributed power control scheme as a means to improve the energy efficiency of routing algorithms also for ad hoc networks. In [7], a methodology for adapting existing multicast protocols to power controlled wireless ad hoc networks was presented. A clustering scheme is used in the adaptation to be energy efficient. [8] introduced and evaluated algorithms for multicasting tree construction for infrastructureless, all-wireless applications. Although [9] minimized the total energy consumption by taking advantage of the physical layer design that facilitates the combination of partial information to obtain complete information, their approach was for broadcasting applications.

In this paper, we investigate a power-controlled transmission scheme for multicasting delay-constrained traffic in single-hop wireless networks. Particularly, we consider packetized data arrives at the server destined for multiple clients within the transmission range. Each arrived packet needs to be transmitted to clients with a stringent delay constraint. We propose a novel formulation to capture the trade-off between transmission power and quality of service (measured by packets received within delay deadline) in this multicasting scenario.

3 Assumptions

To assess the complex trade-offs one at a time, we assume in this paper that there is no mobility. Nevertheless, the impact of mobility can be incorporated into our models because transmitter power can be adjusted to accommodate the new locations of the nodes, as necessary. In other words, the capability to adjust transmission power provides considerable "elasticity" to the topological connectivity, and hence may reduce the need for hand-offs and tracking.

The interferences that clients experience are assumed to be Markovian and unresponsive. Wireless channel in nature is responsive. For example, increasing the current transmission power may increase the interference experienced by the neighbor transmission, causing the neighbor transmitter to increase its power and create more interference for the current transmission. This responsive nature complicates control. By defining the channel interference unresponsive, we assume that the current power level that the transmitter uses will not cause a noticeable effect on the same transmitter. This is relatively more acceptable in the single hop environment than in the multi-hop ad hoc networks [11, 12].

With slotted time, packets arrive in the sender following a Bernoulli distribution. Each arrived packet needs to be correctly received by all clients within a fixed deadline after its arrival. Since the acknowledgement (ACK) is short and ACK packet error rate is small, a perfect feedback channel is usually available. Given the interference level to each client, the information of the server's queue, and the status of each client indicating whether the current transmitted packet is received correctly or not, the server needs to choose the transmission power to minimize the energy consumption while providing a certain QoS captured by the number of packets received within deadline.

4 System Model and Formulation

In this section, we first provide a detailed system model. Figure 1 shows an example of the study scenario. A single server communicates with M clients. Assume time is slotted. Each client experiences different interference i_k, where $k \in \{1,2,...,M\}$, due to their distance to the server and possible obstructions in line of sight. These interferences may fluctuate from current time slot to the next independently or jointly following some Markovian properties. Define $I = \{i_1, i_2, ..., i_M\}$ to be the interference vector and $P(I'|I)$ be the transition probability for channel interference from state I to state I'. Under interference i_k and with a transmission power p, client k receives the packet successfully with probability $s(p, i_k)$. At the beginning of each time slot, the server has perfect estimate of the interference vector I for the current time slot.

At the end of each time slot, data packet arrives at the server for transmission during next time slot. Packet arrives following a *i.i.d* Bernoulli distribution with average arrival rate α. Each packet needs to be transmitted to all clients d time slots after its arrival. At time slot d+1 after the arrival, the server will drop the current packet. Assuming perfect feedback channels are available. Immediately after the transmission, the server will be notified the receiving status of each client on the transmitted packet.

Define $K \in \{0,1\}^M$, the M dimensional binary vector, to record the receiving status of each client. The transition probability, $P(K'|K)$, can be then calculated from $s(p, i_k)$.

At the beginning of each time slot, we also record the server's buffer information. Given the channel state I, receiving status K and the buffer information, the server needs to choose the transmission power to minimize the energy consumption but also to provide a certain QoS. Furthermore, the server may consider some clients more important than others, which may also affect the optimal transmission power. This trade-off between energy consumption and QoS will be captured using a d-period stochastic dynamic programming [10] described in the following two subsections.

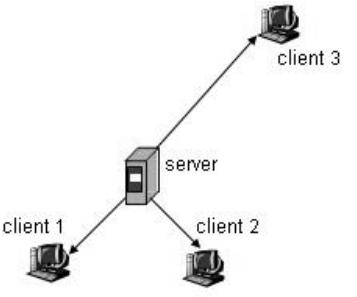


Fig. 1. Study scenario

4.1 Complete Buffer Information

In this subsection, we describe the first approach in casting the power control scheme using a d-period dynamic programming. The time period n is the time stamp for the first packet (head of buffer packet) in the server's buffer. When n reaches d+1, the first packet is dropped. We define the buffer state to record the time stamp of each packet in the buffer, as suggested in [4]. Let $B = (t_1, t_2,\ldots, t_j)$ be the buffer state, where t_k is the time stamp of the k^{th} packet in the queue. Since all packets have the same deadline and arrive in order, t_k decreases with k. Furthermore, at most one packet arrives in each time period. The total number of elements in B should be less than the time period n.

To capture the trade-off between the transmission energy consumption and the QoS, measured by the number of packets received within deadline, we define $C = (c_1, c_2,\ldots, c_m)$ be the packet missing-deadline cost vector. c_k is the cost that client k will suffer if not receiving a packet within period d. This cost may be different for each client since the server may consider some clients more important than others.

Define $V^n(B, K, I)$ be the total cost at buffer state B, receiving status of the current packet K, channel state I and time period n. Then we can formulate the dynamic programming recursion as the following:

For $n < d + 1$,

$$V^n(B,K,I) = \min \begin{pmatrix} p+\sum_{K',I'}^{p}\left(\alpha V^{n+1}(B',K',I')+(1-\alpha)V^{n+1}(B'',K',I')\right) \\ P(I'|I)P(K'|K,p,I) \end{pmatrix} \quad (1)$$

And

$$V^{d+1}(B,K,I) = (1-K)^T C + V^{n'}(B',K,I) \tag{2}$$

In equation (1), $P(K'|K,p,I)$ is obtained by using successful transmission probabilities on all individual links, $s(p, i_k)$. Buffer state B is updated to B' or B'' according to the definition. Equation (2) captures the boundary condition. When the time period reaches $d + 1$, the system suffers a packet missing-deadline cost calculated as $(1\text{-}K)^T C$ and the cost of sending the rest of packet in the buffer, indicated by $V^{n'}(B', K, I)$. $n' = b_2$ is the time stamp of the new head of buffer packet in the buffer. Equation (1) and (2) need to be solved iteratively through each time period and recursively due to the boundary condition in (2). This requires a great effort in calculation. The dimension of the buffer state B further complicates this formulation. In the next subsection, we propose a simplified model that greatly reduces calculation and state complexity and achieves approximately the same optimality.

4.2 Head of Buffer Packet Deadline Model

In the previous subsection, we introduce a buffer state to record the time stamps of each packet in the buffer. This leads to untractable complexity in the formulation. In this subsection, we only consider the deadline of the first packet in the queue instead. In order to reflect the delay constraints of other packets in the buffer, we introduce a buffer holding cost. This model may be suboptimal due to the incomplete buffer state information but simplify the complexity to a manageable level. By choosing the buffer holding cost carefully, this model performs close to the optimal solution.

Since the period n is already the time stamp of the head of buffer packet, we only need to define the number of packets in the buffer, b, as the buffer state. We also define a buffer holding cost $H(b)$, which is an increasing function of b. With large number of packets in the buffer, $H(b)$ is large. The server will increase the power to transmit the head of buffer packet successfully as early as possible. In other words, $H(b)$ models the deadline pressure of other packets in the server's buffer.

With this simplified system model, we redefine the d-period dynamic programming as the following:

For n < d + 1,

$$V^n(b,K,I) = \min \begin{pmatrix} p + H(b)\sum_{K',I'}^{p}\left(\alpha V^{n+1}(b+1,K',I') + (1-\alpha)V^{n+1}(b,K'I')\right) \\ P(I'|I)P(K'|K,p,I) \end{pmatrix} \tag{3}$$

$$V^{d+1}(b,K,I) = (1-K)^T C \tag{4}$$

And for any n,

$$V^n(b, 1, I) = 0 \tag{5}$$

Equation (3) is similar to equation (1), except we include a buffer holding cost $H(b)$ and change the buffer state b to record the number of packets in the buffer. Due to this formulation, the final period cost is only the packet missing-deadline cost shown in

equation (4). Equation (5) defines the terminal cost if all clients receive the packet correctly. With this formulation, we can solve equation (3), (4) and (5) iteratively from period $d + 1$. The optimal power $p^n(b,K,I)$ can be obtained.

We designed a simple scenario to illustrate the relationship between the optimal power and the buffer state, receiving status and channel interferences. We assume that three clients in the system (M= 3) experience different Markovian interferences independently. The interferences are shown in table I. Given interference i_k and transmission power p, client k receives packet correctly with probability $s(p,i_k) = \frac{p}{p|i_k}$.

Table 1. Interference levels and transition matrix

	Value	Transition matrix
i_1	2, 10	$\begin{pmatrix} 0.86 & 0.14 \\ 0.07 & 0.93 \end{pmatrix}$
i_2	2, 10	$\begin{pmatrix} 0.86 & 0.14 \\ 0.07 & 0.93 \end{pmatrix}$
i_3	1, 20	$\begin{pmatrix} 0.6 & 0.4 \\ 0.14 & 0.86 \end{pmatrix}$

We choose packet missing-deadline cost C = [50, 50, 20]. The server considers first two clients more important than the third client. We assume buffer holding cost $H(b)$= 2*b, deadline d=10 and packet arrival rate $\alpha = 0.5$. Figure 2 shows the relationship between time period and optimal power.

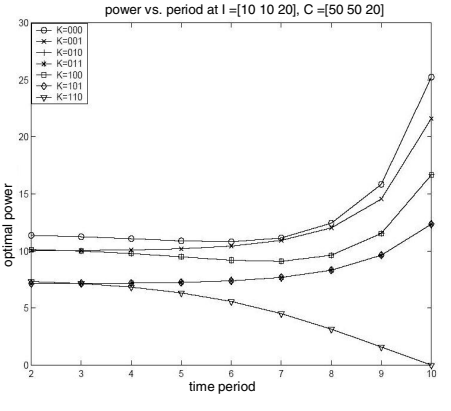


Fig. 2. Optimal power vs. time period n at channel interference I = [10, 10, 20], buffer state b=2 for different receiving status K

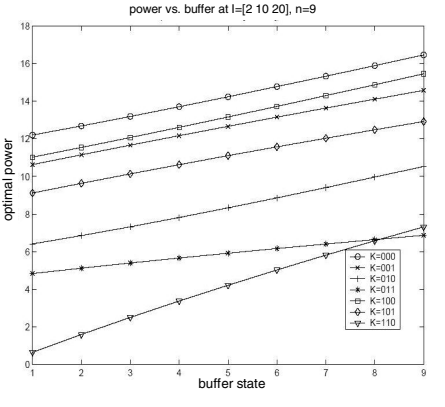


Fig. 3. Optimal power vs. buffer state b at channel interference I = [2; 10; 20], time period n = 9 for different receiving status K

From figure 2, we observe in general the optimal power increases as the time period increases to avoid the packet missing-deadline cost. However, for receiving status K = 110, which only client 3 has not received the current packet, the optimal power decreases with time period. This can be explained as follows: Due to the high channel interference for client 3, $i_3 = 20$, the power required for successful transmission is comparable with the packet missing cost $c_3 = 20$. For the time period close to deadline d+1, the small number of packets in the buffer (in this case, $b = 2$) only represents small deadline pressure because these packets may have just arrived. Then the server will only make minimum efforts to transmit the head of buffer packet to client 3. When the time period is far away from the deadline, due to high deadline pressure from other packets in the buffer, the server attempts to finish the head of buffer packet's transmission as soon as possible to avoid the missing-deadline cost of other packets in the buffer. The optimal power in this case decreases as the time period increases.

Figure 3 shows the relationship between optimal power and buffer state. The optimal power increases almost linearly with the number of packets in the buffer b due to the linear buffer holding cost function. Comparing optimal power for state K = 011 and K = 110, we observe some interesting features of optimal power. When buffer state b is small, the deadline pressure of other packets in the buffer is low. With low packet missing-deadline cost for client 3, the server will only make minimum effort to transmit the head of buffer packet at K = 110. As a result, the optimal power for K = 011 is larger even though the interference i_1 is lower. When buffer state b is large, however, the high deadline pressure of other packets requires the server to transmit the head of buffer packet even at K = 110. Then the optimal power for K = 110 is larger to compensate the high interference level.

5 Performance Evaluation

In this section, we obtain some numerical results through simulation. First, we compare the performance of our power-controlled multicasting scheme with the constant SIR scheme. The server may consider some clients more important than others, which is reflected in the packet missing-deadline cost vector C. We explore the effect of different cost ratios on number of packets received by each client. The effect of buffer holding cost $H(b)$ on number of packets received and average delay will also be discussed.

Our simulation setting is the same as we defined in the previous section, except we keep packet missing-deadline cost C and buffer holding cost $H(b)$ as varying parameters. We further impose a more stringent delay constraint by setting d=5.

5.1 Performance Comparison of SIR and PCMC

The performance is evaluated through two parameters: the total transmission energy and effective number of packets received. The effective number of packets received is calculated by considering different importance of each client.

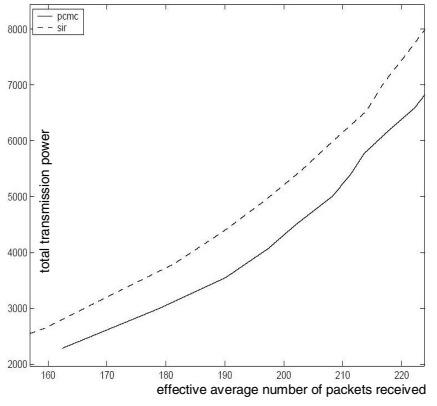


Fig. 4. Total power consumption vs. effective number of packets received for arrival rate $\alpha = 0.5$

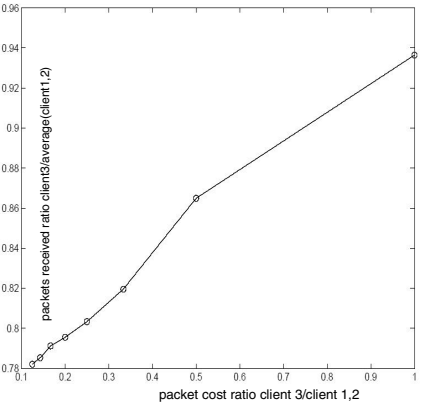


Fig. 5. Ratio of number of packets received by client 3 with client 1, 2 vs. the packet missing deadline cost ratio

Define R to be the vector recording number of packets received by each client, then the effective number is $R^T C / \sum_i c_i$, where C is the packet missing deadline cost vector, c_i is the ith element of C. We compare the following two schemes:

1) Constant SIR: This scheme simply tries to maintain the SIR for each client above some fixed threshold γ. The transmission power is calculated as $p = max_k(\gamma i_k)$. By varying the threshold γ, we obtain the power vs. effective number of packed received curve.

2) Power-Controlled Multicasting: This scheme applies our optimal power solution in section 4. We use packet missing-deadline cost $C = c[5, 5, 2]$, where c varies from 5 to 12. We also define the buffer holding cost $H(b) = \frac{2c}{d} b$.

In both schemes, we perform 100 runs with simulation length 500 time slots. We observe our proposed PCMC saves 20% transmission energy shown in figure 4.

5.2 The Effect of Packet Missing-Deadline Cost Ratio

Since the server may consider some clients more important than others, we also investigate how this relative importance, which is captured by the packet missing-deadline cost ratio, affects the number of packets each client receives. Under the same simulation setting, we define the packet missing-deadline cost C = [50, 50, 50/r], where r is the cost ratio between first two clients with client 3.

Figure 5 shows the effect on the number of packet received. As we expect, as r increases, the number of received packets ratio between client 3 and the average of client 1 and2 also increases. However since client 3 experiences much worse interference, even at r = 1, the number of received packets ratio is only 94%. As r increases, the total energy consumption also increases due to larger packet missing-deadline cost. This is demonstrated in figure 6.

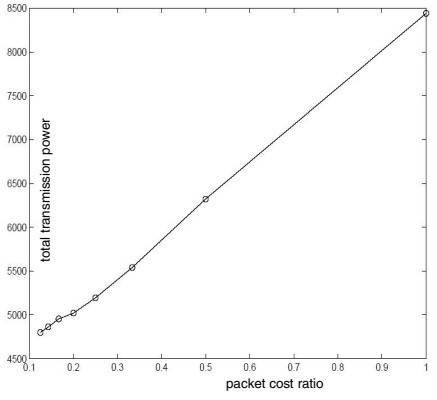


Fig. 6. Total power consumption vs. the packet missing deadline cost ratio

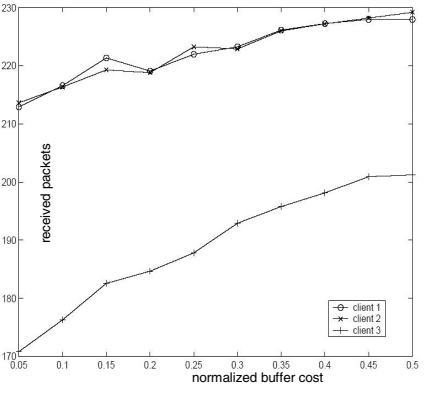


Fig. 7. Number of packets received by each client vs. normalized buffer holding cost ratio

5.3 Effect of Buffer Holding Cost

We introduce the buffer holding cost in our system model to model the deadline pressure of other packets in the server's buffer. Intuitively, as buffer holding cost increases the server will try to finish the head of buffer packet's transmission as soon as possible. We define buffer holding cost function $H(b) = h \cdot b$ be a linear function of the number of packets in the buffer.

Figure 7 shows the effect of h on the number of packets received by each client. The buffer holding cost ratio is defined as $\frac{h}{\min c_k}$, the ration of the buffer holding cost per packet. We observe as h increases and the number of packet received by each client increases. Client 3 experiences the most increase since packet missing-deadline cost c_3 is always the minimum in our simulation. Naturally, as buffer holding cost increases, the delay decreases.

6 Conclusion and Future Work

In this paper, we proposed a modeling framework for power controlled multicasting for wireless networks. We assume data traffic arrives at a server destined for multiple clients within the transmission range. Using dynamic programming approach, we devise a power controlled transmission scheme to minimize transmission energy consumption and provide a certain QoS measured by number of packet received within deadline. Through simulation, our proposed scheme presents a 20% energy savings than the constant SIR benchmark.

We realize even with our simplified model, the state complexity can still be large due to dimension of channel state and packet receiving status. We are looking into some heuristics that can further simply the problem without suffering significant performance loss. The exchanges evolved out of incremental bi-directional increase in

transmission power, though is complex and therefore difficult to model, should be considered for refinement. In a wireless video streaming scenario, each packet may have different importance to the overall quality of video. In future we would also like to include this differentiation in packet missing-deadline cost into our model.

References

[1] Bambos, N.: Toward Power-Sensitive Network Architectures in Wireless Communications: Concepts, Issues and Design Aspects. IEEE Personal Commun. Mag., Vol. 5 (June. 1998), 50-59

[2] Bambos, N., Kandukuri, S.,: Power Controlled Multiple Access in Wireless Communication Networks. IEEE INFOCOM 2000, 386-395.

[3] Bambos, N., Kandukuri, S.,: Globally Constrained Power Control Across Multiple Channels in Wireless Packet Networks. ACM Mobile Networks, Vol. 6 (Aug. 2001), 427-434

[4] Kandukuri, S. Bambos, N.: Multi- Channel Power Control for Data Traffic in Wireless Networks. Proc. of IEEE International Workshop on Mobile Multimedia Communications (Nov. 1999), 83-92.

[5] Ryu, J.-H., Song, S.-H., Cho, D.-H.: A Power-Saving Multicast Routing Scheme in 2-tier Hierarchical Mobile Ad-hoc Networks. IEEE VTC2000, Vol. 4 (Sep. 2000), 1974-1978

[6] Bergamo, P., Giovanardi, A., Travasoni, A., Maniezzo, D., Mazzini, G., Zorzi, M.: Distributed Power Control for Energy-efficient Routing in Ad Hoc Networks. Wireless Networks, Vol. 10 (Jan. 2004), 29-42

[7] Tang, C.-M., Raghavendra, C.-S.: Energy Efficient Adaptation of Multicast Protocols in Power Controlled Wireless Ad Hoc Networks. Mobile Networks and Applications, Vol. 9 (Aug. 2004), 311-317

[8] Wieselthier, J.-E., Nguyen, G.-D., Ephremides, A.: Energy-Efficient Broadcast and Multicast Trees in Wireless Networks. Mobile Networks and Applications, Vol. 7 (Dec. 2002), 481-492

[9] Agarwal, M., Cho, J.-H., Gao, L.-X., Wu, J.: Energy Efficient Broadcast in Wireless Ad Hoc Networks with Hitch-hiking. IEEE INFOCOM 2004, Vol. 23 (Mar. 2004), 2097-2108

[10] Bertsekas, D.: Dynamic Programming. Prentice Hall (1987)

[11] Krunz M., Muqattash A., Lee S.-J.: Transmission Power Control in Wireless Ad Hoc Networks: Challenges, Solutions, and Open Issues. IEEE Network Magazine, Vol. 18 (Sep. 2004), 8-14

[12] Gerharz M., Waal C., Frank M., Martini P.: Influence of Transmission Power Control on the Transport Capacity of Wireless Multihop Networks. IEEE PIMRC2004, (Sep. 2004), 78-83

Distributed Hierarchical Access Control for Secure Group Communications

Ruidong Li, Jie Li, and Hisao Kameda

Graduate School of Systems and Information Engineering,
University of Tsukuba, Tsukuba Science City, 305-8573, Japan
Tel/Fax: 0081-29-853-5156
lrd@osdp.is.tsukuba.ac.jp, {lijie,kameda}@cs.tsukuba.ac.jp

Abstract. Hierarchical access control to ensure multiple levels of access privilege for group members is required in many environments, such as hierarchically managed organizations and multimedia applications. In this paper, to efficiently and effectively achieve this goal, we propose a distributed key management scheme whereby each SG (Service Group) maintains an SG server. This server is utilized to manage the key tree and provide the related session keys for all the users in this SG. Compared with the already existing method employing an integrated key graph to the hierarchical access control problem, there is no complex merging key tree algorithm needed in the proposed scheme, and thus the communication overhead can be greatly reduced. Also the trust and communication burden on one centralized server, KDC (Key Distribution Center), is scattered, and thus better scalability when the number of users increases can be achieved.

1 Introduction

Group communication is an internetwork service that provides efficient delivery of data from a source to multiple recipients. Many emerging network applications are based upon the group communication model, such as multiparty videoconferencing and real-time push-based information delivery systems. But multicast suffers from many problems stemming from the inherent complexity of routing packets to a large group of receivers. Security is one of the most important issues in multicast environment.

Access control is a mechanism to enable each user to determine/obtain the same session key (SK) without permitting unauthorized users to do likewise and securely update keys to prevent the leaving/joining user from accessing the future/prior communications, which is referred to as forward and backward secrecy [7]. For multicast, how to provide the access control plays the crucial role in providing security service, which is achieved by the key management scheme. A prominent scheme, the hierarchical key tree scheme, has been recommended in [11, 12].

In practice, many group applications contain multiple related data streams and have the members with various access privileges [8]. This kind of applications always occur in hierarchically managed organizations and the multimedia applications, such as the military group communications, video broadcast including normal TV and HDTV. It is definite that the simplest way that uses only one hierarchical key tree to solve multicast

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 539–548, 2005.

key management is not suited for such a condition. A novel access control mechanism supporting multi-level access privilege is under development, which is referred to as hierarchical access control. Hierarchical access control is to provide access control which can assure that group members can subscribe different data streams or possibly multiple of them.

Till now, there are two categories of methods provided to achieve hierarchical access control. One is independent key tree scheme (IKTS) ([1]-[7], [9, 10]) and the other is multi-group key management scheme (MKMS) [8].

In this paper, we propose a distributed key management scheme (DKMS) to solve the hierarchical access control problem. In the proposed DKMS scheme, each service group, which is a set of users who share the same access privilege and receive exactly the same set of data streams, maintains one service group key server. The server is used to manage keys for all users in this service group. The proposed DKMS can achieve forward and backward secrecy [7]. It does not require the complex merging tree algorithm, and its communication overhead will be reduced greatly from that of MKMS. In addition, the trust and the storage burden over the centralized KDC (Key Distribution Center) which is used to manage keys in the system has been distributed to many service group servers.

The rest of the paper is organized as follows. In Sect. 2, system descriptions for hierarchical access control are provided. The related works are given in Sect. 3. We propose a distributed key management scheme in Sect. 4. In Sect. 5, we provide detailed performance analysis. Finally, we conclude our work in Sect. 6.

2 System Descriptions

We consider a multimedia distributed system consisting of a set of data groups and a set of service groups. A Data Group (DG) is defined as a set of users who receive the same single data stream. The DGs are denoted by $D_1, D_2, \ldots, D_M$, where M is the total number of the DGs. A Service Group (SG) is defined as a set of users who have the same access privilege. SGs are denoted by $S_1, S_2, \ldots, S_I$, where I is the total number of SGs. Each SG is associated with an 1-by-M binary vector . In particular, the SG S_i is associated with $V_i = [t_1^i, t_2^i, \ldots, t_M^i]$ and $t_m^i = 1$ only when the users in the SG S_i subscribe the DG D_m. Figure 1 provides an example of typical hierarchical scenario in which there are 4 SGs and 4 DGs.

In order to achieve access control for group communication, the data streams are encrypted by the session key (SK). SK may change with time [9]. That is, to protect SK from being released to adversary, it is also necessary to periodically renew SK. There are many methods to manage keys to protect SK, and the most prominent proposal is logic key tree method [7], which is the fundament for the researches in this area. In this paper, K_m^D denotes a data group (DG) key and K_i^S denotes a service group (SG) key.

To achieve hierarchical access control, when a user switches from SG S_i to SG S_j, it is necessary to

1. update the session keys of $\{D_m, \forall m : t_m^i = 1 \; and \; t_m^j = 0\}$, such that the users who switch SGs cannot access the previous communications in those DGs contained in that SG, i.e., ensure the backward secrecy [7];

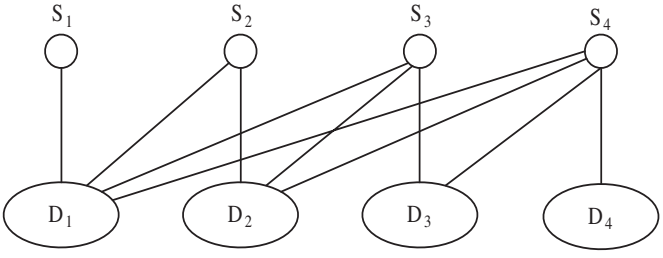


Fig. 1. A scenario containing 4 SGs and 4 DGs

2. and update the session keys of $\{D_m, \forall m : t_m^i = 0 \, and \, t_m^j = 1\}$, such that the users who switch SGs cannot access the future communications in those DGs contained in that SG, i.e., ensure the forward secrecy [7].

3 Related Work

An interesting scheme called independent key tree scheme (IKTS) is proposed in [1]-[7] and [9, 10]. By IKTS, a separate key tree is constructed for each DG, with the root being the data group key and the leaves being the users in this DG. The main advantage of IKTS is its simplicity in implementation and group management. But there is overlap between the different key trees, and thus IKTS brings the redundancy in the key trees belonging to different DGs.

To reduce such redundancy, the multi-group key management scheme (MKMS) has been proposed in [8]. By MKMS, firstly, SG-subtree is constructed for each SG with the leaves being the users and DG-subtree is constructed for each data group with the leaves being the SG keys. Then they are merged into one integrated key graph. MKMS is a good mechanism to achieve hierarchical access control. The merging key tree step is complex, because there are complex relations between data group key (K_m^D) and service group key (K_i^S) in many cases. For example, the relations between SGs and DGs are of full combination. Another arising problem in [8] is that each rekey message will be broadcast to all the users even in the group who cannot decrypt it and actually do not need it. Thus a redundancy for sending rekey messages will incur additional communication overhead. Also there are auxiliary keys in the DG-subtree, which will also bring more communication overhead and storage overhead to users. Since MKMS is obviously better than IKTS, we will only consider MKMS when giving the performance comparison in this paper.

4 Proposed Distributed Key Management Scheme (DKMS)

4.1 Structure for DKMS

To solve the above problems of MKMS [8], in this paper, we propose a distributed key management scheme (DKMS) in which each SG maintains an SG key server. This

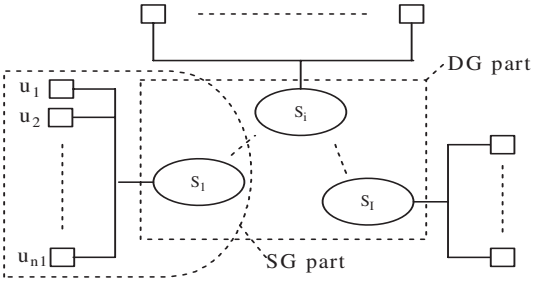


Fig. 2. Structure for DKMS

server in DKMS is utilized to manage the key tree and provide the related SKs for all the users in this SG.

The proposed DKMS structure includes two kinds of parts as depicted in Fig. 2: DG part which is used to manage SG servers, and SG part which is used to manage users who subscribe to an SG as described in Fig. 2.

The DG part is composed of all the SG servers. For example, in Fig. 2, DG part includes $\{S_1$ server, ..., S_i server, ..., S_I server$\}$. The SG part includes an SG server and all users who subscribe to that SG. For example, the SG part symbolized in Fig. 2 includes $\{S_1$ server,$u_1, u_2, \ldots, u_{n1}\}$.

4.2 Structure Construction for DKMS

The structure construction for DKMS includes 3 steps as follows.

Step 1: DG part construction. In this step, an SG server group (SGSG) constituting all SG Servers is constructed. One multicast address and one multicast key are assigned to the SGSG. At the same time, one SG key K_i^S is allotted to each SG server. Also the related SKs should be given to related SG servers during DG part construction.

Step 2: SG part construction. In this step, for each SG S_i, an SG-subtree having the root being associated with an SG key, K_i^S, and the leaves being the users in S_i is constructed. Also one multicast address is assigned to each SG.

Step 3: Combination. In this step, simply combine these two kinds of groups by connecting the SG keys to the roots of SG-subtrees.

After the structure construction and key distribution in DKMS, each user holds a set of keys which include the keys on the path from the leaf user key to the root SG key in the key tree and the needed SKs which are provided by SG server. Each SG server holds a key tree constructed for the users under her service and the needed SKs and an SGSG multicast key.

4.3 Operations for DKMS

When a user wants to switch from S_i to S_j $(i \neq j)$, the system will perform leave and join operations. Here we will describe leave operation and join operation. Without loss

of generality, we will provide the operations that a user, u_k, leaves S_i and u_k joins S_i respectively.

Leave Operation. When a user, u_k, wants to leave SG S_i, the operation will be carried out as follows.

Step 1: After SG server i knows that u_k are leaving the SG, it will update the keys from the leaf to the root in the key tree for this SG.

Step 2: SG servers update the related SKs that is held by users in SG S_i in DG part. SG server i negotiates with other SG group servers with the new SKs, and then multicast the new SKs encrypted by SGSG key (MK) to SG servers via the SGSG multicast address.

Step 3: SG servers perform necessary update on the related SKs in related SG parts. The SG servers who need to update the SKs multicast the new SKs to their users encrypted by the SG keys via the specific SG multicast address. Here SG S_i encrypts the new SKs by the new SG key.

Join Operation. When a user, u_k, wants to join SG S_i, the operation will be carried out as follows.

Step 1: SG server i update the key tree. Firstly, SG server i chooses a leaf position on the key tree to put the joining user. Then SG server i updates the keys along the path from the new leaf to the root by generating the new keys from the old keys using a one-way function. Also SG server i should send rekey messages for the new generated keys to the related users.

Step 2: After the user tree has been updated, the related SKs included in SG S_i should also be updated. This step is similar to the Step 2 plus Step 3 in the leave operation.

5 Performance Analysis

5.1 Performance Metrics

We consider the performance metrics for MKMS and DKMS provided as follows.

- Storage overhead at KDC server or other servers, denoted by R_{SER} and defined as the expected number of keys stored at servers.
- Storage overhead of users, denoted by $R_{u \in S_i}$ and defined as the expected number of keys stored by the users in the SG S_i.
- Rekey overhead denoted by C_{ij} and defined as the expected number of rekey messages transmitted by servers when one user switching from SG S_i to SG S_j.
- Communication Overhead of the network, denoted by TC_{ij} and defined as the expected bandwidth consumed when one user switching from SG S_i to SG S_j.

Here the first three metrics have been considered in [8]. For considering the network configuration, we investigate one more metric, communication overhead.

5.2 Storage Overhead

We denote by $f_d(n)$ the length of the branches and by $r_d(n)$ the total number of keys on the key tree when the degree of the key tree is d and there are n users to accommodate. $f_d(n)$ is either L_0 or $L_0 + 1$, where $L_0 = \lceil \log_d n \rceil$. At the same time, the total number of keys on a key tree can be given as below.

$$r_d(n) = n + 1 + \frac{d^{L_0} - 1}{d - 1} + \lceil \frac{n - d^{L_0}}{d - 1} \rceil. \tag{1}$$

In addition, we use $n(S_i)$ to denote the number of users in the SG S_i and $n(D_m)$ to denote the number of users in DG D_m. We will discuss the storage overhead for servers and users respectively as below.

Storage Overhead for Servers. Firstly, we consider storage overhead on the KDC by MKMS. The number of keys to be stored in SG-subtrees can be calculated as $\sum_{i=1}^{I} E[r_d(n(S_i))]$, and that in DG-subtrees is not more than $\sum_{m=1}^{M} E[r_d(c_m)]$. Therefore, the storage overhead at KDC server when using MKMS is

$$R_{SER}^{MKMS} = \sum_{i=1}^{I} E[r_d(n(S_i))] + e_1, (0 \le e_1 \le \sum_{m=1}^{M} E[r_d(c_m)]), \tag{2}$$

where $c_m = \sum_{i=1}^{I} t_m^i$, which is the number of leaf nodes in DG-subtree. $E[x]$ is used to denote the expectation of a variable, x, in this paper. The results for MKMS obtained in this paper are similar to those given in [8].

The storage overhead of all the servers when using DKMS can be calculated as follows:

$$R_{SER}^{DKMS} = \sum_{i=1}^{I} E[r_d(n(S_i))] + \sum_{i=1}^{I} \sum_{m=1}^{M} t_m^i + I. \tag{3}$$

Without loss generality, we demonstrate the storage overhead of MKMS and DKMS in the applications containing multiple layers as illustrated in Fig. 1. In this case, we can get $t_i^m = 1$ for $m \le i$ and $t_i^m = 0$ for $m > i$. Thus, $\sum_{m=1}^{M} t_m^i = i$ and $\sum_{i=1}^{I} t_m^i = M - m + 1$. We further assume that the number of users in each SG is the same, denoted by $n(S_i) = n_0$. In such environment, $I = M$.

Under such a condition, using (2) and (3), the storage overhead of the server(s) will be:

$$R_{SER}^{MKMS} = M \cdot E[r_d(n_0)] + e_{10}, (0 \le e_{10} \le \sum_{m=1}^{M} E[r_d(M - m + 1)]), \tag{4}$$

$$R_{SER}^{DKMS} = M \cdot E[r_d(n_0)] + \sum_{i=1}^{I} i + I. \tag{5}$$

For study of the scalability for the schemes, we consider the situation that the group size is large, i.e. $n_0 \to \infty$. Under such condition, the above equation implies that:

$$R_{SER}^{MKMS} \sim O(\frac{M \cdot d \cdot n_0}{d-1}), R_{SER}^{DKMS} \sim O(\frac{M \cdot d \cdot n_0}{d-1}). \tag{6}$$

Storage Overhead for a User. Similarly, we calculate the storage overhead of a user when applying MKMS by adding the number of keys to be stored by a user in SG-subtree and DG-subtree. We can get

$$R_{u \in S_i}^{MKMS} = E[f_d(n(S_i))] + e_2, (0 \le e_2 \le \sum_{m=1}^{M} t_m^i (E[f_d(c_m)] + 1)). \tag{7}$$

We calculate the storage overhead of a user when using DKMS by adding the number of keys to be stored by a user in SG-subtree and the needed SKs. We can obtain:

$$R_{u \in S_i}^{DKMS} = E[f_d(n(S_i))] + \sum_{m=1}^{M} t_m^i. \tag{8}$$

Obviously, we obtain

$$R_{u \in S_i}^{MKMS} \ge R_{u \in S_i}^{DKMS}, \tag{9}$$

because there is no DG-subtree by DKMS. That is, the auxiliary keys on DG-subtree that should be stored by users in MKMS are not needed to be stored by users in DKMS. At the same time, all the keys stored by users in DKMS should also be stored by users in MKMS.

Similarly as the discussion for the storage overhead for servers, we analyze storage overhead for a user in a multi-layer scenario with $n(S_i) = n_0$ as in Figure 1. Using (7) and (8), we can obtain the storage overhead of a user, which is given as follows:

$$R_{u \in S_i}^{MKMS} = E[f_d(n_0)] + e_{20}, (0 \le e_{20} \le \sum_{m=1}^{M} t_m^i (E[f_d(M-m+1)] + 1)), \tag{10}$$

$$R_{u \in S_i}^{DKMS} = E[f_d(n_0)] + i. \tag{11}$$

Therefore,

$$R_{u \in S_i}^{MKMS} \sim O(\log_d n_0), R_{u \in S_i}^{DKMS} \sim O(\log_d n_0). \tag{12}$$

Note: From the above discussion, we see that the proposed DKMS scheme has the storage overhead at the same order as what is needed in MKMS, whether for the servers or for one user. In the mean time, it is found that the storage overhead for a user can be reduced by DKMS because there is no DG-subtree in DKMS.

5.3 Rekey Overhead

Here we do not specify the user dynamic behavior, and calculate the amount of rekey messages transmitted by servers when one user switches from S_i to S_j, denoted by C_{ij}.

Similarly as in [8], for MKMS, the rekey overhead will be:

$$C_{ij}^{MKMS} = d \cdot f_d(n(S_i)) + e_3, (0 \leq e_3 \leq \sum_{m=1}^{M} (max(t_m^i - t_m^j, 0) \cdot (d \cdot f_d(c_m) + 1) + t_m^i \cdot t_m^j \cdot d \cdot f_d(c_m)) + 1). \quad (13)$$

When a user switches from S_i to S_j and $i \neq j$, by DKMS, rekey overhead is obtained as (14). Due to the limited space, the calculation procedure is omitted.

$$C_{ij}^{DKMS} = d \cdot f_d(n(S_i)) + e_4, (0 \leq e_4 \leq \sum_{m=1}^{M} max(t_m^i - t_m^j, 0) + \sum_{m=1}^{M} (max(t_m^i - t_m^j, 0) \cdot \sum_{i=1}^{I} t_m^i) + 1). \quad (14)$$

Similarly as in Sect. 5.2, we analyze rekey overhead in a multi-layer scenario with $n(S_i) = n_0$. Using (13) and (14), it is obtained as following.

$$C_{ij}^{MKMS} = d \cdot f_d(n_0) + e_{30}(0 \leq e_{30} \leq \sum_{m=1}^{M} (max(t_m^i - t_m^j, 0) \cdot (d \cdot f_d(M - m + 1) + 1) + t_m^i \cdot t_m^j \cdot d \cdot f_d(M - m + 1)) + 1), \quad (15)$$

$$C_{ij}^{DKMS} = d \cdot f_d(n_0) + e_{40}(0 \leq e_{40} \leq \sum_{m=1}^{M} max(t_m^i - t_m^j, 0) + \sum_{m=1}^{M} (max(t_m^i - t_m^j, 0) \cdot \sum_{i=1}^{I} t_m^i) + 1). \quad (16)$$

When the group size is large, i.e. $n_0 \to \infty$, the above equation tells that:

$$C_{ij}^{MKMS} \sim O(d \cdot \log_d n_0), C_{ij}^{DKMS} \sim O(d \cdot \log_d n_0). \quad (17)$$

Note: That is, the rekey overhead for DKMS is similar to that of MKMS when one user switching from S_i to S_j.

5.4 Communication Overhead

Some assumptions will be given as follows. Firstly, we assume that the mean communication overhead for one rekey message for one user via multicast by MKMS will be C_0, which is also assumed to be the same as that by DKMS. At the same time, we assume

that the mean communication overhead for one rekey message for one user via unicast will be C_1. Apparently, in group communication, $C_1 \geq C_0$.

We also assume that the number of users in each service group is the same: n_0, and the total number of users who subscribe the service is N. Here we also do not consider the member dynamic behavior and only calculate the total communication overhead of switching from SG S_i to SG S_j.

We calculate the communication overhead in the following step. Firstly, multiply the number of multicast messages, which has been calculated in the Sect. 5.3, with C_0 and multiply the number of unicast messages, which has also been calculated in the Sect. 5.3, with C_1. Then add these two values to get the communication overhead. Further, we assume $C_0 = 1$. Thus, $C_1 = h$.

Similarly as in Sect. 5.2, we analyze communication overhead in a multi-layer scenario with $n(S_i) = n_0$. We can obtain:

$$TC_{ij}^{MKMS} = M \cdot n_0 \cdot d \cdot f_d(n_0) + e_{50}(0 \leq e_{50} \leq M \cdot n_0 \cdot (\sum_{m=1}^{M}(max(t_m^i - t_m^j, 0) \cdot (d \cdot f_d(M - m + 1) + 1) + t_m^i \cdot t_m^j \cdot d \cdot f_d(M - m + 1)) - 1) + 2 \cdot h), \quad (18)$$

$$TC_{ij}^{DKMS} = (d \cdot f_d(n_0) - 1) \cdot n_0 + e_{60}(0 \leq e_{60} \leq I \cdot \sum_{m=1}^{M} max(t_m^i - t_m^j, 0) + \sum_{m=1}^{M}(max(t_m^i - t_m^j, 0) \cdot \sum_{i=1}^{I}(t_m^i \cdot n_0)) + 2 \cdot h). \quad (19)$$

When the group size is large, i.e. $n_0 \to \infty$, the above equation implies that:

$$TC_{ij}^{MKMS} \sim O(M \cdot d \cdot n_0 \cdot \log_d n_0), \quad (20)$$

$$TC_{ij}^{DKMS} \sim O(d \cdot n_0 \cdot \log_d n_0). \quad (21)$$

Note: From (20) and (21), we see that DKMS can reduce the communication overhead greatly compared with MKMS.

From above performance analysis, we can see that the storage overhead of each user can be reduced. Additionally, DKMS can achieve better performance than MKMS on the communication overhead. The results are summarized as Table 1.

Table 1. Results Summarization

Metrics	MKMS	DKMS
R_{SER}	$O(\frac{M \cdot d \cdot n_0}{d-1})$	$O(\frac{M \cdot d \cdot n_0}{d-1})$
$R_{u \in S_i}$	$O(\log_d(n_0))$	$O(\log_d(n_0))$
C_{ij}	$O(d \cdot \log_d(n_0))$	$O(d \cdot \log_d(n_0))$
TC_{ij}	$O(M \cdot d \cdot n_0 \cdot \log_d(n_0))$	$O(d \cdot n_0 \cdot \log_d(n_0))$
NOTE	$R_{u \in S_i}^{MKMS} \geq R_{u \in S_i}^{DKMS}$	

6 Conclusions

In this paper, we proposes a distributed key management scheme to achieve hierarchical access control in secure group communications. Compared with multi-group key management scheme proposed in [8], the main advantages of our scheme are summarized as follows. 1. Because there is no DG-subtree in DKMS, there is no complex merging key tree algorithm in our scheme. 2. The communication overhead can be greatly reduced because the rekey messages broadcast can be restricted to the users in the related SGs. This advantage is due to the fact that when multicast is employed, a message is sent to all the users in the group, regardless of whether or not all the users need that message. 3. The storage overhead of each user is reduced for the reason that it is not necessary to store some auxiliary keys in DG-subtree. 4. The system will be more robust, because the trust on one centralized server, KDC, is shared by more servers. 5. Also the better scalability can be achieved by our scheme.

Acknowledgment

The authors would like to thank Prof. Yan Sun for her kind help. This work is supported by JSPS under Grand-in-Aid for Scientific Research.

References

1. S. Banergee and B. Bhattacharjee, "Scalable Secure Group Communication over IP Multicast", *JSAC Special Issue on Network Support for Group Communication*, vol. 20, no. 8, pp 1511-1527, Oct. 2002.
2. R. Canetti, J. Garay, G. Itkis, D. Miccianancio, M. Naor, and B. Pinkas, "Multicast Security: A Taxonomy and Some Efficient Constructions", *Proc. IEEE INFOCOM'99*, vol. 2, pp. 708-716, March 1999.
3. G. H. Chiou and W. T. Chen, "Secure Broadcasting Using The Secure Lock", *IEEE Trans. Software Eng.*, vol 15, pp. 929-934, Aug 1989.
4. S. Mittra, "Iolus: A Frame for Scalable Secure Multicasting", *Proc. ACM SIGCOMM'97*, pp.277-288, 1997.
5. M.J. Moyer, J. R. Rao, and P. Rohatgi, "A Survey of Security Issues in Multicast Communications", *IEEE Network*, vol. 13, no 6, pp. 12-23, Nov.-Dec. 1999.
6. A. Penrig, D. Song and D. Tygar, "ELK, A New Protocol for Efficient Large-group Key Distribution", *Proc. IEEE Symposium on Security and Privacy*, pp 247-262, 2001.
7. S. Rafaeli and D. Hutchison, "A Survey of Key Management for Secure Group Communication", *ACM Computing Surveys*, vol. 35, no. 3, pp 309-329, Sept. 2003.
8. Y. Sun and K. J. Ray Liu, "Scalable Hierrarchical Access Control in Secure Group Communications", *Proc. IEEE INFOCOM'04*,Hong Kong,Mar. 2004.
9. W. Trappe, J. Song, R. Poovendran, and K. J. R. Liu, "Key Distribution for Secure Multimedia Multicasts via Data Embedding", *Proc. IEEE ICASSP'01*, pp. 1449-1452,May 2001.
10. M. Waldogel, G. Caronni, D. Sun, N. Weiler, and B. Plattner, "The VersaKey Framework: Versatile Group Key Management", *IEEE Jounal on selected areas in communications*, vol. 17, no 9, pp. 1614-1631, Sept. 1999.
11. D. Wallner, E. Harder and R. Agee, "Key Management for Multicast: Issues and Architecture", *Internet Draft Report*, Sept. 1998,Filename: draft-wallner-key-arch-01.txt
12. C. Wong, M. Gouda, and S. Lam, "Secure Group Communications Using Key Graph", *IEEE/ACM Trans. On Networking*, vol.8, pp. 16-30, Feb. 2000.

Hierarchical Multicast Tree Algorithms for Application Layer Mesh Networks*

Weijia Jia[1], Wanqing Tu[1], and Jie Wu[2]

[1] Deprtment of Computer Science, City University of Hong Kong,
83 Tat Chee Ave. Hong Kong, China
itjia@cityu.edu.hk
[2] Department of Computer Science and Engineering,
Florida Atlantic University, Boca Raton, F133431, USA

Abstract. This paper proposes a set of novel multicast algorithms for *m*-D mesh overlay networks that can achieve shorter multicast delay and less resource consumptions. In contrast to previous approaches, our algorithms partition the group members into clusters in the lower layer, seeking an *optimal* core (root) to guarantee the minimum routing delay for each cluster and building a shared tree within each cluster to minimize the number of links used. In the upper layer, a shared tree is then constructed using our algorithms to implement the inter-cluster routing. The extended simulation results indicate that the application layer multicast that is constructed by our algorithms is efficient in terms of routing delay and link utilizations as compared with other well-known existing multicast solutions.

1 Introduction

Multicast function was originally implemented in the network layer [1]. In recent years, the *application layer multicast* is considered as an alternative multicast function in the overlay network (i.e. the application layer) by many researchers [2-9] for the following attractive features: 1) no requirement for multicast support in the network layer of OSI reference model; 2) no need to allocate a global group id, such as IP multicast address; and 3) data is sent via unicast which enable flow control, congestion control and reliable delivery services that are available for the unicast can also be employed in the application layer multicast.

Generally, the overlay topologies for the application layer multicast fall into two categories: (1) Topologies consisting of a single tree [3,10-11]; (2) Abstract coordinate spaces obtained from *m*-D Cartesian coordinates on an *m*-torus [5, 12-13]. Such abstract coordinate space is a mesh from which members are assigned the logical addresses. A drawback of using a single tree is that the failure of a single application may cause a partition of the topology. The advantage of building the overlay mesh network is that the next-hop routing information can be encoded in the logical

* This work is supported by Strategy Grant of City University of Hong Kong under nos 7001709 and 7001587 and partially by the National Grand Fundamental Research 973 Program of China under Grant No.2003CB317003.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 549 – 559, 2005.

addresses for the good choice of address space and topology. It shows that the robust communications of the application layer multicast built in the mesh overlay network.

Many well-known multicast schemes based on the mesh network have been presented. Double-Channel XY Multicast Wormhole Routing (DCXY) [14] uses an extension of the XY routing algorithm to set up the routing scheme. Dual-Path Multicast Routing (DPM) [15] is developed for the *2*-D mesh. It assigns a label l for each node in the mesh and partitions the group into two subgroups (i.e. g_h and g_l) such that they are composed of the members with their l greater (g_h) or less (g_l) than the label of the source respectively. The routing paths are constructed through connecting the nodes covered by g_h in the ascending order of the l value and the nodes covered by g_l in the descending order of the l value. CAN-based multicast [5] is developed for the P2P applications that utilize the CAN (Content-Addressable Network) [16] configuration. CAN-based multicast is scalable especially when multiple sources coexist. However, only flooding approach is used for propagating the multicast messages which compromises the efficiency in terms of multicast delay and consumes a large number of network links. We will give the performance comparisons of these well-known multicast solutions with our multicast scheme in Section 3.

Our motivation is to design an application layer multicast scheme in *m*-D mesh overlay networks that can achieve shorter multicast delay and less resource consumptions. The network is partitioned into clusters in terms of some regular mesh area (the issue is omitted due to space limit). After group members are initially scattered into different clusters, a tree is built to connect the cluster members within each cluster. The connection among different clusters is done through hooking the tree roots. To construct such architecture, a set of novel algorithms based on the *m*-D mesh networks are presented: (1) *cluster formation algorithm* that partitions the group members with the "closeness" relationship in terms of *static delay distance* into different clusters; (2) o*ptimal core selection algorithm* that can seek the *optimal* core (i.e. root) for a shortest path cluster tree using the minimum sum of *static delay distances* to all cluster members as the metric; (3) *weighted path tree generation algorithm* that may maximize the link usage (i.e., using the minimum number of links) for creating the shortest path tree to reliably route the multicast message and (4) *multicast routing algorithm* that efficiently dispatches the multicast packets in the group based on the architecture constructed by above three algorithms. Our solution is suitable for both logical address *m*-torus and *m*-D (abstract or physical) mesh networks. To set up such shortest path tree, we apply a heuristic approach to reduce the number of links used so as to utilize the resource effectively. To avoid confusion, we wish to point out that we do not seek the *optimal* multicast tree; instead, we seek the *optimal* core for a cluster of members based on the total *static delay distance*.

The paper is structured into four sections: Section 2 discusses the algorithms for cluster formation, seeking of the *optimal* core(s) for a cluster of nodes, multicast tree generation and routing. Performance results are demonstrated in Section 3 and we conclude the paper with some discussions in the final section.

2 Algorithms for Multicast Architecture and Routing

Denote the multicast group with *n* members as $G=\{u_0,\ldots,u_i\ldots,u_{n-1}\}$, $i\in[0,n-1]$. Suppose the group members are mapped into an *m*-D mesh network by some P2P scheme.

Each member u_i can be identified by m coordinates: $(U_{i,0},\ldots,U_{i,j},\ldots,U_{i,(m-1)})$, where $0 \le U_{i,j} \le k_j-1$ and $0 \le j \le m-1$. End hosts $u_i=(U_{i,0},\ldots,U_{i,j},\ldots,U_{i,(m-1)})$ and $u_{i'}=(U_{i',0},\ldots,U_{i',j},\ldots,U_{i',(m-1)})$ $(i' \in [0,n-1], i' \neq i)$ are neighbors if and only if $U_{i,j} = U_{i',j}$ for all j, except $U_{i,j'} = U_{i',j'} \pm 1$ along only one dimension j'. Thus, in the m-D mesh, an end host may have m to $2m$ neighbors. We also define the *Euclid distance* of two nodes in the mesh as their *static delay distance*. In a 2-D mesh, the *static delay distance* of two nodes (X_0,Y_0) and (X_1,Y_1) is $|X_1-X_0|+|Y_1-Y_0|$. The sum of *static delay distances* from (X_0,Y_0) to other nodes (X_i,Y_i) is $f(X_0,Y_0)=\sum_{i=1}^{n-1}|X_i-X_0|+|Y_i-Y_0|$.

2.1 Cluster Formation Algorithm

In our application layer multicast scheme, the group members are initially split into several clusters by some management nodes (called Rendezvous Points – RP). The cluster size is normally set as

$$S=(k,3k-1) \tag{1}$$

The expression *(k, 3k-1)* represents a random constant between k and $3k-1$. Like NICE, k is a constant, and in our simulation, we also use $k=3$. The definition of cluster size is for the same reason as the one of NICE that is to avoid the frequent cluster splitting and merging (see [4]). Define the state of the end host that has not been assigned into any cluster as *unassigned.* We describe the cluster formation as follows. The RP initially selects the *left lowest end host* (say u) among all *unassigned* members. The *left lowest end host* is the end host who occupies the mesh node that has the minimum coordinates along m dimensions among all nodes occupied by the *unassigned* group members. The cluster member selection is in the dimension order around u by using the following algorithm.

Alg-1: Cluster Formation
Input: *Unassigned group member set* $G'=\{u_0,\ldots,u_i,\ldots u_{n-1}\}, i \in [0,n-1]$ and the RP;
// n is the set size that initially equals to the group size
Output: Cluster set $CS=\{\}$;
1. While $G' \neq \Phi$ do {
2. the RP selects the *left lowest end host* u in G' and removes u from G';
3. for $j=0$ to $m-1$ do { //m is the dimension number of mesh overlay
4. The RP selects *unassigned* closest member in the j-th dimension into the cluster and removes it from G';
5. For $j'=0$ to $j-1$ do {
6. The RP selects the closest *unassigned* member in the sub-mesh $k_{j'} \times k_j$ into the cluster and removes it from G';}
7. The RP selects the closest *unassigned* member in the sub-mesh $k_0 \times \cdots \times k_j$ into the cluster and removes it from G';
8. If (the cluster size equals to S) $\{j=m-1;\}\}\}$

Fig. 1 shows a 2-D mesh. In this mesh, the initial *left lower end host* is (0,0). According to steps *3-4*, the RP firstly selects the end host in (0,1) into the cluster. Because *j=0*, steps *5-7* are neglected. Then, the RP selects the end host in (1,0) into the cluster by steps *3-4*. Based on steps *5-7*, the next selected cluster member is the one in (1,1). The cluster formation guarantees that each cluster contains the closest group members in terms of *static delay distance*. According to our research results in [18], the scheme that assigns closed members into the same cluster will improve the scalability and efficiency of application layer multicast.

2.2 Optimal Core Selection Algorithm

Each cluster will have a cluster core. The core is the root of the tree in the cluster. The following theorem gives the sufficient and necessary conditions to select a cluster core in each cluster that is *optimal* in terms of the minimum sum of *static delay distance*s to all other cluster members.

Theorem 1: *Let u be the cluster member that occupies the node* $(U_0,\ldots,U_j,\ldots,U_{m-1})$ *in a m-D mesh network and* $n_{>j}$, $n_{<j}$, *and* $n_{=j}$ *be the number of cluster members with the j-th coordinates larger than, less than, and equal to* U_j *respectively. Then u is the optimal core if and only if the following m inequalities hold simultaneously:*

$$| n_{<j} - n_{>j} | \leq n_{=j}, \quad j=0, 1,\ldots, m-1. \tag{2}$$

Proof (➔): Suppose $u = (U_0,\ldots,U_j,\ldots,U_{m-1})$ is an *optimal core*, then for any member *u'* in the mesh, there exists $f(u) \leqslant f(u')$. To achieve (5), we first consider a node $u'=(U_0,\ldots,U_j+1,\ldots,U_{m-1})$ and its multicast static delay distance *f(u')*. Given any member $u_i = (U_{i,0},\ldots,U_{i,j},\ldots,U_{i,(m-1)})$ and $U_j \leqslant U_{i,j}$, the distance from u_i to the end host *u* is one unit longer than the distance from u_i to *u'*. Similarly, it can be seen that for any member $u_i = (U_{i,0},\ldots,U_{i,j},\ldots,U_{i,(m-1)})$ and $U_{i,j} \leqslant U_j$, the distance from u_i to node *u* is one unit shorter than the distance from u_i to *u'*. Because there exist $(n_{>Uj} + n_{=Uj})$ members whose *j*-th coordinates are larger than or equal to U_j, and $n_{<Uj}$ cluster members whose *j*-th coordinates are less than U_j, we have

$$0 \leq f(u') - f(u) = \sum_{i=0}^{n'} (d(u',u_i) - d(u,u_i)) = n_{>U_j} + n_{=U_j} - n_{<U_j}$$
$$\rightarrow n_{<U_j} - n_{>U_j} \leq n_{=U_j}$$

By comparing $f((U_0,\ldots,U_j-1,\ldots,U_{m-1}))$ with $f(u)$ in the same way as above, we can achieve the inequality of (2).

(➔): It is easy to demonstrate that if (2) is violated, then u cannot be the *optimal* core. Assume $n_{<Uj} - n_{>Uj} > n_{=Uj}$, then $n_{>Uj} > n_{<Uj} + n_{=Uj}$. This means that the number of end hosts with the *j*-th coordinates greater than U_j is more than the other two cases. Thus the distance from *u* to these end hosts is larger than some other end hosts, a desired contradiction. ■

The *optimal core selection algorithm* in the *m*-D mesh network is given below

---

Alg-2: Optimal Core Selection in m-D Mesh Networks
Input: Cluster member set $C=\{c_0=(C_{0,0}, C_{0,1},\ldots, C_{0,(m-1)}), c_1=(C_{1,0}, C_{1,1},\ldots, C_{1,(m-1)}), \ldots, c_{(n'-1)}=(C_{(n'-1),0}, C_{(n'-1),1},\ldots, C_{(n'-1),(m-1)})\}$; // *n'* is the cluster size
Output: *optimal core* $c^*=(C_0^*, C_1^*,\ldots, C_{(m-1)}^*)\in C$;
1. Initiate $\{a_{(Cj)min},\ldots,a_{(Cj)t},\ldots,a_{(Cj)max}\}=\{0,\ldots,0,\ldots,0\}$; // $a_{(Cj)t}$ records the number of cluster members whose *j*-th coordinates equal to $(C_j)_t$, where $(C_j)_{min}\leqslant(C_j)_t\leqslant(C_j)_{max}$ and $0\leq j\leq m-1$
2. For $k = 0$ to *n'-1* do
3. If (the *j*-th coordinate of $c_k == (C_j)_{td}$) $\{a_{(Cj)t} = a_{(Cj)t} + 1;\}$
4. For *i=0* to *n'-1* do {
5. For *j=0* to *m-1* do {
6. If $(|\sum_{l=(C_j)_{\min}}^{C_{i,j}} a_l - \sum_{l=C_{i,j}}^{(C_J)_{\max}} a_l| \leq a_{(C_{I,J})})$ $\{C_j^*=C_{i,j}; j=j+1;\}$
7. Else $\{j=m-1; i=i+1;\}\}\}$
8. $c^*=(C_0^*, \ldots, C_j^*,\ldots,C_{(m-1)}^*)$.

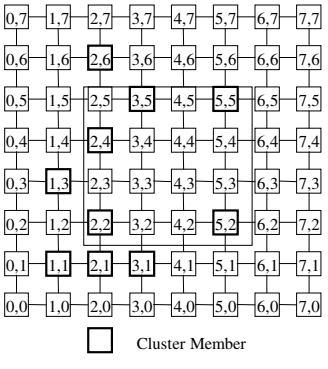


Fig. 1. Selecting the *optimal* core in a *2*-D mesh

In *Alg-2*, steps *1-3* can be executed in time *O(n)*. Steps *4-7* can be improved using binary searching algorithm that yields an $O(ln(n))$ complexity. But for brevity of discussion, we keep the linear search algorithm here. The algorithm may select multiple *optimal* cores. Only one of them will be used at random as the current core and other cores can be the back-up cores for fault-tolerance. Fig. 1 illustrates the *optimal core selection* in a *2*-D mesh. It is known that the core should be in the area $[1,1]\times[5,6]$. It can be checked that the optimal core's *x* coordinate must be 2 while *y* coordinate could be 2 or 3 for $f((2,2)) = f((2, 3)) = 26$. Node (2, 2) is the member and is preferred to (2, 3).

2.3 Weighted Path Tree Generation Algorithm

To multicast the packets in each cluster, a tree using the cluster core as the root is established in each cluster. Because several multicast groups may exist in the network, multicast traffic has to compete with other traffic. It is anticipated that the tree should maximize the sharing of link utilization within the cluster so that the rest of the links may be used for other traffic. Our approach is to connect all members such that (1) the branch on the tree between two adjacent members is the shortest path in the cluster, (2) under the condition (1), the total number of links on the tree should be also minimized.

Table 1. The weights marked '*' belong to the cluster members

Y=6	0	1*	0	0	0
Y=5	0	3	2*	1	1*
Y=4	0	4*	2	1	1
Y=3	1*	5	2	1	1
Y=2	2	10*	4	2	2*
Y=1	1*	3*	1*	0	0
	X=1	X=2	X=3	X=4	X=5

Before the discussion of the *algorithm*, we first define the following terminologies (using a *2*-D cluster as the model):

1. *Shortest path area nodes* (*SPAN*): For any two nodes (X_0,Y_0) and (X_1,Y_1), let $X_{min}=\min\{X_0,X_1\}$, $X_{max}=\max\{X_0,X_1\}$, $Y_{min}=\min\{Y_0,Y_1\}$ and $Y_{max}=\min\{Y_0,Y_1\}$. X_{min}, X_{max}, Y_{min} and Y_{max} uniquely define a *rectangle area* $[X_0,Y_0]\times[X_1,Y_1]$. Each node (X,Y) in $[X_0,Y_0]\times[X_1,Y_1]$ is on one of the shortest paths between $[X_0,Y_0]\times(X_0,Y_0)$and (X_1,Y_1) and is called the s*hortest path area* (*SPAN*) nodes between (X_0,Y_0) and (X_1,Y_1).
2. *SPAN nodes of a cluster member*: When the tree is built in the cluster with the size of n', we call all nodes in the *SPAN* area from the core (i.e. the root of the tree) (X^*,Y^*) to a cluster member $c_i(i\in[0,n'-1])$ as the *SPAN* nodes of c_i. We take Fig. 1 as an example. Assume that the core is in the node (2,2). All nodes in $[2,2]\times[5,5]$ are the *SPAN* nodes of this cluster member.
3. *Node Weight*: A node may be the *SPAN* node of several cluster members. If a node is the *SPAN* node of *k* cluster members, this node is assigned the weight of *k*. Table 1 gives the weights of all nodes in Fig. 1. Take the non-member node (2,5) as an example. Its weight 3 means that 3 cluster members may pass through node (2,5) to (2,2) by the shortest paths. Apparently, the weight of (2,2) is 10.
4. *Path Weight*: Given a shortest path, the path weight is the sum of all on-path node weights. For example, the weight of path <(2,2),(2,3), …,(2,5),…,(5,5)> is 26.

Let the cluster with *n'* members be $C=\{c_0=(C_{0,0},\ldots,C_{0,(m-1)}),c_1=(C_{1,0},\ldots,C_{1,(m-1)}),\ldots,c_{(n'-1)}=(C_{(n'-1),0},\ldots,C_{(n'-1),(m-1)})\}$ and the cluster core be $c^*=(c_0^*,\cdots,c_{m-1}^*)$. We sort the cluster members in a non-decreasing order of the distances from $c^*$to them, thus

$d(c^*, c_i) \leqslant d(c^*, c_j)$ where *ij*. The main idea of the *weighted path tree generation algorithm* can be sketched as follows. Assign a weight for each node in the *rectangle area* $[c_0^*, \cdots, c_{m-1}^*] \times [c_{i,0}, \cdots, c_{i,(m-1)}]$ as described before. After knowing the weight of each node, the RP computes the weight of each shortest overlay path. The *weighted path tree generation algorithm* is shown below:

Alg-3: Weighted Path Tree Generation

Input: Cluster member $CM=\{c_0= (C_{0,0},\ldots,C_{0,(m-1)}),\ldots,c_i=(C_{i,0},\ldots,C_{i,(m-1)}),\ldots, c_{(n'-1)}= (C_{(n'-1),0},\ldots,C_{(n'-1),(m-1)})\}$, $i \in [0,n'-1]$ and the *optimal* core $c^*=(C_0^*,\ldots,C_{(m-1)}^*)$;

Output: Tree *T;*

1. $T= \{ \}$;
2. For any node $c_i=(C_{i,0},\ldots,C_{i,,(m-1)})$ with $((C_j)_{min} \leq (C_j) \leq (C_j)_{max})$, initialize its weight $W_{c_i} = 0$;
3. For $i' = 0$ to $n'-1$ do
   If (c_i is a *SPAN* node of $c_{i'}=(C_{i',0},\ldots, C_{i,(m-1)})$) $\{ W_{c_i} = W_{c_i} + 1; \}$
4. For $i = 0$ to $n'-1$ do
   Select the shortest path $P=<(C_0^*,\ldots,C_{(m-1)}^*),\ldots,(C_{i,0},\ldots,C_{i,(m-1)}) >$ with the maximum weight and add *P* to *T;*

2.4 Multicast Routing Algorithm

To build a tree for each cluster, the *weighted path tree generation algorithm* is employed to construct a tree connecting all the cluster roots for the inter-cluster routing. Then, the *optimal core selection algorithm* is used to select the root of this tree. At last, the following multicast routing is designed to routing the packets among all group members.

Alg-4: Multicast routing for group G:

1. Source *s* sends its multicast messages to its cluster core *c*, *c* then forwards them to the roots *r* of all other trees;
2. *c* routes the multicast packets to its own cluster members along the cluster tree;
3. At the same time, all cluster cores, upon receiving the multicast messages, transmit them along the cluster trees to all cluster members within the clusters.

---

3 Performance Evaluations

3.1 Simulation Model

This section evaluates our multicast algorithms with the simulation developed in *ns-2* [17] and run by a group of SUN SPARC-20 workstations. In this simulation, six multicast routing algorithms for 2-D meshes are used for the performance testing and comparison: *Double-Channel XY Multicast Wormhole Routing* (*DCXY*) [14], *Dual-Path Multicast Routing* (*DPM*) [15], *RCWP*, *OcxyP*, *RcxyP* and our multicast scheme named

as *OCWP*. *RCWP* is the multicast scheme that randomly selects the cluster core for each cluster but constructs the tree by the *weighted path tree generation algorithm*; *OCWP* is the multicast scheme that selects the cluster core by the *optimal cluster core selection algorithm* and constructs the forwarding paths by the *weighted path tree generation algorithm*; *OcxyP* selects the cluster core by the *optimal cluster core selection algorithm* but constructs the forwarding paths by using the *XY routing algorithm*; *RcxyP* randomly selects the cluster core and constructs the multicast paths by using the *XY routing algorithm*. The network topology used in the simulation is a 32×32 *2*-D mesh. The bandwidth of each link is 10Mbps. During the simulation, 20,000 multicast packets are randomly generated as a Poisson process and the average size of the packets is 1200 bytes so that the average time to transmit a packet on the defined link is about 1ms. The following two metrics are employed to evaluate these multicast schemes:

- *Average multicast delay*: Define the message multicast delay at a node as the sum of the routing delay, queuing delay and transmission delay. The *average multicast delay* AD is computed by

$$AD = (\sum_{i=0}^{n-1} d(s,u_i))/n \tag{3}$$

  where $d(s,u_i)$ is the packet delay from the source s to the member u_i and n is the group size.
- *Number of link used*: It refers to the total number of links used in G in order to multicast the messages to all group members.

3.2 Simulation Observations on Regular Mesh Multicasting

The average delay metric under the light load of network is shown in Fig. 2 (a) and (b). The link usage for different algorithms is shown in Fig. 2 (c). It can be seen that the average delay increases with the increase of the network load (Fig. 2 (d)). From these simulation results, we have the following observations:

1. Under the lower load circumstance, the delay is mainly related to the distance from the source to the group members (Fig. 2 (a)). Because the *DCXY* approach always transmits multicast packets to group members along the shortest paths from the source to the group members, it achieves the best delay performance among the other systems when the network is lightly loaded. When *DPM* approach is applied, the delay increases rapidly as the number of group members increases. This indicates that DPM does not scale well (Fig. 2 (b)). When traffic is low, *OCWP* achieves the second best delay performance to *DCXY* but it scales well as the traffic increases (Fig. 2 (d)).
2. Fig. 2 (c) shows the average number of links used by these routing approaches. In general the number of links will be increased with the number of the group members. The figure shows that for the same number of group members, *OCWP* makes use of the minimum number of links for transmitting the multicast packets whereas *DPM* uses the maximum. The shared tree routing approach (such as *RCxyP*) uses almost the same number of links as *DCXY*.

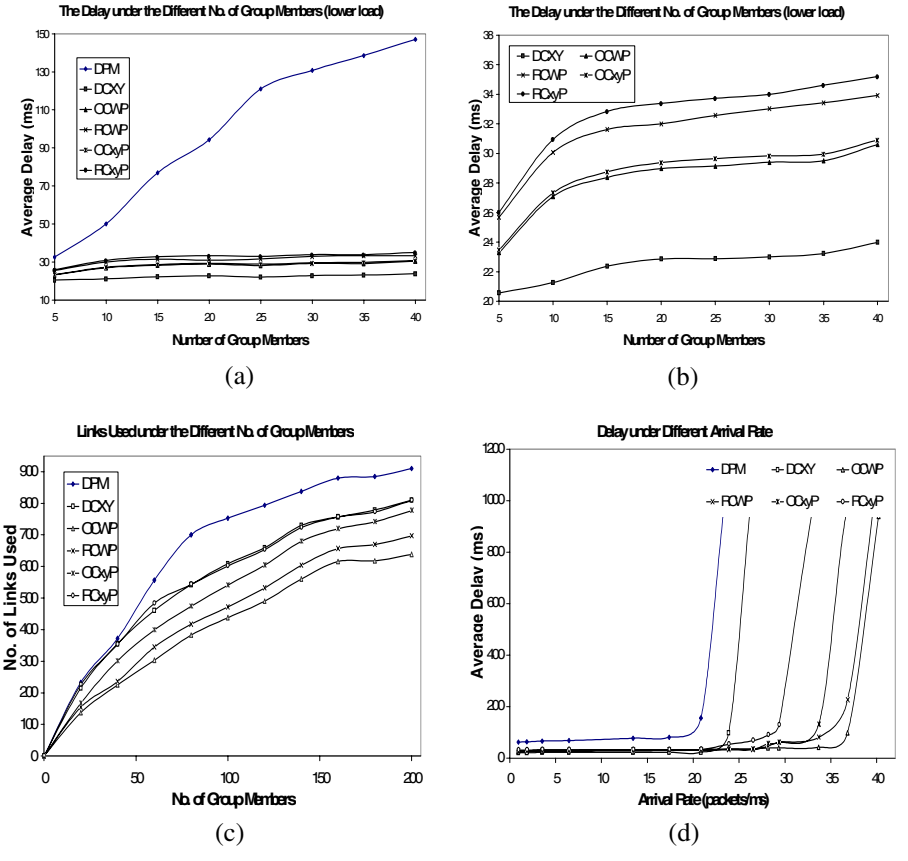


Fig. 2. Simulation results for *DCXY*, *DPM*, *RCWP*, *OcxyP*, *RcxyP* and our *OCWP*

3. Fig. 2 (d) shows that the delay increases as the packet arrival-rate increases. The system saturation points for *DPM, DCXY, RCxyP, OCxyP, RCWP* and *OCWP* are about 21.5, 24, 29.5, 34, 36.5 and 37.5 packets/ms respectively. Our algorithm achieves the maximum throughput. It reveals that under the same condition, *OCWP* obtains the best balance over the performance parameters, i.e., the less resource a system consumes, the higher the throughput and the shorter the end-to-end delay under the high traffic load.

4 Conclusions and Future Work

The *cluster formation*, *optimal core selection* and *weighted path tree generation* algorithms are suitable for multicast communication on (abstract) mesh networks. It is proved that the core selection algorithm is *optimal* in terms of the minimum sum of *static delay distances* from the core to all the members in the cluster. The multicast tree formulated by our tree generation algorithm can effectively utilize the links with

the shorter average delay. As compared with other multicast schemes, our algorithms can select a suitable core, and construct an efficient tree in terms of balancing the less resource a system consumes, the higher the throughput and the shorter multicast delay under the high traffic load. We anticipate that the issues discussed may be applied to ad-hoc network routing where the nodes can move and an *optimal* core may be re-selected or re-positioned.

References

[1] S. Deering and D. Cheriton, "Multicast Routing in Datagram Internetworks and Extended LANs", ACM Transactions on Computer-Systems, pp. 85-110, Vol. 8, No. 2, May, 1990.

[2] H. Chu, S. Rao, S. Seshan, and H. Zhang, "A case for end system multicast", Proc. of ACM SIGMETRICS 2000, pp. 1-12, June 17-21, 2000, Santa Clara, California, USA.

[3] P. FRANCIS, "Yoid: extending the internet multicast architecture", available at http://www.aciri.org/yoid/docs/index.html, April, 2000.

[4] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast", Proc. of ACM SIGCOMM, pp. 205-217, August 19-23, 2002, Pittsburgh, Pennsylvania, USA.

[5] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Application-level multicast using content-addressable networks", Proc. Of The *3*rd International Workshop on Network Group Communication, pp. 14-29, November 7-9, 2001, London, UK.

[6] J. Jannotti, D. K. Gifford, K. L. Johnson, M. Frans Kaashoek, and J. W. O'Toole Jr., "Overcast: reliable multicasting with an overlay network", Proc. of The *4*th Usenix Symposium on Operating Systems Design and Implementation, October 22-25, 2000, Paradise Point Resort, San Diego, California, USA.

[7] D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel, "ALMI: An application level multicast infrastructure", Proc. of The 3rd USENIX Symposium on Internet Technologies and Systems, March 26-28, 2001, Cathedral Hill Hotel, San Francisco, USA.

[8] Y. Chu, S. G. Rao, S. Seshan, and H. Zhang, "Enabling conferencing applications on the Internet using an overlay multicast architecture", Proc. of ACM SIGCOMM 2001, pp. 55-67, August 27-31, 2001, San Diego, California, USA.

[9] B. Zhang, S. Jamin, and L. Zhang, "Host multicast: a framework for delivering multicast to end users", Proc. of IEEE INFOCOM 2002, pp. 1366-1375, June 23-27, 2002, New York, USA.

[10] H. Deshpande, M. Bawa, and H. Garcia-Molina, "Streaming live media over a peer-to-peer network", Stanford Univ. Comput. Sci. Dept., Stanford, CA, June 2001.

[11] D.Pendarakis, S.Shi, D.Verma, and M. Waldvogel, "ALMI:An application level multicast infrastructure", *Proc. 3*rd Usenix Symp. Internet Technologies and Systems, San Francisco, CA, pp.49-60, March 2001.

[12] I.Stocia, R.Morris, D.Karger, M.F.Kaashoek, and H.Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications", ACM SIGCOMM 2001, San Diego, CA, pp.160-172, August 2001.

[13] B.Y.Zhao, J.Kubiatowicz, and A.Joseph, "Tapestry: An infrastructure for fault-tolerant wide-area location and routing", Univ. California, Berkeley, CA, Apr. 2001.

[14] X. Lin, P. K. McKinley, and L. M. Ni, "Deadlock-free multicast wormhole routing in *2*-D mesh multicomputers", IEEE Trans. On Parallel And Distributed Systems, Vol. 5, pp.793-804, 1994.

[15] X. Lin, P. K. McKinley and A. H. Esfahanian, "Adaptive multicast wormhole routing in *2*-D mesh multicomputers", *Proc.* of Parallel Architectures And Languages Europe 93, pp.228-241,1993.

[16] S. Ratnasamy, P. Francis, M. Handley, R.Karp, and S.Shenker, "A scalable content-addressable network", ACM SIGCOM 2001, August 27-31, 2001, San Diego, CA, USA.

[17] UC Berkeley, LBL, USC/ISI, and Xerox PARC, "*ns* Notes and Documentation", October 20, 1999.

[18] W. Tu and W. Jia, “A scalable and efficient end host multicast for Peer-to-Peer Systems”, *Proc.* of IEEE Globecom 2004, pp. 967-971, November 29-December 3, 2004, Dallas, Texas, USA.

A Novel Dual-Key Management Protocol Based on a Hierarchical Multicast Infrastructure in Mobile Internet

Jiannong Cao[1], Lin Liao [1,2], Guojun Wang [1,2], and Bin Xiao[1]

[1] Department of Computing, Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
[2] School of Information Science and Engineering, Central South University, Changsha, P.R. China
{csjcao, cslliao, csgjwang, csbxiao}@comp.polyu.edu.hk

Abstract. This paper describes a secure multicast infrastructure for large-scale group communications in Mobile Internet and proposes a key management protocol based on the infrastructure. The multicast communication domain is logically divided into several administrative areas with a key server associated with each area. All the key servers participate in a Public Key Infrastructure (PKI) as trusted entities known by the subgroup members. Therefore, it's efficient to minimize the re-key overhead implemented in the mobile host tier. The simulation results show that the proposed protocol has better performance compared to the centralized protocols without PKI support. The numbers of the real re-key messages and the re-key events are reduced to approximately 30% and 65%, respectively.

1 Introduction

The proliferation of the Internet technology and mobile computing devices gives rise to the growth of applications emerging in mobile Internet. Its popularity is fuelled by the growing importance of group-oriented and collaborative applications. One of the major challenges of group communications is secure and efficient group key management, where the basic step to secure the traffic data is to provide a cryptographic group key shared by all the members within a group.

However, the group key should be updated when the members change their status during the group communication session. Furthermore, the delivery of the valid key to all the members of a group is a challenging task due to the fact that the group key and group members can dynamically change. Since the communication among the group members may be inconsistent while data encryption keys are being updated, the challenge for any key management schemes is how to generate and distribute new group keys to authorized group members such that the communication remains secure while the overall impact on the system performance is minimized.

In mobile Internet, the frequent mobility of mobile hosts and limited bandwidth add complexity to the security problem in multicast group communications.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 560 – 569, 2005.

Especially when the number of group members becomes larger and the covering area becomes wider, the key distribution and re-keying process can impose a huge overhead. Researchers have proposed many key management approaches to minimize such an overhead in a scalable and secure manner.

In this paper, we firstly investigate the issues of designing key management protocols for multicast communications in mobile Internet. We propose a secure multicast infrastructure for large-scale group communications, and propose a key management protocol based on the infrastructure. Compared to the centralized key management algorithms under the mobile and dynamic environment, the proposed distributed key management protocol shows better performance in terms of the re-key events and the real re-key messages.

2 Related Work

The most important tasks involved in secure group communications include how to reduce the overhead of key distribution, how to minimize the number of encryptions and decryptions, how to reduce the number of re-key messages, and how to share a secure group key in a large-scale group [1,2,3]. Re-keying efficiency is evaluated based on the following aspects: the communication complexity, the time complexity and the storage requirements [4]. In a small-scale group, the tree structure is widely adopted to cope with key management, such as Tree Key Graph (TKG) [5] and Logical Key Hierarchy (LKH) [6]. In such schemes, there is a trusted third party, known as Key Distribution Centre (KDC), which maintains a tree of keys where the change of one sub-tree will inevitably trigger the re-key operation involving other sub-trees. Extending to large-scale groups, such a centralized KDC turns out to be somewhat burdensome and the single server turns out to be the point of attack for intruders.

One established way for enhancing the fault tolerance of centralized components is to distribute the components to a set of servers and use replication algorithms to mask faulty servers. Consequently, hierarchical approaches have recently been proposed to manage the distribution of the Traffic Encryption Key (TEK) in a scalable manner. The main idea of such a mechanism is that the whole group is divided into many disjoint subgroups, each of which is controlled by an Area Key Distributor (AKD), assisting the group key distribution with KDC. It is obvious that the overhead of the KDC will be diminished by means of distributed AKDs. Iolus [7], a hierarchical framework for secure multicast is proposed with this philosophy in a scalable manner. The divided subgroups sketch out a tree hierarchy with individual address and individual subgroup key for every subgroup respectively. In [8], an inter-domain key management protocol is proposed and each “leaf” region in this architecture is connected together through “trunk” region (backbone). There exists an Initiator Key Distributor (IKD) that holds a copy of the multicast-key and a copy of all the subgroup-keys. Thereafter, the IKD is actually the organizer in all the Autonomous Systems (ASs) as well as the initiator of the whole multicast instance. Due to the existence of global multicast-key, the re-keying of any one AS raised by some members’ dynamic change will give rise to the update of the global multicast-key, as well as the delivery of a new multicast-key to other ASs.

All the schemes summarized above focus on the wired environment. Since wireless devices are gaining in popularity with feasible network connections and powerful computing capabilities, the research of extending them to secure multicast group is worthy of being explored.

The impact of mobility on secure multicast is firstly considered in[9]. Besides the common issues in traditional networks, some other issues, such as transparency of the security features and self-efficiency of mobile users who is willing to take part in a secure group are identified as the specific features for mobile multicast. In mobile multicast scenario, we are facing the difficulty that we have to minimize the participation and computation of mobile hosts because of the intrinsic limitations, while the cost of the Group Manager (GM) tends to be minimized. In order to solve the problem, researchers come up with many solutions, such as matching the key management tree to the network topology called the TMKM tree[10], and the enhanced LKH protocol called LKH++[11].

Nevertheless, there are not many schemes, which solve key management in a large-scale group, and even less in a mobile and wireless environment. Based on the PKI [12] and the clustering techniques, in this paper, we propose a dual-key management protocol to combine the two into a hierarchical multicast infrastructure for secure mobile group communications.

3 The Hierarchical Multicast Infrastructure

3.1 The Proposed Infrastructure

The basic infrastructure for multicast communications in mobile Internet is depicted in Fig. 1. The proposed infrastructure consists of three tiers and four classes of network entities. The three tiers are: the Wired Station (WS) tier, the Access Proxy (AP) tier, and the Mobile Host (MH) tier. WS is the top tier of the infrastructure, consisting of some server stations with high computational capability and high stability. The multicast source disseminates data from this top tier. This tier is implemented through network entities typically found on the wired Internet today, such as routers, switches and servers, together with their corresponding network protocols. AP is the middle tier through which the mobile hosts access and connect to

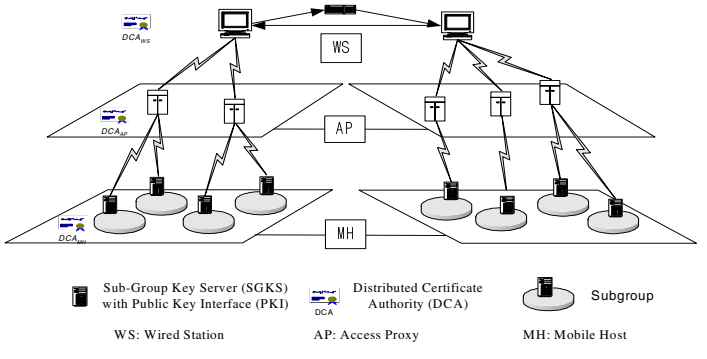


Fig. 1. Hierarchical multicast infrastructure

the WSs to receive multicast data. Suppose it is a cellular network connected to the wired backbone network, the AP tier can act as the Mobile Support Stations (MSS) role to provide the interfaces for mobile hosts. MH is the bottom tier of the infrastructure, consisting of a set of MHs. The MH is a host whose location relative to the rest of the network changes with time, as it is capable of moving between different locations.

Besides the entities of the three tiers mentioned above, another important entity called Sub-Group Key Server (SGKS) in the MH tier involving the dual-key is supposed (see Fig. 2). SGKS acts like the AKD but it differs from AKD in that it works in the PKI infrastructure. All the Sub-Group Key Servers (SGKS) are assumed to be trusted parties known by all the MHs and applied in a PKI infrastructure. To the upper-tier entities, the SGKS is the representation of one subgroup of the MH tier and the direct communication object of the MH tier; while to the MHs, the SGKS acts as the group manager of one subgroup of the MH tier. The MHs that have identified themselves with a particular SGKS are considered local to the SGKS.

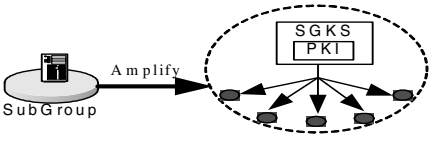


Fig. 2. Subgroup model

3.2 How the Infrastructure Works

As the WS tier is fixed and the AP tier is of lower mobility compared to the MH tier, we divide the MH tier into smaller administrative subgroups, with each subgroup associated with one SGKS as the group manager. When implementing a secure multicast instance in a mobile environment, the SGKS is not the member of the multicast group, but just kind of a known server. In general, the WS tier, the AP tier and the SGKS of the MH tier, are connected and they constitute the backbone of the network infrastructure. As mentioned above, the SGKS is different from the normal GM because it integrates the PKI interface into the unit as shown in Fig. 2. The responsibility of SGKS can be summarized as key distribution of the inner subgroup and the communications with the AP tier. Within each subgroup, the key distribution can be implemented by existing symmetric key management protocols such as Key Graph [5] or LKH [6]. On the other hand, the backbone entities exchange secret data encrypted by asymmetric keys, i.e. Public Key (PK) and Secret Key (SK), due to their low mobility and reliability.

3.3 Assumptions

- All the SGKSs of the MH tier are distributed into the multicast network as service centers, which initially need to register with their DCA for a pair of keys.

- We use $SGKS_i$ to denote the subgroup key server of SG_i (subgroup with sub-index *i*). Two types of keys held by $SGKS_i$ are a pair of asymmetric keys, i.e. $PK(SGKS_i)$ and $SK(SGKS_i)$, and a symmetric subgroup key $K_{S,i}$. Both of them are integrated together in the entity SGKS. Specifically, $K_{S,i}$ stands for the subgroup key shared by the subgroup members, while $PK(SGKS_i)$ stands for the public key held by the $SGKS_i$ in representation of SG_i, and $SK(SGKS_i)$ stands for the other half of the asymmetric keys.
- It is assumed that a cross-tier authentication mechanism exists. Under such circumstances, certificates issued by one certain DCA can be authenticated by DCAs of other tiers. Consequently, DCAs of all tiers are authentic between each other.
- Once the upper tier obtains the PK of the entry it needs to communicate with, the PK is buffered into the buffer box identified by the SGKS's identity number of the certificate.
- We assume that a Distributed Certificate Authority (DCA) is associated with each tier of the infrastructure. Each DCA is responsible for the generation, authentication, expiration and regeneration of the PKs owned by the tier.
- An important requirement is that the available (trusted) SGKSs should be known in advance in order to reduce the possibility of masquerading.

4 The Key Management Protocol

4.1 Adoption of PKI

Many collaborative group settings require distributed key agreement techniques. In the PKI system, all the PKs are public and visible for enquiry and the owner of every PK is a unique one who can decrypt a message by using its secret SK. Unless the SK is expired or disclosed or a fake PK is detected by the DCA, all the PKs are convincing and firm during the valid period. Because of the advantage of the PKI that the security property is high and re-key cost is low, the PKI is widely used in current commercial and educational intranets. Nevertheless, entities in wireless network are not capable of offering the PK computation cost. Therefore, in our assumption, only the backbone hierarchy, which consists of the WSs, APs and all the SGKSs in the MH tier, is applied in the PKI mechanism.

During the initiation, the WSs, APs and all the SGKSs are required to register with the corresponding Distributed Certificate Authority (DCA) they belong to, to announce their identities and parent-children relationships. After the information validation, each DCA issues a pair of keys for all registered members, with a certificate for authentication. The data exchange between different tiers in the infrastructure relies on the PKI mechanism to transmit packets. For example, WS_i has to query DCA_{AP} (DCA of the AP tier) for the PK of its descendant AP by putting the checking information. Notice that WS_i only needs to query the direct downward DCA for efficient and convenient check. As to the SGKS of the MH tier, on receiving data packets encrypted by its PK from its parent entity, it starts to decrypt it and disseminates it to its subgroup members encrypted by its subgroup key. Fig. 3 illustrates of how it works.

Data transmission from AP_i of the AP tier to $SGKS_j$ of the MH tier.
Let P denotes the data packets; $+K(P)$ means using K to encrypt P; and $-K(P)$ means using K to decrypt P.
AP_i : $P' = +PK(SGKS_j)(P)$
$SGKS_j$: $P = -SK(SGKS_j)(P')$ $P'' = +K_{S,j}(P)$
The MHs of $SGKS_j$ with subgroup key $K_{S,j}$: $P = -K_{S,j}(P'')$

Fig. 3. Data transmission process

However, determining the owner of a public key or, conversely, determining the public key for a user, appears to be a basic functionality for executing transactions securely in any large-scale open system. For such authentication issues, many schemes such as DSSA and SPX are found to tackle them [13,14]. It's assumed that the PK searched from the trusted DCA is simply regarded as authentic in our proposed protocol. Once the DCA detects that any PK expires, it will inform the entity of the upper tier to renew its buffer.

4.2 Handling of Changes of Group Members

In this subsection, we describe how to handle the events of members join, members leave and members transfer. Among the three scenarios, the first two belong to *group dynamics* and the last one belongs to *population dynamics* [4].

4.2.1 Join Event

Let's consider the situation that a member needs to join SG_i. Upon approval, it sends to $SGKS_i$ a signal message to notify $SGKS_i$ of its arrival. Then, a new $K_{S,i}$ must be generated by the $SGKS_i$ and multicast to the previous members encrypted by the old $K_{S,i}$ as in most other schemes. $SGKS_i$ is responsible for the re-keying of the SG_i to ensure the backward confidentiality. Approaches for inner subgroup re-keying include logical tree-based algorithms such as key graph [5]. Because the $PK(SGKS_i)$ of $SGKS_i$ is not altering as the change of $K_{S,i}$, and the $K_{S,i}$ is only generated by $SGKS_i$, it's apparent that other subgroups needn't to carry out the re-key operations. All the re-key operations are accomplished by the SGKS within the subgroup, and the whole overhead is only concerned within the changing subgroup, which is apparently reduced compared to the centralized protocols without PKI [8,15,16].

4.2.2 Leave Event

When a member of SG_i tends to leave from the group session, actually it firstly needs to send a request signal of departure to the $SGKS_i$. Upon receiving the signal, the $SGKS_i$ starts the re-keying process to ensure forward confidentiality. Similar to a join event, the re-keying process only happens within the subgroup by multicasting the new $K_{S,i}$ to the remaining group members encrypted by members' individual keys. While in centralized key management schemes without PKI [8,15,16], the group manager still needs to update the global group key and subgroup key database and

deliver the new group key to all the subgroups. From the above analysis, join and leave events merely trigger the re-key processes within that subgroup, and other subgroups remain unaffected.

4.2.3 Transfer Event

Mobility complicates the key management by allowing members to not only leave or join but also transfer between subgroups while remaining in the session (see Fig. 4). Mobility impacts performance only when members cross between subgroups, where re-keying messages must cross the boundaries resulting in performance degradation.

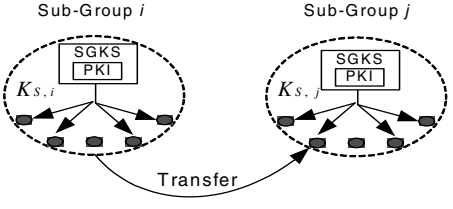


Fig. 4. Mobile nodes are transferring

The algorithms describing a member transferring from one subgroup to another subgroup are outlined as three approaches. It's analyzed that First Entry Delayed Re-key + Periodic (FEDRP) has a low re-key rate and message rate [17]. We adopt the scheme to do our comparison with the centralized schemes without the PKI. In FEDRP, when a member transfers from SG_i to SG_j, SG_i doesn't perform re-keying process right now. Thus, a member may accumulate $K_{S,i}$ as it visits different subgroups. If the entering member has previously visited SG_j, no rekey occurs for SG_j. If there is no visiting record, $SGKS_j$ will send the current $K_{S,j}$ to it by a secure unicast channel as needed. If the member is entering into SG_j for the first time, a new $K_{S,j}$ is generated and distributed through one multicast transmission (to current SG_j members using previous $K_{S,j}$) and one unicast transmission (to the newly entered member using a secure channel). To bound the maximum time that $K_{S,i}$ can be held by a member outside SG_i, each SGKS maintains a timer to bound it. Once the timer reaches the value, the subgroup re-keys itself and the timer is reset to zero. To trace member's movement history, $SGKS_i$ maintains a table of group members that hold a valid $K_{S,i}$ residing outside the subgroup. The table is reset once the member leaves the group or the timer expires. A member is added to the table when it transfers out of it, and a member is removed from the table when it transfers back.

In such a situation, FEDRP behaves with lower re-key rate than merely treated as firstly leave and then join [17]. Since the dual-role of the SGKS and the absence of the global group key, the transfer process is only handled by the two involved SGKSs, which still gets the benefit from the PKI system.

5 Simulation Studies

Because of the introduction of the public key infrastructure, the backbone of the proposed infrastructure relies on the PKI mechanism and re-keying processes occur

within the subgroups. The complexity of our protocol, e.g. the join event, the leave event, is $O(\log n')$, here n' is the number of members of each subgroup; while in the centralized protocols without PKI [8,15,16] the complexity is $O(\log n)$ and n denotes the number of overall group members. In addition, in the centralized protocols without PKI support [8,15,16], the subgroups are not absolutely independent because of the existence of the global key manager managing all the subgroups. It is inevitable that the variation of one subgroup key will give rise to the global key update, and that the GM will still send many re-key messages to the other subgroups.

Three performance metrics are used:

- *Delay time (D_t)* measures the time difference between the time the member sends its willing to join or leave and the time the member is really granted for join or leave after the re-key process completes.
- *Re-key events (NumEvents)* measures the total number of control events to notify a new key when doing the re-key operations. The corresponding re-key events fall into three categories: signal events, unicast events, and multicast events.
- *Re-key messages (NumMsgs)* measures the total number of real re-key messages transmitted to all the mobile members for re-key operations. If a re-key message is a multicast re-key message, then there will be more than one user who receive it and use their correct keys to decrypt the required segments of the message respectively. As to the signal events and unicast events, such a re-key events is equivalent to one real re-key message. Therefore, *NumMsgs* is used to evaluate the real-transmitted number of re-key message packets in terms of the number of receivers in the multicast group.

We conducted the simulation to evaluate the performance with a comparison to the centralized key management protocol without PKI support [8,15,16]. We define the simulation time to be 600s and the whole area to be 600m*400m.

As to the delay time, it is obvious that our proposal performs better. In our proposed protocol, the join and leave procedures complete just after the subgroup re-creates a new subgroup key and distributes it to the valid members. In contrast, the centralized protocols without PKI support need two further steps to update the global multicast key and distribute it to the remaining subgroup controllers.

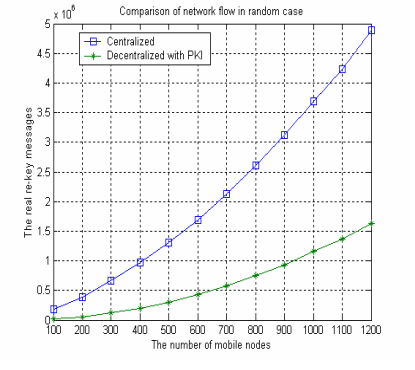


Fig. 5. The real re-key messages in random case

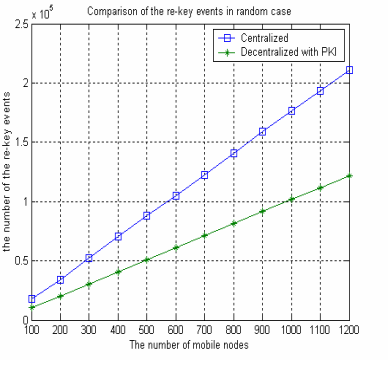


Fig. 6. The re-key events in random case

In Fig. 5 and Fig. 6, we plot the real re-key messages (*NumMsgs*) and the re-key events (*NumEvents*) for the two protocols with the change of network size in the random case, where all the members move at random speed and directions. Each data of the curve is the average result of ten rounds of independent running. We compare the data of the two protocols to get the ratios of improvements for each X-axis value. It's concluded that in average our protocol has 30% and 65% of the real re-key messages and the re-key events respectively in contrast to centralized protocols. The reason is that in our protocol re-keying is almost occurred within the subgroups, while the centralized one needs to have the global key update since the existence of a global group key all subgroups share.

We also carried out simulations for the situation where each member moves back and forth between two subgroups. In the regular case, we can find the differences become more evident in Fig. 7 and Fig. 8 than in the random case. Due to the reason that the movement between two subgroups in the centralized protocols give rise to two subgroup re-keying and the global key update, which is much bigger than our protocol in PKI. In such regular case, our protocol just gets 25% and 55% of the real re-key messages and the re-key events compared to centralized protocols without PKI in average respectively [8,15,16].

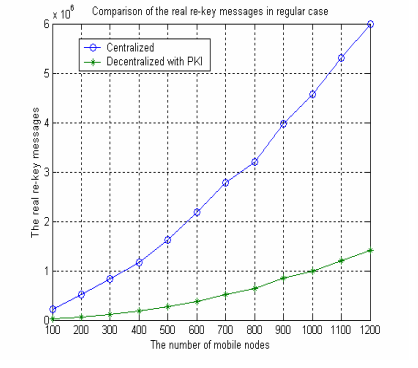


Fig. 7. The real re-key messages in regular case

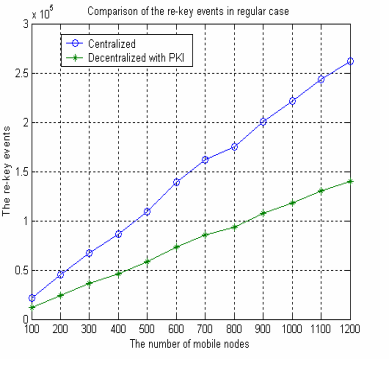


Fig. 8. The re-key events in regular case

6 Conclusions

The proposed dual-key management protocol with PKI support has better performance than the centralized key management protocol without using PKI. Such a conclusion is drawn on the basis of the stability of the PKI system and the trustiness of the SGKSs. However, the proposed protocol requires that the system heavily relies on the PKI infrastructure. Once the authentication of the PKI fails, the consequence will be serious. The delay due to the decryption of SK and encryption of subgroup key may also affect the performance. Nonetheless, the computation power of the SGKS counterbalances the delay. Although some delay cannot actually be avoided, in large-scale multicast communications, such a drawback will not affect the whole performance much and our proposed protocol outperforms centralized protocols without PKI support.

Acknowledgment

This work is supported in part by the Hong Kong Polytechnic University Central Research Grant *G-YY41*, and in part by the University Grant Council of Hong Kong under the CERG Grant PolyU *5170/03E*.

References

1. B. DeCleene, L. Dondeti, S. Griffin. Secure Group Communications for Wireless Networks. *Military Communications Conference (MILCOM 2001).* Vol. 1, pp. 113-117, 2001.
2. T. Hardjono, B. Cain, N. Doraswamy. A Framework for Group Key Management for Multicast Security. *Internet Draft*, draft-ietf-ipsec-gkmframework-03.txt, 2000.
3. S. Rafaeli, D. Hutchison. A Survey of Key Management for Secure Group Communication. *ACM Computing Surveys*, Vol. 35, No. 3, pp. 309-329, 2003.
4. D. Bruschi, E. Rosti. Secure Multicast in Wireless Networks of Mobile Hosts: Protocols and Issues. *ACM/Kluwer Mobile Networks and Applications*, Kluwer Academic Publishers, Vol. 7, pp. 503-511, 2002.
5. C. K. Wong, M. Gouda, S. S. Lam. Secure Group Communications Using Key Graphs. *IEEE/ACM Transactions on Networking*, Vol. 8, No.1, pp. 16-30, 2000.
6. H. Harney, E. Harder. Logical Key Hierarchy Protocol. *Internet draft*, draft-harney-sparta-lkhp-sec-00.txt, 1999.
7. S. Mittra. Iolus: A Framework for Scalable Secure Multicasting. *Proceedings of ACM SIGCOMM'97*, pp. 277-288, 1997.
8. T. Hardjono, B. Cain. Secure and Scalable Inter-domain Group Key Management for N-to-N Multicast. *Proceedings of International Conference on Parallel and Distributed Systems*, pp. 478-485, 1998.
9. L. Gong, N. Sacham. Multicast Security and its Extension to a Mobile Environment. *ACM/Kluwer Wireless Networks*, Vol.1, No. 3, pp. 281-295, 1995.
10. Y. Sun, W. Trappe, K. J. Ray Liu. A Scalable Multicast Key Management Scheme for Heterogeneous Wireless Networks. *IEEE/ACM Transactions on Networking*, Vol.12, No. 4, pp. 653-666, 2004.
11. R. Di Pietro, L. V. Mancini, S. Jajodia. Efficient and Secure Key Management for Wireless Mobile Communications. *Proceedings of the Second ACM International Workshop on Principles of Mobile Computing*, pp. 66-73, 2002.
12. P. R. Zimmermann. The Official PGP User's Guide. MIT Press, 1995.
13. M. Gasser, A. Goldstein, C. Kaufman, B. Lampson. The Digital Distributed System Security Architecture. *Proceedings of the 12th NIST/NCSC National Conference on Computer Security*, pp. 305-319, 1989.
14. J. J. Tardo, K. Alagappan. PX: Global Authentication Using Public Key Certificates. *Proceedings of IEEE Symposium on Research in Security and Privacy*, pp. 232-244, 1991.
15. G. Wang, L. Liao, J. Cao, K. Chan. Key Management for Secure Multicast Using the RingNet Hierarchy. *Proceedings of International Symposium on Computational and Information Sciences (CIS 2004),* LNCS (Spring-Verlag), Vol. 3314, pp.77-84, 2004.
16. T. Kostas, D. Kiwior, G. Rajappan, M. Dalal. Key Management for Secure Multicast Group in Mobile Networks. *Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX'03)*, pp. 1-3, 2003.
17. C. Zhang, B. Decleene, J. Kurose, D. Towsley. Comparison of Inter-area Re-keying Algorithms for Secure Wireless Group Communications. *Performance Evaluation*, Elsevier Science, Vol. 49, pp. 1-20, 2002.

Interdomain Traffic Control over Multiple Links Based on Genetic Algorithm

DaDong Wang[1,2], HongJun Wang[1], YuHui Zhao[1], and Yuan Gao[1]

[1] School of Information Science & Engineering, Northeastern University, 110004 Shenyang, China
wdd@mail.neuq.edu.cn
[2] School of computer, Jilin Normal University, 136000 Siping, China

Abstract. Network operators must have control over the flow of traffic into, out of, and across their networks. Usually, changes of interdomain traffic affect intradomain traffic in other domains. We propose an approach to control interdomain traffic over multiple links based on prefixes. Transit Autonomous System (AS) measure the flow of traffic and cooperate with other ASes in balancing interdomain traffic. Using the data of interdomain traffic and intradomain traffic measured in transit networks, the approach evaluates the cost of transit networks. We present a genetic algorithm to specify a link for prefix in AS neighbors with the objective of minimizing costs and configuration changes. In order to decrease the complex of calculation, we analysis the distribution of traffic and propose an approach to select popular prefixes. A example verify the availability and effectiveness of the algorithm.

1 Introduction

The delivery of IP packets through the Internet depends on a large collection of routers belonged to Autonomous Systems (ASes). Using standardized intradomain routing protocols and interdomain routing protocols, routers compute end-to-end paths in a distributed fashion. The routers inside an AS typically run an intradomain routing protocol or IGP such as OSPF, IS-IS, or RIP. The IGP determines how a network entity (end hosts or router) inside the AS reaches another entity in the same AS via intermediate hops. To reach entities outside the AS, the border routers run an interdomain routing protocol or EGP. Border Gateway Protocol (BGP) is the current de facto standard interdomain routing protocol. Operating a large IP backbone requires continuous attention to the distribution of traffic over the network. Network operators adapt to changes in the distribution of traffic by adjusting the configuration of the routing protocols running on their routers. For the most part, the effects of IGP configuration changes are local to the operator's network. However, traffic engineering grows more complicated if the operates need to alter how packets travel between ASes, since EGP configuration changes have direct effects on the flow of interdomain traffic in other ASes.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 570–579, 2005.

IGPs like OSPF select shortest-path routes based on the sum of integer link weights. Changing the link weights triggers the selection of new paths for some portion of the traffic[1]. However, BGP does not incorporate any performance, load, or capacity information. In today's Internet, operators have indirect control over the flow of interdomain traffic by setting some BGP attributes to configure import policies that favor some routes over others[2] [3]. The state of the art for interdomain traffic engineering is extremely primitive. [4]described several techniques that are used to control the flow of packets in the global Internet, [5] proposed fundamental objectives for interdomain traffic engineering and specific guidelines for achieving these objectives with in the context of BGP, [6] designed a online simulation system which can be applied to adaptively adjust BGP configuration to achieve load balancing. In this paper, we propose an approach to control interdomain traffic over multiple links.

2 Interdomain Traffic Control

BGP is a path-vector protocol that works by sending route advertisements. A route advertisement contains the network reachability information (represented by prefixes:a network address and a netmask), the next-hop and AS_path attributes. A route advertisement may also contain several optional attributes such as the local-pref, Multi-Exit Discriminator (MED) or communities attributes.

Inside a single domain, all routers are considered as equal and the intradomain routing protocol announces all known paths to all routers. In contrast, in the global Internet, all ASes are not equal and an AS will rarely agree to provide a transit service for all its connected ASes toward all destinations. Therefore, BGP allows a router to be selective in the route advertisements that it sends neighbor eBGP routers. A simplified view of a BGP router is shown in figure 1.

BGP allow each AS to choose its own import and export policy in selecting the best route, and announcing and accepting routes. One of the most important factors in determining routing policies is the commercial contractual relationships between ASes. The AS relationships can be classified into customer-provider and peering relationships[9]. Each AS sets its policy according to the AS relationships and some rules[10]. Routing between ASes is determined by the

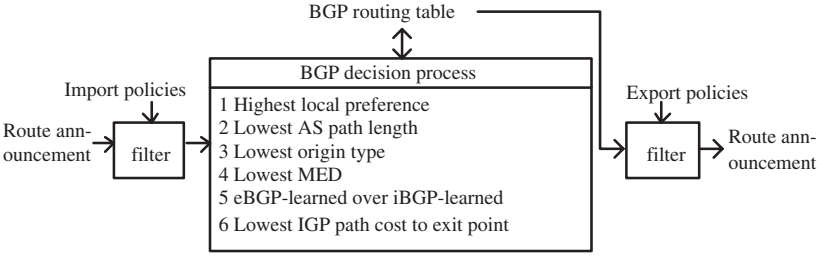


Fig. 1. A BGP router

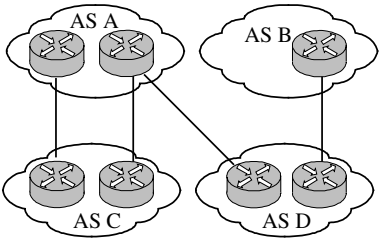


Fig. 2. A example of AS topology

policies, and the routes affect the traffic. As a whole, the distribution of traffic is determined by AS relationships. In general, the outbound traffic of a customer are controlled by the customer and the inbound traffic of a customer are controlled by its provider. However, the outbound traffic changes of an AS maybe affect the traffic of its neighbors. The traffic control need cooperate between neighboring ASes.

The interdomain traffic control includes two aspect:

1. There are multiple links between two ASes (AS A and AS C in figure 2). Balance the traffic over each link.
2. An AS is multi-homed (AS D in figure 2). Select one provider to transit IP packets for each prefix in the AS.

There are two types of ASes in today's Internet. A stub AS is an AS that sends or receives IP packets, but does not transit packets. A transit AS is an AS that agrees to transit IP packets from one of its neighbors to another neighbor. Today, around 90% of customer ASes are stub AS. [7] proposes a method based on multi-objective combinatorial optimization to perform interdomain traffic control for multi-homed stub ASes with minimal BGP configurations. In this paper, we discuss how to balance the traffic over multiple links.

There are four different ways to balance the traffic over multiple links: setting of the MED parameter, prepending of the AS_PATH parameter, usage of communities and defining of more specific routes. To balance the traffic over multiple links, it is important to know how to route each prefix at first. Then, apply one of the above four ways for each prefix to balance the traffic. As the traffic changes of a stub AS affect the traffic of its neighboring transit AS, the cooperation between the transit AS and the stub AS is necessary. In general, the process dominated by a transit AS could repeat less times than the process dominated by a stub AS before setting on a mutually agreeable change in the configuration. In this paper, we focus on how to select a interdomain link for each prefix in a transit AS.

3 A Model

A transit AS is represented by a directed graph $G=(V, E)$ whose nodes and edges represent routers and the links between them. Each edge e has a capacity $C(e)$

which is a measure for the amount of traffic flow it can take. Given the demand $d(s,t)$ which is the amount of traffic flow to be sent from s to t. Obviously, $d(s,t)$ distributes over all the links between s and t. The load $l(e)$ on an edge e is the total flow over e, that is $l(e)$ is the sum over all demands of the amount of flow for that demand which is send over e.

$$l(e) = \sum_{(s,t)\in V\times V} d(s,t) \tag{1}$$

The utilization of a link e is $l(e)/\mathrm{C}(e)$. We associate a cost on link e as a function of the utilization, and use $\Phi_e(l(e))$ represent the cost over link e. The cost function Φ sums the cost of the links.

$$\Phi = \sum_{e\in E} \Phi_e(l(e)) \tag{2}$$

To minimize congestion, our objective is to distribute the flow so as to minimize Φ.Generally, Φ favors sending flow over links with small utilization. The cost increases progressively as the utilization approaches 100% and then explodes when maximum capacity is reached.

The cost function Φ is piecewise linear and convex. For each edges $e\in E$, $\Phi_e(l(e))$ is the continuous function and derivative

$$\Phi_e l(e) = \begin{cases} 0 & (l(e) = 0) \\ \alpha_i l(e) + \beta_i & (l(e) > 0) \end{cases} \tag{3}$$

where for $l(e)/\mathrm{C}(e)\in$ [0,1/3), [1/3,2/3), [2/3,9/10),[9/10,1), [1,11/10) and [11/10,$+\infty$), α_i equals 1,3,10,70,500 and 5000 respectively[8]. β_i can be calculated at vertexes. The idea behind $\Phi_e(l(e))$ is that it is cheap to send flow over an edge with small utilization. As the utilization approaches 100%, it becomes more expensive. If the utilization goes above 100%, we get heavily penalized, and when the utilization goes above 110% the penalty gets so high that this should never happen. The exact definition of $\Phi_e(l(e))$ is not so important for the results, as long as it is a piecewise linear increasing and convex function.

If s and t are border routers, the IP packets sent from s to t come from the neighbor ASes. Extended the directed graph. Add edges to border routers to represent interdomain links. Then

$$\Phi = \sum_{e\in E} \Phi_e(l(e)) + \sum_{e'\in E'} \Phi_{e'}(l(e')) \tag{4}$$

where E' is the collection of the interdomain links between the transit AS and its neighbor ASes.

OSPF is the most commonly used intradomain internet routing protocol. Traffic flow is routed along shortest paths, splitting flow at nodes where several outgoing links are on shortest paths to the destination. The weight of the links, and thereby the shortest path routes, can be changed by the network operator.

The OSPF weight setting problem seeks a set of weights that optimizes network performance.

The interdomain traffic control problem can be described as the follows: Given the extended directed graph of a transit AS $G' = (V, E, E')$, where V is the collection of the routers in the transit AS, E is the collection of the intradomain links and E' is the collection of the interdomain links. Each link e has a capacity C(e). To $\forall(s,t) \in V \times V$, given the intradomain demand $d'(s,t)$ which is the amount of traffic flow to be sent from node s to t. To $\forall p_i, p_j \in \{AS\} \times \{AS\}$, given the interdomain demand $d_p(p_i, p_j)$ which is the amount of traffic flow to be transited by the transit AS from prefix p_i to prefix p_j. The interdomain traffic control problem is to find a OSPF weight w_e [1,65535] for each link $e \in E$, to select a link $e' \in E'$ from multiple links for each prefix transited by the transit AS, making the minimization cost Φ and minimization BGP configuration changes. Finding a solution of the problem is NP-hard. We use a genetic algorithm to find the solution.

4 GA Implement

The algorithm is composed of the inner genetic algorithm and the outer genetic algorithm.

4.1 Outer Genetic Algorithm

The outer genetic algorithm selects a interdomain link from multiple links for each prefix transited by the transit AS. In Internet, the bulk of the traffic is concentrated in a small fraction of prefixes. The top 10% of prefixes accounts for approximately 70% of the traffic in AT&T[5]. A similar rule also exists in the traffic flow between two ASes.

Figure 3 shows the cumulative distribution of the proportion of outbound traffic be-tween the Abilene Network and its neighbor ASes connected by multiple links on October 1, 2004. Although the number of the concerned prefixes belonging to different neighbor ASes ranges from several to several hundreds,

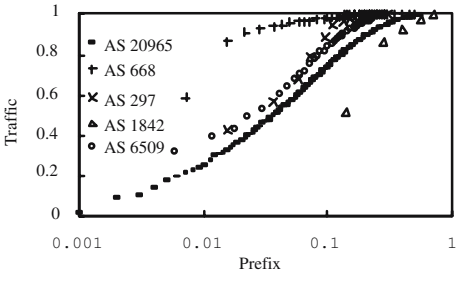


Fig. 3. Cumulative distribution of traffic between Abilene and its neighbor ASes connected by multiple links

the rule that the bulk of the traffic is concentrated in a small fraction of prefixes does not change. In general, balancing the interdomain traffic over multiple links does not need to concern all the prefixes. Adjusting the route of a small fraction of prefixes (popular prefixes) is sufficient.

The outer genetic algorithm uses a serial of integers to represent the selected links of popular prefixes. The initial population is generated by randomly choosing feasible links.

(1) Representation. Use the representation of links $L = l_1, l_2, \ldots l_n$, where $l_i \in$[1,65535] for each popular prefix connected to the transit AS by multiple links. As the number of links between the transit AS and its neighbor AS maybe does not equal, we select multiple-point crossover.

(2) Crossover and mutation. Use multiple-point crossover and 1-point random mutation.

(3) Evaluation function. The association of each solution to a fitness value is done through the evaluation function. The evaluation function is given by the inner genetic algorithm. The outer genetic algorithm is responsible for calculating the amount of traffic between each pair node in the transit AS. $P_t = \{p | p \in AS_t \cup Customer(AS_t)\}$ is the collection of interdomain prefixes transited by node t, where AS_t is the collection of ASes connected the transit AS via node t, Customer(AS_t) is the collection of the customer of AS_t. The demand sent from node s to t

$$d(s,t) = \sum_{p_i \in P_s, p_j \in P_t} d_p(p_i, p_j) + d'(s,t) \tag{5}$$

(4) Parent selection. Use elitist model.

(5) Stopping criterion. MAXGEN denotes the number of generations. The outer genetic algorithm uses this parameter as a stopping criterion.

4.2 Inner Genetic Algorithm

The inner genetic algorithm searches the optimal OSPF weights and calculates the cost.

(1) Representation. Use the representation of weights $W = w_1, w_2, \ldots, w_m$, where $w_i \in$ [1,65535] for each link $e \in E$. Instead of using the upper limit of 65535, we use a user-defined upper limit MAXWEIGHT.

(2) Crossover and mutation. Use 1-point crossover and 1-point random mutation.

(3) Evaluation function. We associate a cost to each individual through the cost function Φ and the number of prefixes whose route change. The evaluation function is complex and computationally demanding, as it includes the process of OSPF routing, needed to determine the link loads resulting from a given set of weights and a given set of links. Using the outer genetic algorithm, we select a interdomain link for each prefix. A given weight setting will completely determine the shortest paths, which in turn determine the OSPF routing, and how much of the demand is sent over which links. The load on each link gives us the link utilization, which in turn gives us a cost from the cost function $\Phi_e(l(e))$.

Add the total cost Φ for all interdomain and intradomain links to the cost of configuration change. We show, in more detail, how to compute the fitness value.

(a) Compute the shortest distances to u from each node $t \in V$, using Dijkstra's shortest path algorithm. Obtain the collection of the nodes between t and u $V_{tu} = \{t, \ldots, u\}$, the collection of links between t and u $E_{tu} = \{e_{vw} | v, w \in V_{tu}\}$.

(b) The demand sent from node t to u will distribute on each link between node t to u . For each edge $e_{vw} \in E_{tu}$, $l_{tu}(e_{vw}) = d(t, u)$.

(c) For each node $u \in V$

$$l_t(e_{vw}) = \sum_{u \in V} l_{tu}(e_{vw}) \tag{6}$$

(d) For each node $t \in V$, the load on each link e_{vw}

$$l(e_{vw}) = \sum_{t \in V} l_t(e_{vw}) \tag{7}$$

(e) Use formula (4) to calculate the cost Φ.

(f) $fitness = F - (\Phi + \alpha N)$. where α is a constant, F is the maximum value of in the process of running, N is the number of prefixes whose route change by the outer genetic algorithm.

(4) Parent selection. Use elitist model.

(5) Stopping criterion. The inner genetic algorithm uses MAXGENIN as a stopping criterion or stops when no evolutionary for a fixed generations.

5 An Example

The Abilene Network is an Internet2 high-performance backbone network that enables the development of advanced Internet applications and the deployment of lead-ing-edge network services to Internet2 universities and research labs across USA. The backbone topology of the Abilene Network is shown in figure 4. The Abilene Network has 77 peer ASes connected via these backbone router on October 1, 2004.

The Abilene Observatory is a program that supports the collection and dissemina-tion of network data associated with the Abilene Network. There is an enormous amount of data collected under the Abilene Observatory program. Abilene collects Netflow statistics every day from its POPs and allows these

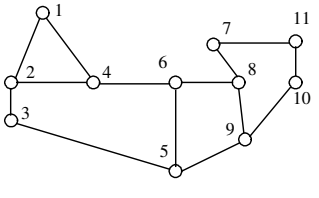


Fig. 4. The Abilene Network backbone topology

statistics to be queried from the web(http://www.itec.oar.net/ abilene-netflow). We use the "Source-Destination-prefix" traffic matrix that provides with total bytes counts for each prefix. Each item of the "Source-Destination-prefix" includes source prefix, destination pre-fix, flows, octets, packets and duration time. The format is shown as the follows.

source-address, destination-address, flows, octets, packets, duration time
130.39/16,130.18/16,1188,10203764100,6900800,35808886

There are about 6.0 millions items on October 1, 2004 and about 1.7 millions items include the prefixes concerning multiple links. Sort the later prefixes of each AS and its customers on flows, the results were shown in table 1.

Table 1. The Abilene Network's neighbor AS connected by multiple links

AS	Access node	Total prefix number	Prefix number(70% inbound traffic)
293	2,7,9,11	32	1
297	2,7,10	54	2
668	2,7,10	135	2
1842	2,7	7	2
3754	7,11	39	4
6509	7,11	163	11
20965	7,10	969	83

As the popular prefixes dominate the bulk of the traffic, we only adjust the route of these popular prefixes over multiple links. Use the following way to select popular prefixes:

(1) Group the sorted prefix list belonging to each neighbor AS into m groups, ensuring the total traffic of each group nearly equals. m is the number of links between the Abilene Network and its neighbor AS.

(2) Select the prefixes whose total traffic exceeds 50% of the total traffic in each group.

Test the above way with prefix-prefix items on October 1, 2004. The number of selected popular prefixes, relevant traffic and the number of relevant prefix-prefix items are 95, 10.32E13 Octets and 271698 respectively. Table 2 lists the percent of traffic in each AS connected to the Abilene network over multiple links, the number of relevant prefixes and prefix-prefix items. Compare the selected popular prefixes with table 2. The traffic controlled by the selected popular prefixes at least exceeds 50% of total traffic.

Remain the selected popular prefixes and replace the other prefixes with the node in the prefix-prefix records. Incorporate those items which have the same source-prefix (node) and destination-prefix (node). At last, we obtain 4710 items to test the outer genetic algorithm.

Set population size of the inner genetic algorithm to 50, MAXWEIGHT to 20, crossover rate to 0.90, mutation rate to 0.01. The average generations of the inner genetic algorithm is 2.15. Set population size of the outer genetic algorithm

Table 2. Traffic and prefixes over multiple links

traffic(%)	Total prefix number	Relevant traffic(10E13 Oct)	P-P item number
10	9	2.90	20233
20	13	3.14	29363
30	21	4.54	59296
40	31	5.15	89319
50	51	6.44	144313
60	74	8.28	211988
70	106	9.34	303022
80	160	11.61	442424
90	260	13.84	685976
100	1399	18.55	1692417

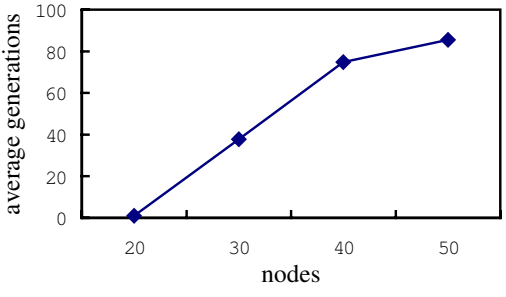


Fig. 5. The average generations of Waxman network

to 100, crossover rate to 0.90, mutation rate to 0.01, the number of crossover points to two times of the number of ASes connected to the Abilene network over multiple links. The average generations of the outer genetic algorithm is 36.2.

Use the random network generated by Waxman model (α=0.1, β=1.0) to test the average generations of the inner genetic algorithm. Set population size to two times of the number of nodes. The result is shown in figure 5.

6 Conclusion

This paper proposes an approach to adjust interdomain traffic over multiple links based on genetic algorithm. Transit ASes measure the flow of traffic and cooperate with other ASes in balancing interdomain traffic over multiple links. The algorithm was tested using the netflow statistics in the Abilene Network. The results show that the algorithm is feasible.

As the structure of the Internet is complex and the amount of traffic in the Internet is diverse, our approach has certain limitations: Although most prefixes received a lot of traffic that varied moderately over large time periods, the absolute amount of traffic can vary a lot from day to day[4]. The approach

can not ensure that the seletion of each prefix over multiple links is optimal from time to time. The running time of the algorithm maybe becomes too long for a complex transit AS.

Despite these limitations, we have shown that our approach provides a view of scheduling the interdomain traffic over multiple links.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under grant No.60273078.

References

1. Fortz, B., Rexford, J., Thorup, M.: Traffic Engineering with Traditional IP Routing Protocols. IEEE Communications Magazine, 2002, 40(10):118-124
2. Awduche, D., Chiu, A., Elwalid, A., et al.: Overview and principles of internet traffic engineering. http://www.ietf.org/rfc/rfc3272.txt
3. ReKhter, Y., Li, T.: A Border Gateway Protocol. http://www.ietf.org/rfc /rfc1771.txt
4. Quoitin, B., Uhlig, S., Pelsser, C., et al.: Interdomain traffic engineering with BGP. IEEE Com-munications Magazine, 2003,41(5):122 128
5. Feamster, N., Borkenhagen, J., Rexford, J., et al.: Guidelines for Interdomain Traffic Engineering. ACM SIGCOM Computer Communications Review , 2003,33(5), 19-30
6. Hema Tahilramani Kaur., et al.: Outbound Load Balancing in BGP Using Online Simulation. http://network.ecse.rpi.edu/ hema/papers
7. Uhlig, S., Bonaventure, O.: Designing BGP-based outbound traffic engineering techniques for stub ASes. ACM SIGCOMM Computer Communication Review, 2004,34(4): 89-106
8. Fortz, B., Thorup, M.: Internet Traffic Engineering by Optimizing OSPF Weights. In Proc. IEEE INFOCOM 2000 Piscataway, NJ, USA v(2):519-528
9. Huston, G.: Interconnection, peering and settlements -part. Internet Protocol Journal, 1999, 23(3): 45-51
10. Gao, L.: On inferring autonomous system relationships in the Internet. IEEE/ACM Transactions on Networking, 2001,Dec:733-745

Congestion Management of IP Traffic Using Adaptive Exponential RED

S. Suresh and Özdemir Göl

School of Electrical & Information Engineering, University of South Australia,
Mawson Lakes Campus, Adelaide, South Australia-5095, Australia
Tel. No.: +61-8-8302-3241
pessuresh@yahoo.com

Abstract. In an IP network, if the source rates are increased beyond the service rates of the routers, then queues of packets waiting to be routed at the buffers, build up and exceed the buffering capacity of these routers leading to packets getting dropped. This results in low throughput and congestion collapse. In such networks, an AQM mechanism manages queue lengths in buffers and enables the end-systems to react to such losses by reducing their packet rate, avoiding severe congestion. Random Early Detection (RED) is one of the first AQM mechanisms to be used to avoid congestion in this manner. In this paper, the existing Normal and Gentle RED algorithms of Floyd as well as the justification for the proposed modified exponential RED algorithm have been discussed along with the results obtained on the functioning of the algorithms. Functioning of the algorithm proposed has also been tested using ns2 Simulator.

1 Introduction

In this paper, we present the results of one part of our research work in the area of traffic congestion management in IP networks [1], [2], [3], [4]. In an IP network, packets generated by a source are delivered to their destination by routing them via a sequence of intermediate routers. If the source rates are increased without constraint, say, beyond the service rates of the routers, then queues of packets waiting to be routed at the buffers of routers, build up leading to high delay. Eventually, the buffering capacity of these routers is exceeded and packets are dropped. This results in low throughput and congestion collapse [5]. In such networks, an AQM mechanism [6] manages queue lengths in buffers by dropping (or marking) packets before the buffer gets full. The end-systems can then react to such losses by reducing their packet rate, thus avoiding severe congestion. Random Early Detection (RED) [7], [8], [9] is one of the first AQM mechanisms that was proposed and has been mostly used to avoid congestion by randomly discarding packets based on the comparison of the average queue length size with two thresholds. In this paper, the existing Normal and Gentle RED [10] [11] algorithms of Floyd as well as the justification for proposing a modified exponential RED algorithm by us have been discussed. The rest of the paper has been organized as follows. In section 2, the existing Normal and the Gentle RED algorithms of Floyd are explained briefly as well as the justification for the modified algorithm proposed by us. Section 3 presents some results obtained on the

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 580 – 589, 2005.

verification of the proposed algorithm, using simulated data for the two cases, viz., P(Ma) – the marking/dropping probability, varying with single slope (Normal RED), and exponentially with queue lengths. Section 4 presents the details of the results on the functioning of the proposed algorithm using ns2 simulation. Section 5 is the concluding section.

2 Congestion Management and the RED Algorithm

In this section, we will briefly discuss the two algorithms proposed by Floyd viz., Normal RED and Gentle RED.

2.1 Normal RED

Floyd [7], [8], first proposed Normal Random Early Detection (RED) for congestion avoidance through controlling the average queue size. This RED scheme is implemented using two separate algorithms. The first algorithm computes the exponential weighted mean average (EWMA) of the queue size (L_a), which determines the degree of burstiness that will be allowed in the router queue. The second algorithm enables comparing the average queue size (L_a) with to two queue length thresholds viz., a *minimum* (L_{min}) and a *maximum* (L_{max}). When the average queue size is less than L_{min}, no packets are marked and also when the average queue size is greater than L_{max}, every arriving packet is marked and if the source is not co-operative, then they are dropped. But when the average queue size is between L_{min} and L_{max} each arriving packet is marked with a probability $P(M_a)$, where $P(M_a)$ is a function of the average queue size L_a, L_{min} and L_{max}. Floyd has proposed a typical marking function $P(M_a)$ as:

$$P(M_a) = 0 \text{ for } L_{min} > L_a \tag{1}$$

$$P(M_a) = F(L_a - L_{min})/(L_{max} - L_{max}) \quad \text{for } L_{max} > L_a \geq L_{min} \tag{2}$$

$$P(M_a) = 1 \text{ for } L_a \geq L_{max} \tag{3}$$

In eqn.(2), F can be any value between 0 and 1 and is same as 'maxP' in the equation proposed by Floyd in his normal RED. This function is being introduced to vary the marking probability from 0 corresponding to L_{min} to a maximum of 1 corresponding to L_{max}.

2.2 Gentle RED

It has been mentioned [10] that RED routers perform best when the packet marking probability changes fairly slowly as the average queue size L_a changes, May, et.al, [12], [13] explain the interaction between the sharp edge in the dropping function and the average queue size and they have recommended avoiding the sharp edge in the dropping function. In the context, the 'gentle RED' algorithm that has been suggested by Floyd [10] in which the marking/dropping probability of a packet varies from 0 to $P(M_a)$, in two rates/slopes, viz., with the first rate when L_a is between L_{min} and L_{max1},

from 0 to some value of $P(M_a)_1$ (<1) and then with a second higher rate between L_{max1} and L_{max2}, from $P(M_a)_1$ to $P(M_a)_2$ (a maximum of 1). This is possible by fixing two values for F – one lower value F_l between L_{min} and L_{max1} and another higher value for F_h, between L_{max1} and L_{max2}. Then the algorithm for the Gentle RED functioning between L_{min} and L_{max2} can be written as under:

$$P(M_a) = 0 \text{ for } L_{min} > L_a$$

$$P(M_a)_1 = F_l[L_a - L_{min}]/[L_{max1} - L_{min}] \quad (4)$$
$$\text{for } L_{max1} > L_a \geq L_{min}$$

$$P(Ma)_2 = F_h[L_a - L_{max1}]/[L_{max2} - L_{max1}] \quad (5)$$
$$\text{for } L_{max2} > L_a \geq L_{max1}$$

$$P(M_a) = 1 \text{ for } L_a \geq L_{max2} \quad (6)$$

In the gentle version of RED [10], [11] proposed the drop probability varies from $P(M_a)_1$ to 1 when the average queue size varies from L_{max1} to 2 L_{max1} .

2.3 Piecewise Linear RED

The natural and logical thinking would seem to be to increase the number of segments from two to say N, for example, N = 5, in the characteristic. All of them will then be piecewise linear. In the example explained below (five segments), the marking/dropping probability of a packet varies from 0 to $P(M_a)$, in five rates/slopes, viz., with the first rate when L_a is between L_{min} and L_{max1}, from 0 to some value of $P(M_a)_1$ (<1) and then with a second higher rate between L_{max1} and L_{max2}, from $P(M_a)_1$ to $P(M_a)_2$ and the last one will have the highest rate between L_{max4} and L_{max4}, from $P(M_a)_4$ to $P(M_a)_5$. Here as explained in the previous section it could be from $P(M_a)_4$ to 1. This is possible by fixing five values for F – one lowest value F_1 between L_{min} and L_{max1} and another higher value for F_2, between L_{max1} and L_{max2} and the last segment having the highest slope. Then the algorithm for the piecewise linear RED functioning between L_{min} and L_{max5} can be written as below:

$$P(M_a) = 0 \text{ for } L_{min} > L_a$$

$$P(M_a)_1 = F_1[L_a - L_{min}]/[L_{max1} - L_{min}] \quad (7)$$
$$\text{for } L_{max1} > L_a \geq L_{min}$$

$$P(Ma)_2 = F_2[L_a - L_{max1}]/[L_{max2} - L_{max1}] \quad (8)$$
$$\text{for } L_{max2} > L_a \geq L_{max1}$$

$$P(Ma)_3 = F_3[L_a - L_{max2}]/[L_{max3} - L_{max2}] \quad (9)$$
$$\text{for } L_{max3} > L_a \geq L_{max2}$$

$$P(Ma)_4 = F_4[L_a - L_{max3}]/[L_{max4} - L_{max3}] \quad (10)$$
$$\text{for } L_{max4} > L_a \geq L_{max3}$$

$$P(Ma)_5 = F_5[L_a - L_{max4}]/[L_{max5} - L_{max4}] \quad (11)$$
$$\text{for } L_{max5} > L_a \geq L_{max4}$$

$$P(M_a) = 1 \text{ for } L_a \geq L_{max5} \quad (12)$$

As mentioned in the gentle version of RED, in this case also the drop probability can be varied from $P(M_a)_4$ to 1, when the average queue size varies from L_{max4} to L_{max}.

3 Adaptive Exponential RED Algorithm

It has already been mentioned that RED routers perform best when the packet marking probability changes fairly slowly with the average queue size L_a and also the interaction between the sharp edge in the dropping function and the average queue size and the recommendation of May, et.al., for modifying the normal RED algorithm. In the consequent 'Gentle RED' also we see sharp edge although it has two slopes. The piecewise linear version is better than Gentle in this respect since the slopes vary gradually. Taking the above two schemes into consideration and also that RED routers perform best when the packet marking probability changes fairly slowly initially as the average queue size L_a changes, and then increases rapidly, a requirement taken from Gentle RED, it is felt that changing the marking/dropping probability from 0 to 1 gradually as an exponential function would be advantageous. Also the algorithm to be used in piecewise linearisation RED becomes more complex as the number of segments gets increased. We hence propose changing the factor F, as an exponential function of L_a, taking into account the values of L_{min} and L_{max}, as boundary values, such that at L_{min} the value of F is 0 and at L_{max} it is 1. This has been done as follows.

3.1 Exponential RED Algorithm

Let F_e be taken as an exponential function of F, given by

$$F_e = F(e^{\beta})/(e^{p}) \tag{13}$$

$$\text{Where } \beta = p*(L_a - L_{min})/L_{max} - L_{min}) \tag{14}$$

The value of p (varying from 0 to say 5) decides the amount of concavity in the characteristic of $P(M_a)_e$ with queue length. As concavity is more we get better packet marking compared to lesser concavity [12],[13]. F = 1, is the maximum value reachable when $L_a = L_{max}$. We now propose the exponential packet-marking algorithm based on queue length, functioning between L_{min} and L_{max}, for this case as under.

$$\begin{aligned} &P(M_a)_e = 0 \text{ for } L_{min} > L_a \\ &P(M_a)_e = F[(e^{\beta})/(e^{p})]\ [L_a - L_{min}]/[L_{max} - L_{min}] \\ &\text{for } L_{max} > L_a \geq L_{min} \\ &P(M_a)_e = 1 \text{ for } L_a \geq L_{max} \end{aligned} \tag{15}$$

From the above equations, it can be seen that when $L_a = L_{min}$, $\beta = 0$ and hence $F_e \approx 0$. Also when $L_a = L_{max}$, $\beta = p$ and hence $F_e = F$. Since the denominator in F_e, viz., (e^p) is a constant for a given p, the value of F_e increases exponentially from ≈ 0 to a maximum value of F, between L_{min} and L_{max}. Accordingly $P(M_a)_e$, also increases exponentially as given by equation (15). It can also be seen that the instantaneous slope adapts to the instantaneous average queue value. Also when p =0, the algorithm reduces to the one given by the Normal RED.

4 Functioning of the Exponential RED Algorithm

We will now consider the functioning of the four algorithms viz., Normal RED, Gentle RED, Piecewise Linear RED and Exponential RED, which have been tested using simulation for normalised queue lengths in the range of 0 to 1. For these tests, L_{min} is taken as 0.25 and L_{max} is taken as 0.75. The maximum value of $P(M_a)$ has been taken as 0.2 for this testing example.

The verification of the four algorithms has been done through simulation using Matlab. We have taken for example, F1 = 0.2 for our calculation, in respect of the Normal RED, although the recommended value by Floyd is 0.1. The Values in Table 1, are for F1=0.2 (Normal RED), F2 = 0.05, F3 = 0.15 (Gentle RED), F4=0.02, F5=0.03, F6=0.04, F7=0.05, F8=0.06 (Piecewise RED) and F9=0.2 and p=1 (Exponential RED).

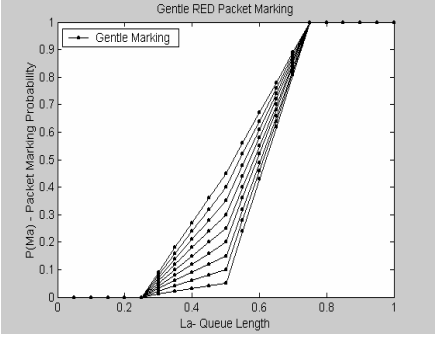


Fig. 1. Graphs for the four algorithms

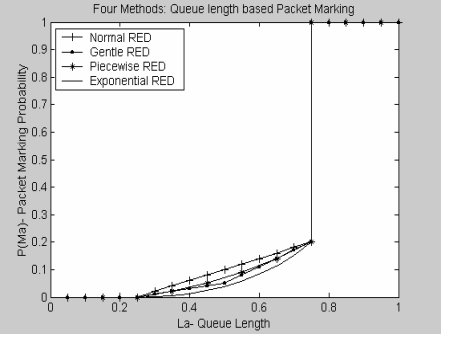


Fig. 2. Gentle RED variation with slopes

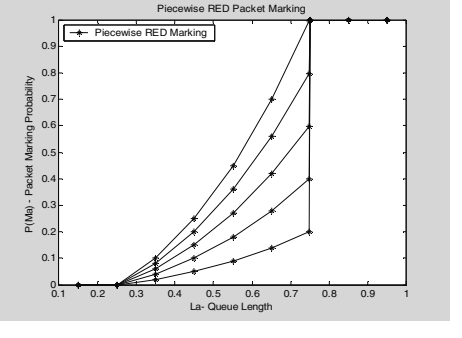


Fig. 3. Piecewise RED-slope variation

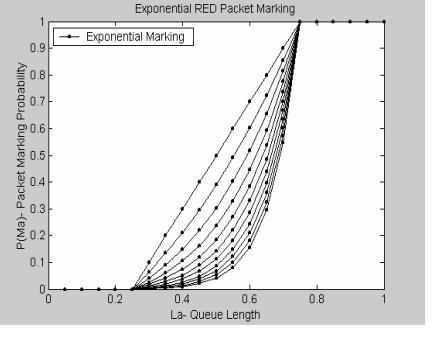


Fig. 4. Exponential RED-slope variation

Fig.1 shows all the four graphs obtained for $P(M_a)$ for various normalised queue lengths and for F=0.2, the maximum value of $P(M_a)$. Furthermore, Fig. 2 provides details on Gentle RED for varying values of F2 and F3 in steps of 0.05, when the

corresponding F1 for the Normal RED is kept as 1. Fig.3 similarly provides the details on Piecewise Linear RED for various values of F1 starting from 0.2 to 1, and Fig. 4 provides details on varying values of p for exponential RED, which is for varying the concavity, again keeping the corresponding F1 for the Normal RED as 1.

Table 1. F1 = 0.2., F2 = 0.05, F3 = 0.15., F4=0.02, F5=0.03, F6=0.04, F7=0.05, F8=0.06., F9 =0.2 and p = 1

L_a (Normalised)	Normal P(Ma) F1	$P(M_a)_1$: Gentle (F2)	$P(M_a)_2$: Gentle (F3)	Gentle $P(M_a)$ = $P(M_a)_1 + P(M_a)_2$	Piecewise Linear P(Ma) F4,F5,F6, F7,F8	Exponential $P(M_a)$ F9
0.25	0.0000	0.000	------	0.000	0.0000	0.0000
0.30	0.0200	0.010	------	0.010	-------	0.0012
0.35	0.0400	0.020	------	0.020	0.0200	0.0052
0.40	0.0600	0.030	------	0.030	-------	0.0122
0.45	0.0800	0.040	-------	0.040	0.0500	0.0229
0.50	0.1000	0.050	------	0.050	-------	0.0378
0.55	0.1200		0.030	0.080	0.0900	0.0574
0.60	0.1400		0.060	0.110	--------	0.0826
0.65	0.1600		0.090	0.140	0.1400	0.1141
0.70	0.1800		0.120	0.170	--------	0.1529
0.75	0.2000		0.150	0.200	0.2000	0.2000

4.1 Validation of Two Algorithms

To analyse the results further, as the simulation platform, we used the Network Simulator 2 (ns2) [14]. Different scenarios were simulated, using the configuration shown in Fig.5. In this simulation we have compared the linear length based RED and our proposed algorithm for their behaviour by simulating CBR and Poisson input through the use of two sources S_1 and S2 in ns environment. L is the bottleneck link connecting the routers R1 and R2, and D1 and D2 are the destination nodes. The CBR and Poisson queue lengths generated using this configuration as well as the shape of the curves for the final packet marking probability P_a [7], [8] values obtained using equation (16), are shown in Figs. 6 to 11.

$$P_a = P(M_a)/[1\text{-count}*P(M_a)] \tag{16}$$

In the above equation (16), P_a is the final marking probability computed based $P(M_a)$, the packet marking probability. The final packet marking probability P_a slowly increases as the count increases since the last marked packet. We have used the following algorithm for computing the P_a values in both cases.

4.1.1 Algorithm for P_a Computation

Initialisation
Avg ← 0
Count ← -1
For each packet arrival
$L_a = (1\text{-alpha})*L_a + (\text{alpha})*L_i$
If $L_a \leq L_{min}$
$P(M_a)_l = 0$
$P(M_a)_e = 0$
else
if $L_{min} \leq L_a < L_{max}$
count = count + 1
$P(M_a)_l = (F)*(L_a - L_{min})/(L_{max} - L_{min})$
$P_{a(l)} = P(M_a)_l/[1\text{-count}*P(M_a)_l]$
$d = (L_a - L_{min})/(L_{max} - L_{min})$
$F_e = F*(\exp(d)/2.718)$
$P(M_a)_e = (F_e)*(L_a - L_{min})/(L_{max} - L_{min})$
$P_{a(e)} = P(M_a)_e/(1\text{-count}*P(M_a)_e)$
count ← 0
else
if $L_a > L_{max}$
$P(M_a)_l = 1$
$P(M_a)_e = 1$
count ← 0
else count ← -1

In the above algorithm, L_i is the current queue size and L_a is the EWMA average of the queue size. Count is the number of packets marked since last packet marked. Alpha is the weight of the queue, which is a parameter of the low pass filter used for EWMA operation. L_{min} and L_{max} are the minimum and maximum threshold of the queue lengths selected. F denotes the maximum value for $P(M_a)$.The number 2.718 is the value of exp(d), when d = 1, i.e., when $L_a = L_{max}$.

4.2 Simulation Results

In all the simulations carried out using ns2, L_a has been kept within 20% and 60% of the normalised maximum buffer value. Fig.6 shows the variation in the computed packet marking probability $P(M_a)$ for a sampling duration of 60 secs, in respect of the linear length based RED and that of the proposed exponential RED. Figs. 7 to 10 show the generated queue and its EWMA values (in red line) for each type of traffic simulated using ns2 as well as the variations in the final marking probability (P_a) values. It can be seen from these results that the average queue length (shown in red line) follows the fluctuations of the instantaneous queue length. Because of this, the final marking probability of the linear length based as well as of the exponential RED, also fluctuates [15] as well.

We know that P_a is based on PM_a and count. Count gives the number of packets not marked or dropped since the last marked packet [7],[8],[16] when L_a is between L_{min} and L_{max}. Now we compare the values of P_a for Normal linear RED and the Exponential RED. From the graphs for P_a, it is evident that P_a in the case of Normal

linear RED is more as PM_a increases linearly. In contrast in Exponential RED PM_a increases exponentially instead of linearly and so the number of packets dropped or marked is less. Obviously the convergence is much better for Exponential RED when compared to Normal RED.

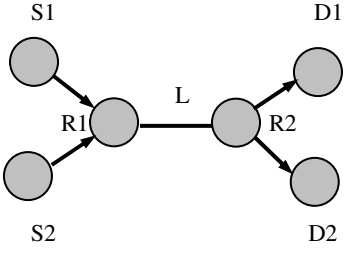


Fig. 5. Scheme used for ns2 simulation

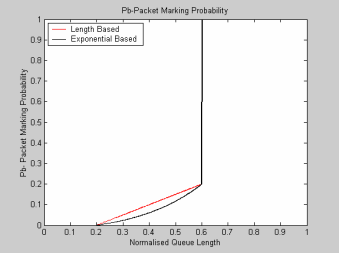


Fig. 6. $P(M_a)$ –vs- Normalised Length

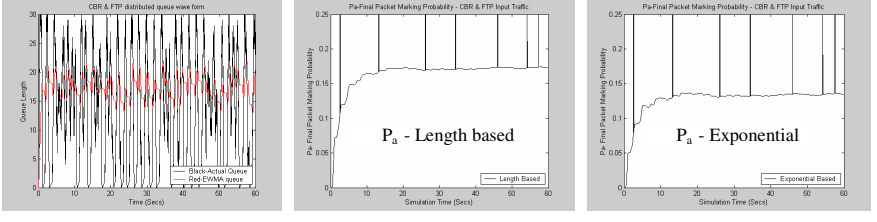


Fig. 7. 700 Kbps – 500 Kbps : CBR + FTP

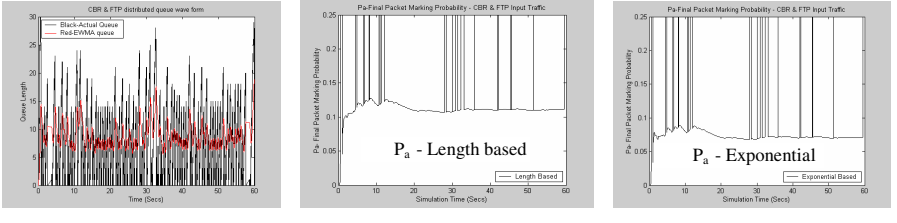


Fig. 8. 2 Mbps – 1 Mbps : CBR + FTP

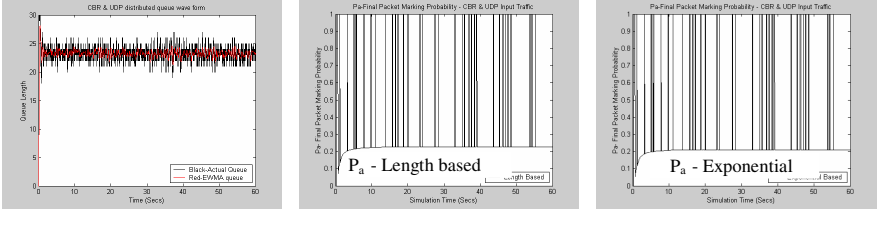


Fig. 9. 700 Kbps – 500 Kbps : CBR

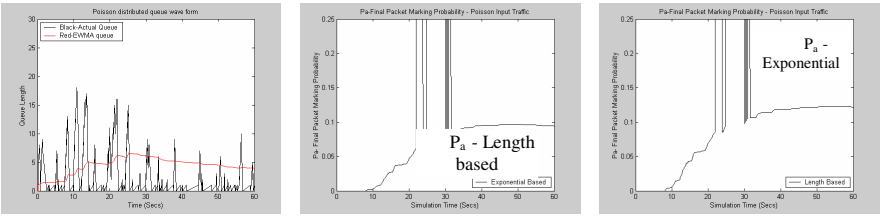


Fig. 10. 1 Mbps – 500 Kbps : CBR + Poisson

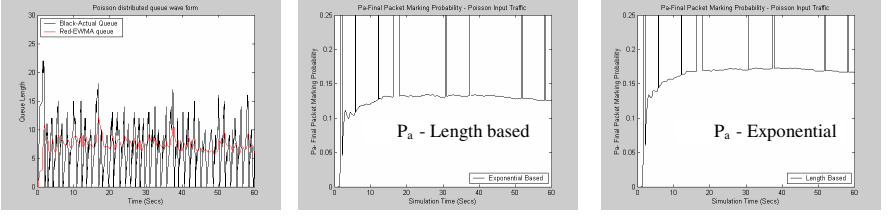


Fig. 11. 1 Mbps – 500 Kbps : FTP + Poisson

As can be seen from the appropriate figures of simulations shown for P_a, it has been found that the final packet marking is much better for exponential RED compared to that of the linear length based RED, for the traffic inputs, in view of the fact that converging value of P_a is lower [15] in the case of exponential RED than in the case of length based RED. In addition, the packet marking probability PM_a for the exponential RED, changes fairly slowly initially as the average queue size L_a changes, and then increases rapidly and there is no sharp edge throughout.

5 Conclusion

Schemes described in the literature on network congestion management are in general based on queue length management. In the context it may also be noted that the Floyd's RED algorithm fixes the value of the marking probability $P(M_a)$ as a function of the values of the average queue lengths. The value of $P(M_a)$ is therefore a linear function of the desired queue length L_a. It has been found that RED routers perform best when the packet marking probability changes fairly slowly as the average queue size L_a changes, and for this reason, it has been recommended avoiding the interaction between the sharp edge in the dropping function and the average queue size. With this in view, in this paper, we have proposed an exponential RED algorithm for traffic congestion management in IP networks. We first presented the basic scheme of normal RED as proposed by Floyd et.al, and then explained the modification to the algorithm proposed by Floyd called Gentle RED. Then we extended the concept of Gentle RED into piecewise linear RED. And then explained the new algorithm proposed by us called Exponential RED. Analysis of functioning of these algorithms has been done using Matlab. Also we have simulated traffic using ns2 simulator and for various combinations of the input traffic, passing through a bottleneck link. The

variations in the final marking probabilities with respect to time, in respect of linear length based RED of Floyd as well as for the exponential RED proposed by us, have been computed and the trends have also been shown graphically. From the results it is concluded, that the algorithm as proposed by us would give a better packet marking due to the increasing concavity and so we can expect better performance compared to the one proposed by Floyd. Result of these have also been tabulated and shown diagrammatically.

References

1. Challinor. S., "An introduction to IP Networks", BT Technology J, Vol.18, No.3., pp.15-22., July, 2000
2. Peterson. L.L. and Davie. B.S., "Computer Networks – A Systems Approach", II Edition, Morgan Kaufmann., pp.446-509, 2000.
3. Stallings. W., "Data and Communications", VI edition, Pearson Education Asia, pp.384-394, 2002
4. Tanenbaum. A.S., "Computer Networks", III edition, Prentice Hall, pp.374-395, 1997
5. Jain. R. and Ramakrishnan. K.K., "Congestion Avoidance in Computer Networks with a Connectionless Network Layer: Concepts, Goals and Methodology", Proceedings Computer Networking Symposium, Washington,D.C., pp.134 -143., April, 1988.
6. Haider.A., et.al., "Congestion Control Algorithms in High Speed Telecommuni-cation Networks"., www.esc.auckland.ac.nz/ Organisations/ ORSNZ/conf36/ Programme2001.htm
7. Floyd. S. and Van Jacobson., "Random Early Detection Gateways for Congestion Avoidance", IEEE/ ACM Transactions on Networking, August 1993
8. Floyd. S., "Random Early Detection (RED): Algorithm, Modeling and Parameters Configuration".,www.ece.poly.edu/aatcn/pres_reps/JTao_ RED_report.pdf.
9. Eddy..W.M, and Ostermann.S., "A Review of Research in RED queueing or RED in a Nutshell", Shawnroland.grc.nasa.gov/~weddy/papers/ redreview.ps
10. Floyd. S., "Description of gentle mode in NS', http:/www.icir.or/floyd/notes/test-suitered.txt
11. Orozco.J., and Ros.D., "An adaptive RIO queue management algorithm", Internal Publication No.1526, April 2003, IRISA, France.
12. May. M., et.al., "Influence of Active Queue Parameters on Aggregate Traffic Performance", Tech.Report.No. 3995, INRIA, Sophia Antipolis, France, 2000
13. May. M et al., "Reasons not to deploy RED", Proc.IEEE/IFIP/WQoS '99, June 1999.
14. Network Simulator, http://www.isi.edu/nsnam/ns/, 2002.
15. Wu-chang Feng, et.al., "The BLUE Active Queue Management Algorithms"., IEEE/ACM Transactions on Networking, Vol. 10, No. 4, August 2002
16. Hui Zhang., "A discrete-time model of TCP with active Queue management"., Master of Applied Science -Thesis, School of sEngineering Science., Simon Fraser University, Canada, August 2004

An Analysis and Evaluation of Policy-Based Network Management Approaches*

Hyung-Jin Lim[1], Dong-Young Lee[1], Tae-Kyung Kim[1], and Tai-Myoung Chung[1]

[1] Internet Management Technology Laboratory and Cemi: Center for Emergency Medical Informatics, School of Information and Communication Engineering, Sungkyunkwan University, Chunchun-dong 300, Jangan-gu, Suwon, Kyunggi-do, Republic of Korea
{hjlim, dylee, tkkim, tmchung}@rtlab.skku.ac.kr

Abstract. This paper provides an analytical framework for comparison between centralized and distributed models of policy-based network management (PBNM). Policy-based networks are being deployed in a variety of applications, in which the policy-enabled architecture may provide appropriate performance control depending on the particular application being used. Therefore, in this paper, each PBNM model is evaluated based on the performance and scalability provided when a policy is provisioned. Our metrics include a qualitative evaluation of the policy provisioning time, the amount of traffic measured between the PBNM components and PDP utilization. Each approach has its own merits and demerits depending on the particular application requirement. We highlight the fact that an appropriate architecture needs to be introduced, in order to optimize the PBNM deployment. The PBNM modeling framework presented in this paper is intended to quantity the merits and demerits of the currently available methodologies.

1 Introduction

Most of the currently available network management systems which are based on SNMP (Simple Network Management Protocol) or CMIP (Common Management Information Protocol) may provide improved efficiency from an administration perspective. The main network management technologies, however, such as fault detection, correction and restoration, are insufficient for complicated network environments and fail to satisfy many of the users' requirements [1]. In order to satisfy these requirements, PBNM (Policy Based Network Management), which is an extension to the existing network management architecture, has been proposed [2]. PBNM started as a network control technology, which was initially deployed in the field of QoS and other security applications [4].

PBNM-based network management has been presented as 2-tiered architecture by the IETF. However, A. Corrente [7] pointed out various PDP bottleneck issues which

* This study was supported by a grant of the Korea Health 21 R&D Project, Ministry of Health & Welfare, Republic of Korea. (02-PJ3-PG6-EV08-0001)

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 590 – 599, 2005.

were caused by the complicated policy structure inherent in 2-tiered architectures. In addition, there has been some discussion of 3-tiered architectures [8] in the IETF standardization working group for QoS deployment. Eddie Law [9] pointed out that the 2-tiered architecture has problems with scalability and Policy Decision Point (PDP) bottlenecks, and proposed an alternative 3-tiered architecture. However, he only investigated the case of PDP utilization that is provided when a policy is implemented among PBNM components within a single domain. The performance of traditional network management systems is dependent upon the processing utilization, the processing load on the NMS (network management system) and the amount of management traffic [1], while the performance of PBNM as a control system may depend on its ability to address policy conflict resolution and distribution problems, as well as issues involving policy representation and device-level policy interpretation. Furthermore, in a multi-PDP environment, to what extent security and policy conflict might influence such performances should also be investigated.

In this paper, various approaches to PBNM are analyzed, involving a single architecture, hierarchical architecture and hybrid-tiered architecture. The architectural efficiency of each model is also validated, based on the performance metrics. Section 2 classifies the models proposed for PBNM. In Section 3, we investigate some of the metrics which influence the performance of the PBNM model. Section 4 includes an analysis of each model. Finally, in the last section, we present our conclusions.

2 Model and Assumptions

2.1 Model

A PBNM model is usually composed of a PDP, Policy Enforcement Point (PEP), policy repository and management console. The key component of the PBNM system is the PDP, which is primarily responsible for controlling the networks [5]. A policy created from policy management tools should be saved in the policy repository and the PEP causes a device to activate such a policy from the repository. There should also be some synchronization based on notification among the PDP, PEP and policy repository. Fig. 1 describes the various policy-based network architectures [4].

Fig 1. (a) refers to a PBNM model in a single box. This basic architecture involves traditional network equipment or systems. Since this approach contains all of the PBNM components within a single piece of equipment, there is no need for any communication protocol between the PDP and PEP, and each policy defined by the manager is saved in the network equipment.

Fig.1 (b) presents a centralized repository and distributed PDP model. In this approach, each device has its own PDP module or, alternatively, those nodes not having PDP modules provide those having remote PDP with the policies. Since PDP and PEP are located in the same device, policy validation and enforcement should be done at each PDP rather than at the domain level. Fig.1 (c) has a policy repository and PDP deployed on a centralized basis, while the PEP is separately located within each device. This model constitutes the basic framework proposed by the IETF. In this architecture, any request to validate changes in the network status or policy conflicts is collected from the NMS and forwarded to the PDP.

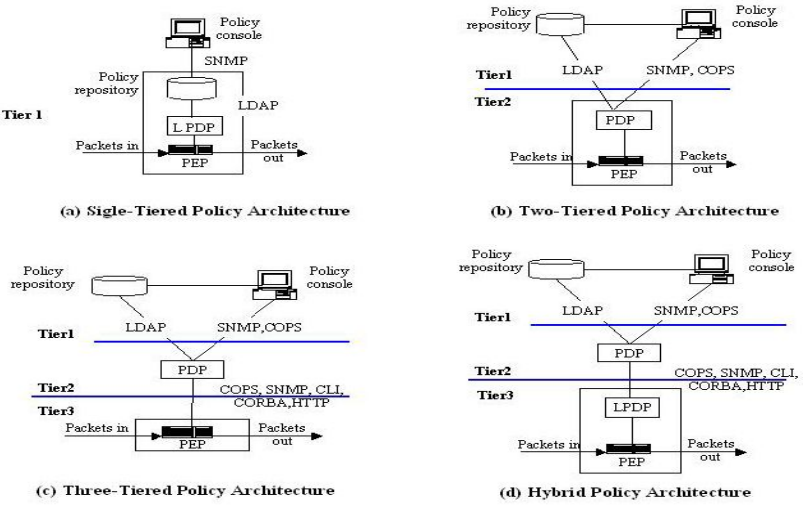


Fig. 1. Policy-based Network Management Model

Generally, a policy originating from a policy repository may be distributed from the PDP to low-level equipment, such as network devices, as PEP. When a distributed PDPs exist a policy conflict arising among the PDPs needs Global Conflict Detection (GCD) validation [8]. The hybrid policy architecture shown in Fig.1 (d), which is a hierarchical extension of the three-tired policy architecture, has an LPDP (local PDP) located within each device. The PEP uses the LPDP for local policy decisions. The communication between the PDP and PEP is carried out using CLI, SNMP, HTTP, CORBA, COPS, etc., while LDAP, SNMP, CLI, etc., may be used between the repository and the PDP.

2.2 Assumptions

We assume that the COPS protocol is used for the policy transfer between the PDP and PEP and that there is an application in the network, such that all resources accept and activate any policy. The performance of the policy repository acts as an independent factor, having no direct impact on policy provisioning. Therefore, it is not assumed that the performance is influenced by how distributed the repository is. Since the PDP is usually deployed as a policy server, it is assumed that it owns most of the policies saved in the policy repository. The PDP and PEP share a temporary repository called the PIB, which is used whenever the policy conversion process needs to be activated. The PIB in the PDP contains those policies applicable to the PEP. Therefore, if i commands are configured in the PEP, there will be i policies shared by the PDP and PIBs. In the case where there is no specific policy in the PDP, the necessary policy should be created either manually by the manager or automatically.

Table 1. Performance Variables

Variables	Description
Iq	Size of request message between PDP and PEP
Ir	Size of response message between PDP and PEP
Tc	Processing time for policy request and response at PDP
Tq	Processing time for policy request at PEP
Tr	Processing time for policy response at PEP
Sd	Average time required when searching for a policy in PDP (=Q/2)
Pd	Average processing time to convert into PIB format in order to recognize the policy transmitted from PDP to PEP
Td	Transmission time required to transfer data from PDP to PEP
Pe	Average processing time required to convert into device-specific format(i.e., commands), in order for the policy(i.e. PIB) to be executed by the device
Sp	Average time required to detect a policy conflict (= $k \times Q^2$)
H	Header size of control message

Generally, the time taken to finish a specific task may be a performance factor. For the purpose of modeling any irregular delay in the network, the packet delay is defined as a random value, T(N), depending upon the network size, which is assumed to be N nodes. Although the packet processing time is actually dependent on the load on processor, a constant value is generally used.

3 Evaluation of Performance Metric

In PBNM architectures, policy provisioning may be performed when an event policy request is received or a new policy needs to be activated at the PDP. Any event can have Fault Management, QoS, Security Violation and Scheduled Job properties according to the PBNM application. Network events requiring policy provisioning at the PDP are assumed to occur randomly in the form of a Poisson process with rate λ. (i.e., times between in λ terval are independent exponential random variables with mean $1/\lambda$). That is, λ is the frequency at which policies are created as a result of network status changes or service request signaling. The metrics having influence on provisioning performance in each PBNM are as follows.

T_i : Average processing time required for policy provisioning

U_i : Average utilization of PDP for policy provisioning

C_i : Average amount of traffic measured as the capacity at the PDP and PEPs

3.1 Single and Two-Tiered Policy Architecture; S & TT

Single-tiered and Two-tiered policy architectures can be considered as a same model such both architectures have PDP embedded together with PEP at the same device. Each device has its own policy server independently. In a PBNM model, there is some performance variables related to policy provisioning as follows Table 1.

When policy provisioning is performed as a result of an event, the PEP transfers the policy request to the PDP. The PDP then verifies the existence of the policy (Sd) in its own repository and performs policy conflict detection based on the verified policy (Sp). When the PDP searches the existence of the policy, assuming the probability of discovering the Q_n th policy from all Q policies is P(Q_n) = $(Q-Q_n)/(Q+1-Q_n)$, the average search time for Q policies will be Q/2 [11]. Also, the PDP checks whether the policy conflicts with other activated policies (sp). Regarding the time required for the detection of conflicts regarding information saved in the PIB, when there are Q policies with k independent property types, the average verification time (Sp) will be O(kQ^2) [11]. After that, the policy will pass through PIB conversion (Pd) and be transferred to the PEP, which will activate the policy in the form of a device-recognizable command.

In the S&TT architecture, since the PDP and PEP are both located in the same device, there is no need to consider the transmission time or propagation time between the PBNM components. The provisioning time in Eq. (1) is the time required to activate a policy transferred from the PDP to the PEP when an event requiring a network policy occurs.

$$T_1 = \mathrm{Pd} + \mathrm{Pe} + \mathrm{k} \times Q^2 \tag{1}$$

Eq. (2) defines the average utilization of the PDP when the PDP processes any event and performs policy provisioning. Since the PBNM module is contained in one device, no message processing time is required.

$$U_1 = \lambda \times \left(\frac{Q}{2} + \mathrm{Pd} + \mathrm{k} \times Q^2\right) \tag{2}$$

The PDP and PEP are located in the same device, so there is no network traffic related to policy provisioning. If any policy conflict detection is needed among the PDPs, however, the issue of network traffic will need to be considered. In this case, when a policy is changed at a particular node and, consequently, other nodes including the PDP have to perform policy conflict detection, it is assumed that the N-1 nodes will do the job sequentially in round-robin fashion. The NMS traffic required for monitoring is not considered here. Therefore, the overall traffic will consist of the control traffic caused by the policy request and response. If each policy is of the same size, we can evaluate the amount of traffic measured among the PDPs, as follows.

$$C_1 = (\mathrm{N}\text{-}1) \times \lambda \times (\mathrm{Iq} + \mathrm{i} \times \mathrm{Ir} + 2 \times \mathrm{h}) \tag{3}$$

3.2 Three-Tiered Policy Architecture; TT

In the TT model, the N-node network is divided into L subnetworks each containing one PDP, in which case each PDP performs policy provisioning for N/L nodes. Each PDP can configure an administration domain, so the security header (Hs) on a transiting message should be filled in. The security header (e.g. IPSec) provides confidentiality, integrity and authentication information for the transfer of data. If L=1, the security header can be ignored. That is, *Hs* represents the overhead of the security header per packet, and *Ts* is the cost of the additional processing required for encryption and authentication at each node. When the PDP performs new policy

provisioning, the processing time in the TT architecture includes costs such as the policy transfer time from the PDP to the PEP, the security header processing/propagation time (i.e., Td+T(N/L)+2×Ts) and the response processing time for the request from the PEP to the PDP (i.e., Tq+2×Tc+Tr), as compared with the S&TT case. Therefore, the TT architecture has to include the processing time required for new policy provisioning, which can be written as Sd+Pd+Td+T(N/L)+Pe+2×Ts+Sp.

$$T_2 = \text{Tq}+2\times\text{Tc}+\text{Td}+\text{Pe}+\frac{Q}{2}+\text{Pd}+T(\frac{N}{L})+2\times\text{Ts}+\text{Tr}+\text{k}\times Q^2 \quad (4)$$

When a policy request is generated by the PEP, the PEP has to request the policy be transferred from the PDP. In the TT architecture, the policy provisioning time includes the processing time required for the transmission of the data from the PDP. Therefore, the time required to transfer the data to the PEP (Td) and the time required to process the request message (Tc) constitute additional overhead, as compared with the S&TT case. In addition, the security header encryption/decryption time (Ts) should also be considered.

$$U_2 = \lambda \times \frac{N}{L} \times (2\times\text{Tc}+\text{Td}+\frac{Q}{2}+\text{Pd}+2\times\text{Ts}+\text{k}\times Q^2) \quad (5)$$

During policy provisioning, the amount of traffic between the PDP and PEP will be i × Ir+h (where i=the number of policies provisioned). Meanwhile, the request/response message headers for these *i* policies will involve traffic from N (L≥1) nodes $((N/L)\times\lambda\times(Iq+I\times Ir+2\times h)$. Here, we do not consider the traffic at other PDPs, in order to focus on the GCD at a specific PDP. If an individual administration domain is composed of multiple PDPs, a secure channel will be configured between the sub-domains, generating security header (Hs) overhead.

$$C_2 = \lambda \times \text{N}\times(\text{Iq}+\text{i}\times\text{Ir}+2\times\text{h}+2\times\text{Hs}) \quad (6)$$

3.3 Hybrid-Tired Policy Architecture (HT)

In the HT architecture, the processing time depends on the number of activated policies at the Local PDP (LPDP). While the PDP acts as a policy server, the LPDP generally only has a limited number of policies, due to its small memory capacity, so that the number of LPDP policies (q) is equal to or less than the number of PDP policy (Q). In the case of new policy provisioning, the processing time will be the same as that in the TT architecture (i.e., Sd+Pd+Td+T(N/L)+Pe+2×Ts+Sp), since the policy does not exist in the LPDP. In the case of an external event-driven policy request, however, the provisioning performance may depend on whether or not the policy is included in LPDP. Therefore, the comparison phase and policy search should be considered in the case of both the LPDP and PDP, although the consumed time may vary depending on the algorithm involved.

The PEP applies the policy as soon as it receives the requested policy from the LPDP, and only receives an Acknowledgement from the PDP after the validation of the detection of a policy conflict. If the PEP cannot obtain the policy from the LPDP, it creates a new policy and performs policy conflict detection at the PDP, and then

applies it to the PEP. So if $P_{\Pr obability}$ is the probability that the requested policy exists in the LPDP, the processing time can be written as follows.

$$T_3 = P_{\Pr obability} \times \{(\frac{Q}{2} + Pd + Pe)\} + (1 - P_{\Pr obability})$$
$$\times \{(\frac{Q+q}{2} + Td + Pd + Pe + T(\frac{N}{L}) + 2 \times Ts + k \times Q^2 + Tq + Tr + 2 \times Tc\} (Q \geq q) \quad (7)$$

In the TT architecture, the utilization of each LPDP is not considered, since it does not have any direct influence upon the performance of the PDP. When a policy originating from the LPDP is identified, the PDP does not perform GCD and only needs to check whether the policy exists or not.

$$U_3 = \lambda \times \frac{N}{L} \times \{P_{probability} \times (2 \times Tc + \frac{Q}{2} + 2 \times Ts) + (1 - P_{probability})$$
$$\times (\frac{Q+q}{2} + 2 \times Tc + Td + Pd + 2 \times Ts + k \times Q^2\} \quad (8)$$

The TT architecture always performs GCD during the policy provisioning process. This is the same as the traffic issue. The probability that the requested policy exists in the LPDP, $P_{\Pr obability}$ may have an influence upon the occurrence rate used for provisioning traffic. In the case where there is no such policy in the LPDP, additional traffic will be provided during the GCD process performed by the PDP.

$$C_3 = \frac{N}{L} \times \lambda \times \{P_{probability} \times (Iq + Ir + 2 \times h + 2 \times Hs)$$
$$+ (1 - P_{probability}) \times (L \times (2 \times h + 2 \times Hs + Iq + i \times Ir) + (1 - i \times Ir))\} \quad (9)$$

4 Analysis of Evaluated Results

4.1 Policy Provisioning

In a network composed of Multiple PDPs, the provisioning time is affected by the average propagation time, due to there being one sub-administration domain for each PDP. However, in the S&TT model, no such costs are incurred between the PDP and PEP. As in the distributed model, the TT model requires a greater transmission time, which can be written as = Tq+2 Tc+Tr+Td+2 Ts+T(N/L).

The LPDP reference rate is related to the provisioning cost in the HT model. Compared with TT model, the HT model with a high LPDP reference rate can reduce the provisioning cost, as described by $\Delta_2 = \Delta_1 + k \times Q^2$, while in the case of a low reference rate the provisioning cost may be increased by Δ_3=q/2 during the search time in the LPDP. This variation in the cost depending on the model results from the

transmission cost between the PDP and PEP (i.e., Δ_1), the policy conflict detection time (i.e., $\Delta_2 - \Delta_1$) and the average time (i.e., Δ_3) required to search for a policy held in the LPDP. In particular, in the case of the HT model, the policy provisioning time varies according to the LPDP reference rate ($P_{\Pr obability}$) can define the policy provisioning time as follows:

$$T_3 = P_{\Pr obability} \times (T_1 - (\Delta_2 - \Delta_1)) + (1 - P_{probability}) \times (T_2 + \Delta_3) \qquad (10)$$

That is, the effect of the HT model in the provisioning cost prospective depends on the reference rate.

4.2 PDP Utilization

The use of PDPs affects the network control performance in the distributed PBNM model, since the entire network has to be monitored and controlled by a PDP control domain. It has been found that the performance associated with PDP utilization varies according to the size of the network, the policy occurrence rate and the specific PBNM model being used. Therefore, the PBNM performance depends on the PDP capability in terms of the provisioning for policy requests and policy conflict detection. If we suppose that in the S&TT model, a node including PDP manages the policies being requested by other nodes (N-1) at a rate of λ , we should consider the communication cost as $2 \times$Tc+Td and a number of communication as (N-1)$\times \lambda$. Therefore, it is shown that the exponential overhead cost occurs as the network size and the event occurrence rate (λ) increase. As a distributed architecture, the TT model requires the least propagation cost(i.e., T(N/L)) since it controls N/L nodes per PDP, however the total processing cost involves more overhead, due to the cryptography process required for the additional security header(e.g., IPSec), the processing required for requests and responses and the data transmission. In the HT model, the LPDP reference rate has an influence on the PDP utilization. Consequently, the PDP utilization cost is in the order HT $\leq$ S&TT<TT when the LPDP reference rate is a high, but is in the order as S&TT<HT $\leq$ TT, in the case of a low LPDP reference rate. It is assumed that for a given set of parameters affecting the PDP utilization, the processing cost (i.e., $2 \times$ Tc+Td+Pd+$2 \times$ Ts) for the data transmission between the nodes is constant (ϕ>0), and the PDP utilization is fixed. Then, the number of policies that the PDP can afford to process is defined as:

$$Q \leq \sqrt{\frac{1}{k}\left(\frac{L}{\lambda N} - \phi\right)} \qquad (\text{L} > 0, \text{ integer}) \qquad (11)$$

4.3 Amount of Traffic

The variation in the amount of control traffic as a function of the network size (N/L) and the event rate λ has an influence on the capacity of the PDP to act as a control manager in the network management model. Traffic is also generated during GCD or

data transmission between the PBNM modules. The TT model shows that a relative decrease in the number of managed nodes for a uniform rate, λ causes the amount of control traffic to decrease. That is, when L is a large, a specific PDP has to perform GCD together with the other L-1 PDPs, in which case the amount of traffic can be written as (N/L)×(L-1)×(Iq+i×Ir+2×h). Independently of this specific PDP, the other PDPs also perform GCD, where the amount of traffic is (N/L)×(L-1) for a uniform rate λ rate.

However, the amount of traffic within the sub-network decreases starting with a specific PDP, not having an effect on the amount of overall traffic, due to their increase through GCD. Let us assume that ω, is the number of executions (L-1) of an algorithm referring GCDs to other PDPs. Then, if Eq. (6) in the TT model is able to produce an efficient GCD algorithm that can converge to Eq. (12), it can reduce the control traffic by L.

$$\omega \le \frac{L}{N\lambda\phi} - 1 \text{ (L>1, integer)} \tag{12}$$

In the HT model, like other metrics, the LPDP reference rate has an influence on the amount of traffic. Therefore, changing the value of ω in the TT model may cause a reduction in the amount of traffic by varying the LPDP reference rate. As shown in Eq. (11), it's assumed that the processing cost (i.e., 2×Tc+Td+Pd+2×Ts) for the data transmission between the nodes is a constant, such that $\phi_1 > 0$. This assumption is followed by the assumption that the processing cost for the requested data transmission, when it does not reference the requested policy in the LPDP, is a constant, such that $\phi_2 > 0$, Then, if the LPDP reference rate ($P_{probability}$) is high, the number of GCD executions decreases, while if the reference rate is low, the number of GCD executions increases, as shown in the following equation:

$$\omega \le \frac{1}{P_{probability}} \times \left(\frac{L}{N\lambda\phi_2} - 1 - P_{probability} \times \left(\frac{\phi_1}{\phi_2} - 1\right)\right), (\phi_1 \le \phi_2) \tag{13}$$

Therefore, when the LPDP reference rate is high, the amount of traffic between the PBNM components is in the order S&TT < HT ≤ TT, whereas in the case of a low LPDP reference rate, the corresponding order is HT ≤ S&TT < TT.

5 Conclusion

This paper provides the analytical framework for the comparison between centralized and distributed approaches for policy-based network management (PBNM) and their modeling. Policy-based networks are being deployed in a variety of applications. As mentioned above, many of the current PBNM methodologies tend to focus on specific model-oriented application development. Whenever it is necessary to maintain and manage a large number of policies, the utilization of the PBNM architecture will be one of the key factors in managing and controlling the networks. Each model is evaluated with such metrics as the policy provisioning time, the amount of traffic, and PDP utilization.

According to this study and evaluation, the S&TT architecture is ideal for a single-system environment, however the association of policies with multiple nodes may result in the exponential growth of the processing time, as the size of the network and λ increase. The TT architecture can address the problem of the overhead associated with the S&TT architecture through the use of distributed PDPs, although the cost of each metric would be further increased, depending on the degree of communication between the PBNM components and the extent of distribution of the PDPs when policy conflicts occur. The HT architecture may provide better performance than the S&TT or TT architecture, in the case where the reference rate from the LPDP in the PEP is optimized.

As presented in this paper, each approach has its own merits and demerits depending on the particular application. Therefore, when providing PBNM-based services, a careful evaluation of the characteristics of the application and the design of proper control architecture are needed. Further analysis and modeling will be needed concerning the application service requirements and correlations among the various PBNM architectures.

References

1. Thomas M. Chen, Stephen S. Liu, "A Model and Evaluation of Distributed Network Management Approaches", IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 20, MAY 2002.
2. Jude, M., Policy-Based Management: Beyond the Hype, Business Communications Review, March 2001.
3. Kosiur, D., "The Future of Policy-Based Network Management on the Internet", The Burton Group, May 2000.
4. John Strassner, et. al., "Policy-Based network management: solution for the next generation", ELSEVIER, 2004.
5. Emil Lupu, Morris Sloman, et. al., "An Adaptive Policy Based Framework for Network Services Management", Journal of Networks and Systems Management, 2003.
6. Gai, S., et al. “QoS Policy Framework Architecture”, draft-sgai-policy-framework-00.txt, February 1999.
7. Corrente, A., et. al., “Policy provisioning performance evaluation using COPS-PR in a policy based network”, Integrated Network Management, IFIP/IEEE 8th International Symposium on, March 2003.
8. R. Yavatkar, et. al., "A Framework for Policy-based Admission Control", IETF RFC 2753, January 2000.
9. K.L. Eddie Law, Achint Saxena, "Scalable Design of a Policy-Based Management System and Its Performance," IEEE Communications Magazine, 2003.
10. K. Chan, et. al., “COPS Usage for Policy Provisioning (COPS-PR), IETF RFC 3084, March 2001.
11. Verma, D.C., “Simplifying network administration using policy-based management”, Network, IEEE, Volume: 16, Issue: 2, April 2002.

An End-to-End QoS Provisioning Architecture in IPv6 Networks

Huagang Shao[1,2] and Weinong Wang[2]

[1] Department of Computer Science and Engineering, Shanghai Jiaotong University, Mailbox 157, 1954 Huashan Road, Shanghai 200030, China
[2] Network Center, Shanghai Jiaotong University, 1954 Huashan Road, Shanghai 200030, China
{hgshao,wnwang}@sjtu.edu.cn

Abstract. Differentiated Service (DiffServ) is a scalable solution to provide class-based differentiated Quality of Services (QoS). However, the basic DiffServ model lacks mechanisms to reserve resource for the class-based traffic and perform per-flow admission control. In this paper, we propose an end-to-end QoS provisioning architecture in IPv6 networks supported by a QoS-aware path selection algorithm. We design a novel flow label mechanism to achieve the effect of per-flow reservation paradigm and keep the scalability of DiffServ. In addition, the proposed QoS-aware path selection algorithm can optimize multiple QoS objectives simultaneously by exploiting genetic algorithm technique in conjunction with concept of Pareto dominance. The simulation results demonstrate the efficiency and scalability of the proposed architecture and algorithm.

1 Introduction

The enlargement of the Internet user community has generated the need for IP-based applications requiring guaranteed Quality of Service(QoS) characteristics. IETF has standardized Differentiated Services(DiffServ) technology as RFC2475. Packets that require higher QoS are classified as higher priority, and are forwarded in order of priority at nodes along their path. However, DiffServ cannot offer end-to-end QoS by itself, because it controls the per-hop packet forwarding order with relative priority according to its class and does not consider the route and allocate the resource for aggregated traffic. DiffServ requires other mechanisms to achieve end-to-end QoS.

To achieve stringent end-to-end QoS in DiffServ IPv6 networks, we introduce an architecture in conjunction with a novel IPv6 flow label mechanism in this paper. Based on flow label mechanism, the proposed architecture is capable of providing resource reservation for aggregate traffic by setup QoS-aware path in advance. With the capability to setup explicitly routed paths, the flow label mechanism effectively complements the DiffServ architecture. In addition, the proposed architecture provision per-flow QoS guaranteed without maintaining per-flow state in the core routers, which remain the scalability characteristic of DiffServ scheme.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 600–609, 2005.

Furthermore, we design a multiobjective optimization QoS-aware path selection algorithm based on Genetic Algorithm(GA) for the per-flow admission control and explicit routed paths setup. During our QoS-aware path select procedure, multiply QoS objectives can be optimized simultaneously. Whereas, existing QoS-aware path selection algorithms often optimize one objective and check satisfaction of the rest objectives, or optimize a combination objective of multiobjective, usually through a linear combination(weighted sum) of multiple attributes. Therefore, the solution not only becomes highly sensitive to the weight vector but also demands the user to have certain knowledge about the problem(e.g. influence of one parameter over another, priority of a particular objective, etc.). Moreover, the real intuition and logic behind the combinations is often fuzzy. In the proposed algorithm, we eliminate the these fuzzy logic behind the those optimization procedure.

The remainder of this paper is organized as follows. Section 2 discusses some related work. Section 3 give the details of the proposed architecture. A multiobjective optimization QoS-aware path selection algorithm is proposed in Section 4. Section 5 shows results based on our simulations. The paper is concluded by Section 6.

2 Related Work

Over the last three years, several DiffServ based end-to-end QoS provisioning architectures have been proposed. An approach based on the MPLS architecture has been considered in [1,2]. In these architectures, reservations for aggregate traffic are made between pairs of edge routers on specific Label Switched Paths (LSPs) inside the domain. All the QoS-sensitive flows, then, followed the appropriate LSPs.

Spiridon Bakiras et al. proposed a scalable architecture for providing end-to-end QoS guarantees in DiffServ-based IPv6 networks [3]. This architecture enhanced control planes by using the flow label fields. But it adopted a source routing framework and k-shortest paths algorithm to pre-compute paths between the pair of two edge routers, which has been found inefficiency in consideration of throughput[4].

On the other hand, the research community has extensively studied the QoS path selection problem in general, namely QoS routing problem. Several works in the literature have aimed at addressing special yet important sub-problems in QoS routing. For example, researchers addressed QoS routing for DiffServ-MPLS networks with the content of bandwidth and delay, see [5]. Routing with these two measures is not NP complete. Only a few dealt with the general QoS routing problem. Among them, some algorithm were based on GA, see [6,7,8] etc. Reference [7,8] have given a multiobjective optimizations QoS routing algorithm. Both algorithms used sharing function approach when performing genetic operation, which need some user-defined parameters and had higher computational complexity.

3 An Architecture for End-to-End QoS Provisioning

Our assumption is that the Internet consists of several independent administered DiffServ domains that are interconnected in order to provide global connectivity. One typical example is shown in Fig. 1, where each domain consists of a BB, the core routers, and the edge routers.

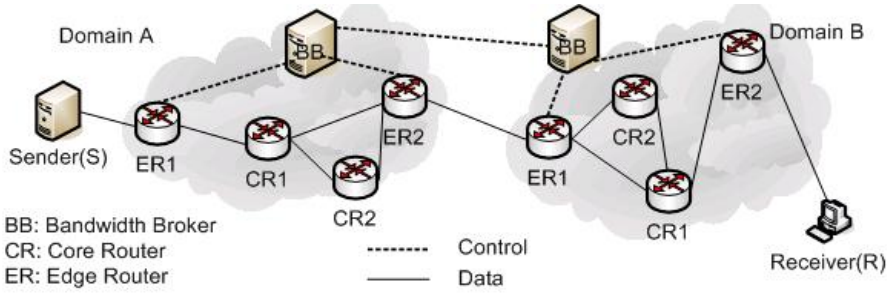


Fig. 1. The Differentiated Services architecture

3.1 Packet Forwarding with Flow Label

Traditionally, the slowest process in the forwarding path of an IP router is the multi-field classification. Specifically, when a packet is received at a router, the next hop behavior is decided by looking into several fields on the packet header (e.g. the source and destination addresses, ports, and the transport protocol type), and then finding the appropriate entry at the local database. However, IP addresses longest prefix match-up is a both CPU time and memory storage consuming process. In addition, this operation will be even more complicated for QoS-sensitive flows, since their packets should follow exactly the same path. The worst situation is that these fields may be unavailable due to either fragmentation or encryption.

Using IPv6 20-bit flow label, which does not exist in the earlier IPv4, can be a big help in alleviating this problem. For the each pair of edge routers inside a domain, there will be x path connecting them. We may then assign one flow label value to each one of these paths, and construct new (much smaller) routing tables inside the core routers of the domain, based only on flow labels. We should emphasize that we use the flow label field in the IP header, in order to identify a unique path within an DiffServ domain. As a result, any path within a domain will be assigned a specific flow label value, and all the packets (from any packet flow) that have to follow this path will be marked with that exact flow label value.

3.2 Resource Reservation and Admission Control

Resource reservation is an essential part of any network that provides QoS guarantees, and an appropriate signaling protocol is necessary in order to perform

this function. In our architecture, the receiver nodes will initiate the signaling procedure for the resource reservation, while the BB will be responsible for selecting a feasible path or rejecting the reservation requests. In the following paragraphs we illustrate how resource reservation be performed across multiple DiffServ domains.

Let us consider the scenario in Fig. 1, and assume that R, at domain B, wishes to receive some QoS sensitive data from the sender S at domain A. Then, the end-to-end resource reservation will be performed as follows.

(1) R will send a PATH message towards S, indicating the requiring QoS parameters, such as bandwidth, end-to-end delay, and packet loss ratio.
(2) The PATH message will reach ER2 of domain B, namely B.ER2, which will be the ingress router for that particular flow. B.ER2 then forward the PATH message to the BB of domain B, namely B.BB. B.BB will perform a QoS-aware path selection algorithm(see Section 4) to check whether there is a feasible path to provide requiring QoS guarantee. If there are not any sufficient resources, the request will be rejected. otherwise B.BB will find a feasible path from B.ER1 towards B.ER2, and PATH message will be forwarded towards S.
(3) The PATH message will reach A.ER2 which will also perform the admission control as in step(2).
(4) If this request can be accommodated, A.ER1 will forward the PATH message to the source node S. If S wishes to establish this connection, it will send the RESV message back to R.
(5) While the RESV message travel back to the destination node, BB will check whether there are the same reservation path between the same edge router pairs in the domain. If BB can not find this path, BB will generate a unique flow label value in the domain, and generate a correlation entry between the path and flow label value, then BB will send this entry to all the routers along this path. When the routers receive this entry, routers will insert this entry into their local database for the packet forwarding. Otherwise, BB can find this path, BB will combine this new connection into a resource reserved aggregated traffic, and send nothing to the routers. In addition, whenever BB can or can not find the same reservation path, BB will update its resource allocation database and send message to the edge routers to configure their traffic shapers, policies and markers to accommodate this new connection.

After completing of resource reservation successfully, the edge routers will classify the arriving packet, and label the packet with corresponding flow label value and DSCP(e.g. EF). When labelled packets enter into the core of domain, they will be forwarded exactly along the resource reserved path according to the flow label value and DSCP in their header fields.

4 The Proposed QoS-Aware Path Selection Algorithm

In this paper, we take advantage of multiobjective optimization technique to perform QoS-aware path selection. Multiobjective optimization is a multiobjective

minimizes (or maximizes) problem. GAs have been recognized to be well-sited to multiobjective optimization, because many individuals can search for multiple good solutions in parallel. The steps involved in solving a optimizations problem using GA is consist of encoding, initialization, selection, crossover, mutation, and decoding. Multiobjective genetic algorithms (MOGAs) vary from the ordinary GAs about their selection. The procedure of selection in MOGAs is based on nondominated sorting operation[9].

4.1 Problem Formulation

A QoS-aware path selection problem is to find a path between a source node to a destination node, which will satisfy multiple constraints QoS parameters simultaneously. Because the bandwidth constraints can be pre-processed by topology pruning operation. Therefore, we focus our ideas to determine the QoS routes that satisfy the three major objective parameters, namely: (1) end-to-end delay, (2) packet loss rate, and (3) proper load balance. We follow the general procedure to represent the network by a graph $G = (V, E)$. A path between a source (v_s) and a destination (v_d) is represented by $P = \{v_s, v_1, v_2, ..., v_d\}$, where $v_i \in V$.

Our network links are assumed to be service queues. The service arriving is assumed to follow Poisson distribution, then the service time obeys Exponential distribution. Therefore, the link delays, which are introduced due to service time should also follow an Exponential distribution with parameter equal to λ. Since, the path consists of a chain of k hops, the delay along the entire path should follow Erlang-K distribution, which is the convolution of k independent exponential random variables, each having the same mean. The probability(Pr^{delay}) that the delay (D_P) over a path P (from the source to destination) of length k is less than t is given by $Pr^{delay}(P) = Pr^{delay}(D_P < t) = \frac{\lambda^k t^{k-1} e^{-\lambda t}}{(k-1)!}$. Hence, to find the optimal path, our algorithm will try to minimize this probability.

For network link l, there have a related packet loss ratio $Pr^{loss}(l)$, The probability of total packet loss ratio $Pr^{loss}(P)$ among P is calculated as $Pr^{loss}(P) = 1 - \prod_{l \in P}(1 - Pr^{loss}(l))$. Our designed algorithm will also try to minimize this probability.

Selecting higher residual bandwidth links can balance the traffic load in the network. we denote C_l and U_l as the capacity of link l and current load of the link l respectively. Then, the total residual bandwidth after allocating bandwidth for P is given by $\sum_{l \in P}(C_l - B_l - U_l)$, where B_l are the bandwidth requirement of path P. The load balance factor can be defined as $M(P) = \sum_{l \in P} \frac{(C_l - B_l - U_l)}{C_l}$. Our designed algorithm will try to maximize $M(P)$.

According to the conditions mentioned, the mathematical model of multiobjective optimization is designed as Equation 1, where R is universe of decision variable vector.

$$\begin{cases} V - min\ f(P) = [f_1(P), f_2(P), f_3(P)]^\top \\ f_1(P) = Pr^{delay}(P) \\ f_2 = Pr^{loss}(P) \\ f_3 = -M(P) \\ s.t. \qquad P \in R \ \ \wedge \ \ (\forall P \in R, min\{C_l - U_l | l \in P\} \geq B) \end{cases} \tag{1}$$

4.2 Algorithm Design

The coding is first step to solve QoS-aware routing problem using the MOGAs. In our coding scheme, all possible paths between the source and destination will be stored. The path is mapped to a solution string consisting of the sequence of nodes along the path. These solution strings can be called chromosomes, and a node in the solution strings can be called a gene. The set of all such strings constitute the initial population.

The performance of a selection individual can be expressed by its fitness. The fitness calculation includes two parts, namely Pareto ranking and diversity processing. Before the Pareto rank of an individual is sorted, the values of the three pre-define objectives are calculated independently. The rank of a certain individual corresponds to the sum of individuals in the current population by which it is dominated. If individual i is dominated by p_i individuals on the sub-objectives, its current Pareto rank is given by $i_{rank} = p_i + 1$. Another key problem is to take a measure to preserve diversity in the population.

The crowded-comparison operator ($\prec_n$) guides the selection process at the various stages of the algorithm toward a uniformly spread-out Pareto-optimal front[9]. We assume that every individual i in the population has another attribute, namely, crowding distance ($i_{distance}$).

Denotes k as the number of solutions, for each objective m, we sort individuals by each objective value, and then we have $1_{distance} = \infty$, $k_{distance} = \infty$, and

```
MOGAPathSelection(G < N, E >, UserRequest, Path)
 1: Difference ← TRUE
 2: C ← initPopulation(G, UserRequest)
 3: while Difference = TRUE do
 4:    l ← |C|  ; l is number of population.
 5:    for each objective m do
 6:       O_m ← calculateFitness(C, UserRequest)
 7:       D_m ← sortParetoRank(C, O_m)
 8:       D_m(1) ← ∞ , D_m(l) ← ∞
 9:       for (i ← 1) to (l − 1) do
10:          D_m(i) ← D_m(i) + (D_m(i + 1) − D_m(i − 1))/(Max(O_m) − Min(O_m))
11:       end for
12:    end for
13:    C ← sortCrowdDistance(C, D) , U ← selectOperation(C)
14:    Difference ← terminateDecision(C, U, UserRequest)
15:    if Difference = TRUE then
16:       C ← makeNewPopulation(U) ;performs crossover and mutation.
17:    else
18:       C ← U
19:    end if
20: end while
21: Path ← decode(C)
```

Fig. 2. Pseudo-code of MOGA for QoS-aware path selection algorithm

$i_{dis\tan ce} = i_{dis\tan ce} + \frac{M_{i+1}^m - M_{i-1}^m}{f_m^{\max} + f_m^{\min}}$ when $1 < i < k$. Here, M_i^m refers to the mth objective function value of the ith individual, and parameters $f_m^{\max}$ and $f_m^{\min}$ are the maximum and minimum values of the mth objective function, respectively.

Now, we define a partial order $i \prec_n j$, if $i_{rank} < j_{rank}$ or $i_{dis\tan ce} < j_{dis\tan ce}$ when $i_{rank} = j_{rank}$. That is, between two solutions with different nondomination ranks, we prefer the solution with the lower (better) rank. Otherwise, if both solutions belong to the same front, then we prefer the solution that is located in a lesser crowded region.

The crossover and mutation operations are same as normal GAs, and the probability of crossover and mutation is corresponding to 0.75 and 0.05 respectively in this paper. But, we have to take care of the fact that these operations must not produce any illegal paths. The crossover operations can only be performed at the same gene (excepting of source and destination node) of two chromosomes.

At the end of every iteration, we get a set of nondominated solutions which are better than all other existing solutions when all the three optimization functions are considered simultaneously. Finally, the program terminates when the improvement of fitness values is less than a specified precision. Based on the components of algorithm, the integral pseudo-code of MOGA for QoS-aware path selection algorithm is given as Fig. 2.

5 Simulation Results

Simulation experiments are performed over a test networks which was generated according to Waxman's model. The model can be expressed by $p(i,j) = \beta \times \exp(-\frac{distance(i,j)}{\alpha \times L})$. Table 1 presents the parameter setting for generating simulation network topology. In addition, We assume the resource reallocation interval T as 1 minute.

In the first experiment we investigate the convergence of the proposed QoS-aware path selection algorithm. The algorithm attempts to minimize end-to-end delay and packet loss ratio, and attempts to balance the network load. We have compared the convergence of our proposed algorithm, with an existing heuristic algorithms [11] and an exhaustive search approach [10]. The exhaustive search method finds the optimal values of the three sub-objectives by exhaustively searching them one after another, which is used to compare our results and act as performance benchmark. The novelty of the algorithm is that it is optimizing all three objective simultaneously by building the set of non-dominated solutions. For the sake of clarity, we demonstrate it in three separate plots. The three plots (one for each QoS objective) in Fig. 3, Fig. 4, and Fig. 5 explain how the non-dominated solutions are proceeded, towards convergence, in a very short time.

After that, we study the scalability of the proposed algorithm. As delivering in Fig. 6, our algorithm exhibits a linear and stable pattern in comparison with another heuristic algorithms when the networks expand from a middling scale topology to a relatively larger scale topology.

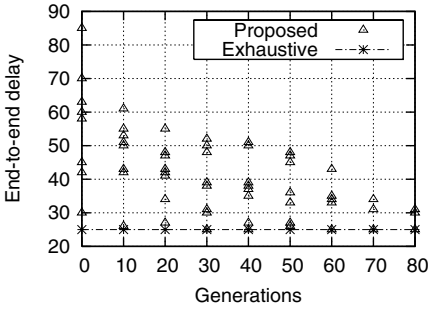


Fig. 3. Convergence of End-to-End delay objective

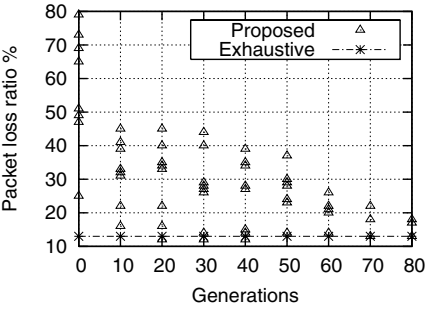


Fig. 4. Convergence of packet loss ratios objective

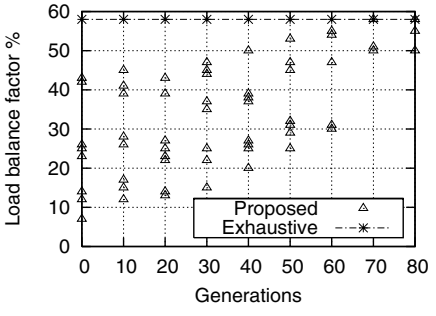


Fig. 5. Convergence of load balance objective

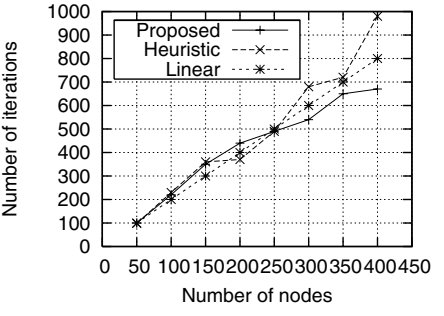


Fig. 6. Comparison of algorithm scalability

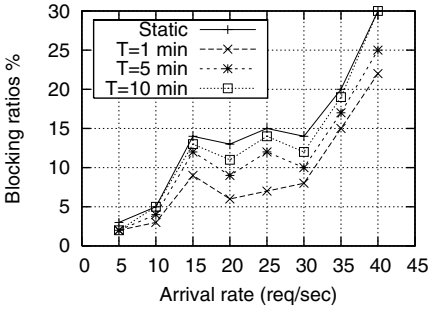


Fig. 7. Blocking ratios with the increasing of resource reallocation interval

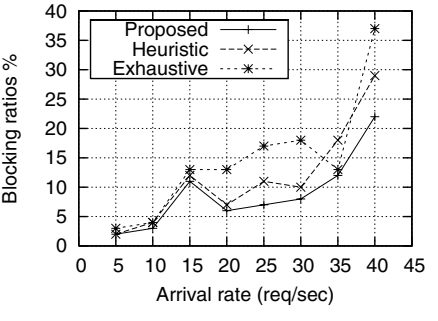


Fig. 8. Blocking ratios with increasing resource reservation request arrival rate

Table 1. Parameters for generating network topology

α	β	P	$delay(ms)$	$capcity(Mbps)$	nodes
0.3	0.6	0.4	(10, 20)	(90, 110)	50

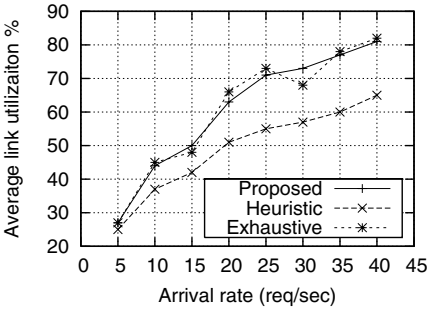


Fig. 9. Average link utilization with the increasing of resource reallocation request arrival rate

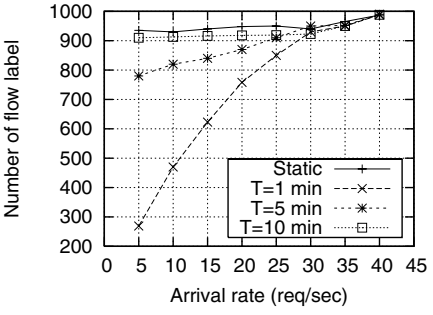


Fig. 10. Flow label allocation situation with increasing resource reservation request arrival rate

And then, we inspect the impact of the reallocation interval T on the performance of the resource allocation algorithm. The static curve in Fig. 7 corresponds to a system where the resource allocation is manually configured according to some long term traffic measurements. In the simulation experiments, this curve is produced by running QoS-aware path selection algorithm set the start of expectation value of each source. We can find that the static and dynamic resource assignment have very similar performance. And performance of scheme is not affected distinctively by changing length of the resource reallocation intervals. Therefore, we can keep the length of the interval T at reasonably value for the balance of control overheads and resource utilization.

Finally, we examine link utilization, blocking ratios, and flow label allocation situation respectively with the increasing resource reservation request arrival rate. The mean rate of arrival of request is assumed to be 25 requests per second, and the average data rate for this experiment is taken as 20 Mbps. We can find(see Fig. 8) that blocking ratios of our algorithm have a clearly better performance than two existing algorithms. Moreover, Fig. 9 indicates that the proposed algorithm has a good link utilization as well as exhaustively searching algorithm. In addition, the number of flow label allocation have a relative stable pattern with increasing the resource reservation request arrival rate in Fig. 10, which exhibits that the most of single flow at the edge of domain can be combined into a few aggregate traffic in the core of domain. Therefore, our resource reservation scheme has a good scalability.

6 Conclusions

With booming of internet real-time applications, end-to-end QoS guarantee in IPv6 networks will continue to be an active research area. In this paper, we propose a end-to-end QoS provisioning architecture supported by a novel IPv6 flow label mechanism and a multiobjective optimization QoS-aware path selection algorithm. The simulation results show the efficiency and scalability of the proposed algorithm. The possible future research includes providing the multicast support in this architecture.

References

1. Faucheur, F. et al.: MPLS support of differentiated services. IETF RFC 3270, (May. 2002).
2. Berghe, V., Turck, F., Piet, D.: SONAR:A platform for flexible DiffServ/MPLS traffic engineering. IEEE International Performance, Computing and Communications Conference, Proc., v23, p.195-201, (2004).
3. Bakiras, S., Li, V.: A scalable architecture for end-to-end QoS provisioning, Elsevier computer commnicaitons, Vol.27, p.1330-1340, (2004).
4. Acharya, S. et al.: Precomputing High Quality Routes for Bandwidth Guaranteed Traffic, IEEE Globecom2004, p.1202-1207, (Dec. 2004).
5. Kimura, T., Kamei, S.: Qos evaluation of diffserv-aware contraint-based routing schemes for multi-protocol label switching networks. Computer Communications, Vol. 27, p.147-152, (2004).
6. Xiang, F., Junzhou, L., Jieyi, W., Guanqun, G.: QoS routing based on genetic algorithm. Computer Communications, Vol. 22, p. 1394-1399, (1999).
7. Joshua, D., Martin, J., David, W.: Multiobjective evolutionary algorithms applied to two problems in the telecommunications. BT Tech., Vol. 18, p51-64, (2000).
8. Xunxue C., Chuang L., Yaya W.: A Multiobjective Model for QoS Multicast Routing BAsed on Genetic Algorithm. Proceedings of the 2003 international conference on computer networks and mobile computing. p.49-53, (2003).
9. Deb, K. et al.: A Fast and Elitist Multiobjective Genetic Algorithm(NSGAII). IEEE Trans. Evolutionary Computation, Vol. 6, No.2, (Apr. 2002).
10. Widyono, R.: The design and evaluation of routing algorithms for real-time channels. International Computer Science Inst., Univ. of California, Berkeley, CA, Tech. Rep. TR-94-024, (1994).
11. Yuan, X.: Heuristic Algorithms for Multi-constrained Quality-of-Service Routing. IEEE/ACM Trans. Networking, Vol. 10, No. 2, p.244-256, (Apr. 2002).

Introducing Public E-Mail Gateways: An Effective Hardening Strategy Against Spam

Francesco Palmieri and Ugo Fiore

Federico II University, Centro Servizi Didattico Scientifico,
Via Cinthia 45, 80126 Napoli, Italy
{fpalmieri, ufiore}@unina.it

Abstract. With the increasing popularity of the Internet, unsolicited electronic mail (spam) has become a major concern. It fills up user's mailboxes, clogs mail relays, wastes postmaster time, and creates inconveniences for sites that have been used as a relay. This seems to be a growing problem, and without appropriate countermeasures, spam messages could eventually undermine the usability of e-mail. In this paper, we propose a cooperative spam-avoidance strategy based on the concept of restricting, at the network border and mail relay level, the mail sending function through properly authorized mail gateways registered as proper new Resource Records in the DNS System. If accepted and enforced by the largest number possible of network and mail administrators on the Internet, this strategy may result in a substantial reduction of the worldwide e-mail spam phenomenon.

1 Introduction

Today, e-mail has emerged as the most commonly used form of communication. Businesses are increasingly recognizing the benefit of using e-mail in their daily work. Though e-mail has provided a cheap and convenient means for businesses to contact with customers and partners all over the world, there are many problems caused by the e-mail which will disturb the businesses. One of the problems most concerned is so-called spam, which is commonly known as "junk" or "unsolicited" e-mail. Spam messages are annoying to most users, as they waste their time and fill-up or clutter their mailboxes quickly. They also waste bandwidth, and may expose users to unsuitable content (e.g. when advertising pornographic sites). Spam seems to be a growing problem, and without appropriate counter-measures, spam messages could eventually undermine the usability of e-mail. Recently, a study has shown that 52% of email users say spam has made them less trusting of email, and 25% say that the volume of spam has reduced their usage of email [1]. To propagate spam, senders are increasingly relying on various tactics such as unauthorized BGP route injection, Autonomous System route hijacking, and asymmetrical routing with spoofed IP addresses. This crisis has prompted proposals for a broad spectrum of potential solutions, ranging from the design of more efficient anti-spam software tools and practices to calls for anti-spam laws at both the federal and state levels. All of the above solutions are characterized by the same, and apparently unsolvable scalability problems:

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 610 – 619, 2005.

since potentially any host can directly connect on the SMTP port of any mail relay to send mail, the number of hosts that can be used for sending spam, consciously or not (a host may be compromised or infected by a virus or worm that can send spam without the user knowing) is practically unlimited and furthermore these hosts may continuously change, such that the application of any host-based anti-spam countermeasures is practically unfeasible. Consequently, the objectives of the various legal and technical solutions are the same: operate on the largest scale possible to make it unfeasible or at least unprofitable to send spam and thereby destroy the spammers' underlying action and business model. Accordingly, in this paper we propose a cooperative spam-avoidance strategy operating only at the network border and mail relay level, thus restricting the problem to a smaller and consequently manageable scale. If accepted and enforced by the largest number possible of network and mail administrators on the Internet, this strategy may result in a substantial reduction of the worldwide e-mail spam phenomenon.

2 Defeating Spam: The Overall Strategy

There is a large number of popular solutions for individual and group spam blocking available. The most basic are: don't run an open relay; don't allow multiple recipients for null sender; and verify that envelope sender contains a valid domain. Yet the spammers seem to have worked around them. Junk e-mailers routinely falsify sender envelopes in order to misdirect complaints about their junk e-mail. Furthermore, authors of viruses that propagate via e-mail falsify sender envelopes to hide the origins of the virus-infected computers. Using a blocking list involves handing a fair amount of responsibility and control to a third party - something that would not make for a good response to a customer unhappy with missed mail. Content analysis tools bring up privacy issues, cost a lot in processing power, generates lots of false positives and consequently tend to require tuning for each individual recipient. Furthermore, all the above solutions will be effective, albeit partially, if applied to all the hosts allowed to send mail toward external mail exchangers (MX). This is practically unfeasible, due to the huge number of hosts (usually all!) that in a typical network are allowed to send mail directly, since anybody can send email by simply connecting to the SMTP port of an MX server. When a PC is compromised, that single PC can let loose megabytes of spam. The administrator of a server targeted by spam messages may analyze log files to identify machines sending anomalous traffic and blacklist them, but this process is lengthy. Further, as soon machines are "cleaned", their legitimate users will want the email service to be immediately reactivated for them, an additional clerical work for the postmaster. Thus, the main question we should try to answer is: "Is there a simple technique, relying on existing mechanisms and protocols, that can be used to limit unrestrained email sending from unauthorized hosts?" The answer, at our advice may be simpler that it can be thought. If some machines were registered with global scope (and the DNS worldwide distributed database will be the best place) as authorized mail transfer gateways, and only those machines were allowed to connect to mail relay servers, mail would be forced to pass through the mail gateways, where user identification and enhanced anti-spam techniques may be in place. Of course, the mail exchanger/relay servers accepting input mail transfer connections should verify through the DNS if the sender is an official Mail

Gateway for its domain before accepting mail. Accordingly, we propose a spam-avoidance strategy that requires cooperative work of network administrators, that must selectively block, by filter at the network border level, all the outgoing SMTP connections, except for the explicitly authorized mail gateways, and of the postmasters and DNS administrators that must properly configure the DNS to define all the official mail gateways and update MTA software on all the mail exchangers to properly check for registered senders in the DNS.

3 Implementation Details

Our strategy requires three areas of intervention: the network border, to define outgoing SMTP filtering policies, the Domain Name System, to introduce the new type of Resource Record and define/register the official mail gateways and on the MTA software on the mail exchanger (MX) servers (aka *sendmail*, *qmail* etc.) to enforce the mail gateway checking for any incoming connection.

3.1 Enforcing SMTP Filtering Policies

First, the line of defense must be shifted nearer to the source of spam. If incoming traffic is discarded unless it comes from a trusted gateway, spammers would be forced to channel their traffic through the official gateways for their domain. When the gateway performance slows down, legitimate domain users will complain. So the administrators at the sending end are called upon for intervention. They have means to locate and identify abusers, and can also act against them. Anti-spam techniques can be successfully used on both the sending and the receiving end. A well crafted filtering policy on the border router of an administrative domain can block all the outgoing e-mail traffic, identified by connections to the SMTP port (port TCP/25) of any outside host, enabling only the official Mail Gateways to send out mail. Furthermore, any anomalous increase in SMTP traffic or filtering policy violation can be noticed and fought by two cooperating network and/or e-mail administrators, the one at the source and the one at the target, instead of just one (the victim).

3.2 Extending the Domain Name System

The Domain Name System is basically a distributed database, or directory service of host information that is indexed by domain names. Each domain name is essentially just a path in a large inverted tree, called the domain name space. The data associated with domain names is contained in resource records, divided into classes, each of which pertains to a type of network. Within a class, several types of resource records (RR) can be defined, which correspond to the different varieties of data that may be stored in the domain name space. Different classes define different record types, though some types are common to more than one class. Each record type in a given class defines a particular record syntax, which all resource records of that class and type must adhere to. DNS is used mostly to translate between domain names and IP addresses, but it is also very useful to control Internet email delivery, since it provides a method for publicly registering the incoming mail relay servers associated to each

domain (the MX RR) and consequently to derive mail routing information. Each RR has the following format [2]:

NAME	a domain name to which this resource record pertains
TYPE	two octets containing one of the RR type codes. This field specifies the meaning of the data in the RDATA field.
CLASS	two octets which specify the class of the data in the RDATA field.
TTL	a 32 bit unsigned integer that specifies the time interval that the resource record may be cached before it should be discarded.
RLENGTH	a 16 bit integer that specifies the length of the RDATA field.
RDATA	a variable length octets string that, according to the TYPE describes the resource.

Fig. 1. Resource Record layout

3.2.1 The Resolution Process

Name servers are adept at retrieving data from the domain name space. They return data from zones for which they're authoritative, but can also search through the domain name space to find data for which they're not authoritative. This process is called name resolution. Because the namespace is structured as an inverted tree, a name server needs only one piece of information to find its way to any point in the tree: the domain names and addresses of the root name servers. A name server can issue a query to a root name server for any domain name in the domain name space, and the root name server return information about who is authoritative to respond from that domain such that the querying name server can directly start and request all the information he needs on its way. Each query refers to a specific RR in a domain, referenced by its type code, the corresponding value that must match in the required records, and the class to which the RR belongs to (usually INET), and results in one or more resource records matching the query itself. Consequently, the introduction of new information and concepts in the Domain Name System implies the definition of new resource records and their proper treatment in query and response messages.

3.2.2 Introducing the New DNS Mail Gateway Concept

We propose that any domain must publicize its servers authorized to relay for sending mail, just as it is done for the Mail Exchanger (MX). A target MX may then query the DNS to recognize whether the host trying to send mail is registered or not. This can be easily done by adding a new resource record type, named MW, to the DNS, represented in the standard format, with the type-code field (49) followed by a variable length character string which specifies an host authorized for sending e-mail from the specified domain. The syntax of the RR will be the usual: *owner*, *ttl*, *class*, *record-name (MW)*, *MW-dname.* A more natural choice for the new Mail Gateway RR name would be MG, but unfortunately the acronym MG is used in RFC 1035 for the Mail Group Member Resource Record (code 8), not widely used but still valid. Essentially, two distinct checks (in the worst case) may be done before accepting or rejecting an incoming mail connection. First the alleged sender (claimed) domain should be ob-

tained by the envelope sender and a successful MW check against this domain allows immediate verification. This is the simpler case of direct domain-based relay. If otherwise the SMTP outgoing relay host is relaying on in IP address basis, the first check may be unsuccessful and a following check against the domain obtained by reverse DNS resolution becomes useful. Thus, if the sending host is an authorized MW at least for the domain resulting from its reverse DNS resolution we guess that it is relaying on an IP address basis (and obviously enforcing anti-spam policies), thus the incoming mail transaction can be accepted by the MX. This implies that any MW needs a reverse mapping on the DNS and that any MW acting on a domain must be registered as a MW for that domain, even if it only relays mail (on an IP address basis) for users outside the domain. A domain can have more than one Mail Gateway and each one can serve more than one domain. In the first case, the MW entries in the DNS are checked in the order in which they have been defined in the zone file. If the order of the entries matters, it is the zone maintainer's responsibility to keep those entries in the correct order. In the latter case, the involved host can act as a Mail Gateway for different domains provided that it is registered as a MW and a host in each of these domains or it can correctly perform relay based on IP address authorization. Here, as explained before, a reverse DNS check on its IP address is necessary and the host must be an authorized Mail Gateway at least for the domain it belongs to. Moreover, to prevent spoofing, the domain name resulting from the reverse DNS check is accepted only if a subsequent address query to the given domain name points to exactly the IP address of the sending MTA (usual procedure to verify PTR records). With this scheme, a host can be designated as an MW for a domain even if it does not belong to that domain. Traffic originated outside the authorized MW need not be automatically discarded. This is certainly an option, and some "critical" sites may elect to adopt it. Nevertheless, another line of action that may be pursued involves assigning lower priority to "unregistered" traffic, or setting volume limits. Message processing or session establishment could also be deliberately delayed, as several *MTA* do. The MX administrator may decide the policy to apply, based on a number of dynamics, including e.g. the number of messages over a time unit.

4 Modifications to the MTA SW

The scheme can be made to work with very small modifications on the MTAs. Usually, an MTA tries to reverse-lookup the connected party by using, for instance, a *getpeername()* call to retrieve the name of connected peer. Then the MTA makes some consistency checks against the envelope sender and verifies if it belongs to the list of unacceptable senders and that is the place where the MW query should be issued. From the envelope sender, the MTA can determine the alleged domain of the sending party. The MW query returns the list of the authorized Mail Gateways for that domain. If the name of the connected peer belongs to the list, the sending host is registered as an authorized Mail Gateway and message processing can proceed. Should the connected party name not be available (no PTR resource record), the MTA can lookup the names of the official Mail Gateways retrieved and compare their IP addresses with that of the connected peer. However, we feel that enabling reverse lookup on an official MW is not a burdensome requirement, and instead is a good

practice. No provision should be made for recursive queries. Just as it happens with MX queries, recursion should be handled by name servers. In some cases, the receiving MTA trusts the sending MTA not to fake messages and does not need to check it against the MW records at message reception. As a typical example, a company might have an outer mail relay which receives messages from the Internet and checks the MW records. This relay then forwards the messages to the several department's MX servers. It does not make sense for these department mail servers to check the outer relay against the MW records, because they know that it is a trusted gateway. It may even lack any MW record registration if it is not configured for sending inter-domain outgoing mail. In this case there is a trust relationship between the department relays and the outer relay. Another common situation is that of the low-priority MX relays, which receive and cache e-mails when the high-priority relays are down. In this case, the high-priority relay would trust the low-priority relay to have verified the sender authorization and would not perform another MW verification (which could obviously fail). Consequently MW checking should be turned off for trusted relays, that can be defined by adding their IP address in the database of hosts and networks that are allowed for relay. Thus, if the -relay check by address or domain name against the "open access" database is successful no further check must be done against the MW record. This is used also to disable MW checking when the mail is sent from internal host directly to the mail gateway that has also the role of MX.

4.1 Relay Scenario

The following scenario describes in detail the MTA behavior when receiving an incoming connection on port SMTP (25) requesting e-mail relay:

1. Accept a connection request on port TCP/25 from IP address **x.y.w.z**
2. Find out the alleged sending domain from the envelope sender
3. Issue an MW DNS query upon the alleged domain and check the results
4. If the sender is an authorized Mail Gateway for the above domain then accept the incoming mail and proceed on relaying it.
5. Otherwise, reverse lookup the IP **x.y.w.z** (as already done by almost all the MTA agents), obtaining an *FQDN* and issue another MW query upon this domain. If the sender is an authorized MW for it, then accept the incoming mail (the sender is relaying on an IP address basis) and proceed. In any other case choose the appropriate rejecting action (e.g., reject the message with the new reason "*553 Not a registered Mail Gateway*" or downright the usual "*550 Relaying denied*").

Unfortunately, SMTP allows empty envelope sender addresses to be used for error messages. Empty sender addresses can therefore not be prohibited. As observed, a significant amount of spam was sent with such an empty sender address. To solve this problem, also in this case we can behave as described in point 5, starting directly from point 2, since straightforwardly the MW lookup of an empty domain will be always unsuccessful. Consequently the domain name obtained by the reverse DNS lookup of the sending MTA can be used instead of the domain obtained from the *"Mail From"* header to lookup the MW records. This makes sense, since such messages were generated by the machine, not a human.

4.2 Host-Less Domains and Corporate Mobile Users

Users in host-less domains should relay all of their outgoing email through one of the ISP's registered Mail Gateways, that would have a valid MW record. Generally, ISPs disallow relaying when the sending host is not in the ISP's domain. However, many ISPs support outgoing mail server authentication. An ISP may also opt for IP address based relaying, for their customers provided with static IP addresses. Corporate users working on their laptops while travelling or in locations away from the office should also have their email relayed through a trusted Mail Gateway. In our opinion, security issues involved with authentication are not a strong argument against Mail Gateways, since such kind of problems are much better managed by VPNs.

5 Prototype Implementation and Performance Analysis

We set up a very simple testing environment with two fake domains, each with its own name server and mail exchange server connected on a separate interface of a two Ethernet Cisco 3640 router. For simplicity, the mail gateway was supposed to be the same machine as the mail exchange server. One side was running *qmail* and the other *sendmail*. The DNS software was ISC BIND on both ends. A mail generator automatically originated, at a specified rate, fake e-mail messages starting from the first domain and directed to the second domain MX, continuously changing its IP address at random time intervals, to simulate e-mail coming from different sources. Only a specified percentage of the messages were relayed through the official domain MW while the others were directly sent to the destination MX, simulating spam coming from different unauthorized sources. SMTP connections coming from odd IP addresses were filtered at the border router level, while the other were passed through incoming MW verification, as specified in section 4, thus messages coming from unauthorized sources were stopped, depending from their source addresses, on the domain border or on the receiving side. To approximate the e-mail sending rate and the total ratio between authorized and unauthorized messages with a value as close as possible to the real-world value, we measured outbound connections to port 25 originating at a real domain (the whole *unina.it* network), and filtered them against a list of machine officially known to be mail gateways. We observed that unauthorized connections constitute a significant portion of the total connections. To be more precise, we averaged separately the data collected on weekdays and on holydays. Again, we found that outbound connection to port 25 from unauthorized sources more that outweigh legitimate traffic: 78.5% of traffic came from machines not known to be mail gateways. As a further validation, we aggregated data collected during weekdays and weekends. Authorized mail gateways continue working during weekends, even if they reduce their activity, because people keep connecting from the home to use their mailboxes. Instead, the majority of users shut their (perhaps compromised) machines off for the weekend. Unauthorized connections indeed dropped down to 55.4% during weekends and raised to 82.8% in weekdays. The above data can also give a useful estimate of the traffic that can be blocked by the widespread use of our strategy. Finally, we ran several 24 hours tests, generating e-mail traffic with the measured rate and distribution. As expected, all the messages coming from the unauthorized source were stopped, without performance degradation or loss for the legitimate traffic.

6 Vulnerabilities and Open Issues

Although the proposed anti-spam strategy looks quite simple and complete, there are some in-depth considerations that need to be done before its deployment in the whole Internet.

6.1 DNS Vulnerabilities

DNS is an essential part of the proposed spam avoidance framework, since it requires a worldwide deployed directory service, and DNS is currently the only one available. Unfortunately, DNS is vulnerable and can be easily spoofed and poisoned. DNS security flaws are commonly known and introduce a certain degree of weakness in the whole architecture, but there is no feasible alternative to using the DNS to publish the Mail Gateway information thus some additional DNS security enforcement facility such as DNS Security Extensions (DNSSEC) [3] or Transaction Signatures (TSIG) [4] is recommended in conjunction with our framework. Anyway, the Internet needs by itself better security in the Domain Name System, and the proposed framework

6.2 Open Relay Mail Gateways

Our anti-spam strategy gives a great deal of trust to the Mail Gateway that assumes a central role in the entire chain of defense. However, it can happen that some hosts that are defined as Mail Gateways may become Open SMTP relays (i.e. machines which accept any e-mail message from anyone and deliver to the world), or by unconscious misconfiguration or by malicious actions after that the host is compromised. Of course they may be abused for sending spam, but in this case, the IP address of the relay machine and the MW records of the domain directly track back to the persons responsible. Both can be demanded to fix the relay or remove the MW record for this machine. Furthermore, the IP address of the open relay may easily be filtered or blacklisted. An open relay is a security flaw like leaving the machine open for everybody to login and send random mails from inside. Should the administrative persons refuse to solve the problem, they may be identified as spammers and held responsible, if that is not in conflict with local laws and policies.

6.3 Graceful Deployment

Obviously, for reasons of backward compatibility and smooth introduction of this scheme, MW records can't be required immediately. Domains without MW records must temporarily be treated the same way as they are treated right now, i.e. e-mail must be accepted from anywhere, eventually enforcing some “safety” limit about the number or frequency of messages that can be received from unregistered outgoing relays. But once the scheme becomes sufficiently widespread, mail relays can start to refuse e-mails from sender MTAs that are not associable to a registered MW record, thus forcing the owners of any domain to include at least a statement of Mail gateway authorization into the domain's zone table. That allows accepting e-mails only from domains with a reasonable security policy. However, the main problem hampering the widespread application of our anti-spam strategy is that it requires a new RR entry

type and consequently an upgrade of all DNS servers. Therefore, as a temporary workaround, an alternative, but equivalent, encoding scheme can be proposed. Instead of using a new RR type, the "*Mail Forwarder*" (MF, code 4) obsolete record type, now replaced by the MX RR in its usage and having exactly the same format of the proposed MG record, can be used to contain the Mail Gateway information. The MF record is still supported by almost all the widely deployed DNS implementation, including the traditional BIND and consequently this migration strategy will be immediately viable. Thus, to allow smooth introduction of Mail Gateways without the need to immediately upgrade all DNS servers, all MTAs (which have to be upgraded anyway) should support both the MF and the MW records to check first the existence of an MW and then, if unsuccessful because recognized as an unsupported RR, (a return code "Not Implemented" is received in the answer), perform an MF query.

7 Related Work

Cryptographic techniques have been suggested to guarantee the legitimacy of SMTP traffic. The system may operate at the end-user level, where everybody wishing to send email must hold a key, or at the server level, so that every server can authenticate its connected parties. Transfer of e-mail would be limited to the authorized hosts and agents. This model, that is substantially peer-to-peer, imposes a significant computational and operational burden and presents many open issues in key management and distribution since there aren't yet the conditions in place for establishing a single worldwide authority handling all the necessary keys. Yahoo's DomainKeys [4] is a hybrid system that combines cryptographic techniques and the DNS. A domain is expected to place the keys they will use for signing messages in the DNS. The sending server signs messages with the appropriate key and includes the signature in the message headers. The receiving server may then verify the signature. The administrative tasks are simplified since they are circumscribed within a single domain. However, the perplexities about performance related to massive use of cryptographic techniques still hold. On the other side, an "electronic stamp", that will imply taxing of each e-mail sent, is also something much talked about, but it is unclear how such a stamp should be administered and if it can be accepted worldwide for a service that has always been totally free. Of course, the core argument is that there should be a deterrent for high-volume mailing, so a monetary charge is not really needed. Something more immaterial could be used as well. Back in 1992 Dwork and Naor [5] proposed that e-mail messages be accompanied by proofs of computational effort, a moderately hard to compute (but very easy to check) function of the message, the recipient's address, and a few other parameters. In contrast with ours, this approach requires extensive modifications to be carried out both at the server side and the client side. Aside from the consideration that using a resource waste as a weapon against another waste is not very appealing, the problem remains as to the performance degradation that legitimate mail gateways would incur, should they be forced to perform a memory-intensive computation for each message.

8 Conclusions

In our opinion the most critical factor influencing the worldwide uncontrollable growth of the spam phenomenon is the fact that potentially any host can establish a connection to any mail exchanger in the world and directly sending mail without any check about the trustiness of the origin. Accordingly we conceived a cooperative spam-avoidance strategy operating at the network border and mail relay level, based on the definition of a new concept of "mail gateways" or "trusted" MTAs, defined for each domain and publicly registered on the Domain Name system, that will be the only hosts that are authorized to send cross-domain e-mail. This will drastically reduce the number of hosts potentially able to perform spam to a limited number of known trusted hosts in any domain that could be strictly controlled by their administrators. If accepted and enforced by the largest number possible of network and mail administrators on the Internet, this strategy may result in a substantial reduction, and over the longer term to the disappearance of the worldwide e-mail spam phenomenon.

References

[1] D. Fallows, Spam: How it is hurting email and degrading life on the internet. Technical report, Pew Internet and American Life Project, 2003
[2] P. Mockapetris, Domain names – implementation and specifications, IETF RFC 1035, 1987.
[3] D. Eastlake, Domain Name System Security Extensions, IETF RFC 2535, 1999.
[4] P. Vixie et al, Secret Key Transaction Authentication for DNS (TSIG), IETF RFC 2845, 2000.
[5] M. Delany, Domain-based Email Authentication Using Public-Keys Advertised in the DNS (DomainKeys), IETF Draft, May 2004.
[6] C. Dwork, M. Naor, Pricing via Processing, Or, Combatting Junk Mail, Advances in Cryptology – CRYPTO'92, LNCS vol. 740, Springer Verlag, 1993.

A Protection Tree Scheme for First-Failure Protection and Second-Failure Restoration in Optical Networks

Fangcheng Tang and Lu Ruan

Department of Computer Science, Iowa State University, Ames,
IA 50011, USA
{tfc, ruan}@cs.iastate.edu

Abstract. Recently, a single-link failure protection scheme using protection tree has been proposed in [6].This scheme is scalable in that the protection tree can be adjusted dynamically as the network topology changes. A drawback of this scheme is that it may not be able to find a backup path for certain network links even if the network is 2-connected. In this paper, we first present a new protection tree scheme that can recover from any single-link failure in a 2-connected network. We then give an Integer Linear Program (ILP) formulation to compute a protection tree and allocate spare capacity on the network links so that the total spare capacity required is minimized. Finally, we present a distributed algorithm for fast double-link failure restoration using a protection tree. Simulation results show that around 70% of the double-link failures can be restored by our scheme even though the spare capacity in the network is planned for single-link failures. In addition, our scheme can achieve almost 100% restorability for double-link failures when spare capacity in the network is sufficiently large.

1 Introduction

Optical networks with ultra-high capacity are believed to be the backbone transport network for the next generation Internet. With the explosive growth of the demand for higher bandwidth services, efficient protection of these services becomes an important issue. A variety of protection and restoration schemes have been proposed for optical transport networks and they can be classified as either ring-based [1] or mesh-based [2]. Ring-based schemes such as SONET self-healing rings can provide fast restoration upon network component failure; however, it requires 100% spare capacity redundancy and it is hard to maintain the ring structure as the network grows (i.e. new nodes/links are added to the network). On the other hand, mesh-based schemes are more capacity efficient but have slower restoration speed. *p*-Cycle protection [7], [8] can achieve ring-like recovery speed while retaining the capacity efficiency of mesh-based schemes. Recently, there have been some studies on tree-based protection schemes [4]-[6]. The main advantage of these schemes is that protection trees can be constructed using distributed algorithms, which allows a protection tree to be adjusted dynamically as the network topology changes [6].

The concept of hierarchical protection tree (p-tree) is introduced in [4]. A hierarchical p-tree is a spanning tree in the network for link protection in which links in the higher layers of the tree provide more protection capacity than the links in the lower layers of the tree. A link protection scheme based on hierarchical p-tree was proposed

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 620 – 631, 2005.

in [6] where a node restores traffic through its primary parent or backup parent when an adjacent link of the node fails. For any node u in the network other than the root of the hierarchical p-tree, it has exactly one *primary parent*, which is its parent in the tree. Other than the primary parent, the neighbor nodes that are not u's children are called the *backup parents* of u. Fig. 1a) shows a hierarchical p-tree in an arbitrary network, where thick lines make up the tree. In this example, node G's primary parent is node D, and node G's backup parents are nodes B, C, and H. The protection scheme works as follows:

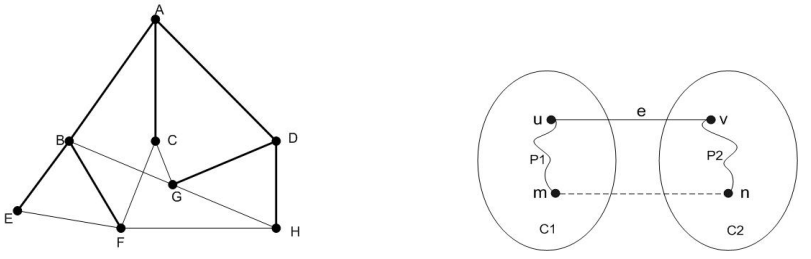


Fig. 1. a) A hierarchical p-tree b) Proof of Theorem 1

– If a non-tree link (i.e., a link not on the p-tree) fails, the nodes adjacent to the failure will reroute the traffic through their primary parents so that the traffic is restored through the p-tree. For example, if link (B, G) in Fig. 1a) fails, the traffic on it will be restored through B-A-D-G.
– If a tree link (i.e., a link on the p-tree) fails, one node adjacent to the failed link will be disconnected from its primary parent. This node will restore the traffic to the p-tree through a designated backup parent. For example, if link (D, G) in Fig. 1a) fails, G will switch the traffic to its backup parent C so that the path G-C-A-D is used to restore the traffic.

A problem with this scheme is that it cannot restore the traffic for certain link failures even if the network is 2-connected. In particular, if a node has no backup parent, this scheme cannot find a backup path for a link failure between the node and its primary parent. For example, when link A-D in Fig. 1a) fails, the scheme cannot find a backup path to restore the traffic since node D has no backup parent.

The rest of the paper is organized as follows. In section 2, we propose a new p-tree protection scheme that can recover from any single link failure in any 2-connected network. In section 3, we present an Integer Linear Program (ILP) formulation to compute an optimal p-tree and spare capacity allocation for a given working capacity distribution so that the spare capacity required is minimized. We present a distributed algorithm for fast double-link failure restoration using p-tree in section 4 and discuss simulation results in section 5.

2 A New P-Tree Protection Scheme

In this section, we present a new p-tree protection scheme that can survive any single link failure in an arbitrary 2-connected network based on the following theorem.

Theorem 1: Given a 2-connected graph G and a spanning tree T of G, for any tree link, there exists a path between the two end nodes of the link that contains exactly one non-tree link.

Proof: Let e = (u, v) be a tree link. Removing e from T will break T into two components C1 and C2 as shown in Fig. 1b). Since G is 2-connected, by Menger's theorem [3] there must exist a path between u and v in G that does not contain e. Thus, there must exist a non-tree link, say (m, n), with one end node in C1 and the other end node in C2. Without loss of generality, assume m is in C1 and n is in C2. (It's possible that m = u or n = v.) Since both C1 and C2 are part of T and each of them is a connected component, there exists a tree path P1 between u and m in C1 and there exists a tree path P2 between v and n in C2. Hence, the path P1 ∪ (m, n) ∪ P2 is a path between u and v that contains exactly one non-tree link (m, n). □

As shown in Theorem 1, for any tree link in a p-tree, there is a path between its two end nodes that contains exactly one non-tree link; such a path can be used to restore the traffic when a tree link fails. In the scheme proposed in [6], when a tree link fails and the node disconnected from its primary parent has no backup parent, the affected traffic can't be restored. To solve this problem, we introduce two types of node called *primary descendant* and *backup descendant* through which a backup path for the failed tree link that contains exactly one non-tree link can be found. It can be shown from Fig. 1b) that for a node u that has no backup parent, there must exist a non-tree link (m, n) such that m is u's descendant and n is connected with u's primary parent v through tree links other than (u, v). We call node m the primary descendant of u and node n the backup descendant of u. When link (v, u) fails, the path between u and v that consists of the tree path from u to m, the non-tree link (m, n), and the tree path from n to v can be used to restore the traffic. For example, in Fig. 1a), node G and C are the primary descendant and backup descendant of node D respectively. When the tree link (A, D) fails, the path D-G-C-A can be used to restore the traffic on link (A, D).

Below is our new p-tree protection scheme where a third rule is added to the scheme proposed in [6]. Our scheme guarantees that any single link failure in a 2-connected network can be survived.

1. If a non-tree link fails, the end nodes of the failure link will reroute the traffic through their primary parents to the p-tree.
2. If a tree link fails and the node disconnected from its primary parent has a backup parent, the node will reroute the traffic through its backup parent to the p-tree.
3. If a tree link fails and the node disconnected from its primary parent has no backup parent, the node will reroute the traffic through its primary descendant and backup descendant to the p-tree.

As in [6], we assume each node is assigned a tree ID represented in a dotted decimal notation. For example, the root node has a tree ID of 1. The tree Ids of its children are 1.1, 1.2, 1.3, etc. A node with tree ID 1.2.3.2 is a child of the node with tree ID 1.2.3. A distributed algorithm was proposed in [6] to build a p-tree by identifying the primary parent and the backup parents of each node and assign a tree ID to each node. We can extend the algorithm to support our new protection scheme by identifying the

primary and backup descendants for each node as follows. For each node x, if it has a neighbor y such that (x, y) is not a tree link, then it sends a message containing its tree ID and y's tree ID towards the root of the tree. When an intermediate node v along the path receives the message, it checks whether x and y are both in the sub-tree rooted at the child from which v receives the message. If yes, v will discard the message. If not, v will record x and y as its primary descendant and backup descendant respectively and forward the message towards the root.

After a p-tree is constructed, each node will know its primary parent, backup parent, primary descendant, backup descendant, as well as the tree ID of every node in the network. It can then calculate the backup paths for its adjacent links and store the paths to a lookup table.

The following procedure BACKUP_PATH_CALCULATION(x, y) can be used to calculate the backup path of link (x, y). In the procedure, function IS_TREE_LINK(x, y) returns *true* if link (x, y) is a tree link, *false* otherwise. P_TREE_PATH(x, y) returns the unique path between node x and node y on the p-tree.

```
BACKUP_PATH_CALCULATION(x,y){
1       if NOT IS_TREE_LINK(x,y)
2             return P_TREE_PATH(x,y)
3       else if y has backup parent
4             return P_TREE_PATH(x, backup parent of y)U{(y, backup parent of y)}
5           else return P_TREE_PATH(y, primary descendant of y)
                    U {(primary descendant of y, backup descendant of y)}
                    U P_TREE_PATH(backup descendant of y, x)
```

3 An ILP Formulation

In this section, we give an ILP formulation for the following problem: given a 2-connected network G(V,E) and the working capacity on each link e in E, compute a spanning tree T(V, E_T) of G and determine the backup path for each link e in E following our p-tree protection scheme so that the total spare capacity required to survive any single link failure is minimized. Such an ILP is useful to preplan the spare capacity to support a projected traffic load in the network.

Let (i, j) in E denote the bidirectional link between node i and node j where i< j. Each bidirectional link (i, j) in E is associated with two arcs, (i→j) and (j→i).

The following are inputs to the ILP.

G: Topology of the network. V: set of the nodes in G numbered 1 through N. E: set of the bidirectional links in G. (i, j) in E denotes a bidirectional link between node i and node j where i< j. E': set of the arcs in G. (i→ j) in E' denotes an arc from node i to node j. $w_{i,j}$: working capacity on link (i, j). $c_{i,j}$: cost of a unit of capacity on link (i, j). Unit cost is assumed here, i.e., $c_{i,j} = 1$.

The following are variables to be solved by the ILP. $T_{i,j}$: take on the value of 1 if link (i, j) is on the p-tree, 0 otherwise. $F^{m,n}_{i,j}$: take on the value of 1 if the backup path of link (m, n) goes through arc (i→j), 0 otherwise. $\Phi^{m,n}_{i,j}$: take on the value of 1 if link (m, n) is protected by link (i, j). $\delta^{m,n}_{i,j}$: take on the value of 1 if link (m, n) is protected by a tree link (i, j). $s_{i,j}$: spare capacity reserved on link (i, j).

Our objective is to minimize the total spare capacity reserved:

$$\text{Minimize} \sum_{(i,j)\in E} c_{i,j} s_{i,j}$$

Subject to the following constraints:

$$\sum_{(i,j)\in E} T_{i,j} = N - 1 \tag{1}$$

$$\sum_{i} F_{i,j}^{m,n} - \sum_{k} F_{j,k}^{m,n} = \begin{cases} -1 & m = j \\ 1 & n = j \\ 0 & otherwise \end{cases}, \forall (m,n) \in E, \forall j = 1..N \tag{2}$$

$$F_{m,n}^{m,n} = 0, \forall (m,n) \in E \tag{3}$$

$$\phi_{i,j}^{m,n} = F_{i,j}^{m,n} + F_{j,i}^{m,n}, \forall (m,n) \in E, \forall (i,j) \in E \tag{4}$$

$$\sum_{(i,j)\in E} \phi_{i,j}^{m,n} - \sum_{(i,j)\in E} \delta_{i,j}^{m,n} = T_{mn}, \forall (m,n) \in E \tag{5}$$

$$\delta_{i,j}^{m,n} \le \phi_{i,j}^{m,n}, \forall (m,n) \in E, \forall (i,j) \in E \tag{6}$$

$$\delta_{i,j}^{m,n} \le T_{i,j}, \forall (m,n) \in E, \forall (i,j) \in E \tag{7}$$

$$\phi_{i,j}^{m,n} + T_{i,j} - 1 \le \delta_{i,j}^{m,n}, \forall (m,n) \in E, \forall (i,j) \in E \tag{8}$$

$$w_{m,n} \times \phi_{i,j}^{m,n} \le s_{i,j}, \forall (m,n) \in E, \forall (i,j) \in E \tag{9}$$

Constraints (1) and (5) ensure a p-tree is set up based on the following theorem.

Theorem 2: Let $T(V_T, E_T)$ be a sub-graph of a connected graph G(V,E). T is a spanning tree of G if the following two conditions hold: (1) $|E_T| = |V| - 1$. (2) For all (u, v) in $E - E_T$, there exists a path P between u and v such that for all link l in P, l is in E_T.

Proof: First, we prove T is connected by contradiction. Assume T is not connected, then T has at least two components, say T1 and T2. Since G is connected, there must exist (u, v) in $E\text{-}E_T$ with u in T1 and v in T2.

By condition 2, there is a path in T that connects u and v. This means that T1 and T2 are connected in T, which is a contradiction.

Next, we prove $|V_T|=|V|$ by contradiction. Assume $| V_T | \neq |V|$, then there exists a node u in $V - V_T$. Since G is connected, u must be connected with some node v in V_T and (u, v) in $E- E_T$. By condition 2, there exists a path in T that connects u and v. This means that u is in V_T which is a contradiction.

We have proved that $T(V_T, E_T)$ is connected and $|V_T| = |V|$. Combined with condition 1, T must be a spanning tree of G. □

The left side of equation (5) denotes the number of non-tree links that protect link (m, n). For a non-tree link (m, n), the right side of equation (5) equals 0, which ensures that a non-tree link is protected by a backup path containing only tree links. Constraint (1) ensures that the number of tree links is |V|-1. According to Theorem 2, constraint (1) and (5) ensure that a spanning tree of G is found.

Using standard network flow formulation, constraint (2) ensures that for all (m, n) in E, there is a backup path R from m to n. Constraint (3) ensures that R does not contain link (m, n).

Constraint (4) ensures that link (m, n) is protected by link (i, j) if and only if the backup path of link (m, n) goes through either (i→j) or (j→ i).

Constraint (5) ensures that a path containing exactly one non-tree link is selected as the backup path of a tree link and a path containing only tree links is selected as the backup path of a non-tree link.

Constraint (6)-(8) ensure that $\delta^{m,n}_{i,j} = 1$ if and only if link (m, n) is protected by link (i, j) and link (i, j) is a tree link.

Constraint (9) ensures that sufficient spare capacity is reserved on each link to protect against any single link failure.

4 Double-Link Failure Restoration

When a link failure occurs, it may take a few hours to a few days to repair the failed link. It is conceivable that a second link failure might occur in this duration, leading to double-link failure in the network. Double-link failure recovery in optical networks has been studied in [10], [11].

In this section, we extend our p-tree protection scheme to deal with double-link failures using the technique of *first-failure protection and second-failure restoration* (1FP-2FR) [9]. The idea of 1FP-2FR is the following: when a p-tree is constructed, the backup path for each link is determined. When the first link failure occurs, the pre-determined backup path for the failed link will be used for traffic restoration. When a second link failure occurs, the affected traffic is rerouted to either the pre-determined backup path of the second failed link or a dynamically computed backup path depending on the relative position of the two failed links.

4.1 Double-Link Failure Recovery Model

Our double-link failure recovery model is based on the recovery method II in [10]. The difference is that our model pre-computes one instead of two backup paths for each link, and dynamically computes a secondary backup path for the second failed link if the pre-computed backup paths cannot be used to restore the second link failure.

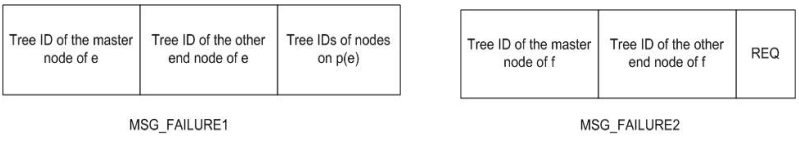


Fig. 2. Message format of MSG_FAILURE1 and MSG_FAILURE2

Our scheme works as follows. Suppose link e and f fail successively, and f fails before e is repaired. When the failure of e is detected, its pre-computed backup path p(e) is used to reroute the traffic on e. Meanwhile, the master node of e sends the message MSG_FAILURE1 (shown in Fig. 2) to all other nodes in the network to inform them of the failure of e. (The master node of a link is one of the two end nodes that has the smaller tree ID.) MSG_FAILURE1 includes the tree ID of the master node of e and the tree ID of the other end node of e. It also includes the tree IDs of the nodes on the backup path p(e) of e so that all nodes in the network are informed of the backup path of link e. When f fails, the traffic on f will be restored in the following four cases.

1. p(f) does not use e and p(e) does not use f. In this case, p(e) will continue to be used to reroute the traffic on e, and p(f) will be used to reroute the traffic on f.
2. p(f) does not use e, but p(e) uses f. Since p(f) is not affected by the failures, the traffic on f (including both the working traffic on f and the traffic rerouted to f due to the failure of e) will be switched to p(f). Thus, the working traffic on e will now be rerouted on p(e) - {f} ∪ p(f).
3. p(f) uses e, but p(e) does not use f. In this case, traffic on f will be rerouted to p(f). However, since e on p(f) is down and p(e) is used to reroute the traffic on e, the working traffic on f will be routed on p(f) - {e} ∪ p(e).
4. p(f) uses e and p(e) uses f. In this case, both p(f) and p(e) are down. Thus, a real-time search for a secondary backup path p'(f) of f that does not use e is needed. When p'(f) is found, the traffic on f will be switched to p'(f). Thus, the working traffic on e will be rerouted on p(e) - {f} ∪ p'(f).

When the master node of f detects the failure of f, it first determines which one of the four cases has occurred. (Note that this requires the master node of f to know the backup path p(e) of e, which can be obtained from MSG_FAILURE1.) If it finds that p(f) uses e and p(e) uses f (case 4), it will broadcast message MSG_FAILURE2 (shown in Fig. 2) to the network with REQ set to *true*. For the other three cases, the master node of f will broadcast message MSG_FAILURE2 to the network with REQ set to *false* and switch the traffic on f to the pre-computed backup path p(f). When a node u receives the message MSG_FAILURE2, it will record the failure of f. If the REQ bit in the message is set to t*rue*, u will run the procedure NEIGHBOR_SEARCH (described in the next section) to search for a neighbor v with certain desired property. If such a neighbor v can be found, u will send a message containing the tree IDs of u and v to the master node of f. Upon receiving the message, the master node of f will compute the secondary backup path p'(f) of f and switch the traffic on f to p'(f).

4.2 Algorithm for Finding the Secondary Backup Path of f

In this section, we describe the detail of the NEIGHBOR_SEARCH procedure.

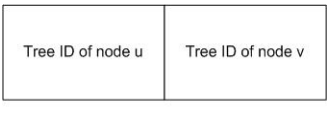


Fig. 3. The message format of MSG_ACK

Let mf be the master node of link f and nf be the other end node of link f. Let T denote the original p-tree before a failure occurs and T' denote the remaining p-tree after link e and f fail, i.e., T'=T-{e}-{f}. When a node u receives a message MSG_FAILURE2 with REQ = *true*, it runs the NEIGHBOR_SEARCH procedure to try to find a neighbor node v such that there is a path from mf to u in T' and there is a path from nf to v in T'. If such a neighbor v can be found, u will create a message MSG_ACK containing the tree ID of node u and the tree ID of node v (shown in

Fig. 3) and send the message to mf. When mf receives MSG_ACK, mf will run algorithm RECOVERY_PATH to compute a secondary backup path p'(f) of f that consists of the tree path from mf to u, the link (u, v), and the tree path from v to nf. After p'(f) is computed, mf will reroute the traffic on f to p'(f). Note that NEIGHBOR_SEARCH, requires u is in the same component as mf in T' so that a tree path between u and mf can be found. We refer to this constraint as *connectivity constraint*.

To find the desired neighbor node v in NEIGHBOR_SEARCH, three scenarios are considered as depicted in Fig. 4. In these figures, thick lines denote tree links and thin lines denote non-tree links; me and ne denote the master node and the other end node of link e respectively; T(x) denotes the sub-tree of T rooted at node x.

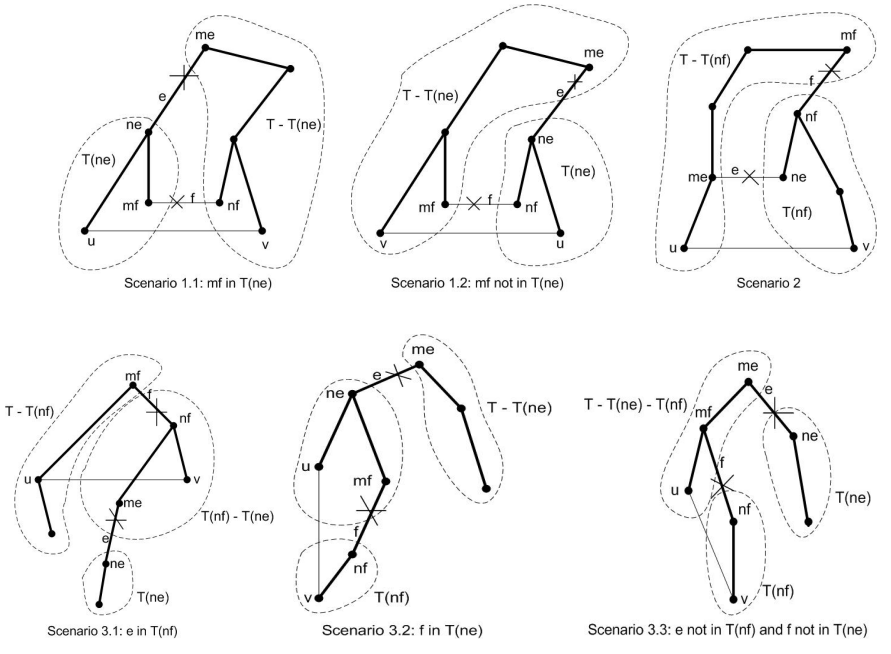


Fig. 4. Three Scenarios in NEIGHBOR_SEARCH

Scenario 1: e is a tree link and f is a non-tree link. As shown in Fig. 4 the failures of e and f divide T into two components T(ne) and T - T(ne). If there is a non-tree link (u, v) between these two components, then there exists a path between mf and nf that goes through some tree links plus link (u, v), and the path can be used as the secondary backup path p'(f) of f. Thus in this case, node u checks whether it is in T(ne) and has a neighbor v in T - T(ne). If so, u will create message MSG_ACK and send it to mf. Note that in scenario 1.2, u is not in the same component as mf, which violates the connectivity constraint. To fix this problem, the tree IDs of u and v will be switched when they are encapsulated into the message MSG_ACK.

Scenario 2: e is a non-tree link and f is a tree link. As shown in Fig. 4, the failures of e and f divide T into two components T(nf) and T - T(nf). If there is a non-tree link (u, v) between these two components, then there exists a path between mf and nf that goes through some tree links plus link (u, v), and the path can be used as the secondary backup path p'(f) of f. Thus, node u checks whether it is not in T(nf) and has a neighbor v in T(nf). If so, u will create message MSG_ACK and send it to mf. Note that in this scenario, u is not in T(nf) ensures u is in the same component as mf, therefore the connectivity constraint is satisfied.

Scenario 3: both e and f are tree links. As shown in Fig. 4 the failure of e and f divide T into three components. If there is a non-tree link (u, v) between the component containing mf and the component containing nf, then there exists a path between mf and nf that goes through some tree links plus link (u, v), and the path can be used as the secondary backup path p'(f) of f. There are three cases based on the relative position of link e and f on T. In scenario 3.1, the three components are T(ne), T(nf) - T(ne) and T - T(nf). Node u checks whether it is not in T(nf) and has a neighbor v in T(nf)-T(ne). If so, u will create message MSG_ACK} and send it to mf. In scenario 3.2, the three components are T(nf), T(ne) - T(nf) and T-T(ne). Node u checks whether it is in T(ne) - T(nf) and has a neighbor v in T(nf). If so, u will create message MSG_ACK and send it to mf. Scenario 3.3 covers the third case, where the three components are T(ne), T(nf) and T - T(ne) - T(nf). Node u checks whether it is neither in T(ne) nor in T(nf) and has a neighbor v in T(nf). If so, u will create message MSG_ACK and send it to mf. Note that in all three cases, u is in the same component as mf, therefore the connectivity constraint is always satisfied.

```
NEIGHBOR_SEARCH{
//Scenario 1
1 if IS_TREE_LINK(me,ne) AND NOT IS_TREE_LINK(mf,nf)
2 {
3      if IS_ON_TREE(ne,u) AND u has a neighbor node v s.t. NOT IS_ON_TREE(ne,v) {
4             if NOT IS_ON_TREE(ne, mf) //Scenario 1.2
5                    SWITCH_ID(u,v)
6             Send MSG_ACK to mf }
7 }
//Scenario 2
8 if NOT IS_TREE_LINK(me,ne) AND IS_TREE_LINK(mf, nf)
9 {
10     if NOT IS_ON_TREE(nf, u) AND u has a neighbor node v s.t. IS_ON_TREE(nf,v)
11           Send MSG_ACK to mf
12}
//Scenario 3
13 if IS_TREE_LINK(me,ne) AND IS_TREE_LINK(mf, nf)
14 {
15     if IS_ON_TREE(nf,me) { //Scenario 3.1
16           if NOT IS_ON_TREE(nf,u) AND u has a neighbor node v
             s.t. IS_ON_TREE(nf,v) AND NOT IS_ON_TREE(ne, v)
17                  Send MSG_ACK to mf }
18     else if IS_ON_TREE(ne,mf) { //Scenario 3.2
19           if IS_ON_TREE(ne,u) AND NOT IS_ON_TREE(nf,u)
             And u has a neighbor v s.t. IS_ON_TREE(nf,v)
20                  Send MSG_ACK to mf }
21     else { Scenario 3.3
22           if  NOT IS_ON_TREE(ne,u) AND NOT IS_ON_TREE(nf,u)
             AND u has a neighbor node v s.t. IS_ON_TREE(nf,v)
23                  Send MSG_ACK to mf }
24 }
25}
```

Procedure NEIGHBOR_SEARCH

The pseudocode of procedure NEIGHBOR_SEARCH is proposed where function IS_ON_TREE(x, y) checks whether node y is on the sub-tree T(x) which is rooted at node x and function SWITCH_ID(x, y) is used to switch the tree IDs of x and y when they are encapsulated into the message MSG_ACK.

On receiving the first message MSG_ACK, mf computes a secondary backup path p'(f) of f using the RECOVERY_PATH procedure given below, where function P_TREE_PATH(x, y) is used to find a tree path between x and y.

```
RECOVERY_PATH{ return P_TREE_PATH(mf, u) ∪ {(u, v)} ∪
P_TREE_PATH(v, nf) }
```

5 Numerical Results

Three test networks Fig. 5 are used to evaluate the performance of our p-tree scheme. Net1 is an artificial 10-node 22-link network taken from [12]. Net2 is the 15-node 28-link Bellcore New Jersey LATA network, which is a widely used metropolitan area model. Net3 is a modified NJ LATA network with 11 nodes and 22 links taken from [11]. A uniform demand matrix with 2 demand units between every node pair is used for all three test networks and shortest path routing is used to route the demands.

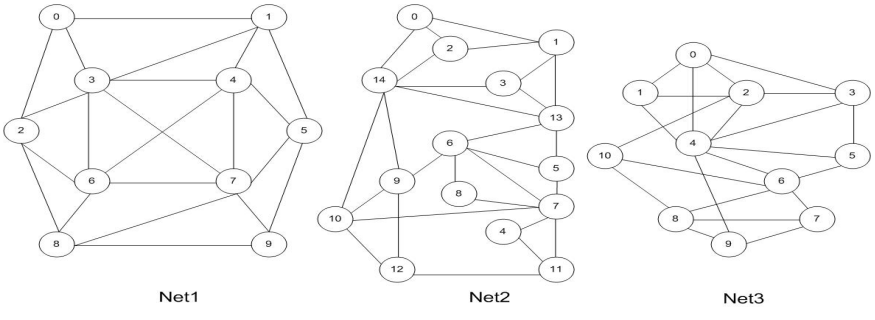


Fig. 5. Topology of the test networks

Table 1. Results of the ILP Solution

Network	Working Cap.	Spare Cap.	Redundancy
NET1	142	96	0.68
NET2	456	424	0.93
NET3	190	178	0.94

Table 1 shows the performance results of our p-tree scheme for single link failure protection, which are obtained by solving the ILP given in section 3. Table 1 gives the total working capacity, the total spare capacity, and the redundancy (i.e. the ratio of the total spare capacity to the total working capacity). Due to the sharing of spare capacity on links used by multiple backup paths, the p-tree scheme leads to less than 100% redundancy.

Table 2. Results of Double-link Failure Restoration

Network	# Total pairs	# NS pairs	R1	R2
NET1	462	400	0.72	0.99
NET2	756	572	0.67	0.99
NET3	462	386	0.71	0.98

Table 2 shows the performance results of our p-tree scheme for double-link failure restoration. In the table, #Total pairs is the number of all possible <e, f> pairs assuming link e and link f fail successively. #NS pairs is the number of <e, f> pairs that do not require a search for the secondary backup path of f (such pairs are covered in case 1-3 in section 4.1). R1 is the ratio of the number of fully restorable <e, f> pairs to the number of total pairs when the spare capacity is allocated for protecting single link failures (i.e. the spare capacity allocation is computed by the ILP given in section 3 R2 is the ratio of the number of fully restorable <e, f> pairs to the number of total pairs when sufficient spare capacity is allocated on each link. (A <e, f> pair is fully restorable if all the working capacity on e and f can be restored by our double-link failure restoration scheme.) As shown in Table 2 for NET1, NET2, and NET3, 87%(400/462), 76%(572/756), and 84%(386/462) of all possible double-link failures can be restored without a search for the secondary backup path of f. Therefore, our scheme can achieve fast double-link failure recovery in most of the cases. With the spare capacity planned for single link failure, our double-link failure restoration scheme can fully restore 72%, 67% and 71% of all the double link failures in NET1, NET2 and NET3 respectively. With sufficient spare capacity available in the network, the percentage of fully restorable double-link failures reaches 99%, 99%, and 98% in NET1, NET2, and NET3 respectively. There are two reasons that the percentage is less than 100% even though sufficient spare capacity is available in the network. First, a secondary backup path for link f may not exist. Second, our algorithm may not be able to find a secondary backup path for link f even though such a path exists because we require that the secondary backup path can only use one non-tree link.

The results in table 2 show that around 70% of the double-link failures can be fully restored by our scheme even when the spare capacity is planned for single link failure. In addition, our scheme can achieve almost 100% restorability for double-link failures when spare capacity in the network is sufficiently large. Thus, the proposed scheme is very effective for double-link failure restoration.

6 Conclusion

In this paper, we propose a new p-tree scheme that can protect against any single-link failure in a 2-connected network. To minimize the spare capacity requirement, we give an ILP formulation to compute the optimal p-tree and spare capacity allocation for a network with given working capacity distribution. We also develop a distributed restoration algorithm for dealing with double-link failures, which searches for a secondary backup path for the second link failure in real-time when necessary. Numerical results show that around 70% of the double-link failures can be fully restored by our scheme even though the spare capacity is planned for single link failure. In

addition, our algorithm can achieve almost 100% restorability for double-link failures when spare capacity in the network is sufficiently large.

Acknowledgments

This work is supported in part by the National Science Foundation under CAREER Award #ANI-0237592.

References

1. T. H. Wu, "Emerging technologies for fiber network survivability", IEEE Communications Magazine, Vol. 33, No. 2, pp. 58-74, February 1995.
2. S. Ramamurthy, Laxman Sahasrabuddhe, "Survivable WDM Mesh networks", Journal of Lightwave Technology, Vol. 21, No. 4, April 2003.
3. J. A. Bondy and U. S. R. Murty, "Graph Theory with Applications", American Elsevier Publishing, 1976.
4. Shahram Shah-Heydari, Oliver Yang, "A tree-based algorithm for protection & restoration in optical mesh", Proc. of Canadian Conference on Electrical and Computer Engineering, CCECE'2001, Toronto, Ontario, Canada, May 2001, pp. 1169-1173.
5. M. Medard, et al., "Redundant tress for preplanned recovery in arbitrary vertex-redundant or edge-redundant graphs", IEEE/ACM Transactions on networking, Vol. 7, No. 5, October 1999.
6. Shahram Shah-Heydari, Oliver Yang, "Hierarchical Protection Tree Scheme for Failure Recovery in Mesh Networks", Photonic Network Communications, 7:2,145-159, March 2004.
7. W. D. Grover, D. Stamatelakis, "Cycle-oriented distributed preconfiguration: ring-like speed with mesh-like capacity for self-planning network restoration", Proc. of IEEE ICC 1998, pp. 537-543, June 1998.
8. D. A. Schupke, C. G. Gruber and A. Autenrieth, "Optimal configuration of *p*-cycles in WDM networks", Proc. of IEEE ICC 2002, pp. 2761-2765, April 2002.
9. W. D. Grover, "Mesh-based Survivable Networks: Options and Strategies for Optical, MPLS, SONET and ATM Networking", Prentice Hall PTR, Upper Saddle River, New Jersey, 2003, chapter 8, pp. 529.
10. H. Choi, S. Subramaniam and H.A. Choi, "On Double-Link Failure Recovery in WDM Optical Networks", IEEE INFOCOM 2002
11. W. He and A. K. Somani, "Path-based Protection for Surviving Double-Link Failures in Mesh-Restorable Optical Networks", Proceeding of IEEE Globecom 2003, pages 2558-2563, Dec. 2003.
12. P.R. Iraschko, M.H. MacGregor, W.D. Grover, "Optimal capacity placement for path restoration in STM or ATM mesh-survivable networks", IEEE/ACM Transactions on Networking Volume: 6, Issue: 3 , June 1998 Pages:325 - 336.

Distributed Dynamic Resource Management for the AF Traffic of the Differentiated Services Networks*

Ling Zhang[1], Chengbo Huang[1,2], and Jie Zhou[1]

[1] Network Engineering and Research Center, South China University of Technology, Guangzhou 510640, China
[2] School of Communication and Electronics, JiangXi Science & Technology Normal University, Nanchang 330013, China
{ling, chbhuang, jiezhou}@scut.edu.cn

Abstract. This paper presents a fully distributed dynamic resource management scheme for the assured services based on Assured Forwarding Per Hop Behavior (AF PHB) of Differentiated Services (DiffServ) networks. Our scheme combines the ideas of state-based approaches and stateless approaches and overcomes some drawbacks of the current proposed schemes. It's scalable because no per-flow states are required in the core routers. This scheme includes a light weighted signaling protocol, the definitions of aggregate states and the algorithms for managing aggregate states. The simulation results demonstrate that the proposed scheme can accurately control the admissible region, achieve high utilization of network resources and simultaneously provide statistical end-to-end Quality of Service (QoS) guarantees for the aggregate traffic.

1 Introduction

It is known that the Integrated Services (IntServ) approach, while allowing hard Quality of Service (QoS) guarantees, suffers from scalability problems in the core network. To overcome this and other limits of IntServ, the Differentiated Services (DiffServ) paradigm has been proposed [1]. By leaving untouched the basic Internet principles, DiffServ provides supplementary tools to further move the problem of Internet traffic control up to the definition of suitable pricing/service level agreements (SLAs) between peers. However, DiffServ lacks a standardized admission control and resource reservation scheme. Upon overload in a given service class, all flows in that class suffer a potentially harsh degradation of service. RFC2998 recognizes this problem and points out that *"further refinement of the QoS architecture is required to integrate DiffServ network services into an end-to-end service delivery model with the associated task of resource reservation"* [2]. RFC2990 suggests defining an *"admission control function which can determine whether to admit a service differentiated flow along the nominated network path"* [3].

* This research is supported by the National Basic Research Program of China (No.2003CB314805) and Guangdong Key Laboratory of Computer Network under China Education and Research GRID (ChinaGrid) project CG2003-CG2005.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 632 – 641, 2005.

Recent papers [4,5] have shown that dynamic resource management can be provided over DiffServ networks by means of explicit signaling protocol and aggregate states. These schemes extend the DiffServ principles with new ones necessary to provide dynamic resource reservation and admission control in DiffServ domains. The RMD (Resource Management in DiffServ) proposed in [5] uses the soft state refresh paradigm to account for and time-out resource usage, but it's difficult to eliminate the aggregate state deviation caused by the network jitter that is introduced by routers' queues. Currently, only an approximate approach is proposed to solve this problem [6]. The scheme proposed in [4] doesn't use soft state in core routers and relies instead on garbage collection. But this scheme doesn't solve the problems caused by the lost signaling messages. The lost signaling messages degrade the utilization of resources and even lead to the situation that the node collapses at last.

In the recent years a new family of admission control solutions named EAC (End-point Admission Control) [7-11] has been proposed. Initially, EAC builds upon the idea that admission control can be managed by pure end-to-end operation, involving only the source and destination hosts. It's stateless in the core routers and applicable to the DiffServ networks. EAC can provide statistical end-to-end QoS guarantees and achieve higher network utilization. In the later work, in order to achieve more reliable network state information, inner router can determine whether a new call can be locally admitted by means of suitable Measurement Based Admission Control (MBAC) [9,10]. Because there are not any states in the core routers, stateless EAC exists transient phenomena. If a large number of connection requests arrive at a router simultaneously, the measurement mechanism is not capable of protecting the system from over-allocation, i.e., all the requests may be accepted. In [11], the communication paths are reserved in a centralized Bandwidth Broker (BB) [12] for the requests that are being processed to eliminate this problem. The requests are processed in sequence. But the exclusive use of paths will cause the drawback that the new arrived requests cannot be processed in time, result in inefficient admission control and increase the burden of BB at the same time.

In this paper, we introduce a fully distributed dynamic resource management scheme for the AF traffic [13] of the DiffServ networks. It is scalable because no per-flow states are required in the core routers. Our approach combines the ideas of state-based schemes and stateless schemes. The goal of our proposed solution is ⅰ) to design a light weighted dynamic resource reservation protocol, accomplish the task of admission control and resource reservation by means of explicit signaling protocol and aggregate states, ⅱ) to define aggregate states properly, avoid using the soft state refresh paradigm and solve the transient phenomena of EAC, ⅲ) to solve the problems caused by the lost signaling messages, and ⅳ) to provide statistical end-to-end Quality of Service (QoS) guarantees for the aggregate traffic, achieve the statistical multiplexing gain and high utilization of network resources.

The remainder of this paper is organized as follows. Section 2 describes the signaling protocol and the end system (hosts or servers) behavior. Section 3 gives the definitions of aggregate states, the algorithms for processing signaling messages, admission control and resource reservation. The method for processing the lost signaling messages is also described in this section. In section 4 we present simulation results and performance analysis. Finally, we conclude in section 5.

2 Signaling Protocol

This section introduces our signaling protocol and end system behavior.

Our scheme currently only considers a single DiffServ domain. We assume that end system receives services directly from the DiffServ network. The QoS requirements of end system can be denoted as

$$QoS = \{r_e, \varepsilon_e, d_e\} \tag{1}$$

Where r_e is the peak rate, ε_e is the end-to-end loss rate, and d_e is the end-to-end delay bound.

Our signaling protocol is simple and it's a sender-initiated protocol. There are only three signaling messages described as follows.

- Req: Generated by the sender to request resource reservation. This message includes the QoS requirements. The result of admission control and resource reservation is transported by this message hop by hop and arrives at the receiver at last. The number of hops where the resource was successfully reserved for a connection request is also transported to the receiver by this message.
- Ack: Sent by the receiver to the sender. This message is an acknowledgment to Req. It returns the reservation result of a connection request and the hops.
- Clr: Sent by the sender. The routers' resource states are regulated by this message (see section 3). This message includes the peak rate of the connection request and the hops returned by Ack message.

Fig. 1 indicates a successful session of end system.

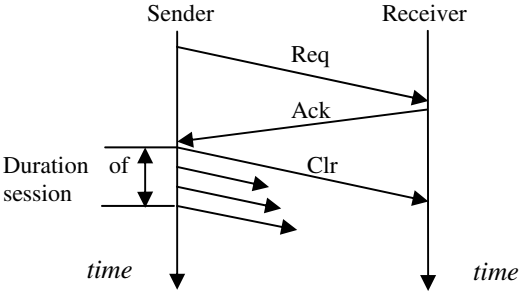


Fig. 1. A successful session of end system

In order to establish a guaranteed service connection, the initiator sends Req towards the destination. The routers on the communication path process Req and save the result of admission control and resource reservation and the hops into Req. The receiver sends back Ack to the sender when it receives Req. When Ack indicates a successful reservation, the sender may immediately start to send data and simultaneously send Clr to the receiver.

The sender starts a waiting timer with a value T after sending a resource request. The value T can be defined as maximum RTT. If no acknowledgement arrives during

T seconds, the sender can send a new reservation request again. The establishment of a connection is repeated until an acknowledgment arrives or the sender decides to stop. The acknowledgement for the old reservation request will be ignored. This case is handled as the case of lost signaling messages (see Section 3.3).

When the connection request is rejected, the sender will send Clr at once to release the resources reserved in the nodes that have successfully processed this request. In this way, the degradation of link utilization will be minimized. The sender may restart the resource request procedure after waiting a stochastic time, or degrade the QoS requirements to request again or stop.

The router behavior will be described in the next section.

3 Aggregate States

This section describes the definitions of aggregate states, the algorithms for processing signaling messages, admission control and resource reservation. This section also describes the method for processing the lost signaling messages.

3.1 Definitions of Aggregate States

In EAC schemes, we can observe that the aggregate traffic characteristics achieved by the run-time traffic measurements performed within each router can characterize the aggregate traffic state, and this state refreshes itself without the help of other mechanism (such as refresh signaling message). On the other hand, in the state-based schemes, the state variable can record the resource requests that are being processed in the router and the connection requests can be processed in sequence. Based on the above consideration, we give the definitions of aggregate sates.

Our scheme uses capacity reservation. In each router, there are two state variables for each AF aggregate traffic class defined as follows.

A. The available capacity Ca that represents the resources left for admitting new resource requests. Ca is estimated from run-time traffic measurements performed within each router. The algorithms used for computing Ca are described as follows.

We currently consider only fluid flow. The AF aggregate traffic classes are described as a cluster of nonnegative stationary stochastic processes and independent each other. We assume that all the routers use a schedule algorithm like CBQ [14] to guarantee the minimum bandwidth for each aggregate traffic class.

We get the mean $\overline{R}_k$ and the variance σ_k^2 of the aggregate maximal rate envelope based on the run-time measurements performed in the routers [10] and estimate the available capacity Ca [11] using these characteristics of the aggregate traffic flow.

Consider that an aggregate traffic class is serviced at rate c, delay bound d and loss rate ε in a router; the available capacity Ca can be estimated using the characteristics of the aggregate traffic flow. Denote $k^*\tau$ as the dominant interval of all time intervals that produces the maximum loss probability. If the current maximum loss probability excesses a threshold value, the available capacity Ca of the aggregate traffic class is set to zero. Otherwise, Ca can be approximated as

$$Ca \approx \frac{1}{k^* \tau}(c(k^* \tau + d) - k^* \tau \bar{R}_k^* - zk^* \tau \sigma_k^*) \tag{2}$$

Where z is defined as $1-\Phi(z) \approx \varepsilon$, approximated as $z \approx \sqrt{|\log(2\pi\varepsilon)|}$. Φ is the probability distribution function of standard normal distribution.

In another word, the value of Ca represents the amount of traffic that is transmitting in an aggregate traffic class. The capacity of a departed flow is automatically released in the value of Ca, and then no explicit tear down message is needed to release the resource when the sender finishes the connection. Using Ca as an aggregate state, we achieve the statistical multiplexing gain and higher network utilization.

B. The requested capacity Rr reserved for the connection requests that are being currently processed on the communication paths. The requested rate is added to this variable when a router receives a request and there is enough local capacity to admit this resource request. When a connection request has been accepted and data transmission has started, the capacity reserved for this request is removed to ensure that only currently requested capacity is stored in Rr. Using the state variable Rr, the requests can be processed in sequence to eliminate the transient phenomena. The requests are processed in time and the efficiency of admission control is guaranteed.

Admission control and resource reservation are carried out by means of the cooperation of these two state variables and the protocol. During the connection request phase, Ca and Rr take part in the admission control decision. The admitted resources are reserved in Rr. During the data transmission phase, the capacity of the admitted connection requests is confirmed in Ca and needn't be reserved in Rr again. The value of Rr is regulated by signaling message Clr. The algorithms for managing the aggregate states are described in detail in the next section.

3.2 Management of Aggregate States

We assume that every node provides the guarantees of minimum bandwidth, delay bound and loss rate for each aggregate traffic class and these values are pre-configured in the local information base of the routers.

Consider that bandwidth C is allocated to an aggregate traffic class in a router, r_e, ε_e and d_e are the QoS requirements of end system (see Eq. (1)). The algorithms for processing signaling message, admission control and resource reservation in the router are described as **Fig. 2**.

In **Fig. 2**, ε_N and d_N respectively denote the guarantees of the end-to-end loss rate and the end-to-end delay bound provided by the communication path from the first node to the local node n. The algorithms for computing ε_N and d_N are

$$\varepsilon_N = 1 - e_N = 1 - (1-\varepsilon_n)e_{N-1} = 1 - (1-\varepsilon_n)\prod_{i=1}^{m}(1-\varepsilon_i) \tag{3}$$

and

$$d_N = d_{N-1} + d_n + t_n = \sum_{i=1}^{m} d_i + \sum_{i=1}^{m} t_i + d_n + t_n \tag{4}$$

Where d_i and ε_i are respectively the guarantee of delay bound and loss rate for this aggregate traffic class in the node i, t_i is the transport delay of the link between node i and $i+1$, m is the number of nodes on the communication path from the first node to the node $n-1$. When the request is locally admitted, e_N, d_N and *hops* are saved into Req and transported to the next node.

```
void server ( ) { //message type: 0 - Req, 1 – Clr
Rr = 0 ; Ca = C ;
while (1) {
  int type=receive ( );
  switch type {
   0: if ( acc ) // local admission control
        if ( ε_N < ε_e and d_N < d_e )
          if (Ca < Rr + r_e)
              mark Req with rej
          else
              Rr+ = r_e ; hops+ = 1;
              save hops, e_N , d_N into Req ;
   1: Rr− = r_e ; hops− = 1 ; // regulate state
      if (hops == 0) drop (Clr);
     } //switch type
  if (Ca < Rr + Th) { // starts the resource cleaning procedure
       wait (W ) ; Rr = 0 ;
    } // if (Ca < Rr + Th)
  } //while (1)
} //void server ( )
```

Note: *hops* -The hops

r_e - Peak rate; *acc* -Accept; *rej* -Reject

Th -Minimum resource unit that can be allocated

Fig. 2. Algorithms for managing the aggregate states

When a resource request is received, the router will compute ε_N, d_N and Ca. The request will be accepted locally if the guarantees of the end-to-end loss rate and the end-to-end delay bound provided by the communication path from the first node to the local node are satisfied and there is enough capacity to admit this resource request. If the request is rejected, Req is marked with *rej* (reject) and downstream

nodes along the communication path won't process this request. The sender sends the message Clr simultaneously when it begins to transmit data. Rr is regulated by Clr to ensure that only currently requested capacity is stored in Rr. When the connection request is rejected, using Clr, the routers release the resources reserved for this rejected request. In this way, the degradation of link utilization will be minimized.

3.3 Processing the Lost Signaling Messages

The signaling messages might be lost due to the network congestion or link failure. The lost signaling messages will result in the situation where the reserved resource in state *Rr* cannot be released forever. This case degrades the utilization of resources and even leads to the situation that the node collapses at last when the resources that cannot be released accumulate to the amount of capacity C. In our approach, a resource cleaning procedure (see **Fig. 2**) is started in time to clean the resources that cannot be released. In the period of waiting time *W*, all Req messages passing the nodes that are doing resource cleaning procedure are marked with *rej* and the message Clr is processed in normal. After the time period *W* (*W*>=RTT), only the resources that cannot be released are left in Rr, thus $Rr = 0$.

The resource cleaning procedure happens in following cases: □) the capacity itself is not enough to admit new requests because all the capacity is in use,□) the capacity that cannot be released accumulates to some degree. During the resource cleaning procedure, all new requests will be blocked. In the first case, the router blocks the new requests is reasonable. In the second case, the block of new requests will degrade the utilization of resources. However, the influence is limited because the duration of resource cleaning procedure is a shot period (about RTT). On the other hand, it releases more capacity and results in fewer activations of the resource cleaning procedure. Our approach has good robustness because the node will not collapse forever.

4 Simulation Results and Performance Analysis

In this section we evaluate the performance of our scheme with NS-2 [15].

The source used in our simulation is a trace file that is derived from Mark Garrett's MPEG encoding of the Star Wars movie [16]. The topology is a five-node network path. There is a FIFO queue in each node and the queue length is the product of link capacity and queuing delay bound. The link capacity is 45Mb/s. We only consider the queuing delay for the link transport delay is constant. In the first and second experiments, the resource requests arrive with exponential inter-arrival time with mean 3 seconds and each flow has exponential holding time with mean 600 seconds. The measurement window is set to 2 seconds and the time interval τ is set to $10ms$. In the third experiment, the mean inter-arrival time of the requests is changed and other parameters are the same as the previous two experiments.

First, we use the average utilization of the link as the performance metric to evaluate the admissible region and the network utilization. The guarantee of the end-to-end loss rate of aggregate traffic class is set to $\varepsilon = 10^{-4}$ and $\varepsilon = 10^{-2}$. The guarantee of the end-to-end delay bound of the aggregate traffic class changes in a range. The result is illustrated in **Fig. 3**. We can observe that: i) our approach has high

network utilization. For example, in the case of $\varepsilon = 10^{-2}$, the utilization reaches respectively 85% and 91% when $d = 20ms$ and $d = 50ms$. ii) our approach can accurately control the admissible region under the condition of the available network resources and QoS constraints of the aggregate traffic class. For example, the utilization of $\varepsilon = 10^{-2}$ is higher than $\varepsilon = 10^{-4}$. The bigger the end-to-end delay bound is the higher the utilization reaches.

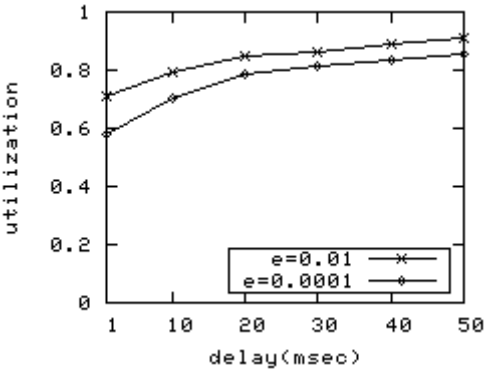


Fig. 3. Average utilization of the link

Second, we measure the end-to-end loss rate of the aggregate traffic class on run-time under the condition that the guarantees of the end-to-end loss rate and the end-to-end delay bound of the aggregate traffic class are respective $\varepsilon = 10^{-4}$ and $d = 20ms$. The result is illustrated in **Fig. 4**. In the stationary state, we can see from the result that the measurement values fluctuate around the target value. This result demonstrates that our approach can provide statistical end-to-end QoS guarantees for the aggregate traffic.

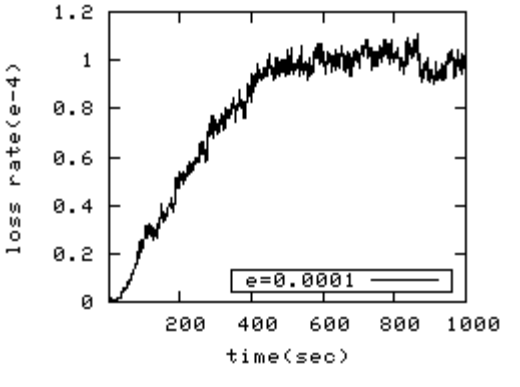


Fig. 4. End-to-end loss rate of the aggregate traffic based on run-time measurements

Finally, we change the mean inter-arrival time of the requests to compare the efficiency of admission control of our approach with 'PSG03' [11]. The guarantees of the end-to-end loss rate and the end-to-end delay bound of the aggregate traffic class are respective $\varepsilon = 10^{-4}$ and $d = 20ms$. The duration of simulation is set to 600s. We can see from **Fig. 5** that the utilization decreases as the mean inter-arrival time increases, but our approach 'DRMD' can achieve higher utilization than 'PSG03'. This result demonstrates that our approach is more efficient than 'PSG03'.

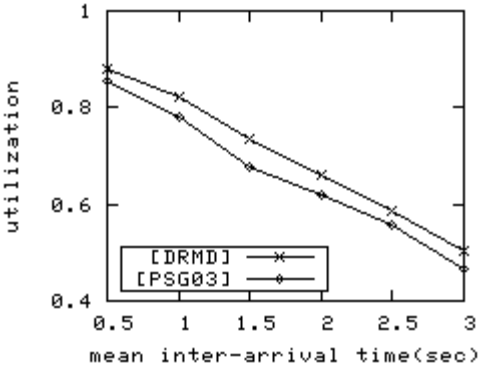


Fig. 5. Experiment of comparing the efficiency of admission control

5 Conclusions

Our fully distributed dynamic resource management scheme presented in this paper extends the DiffServ principles with new ones necessary to provide dynamic resource reservation and admission control in DiffServ domains. By properly defining the aggregate states and an simple explicit signaling protocol, our proposed scheme avoids using the soft state refresh paradigm, solves the transient phenomena of EAC and the problems caused by the lost signaling messages. We demonstrate the performance of our scheme by simulation experiments. In our future work, we will implement our scheme on a prototype router built on Intel IXP2400 network processor and evaluate the processing delay of signaling messages.

References

1. S. Blade, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, An Architecture for Differentiated Services, RFC2475, December 1998.
2. Y. Bernet, R. Yavatkar, P. Ford, F. Baker, L. Zhang, M. Speer, R. Braden, B. Davie, J. Wroclawski, E. Felstaine, A Framework for Integrated Services Operation Over DiffServ Networks, RFC 2998, November 2000.
3. G. Huston, Next Steps for the IP QoS Architecture, RFC2990, November 2000.
4. E. Ossipov , G. Karlsson, A Simplified Guaranteed Service for the Internet, PfHSN 2002, LNCS 2334, pp. 147–163, 2002.

5. L. Westberg, A. Császár, G. Karagiannis, et al, Resource Management in Diffserv (RMD): A Functionality and Performance Behavior Overview, PfHSN 2002, LNCS 2334, pp. 17–34, 2002.
6. Marquetant, A., Pop, O., Szabo, R., Dinnyes, G., Turanyi, Z,: Novel enhancements to load control - a soft-state, lightweight admission control protocol, QofIS2001, LNCS 2156,PP. 82–96,2001.
7. Breslau L , Knightly E , Shenker S , Stoica I , Zhang H, Endpoint admission control : Architectural issues and performance, In : Proceedings of ACM SIGCOMM ' 00 , Stockholm, Sweden , 2000. PP. 57 ~ 69.
8. C. Centinkaya, V. Kanodia, E. Knightly, Scalable services via egress admission control, IEEE Transactions on Multimedia,VOL. 3, NO.1, MARCH 2001. PP. 71~81.
9. L. Breslau, S. Jamin, S. Schenker, Comments on the performance of measurement-based admission control algorithms, IEEE Infocom 2000, Tel-Aviv, Israel, March 2000.
10. Qiu J , Knightly E, Measurement-based admission control with aggregate traffic envelopes, IEEE/ACM Transactions on Networking , 2001 , 9 (2) : 199 ~ 215.
11. PANG Bin, SHAO Huai-Rong, GAO Wen, A Measurement-based Admission Control Scheme For DiffServ Network: Design and Application, Chinese Journal of Computers, Vol. 26, No.3, Mar. 2003.257-265.
12. K. Nichols, V. Jacobson, and L. Zhang, A Two-bit Differentiated Services Architecture for the Internet, RFC2638, July 1999.
13. J. Heinanen, F. Baker, W. Weiss, J. Wroclavski, Assured Forwarding PHB Group, RFC 2597, June 1999.
14. Floyd S, Jacobson V. Link sharing and resource management models for packet networks. IEEE/ACM Transactions on Networking, VOL.3, NO.4, AUGUST 1995, pp. 365 ~ 386.
15. The Network Simulator (NS), http://www.isi.edu/nsnam/
16. http://www.research.att.com/~breslau/vint/trace.html

Constructing Correlations of Perturbed Connections Under Packets Loss and Disorder*

Qiang Li, Qinyuan Feng, Kun Liu, and Jiubin Ju

Department of Computer Science, JiLin University, ChangChun JiLin 130012, China
sckextjg@mail.jlu.edu.cn

Abstract. One of the key problems of detecting stepping stones is the construction of connections' correlations. We focus on the use of detecting windows and propose two methods for constructing correlations of perturbed connections. Within the attacker's perturbation range, the first method uses packet-based window and the average value of the packets in the detecting window is set to increase periodically. The method can construct correlations in attacking connection chains by analyzing the increase of the average value of the inter-packet delay between the two connection chains. The second method uses time-based windows. It divides time into segments, forms segments into groups and uses pairs of groups to take the watermarks. These methods can reduce the complexity of correlation computations and improve the efficiency of detecting. The second method can even work under packets loss and disorder.

1 Introduction

Network attacks have become a severe problem to current network systems. Usually network attackers conceal their real attacking paths by establishing interactive connections along a series of intermediate hosts (stepping stones) before they attack the final target[1]. Since it is very easy to implement and use connection chain attacking techniques, source tracing on network attacks remain one of the most difficult problems in network security.

To identify the real source of attack, tracer can execute a complex tracebacking process from the last host of connection chain using each host's logs. But this approach is not available because attackers usually destroy the tail. Tracer can also install a passive connection traffic monitor in the networks and construct correlations through analyzing input or output traffic of each host. The key problem of connection chain tracebacking is connection correlation in the intermediate hosts (stepping stones).[1,2] However, the correlation process is more difficult because traffic's encrypt or compression changes the connection content and delay changes the connection time. And the correlation process on the stepping stones must be quick because the network intrusion often happens in high speed networks.

* Supported by NSFC(90204014).

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 642–651, 2005.

In this paper, we propose two methods with windows to construct correlations of perturbed connections under packets loss and disorder.

The first method uses packet-based window. Within the attacker's perturbation range, this method analyzes the activity degree of the correlation windows and monitors increasing characteristic of inter-packets delay. The stepping stone connection in each detecting window can have an increasing average value of the inter-packets delay through changing a part of the packets' arrival delay at the network's ingress. The method can construct correlations in attacking connection chains through detecting these increases at the network egress. The method uses actively perturbed correlation algorithm based on passively monitoring the network egress, it can reduce the complexity of correlation computations and improve the efficiency of detecting stepping stones when the attackers use the encrypting connection and timing perturbation.

The second method uses time-based window much more novel, it can deal with the loss and disorder of the packets. It divides the time is small segments and uses the average time of the packets in the segments to delegate the segments. then it adds watermark in the segments. By limiting the attacker's ability, it will be easy to detect the watermark and construct correlations.

The remainder of the paper is organized as follows. In section 2, we give the definitions and assumptions of our method. In section 3, we propose packet-based window method. In section 4, we propose time-based window method. In section 5, we evaluate correlation effectiveness of our proposed correlation metrics through experiments. In section 6, we give a summary of related works. In section 7, we conclude with summary of our findings.

2 Definitions and Assumptions

Given a series of computer hosts H_1, H_2,..., $H_n (n > 2)$, when a person (or a program) sequentially connects from H_i into H_{i+1}(i=1,2,..n-1), we refer to the sequence of connections on $< H_1, H_2, ... H_n >$ as a connection chain, or chained connection. The tracing problem of a connection chain is, given H_n of a connection chain, to identify H_{n-1}, H_{n-2},...,H_1. We define the intermediate hosts of the connection chain as stepping stones. A pair of network connections is defined as stepping stones connection pair while the pair of connections are both of one part of a connection chain.

We use t_i and $t'_i$to represent the arrival and departure times, respectively, of the i th packet. We define the arrival inter-packet delay of the i th packet as $d_i = t_{i+1} - t_i$ and the departure inter-packet delay as $d'_i = t'_{i+1} - t'_i$. We further define the perturbation by the attacker as c_i. Then we have $t'_{i+1} = t'_i + c_i + u$. In this paper, u represents the delay of system(such as processing time, waiting time, etc). Assume the delay range that the attacker can add is [-D, D][3].

We use T to represent $t - t'$ while t means the arrive time of a packets and t' means the departure time of the packet. Because the attacker's ability is limited, so T will have a limited value, and we will use δ to represent Max(T).

From the analysis in reference [3], the influence created by attackers perturb the connection timing and insert extra packets has a theoretic limitation. The probability that the overall impact of iid random delays on the average of inter-packet delay is outside the tolerable perturbation range $(-s/2, s/2]$ is bounded. Let $d_{i,k}$ and $d_{j,k}$ be the random variables that denote the random delays added by the attacker to packets $P_{i,k}$ and $P_{j,k}$ respectively for k=1,...,m. Let $x = d_{j,k} - d_{i,k}$ be the random variable that denotes the impact of these random delays on k th inter-packet delay and X be the random variable that denotes the overall impact of random delay on the average of inter-packet delay. Then we have $X = \frac{1}{m}\sum_{k=1}^{m}(d_{j,k} - d_{i,k}) = \frac{1}{m}\sum_{k=1}^{m} X_k$. Similarly we define the probability that the impact of the timing perturbation by the attacker is out of the tolerable perturbation range $(-s/2, s/2]$ as $Pr(|X| < s/2)$. They show the probability can be reduced to be arbitrarily close to 0 by increasing m and s.

3 Packet-Based Window

We assume that the packets in the attacking connection keep their original sequence after through the stepping stones and there are no dropped and reordered packets. We only consider the situation that the attackers do not change the number of packets.

3.1 Method Description

The method for satisfying the increasing characteristic by adjusting the inter-packet delay is responsible for both incremental rule injection and detection. To achieve this, actively perturbation is exerted on the average inter-packet delay sequence of the being-guarded connection chain at ingress, by which certain of incremental characteristic is injected, while still maintain a certain robustness when the attacker perturbs the timing characteristics of the attacking connection traffic.

Supposed that, in the incoming connection chain, the packet's arrival time sequence is denoted as $\{t_1, t_2, t_3, ...\}$and the outgoing $\{t'_1, t'_2, t'_3, ...\}$. When monitor the ingress, for each m+1 received packets, average IPD is computed, and an average IPD array is obtained, denoted as $\{\overline{d_1}, \overline{d_2}, ..., \overline{d_{n-1}}, \overline{d_n}\}$. In this array active perturbation is performed, to each $\overline{d_i}$, we make it satisfy the inequation of $\overline{d_{i+1}} - \overline{d_i} \geq s, (i > 1)$, by which an incremental rule is injected actively. Also it is needed to limit the increase, at where factor P is defined, according to which the active perturbation is reset by every P times to sustain the synchronization between the characteristic-injected traffic and the original non-injected traffic. As figure 1 shown that the active perturbation scheme, with X-axis denotes the index of the array of computed average IPDs of m+1 packet, and Y-axis is the value. After every 4 times of delay injection, one reset is committed, by which synchronization between incoming traffic and outgoing traffic is accomplished.

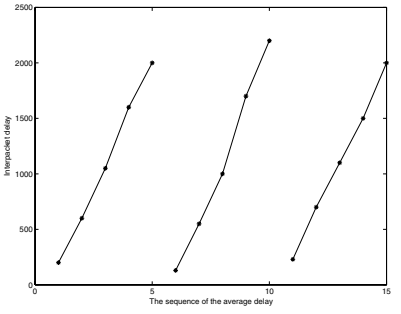


Fig. 1. Ideal model

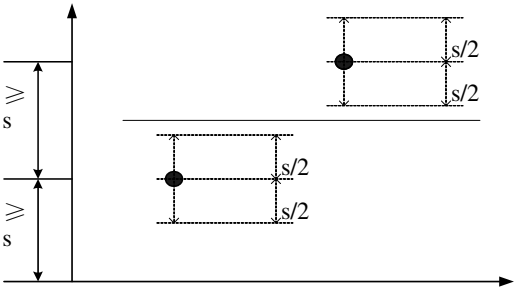


Fig. 2. Attack perturbation and incremental delay

However figure 1 is an ideal mode of timing adjustment. In fact the attacker can also perturb the inter-packet delay. However, if m and s are properly controlled, the perturbation from attacker can be confined in the range of $(-s/2, s/2]$. From figure 2, while take the attacker's countermeasure into account, in the worst situation, while $s/2$ increment is exerted to the preceding average IPD and $s/2$ decrement to the following average IPD by the attacker, the last influence on the sequence is s approximately. For an increment of s is exerted to the sequence, by which sustain the incremental trend, the countermeasure of the attacker can be ignored.

3.2 Adjusting the Inter-packet Delay

As to the first selection of $m + 1$ packets, no active perturbation is introduced, while simply record the IPDs, for example, the corresponding IPDs are $d_{1,1}, d_{1,2}, d_{1,3}, ..., d_{1,m}$. While timing adjustment is performed, all the packets are pushed into a waiting stack, by which a small delay is exerted to the sequence. Where P is defined as a cycle factor, referring to which a reset for adjustment of $m+1$ packets is pursued at the beginning of each cycle. If there is any packet in the waiting stack, send it out as soon as possible, otherwise, keep the transmit characteristic as what it is before.

Also where H is defined as a referential delay factor (with an init of the average IPD of the preceding $m+1$ packets), and g the comparative factor (init as s), f amendment factor(init as 0). In order that every sequence's average IPDs is s bigger than the preceding one, each of the IPD in this sequence must be s bigger than the corresponding one in the preceding sequence. So at here, every departure time is adjusted to make that the delay is bigger by a quantity of g. But in other occasions, the IPD turns to be large enough, and no delay is needed. To decrease the influence on the connection exerted by us, the excess delay is cumulated to the next periods, where f is used to control the amendment factor. The algorithm is described as the following:

1. Set increase count factor p.
2. Let g=s, f=0, i=1. For the first m+1 packets, if there are packets remained in the delay queue, forwarding them as soon as possible; if there is none packet remained in the queue, forwarding the packets according to its original rule. At the same time record the IPDs of the fist $m+1$ packets, denote as $d_{1,1}, d_{1,2}, d_{1,3}, ..., d_{1,m}$.
3. i++; Adjust the IPD of the m packets in the next cycle.

3.1 For the first packet, none adjustment is pursued. When it is not in the delay queue, then simply forwarding the packet according to its original characteristic; if it is in the delay queue, then forward the packet directly.

3.2 initialize factor j with 1, which is utilized to denote the index of the IPDs.

3.2.1 When a packet is received, compute the IPD between this packet and the preceding one, which is denoted as $d_{i,j}$.

3.2.2 compare $d_{i,j}$ with $d_{i-1,j} + g$. a) if $d_{i,j} \geq d_{i-1,j} + g$, then none perturbation is committed, and let $f = (d_{2,1} - (d_{1,1} + g))/q$, where q denote the count of packets that need to be adjusted but not yet(eg. If $m = 20$,and the preceding 5 packets have been dealt, then $q = 20 - 5 = 15$),$g = g - f$; b) else if $d_{i,j} \neq d_{i-1,j} + g$, then $d_{i,j}$ shall be delayed, and the delay time is $d_{i-1,j} + g - d_{i,j}$

3.2.3 j++; if $j \neq m + 1$, then go to 3.2.1

3.3 if $i = p$,then go to 2; else go to 3.

3.3 Detecting the Incremental Delay

When correlation is performed, packets departure time shall be recorded at the egress, and for every m+1 packets, the average IPD shall be computed. Correlation is constructed only when the incremental rule was detected at the fluctuation of the computed average IPDs sequence. Yet while consider the timing synchronization, it is hard to determine from where active perturbation is injected, for that when randomly injected, the incremental characteristic may be evadable in correlation detection. To deal this flaw, a begin point shall be taken as tentative.

1. When receiving the packets, compute the preceding packet's IPD, denoting as $d_1, d_2, ...$
2. Compute the IPD in turn.

2.1 From every m+1 packets, m IPDs can be computed. Let $T_{1,1}$ denotes the average IPD of the packets selection of $\{P_1, P_2, ..., P_m, P_{m+1}\}$, and $T_{1,2}$ of $\{P_2, P_3, ..., P_{m+1}, P_{m+2}\}$, ..., and so on. So $T_{i,j}$ denotes

$$\{P_{m(i-1)+j}, P_{m(i-1)+j+1}, ..., P_{mi+j-1}, P_{mi+j}\}$$

2.2 From the above definition, we get the arithmetic as

$$T_{i,j} = \sum_{j=(m+1)(i-1)+j}^{(m+1)i-2+j} d_j.$$

3. Detect incremental characteristic in $T_{i,j}$ array.

3.1 If incremental characteristic is detected, then the tentative synchronization point is the real synchronization point.

3.2 Perform the correlation detection, if the following IPDs still satisfy the incremental rule, then the connection chain is correlated chain that is being sought for; else go to 3.3

3.3 Forward the sensitive synchronization point to the next position, go to 3.If the tentative synchronization point has been moved for m times, then it turns to be decided that this connection chain is not a correlation connection chain.

3.4 Analysis

This incremental signal injected method adopts a real-time strategy, which can deal with encrypted traffic, even when the attacker pursues some timing perturbation in the traffic. When factors s, m, p are properly set, this method can achieve good performance in practice. And according to our experiments, it proves that when the values of s, m are raised, we will get smaller TPR and FPR. On the other hand, when workload of the arithmetic is taken into count, the value of s must be confined to a certain small range; also m should be set to a comparative small value to avoid mass packet that we have to analysis, which give flaw to its real-time characteristic.

4 Time-Based Window

We assume that the packets in the attacking connection may not keep their original sequence after through the stepping stones and there are dropped and reordered packets. So We will also consider the situation that the attackers change the number of packets.

4.1 Method Description

We will first divide the time into segments with δ and use $w_1 w_2 ... w_n$ to represent each segment. So the packets which arrive in segment w_k will only departure in segments w_k and w_{k+1}. If we only delay the packets in segment w_k and do not delay the packets in segment w_{k+1}, then the average departure time of packets in segments in segments w_k and w_{k+1} will increase will all the others will not change. With this character, we can check our watermark to construct the correlations.

4.2 Injecting the Watermark

The consequence segments will form into groups. So with n segments, n/2 groups will be formed which is (w_1, w_2), (w_3, w_4) ... (w_{n-1}, w_n). And we will use t_i to represent to average arrive time of all the packets in group (w_i, w_{i+1}) and t'_i to represent to departure time of all the packets in group (w_i, w_{i+1}). Stochastic pair of groups will be chosen to be injected in watermark. Take (w_i, w_{i+1}) and (w_j, w_{j+1}) for example. If 0 is to be injected, all the packets in w_i will be delayed. So $\frac{(t'_j - t_j) - (t'_i - t_i)}{j-i}$ will be smaller. If 1 is to be injected, all the packets in w_j will be delayed. So $\frac{(t'_j - t_j) - (t'_i - t_i)}{j-i}$ will be larger.

4.3 Detecting the Watermark

After detecting the normal flow for a while an average of delayed time t_{ave} will be got. Now the answer of $res = \frac{(t'_j - t_j) - (t'_i - t_i)}{j-i}$ will be operated. If $res - t_{ave} > 0$, the corresponding bit of the watermark will be set to 0. If $res - t_{ave} <= 0$, the corresponding bit of the watermark will be set to 1. After constructing the watermark, the hamming distance with the real watermark will be operated. If the answer is smaller than a threshold, it can be said to be correlation.

4.4 Analysis

This time-based window is novel because it can deal with a lot of real conditions with high exactness such as packet loss and Disorder. And $j - i$ will affect the answers, so we should choose it carefully. If we set $j - i$ to a determinate value, it will be easy to operate and get the result and it will be more affected by the attackers. On the other hand, if we choose $j - i$ stochastic, if will be more complex and more precise.

5 Evaluation

We derive test data from over 49 million packet headers of the Bell Labs-1 Traces of NLANR[4]. It contains 121 SSH flows that have at least 600 packets and are 120 seconds long at least. We use these 121 SSH flows for 30 times to evaluate active delay approach.

From lots of experiments, we can prove the true positive of the method introduced in this paper changes while the value of m and s are changed. The positives tested when the number of each group is 20, 15, 10, 5 respectively are shown in figure 3, 4, 5, 6. The horizontal axis coordinate represents the incremental change, and the vertical axis coordinate represents the true positive.

From the four figures above, we can observe that true positive increase while raise the value of s or m. At the points of m=20(15),s=200(150), m=10,s=200, all true positives can reach 100%.

6 Related Works

Existing tracing approaches for a connection chain can be divided into two categories[5] based on tracing object: host-based [1,6,7] and network-based [2,5,8,9,10], each of which has characteristic on tracing area, performance overhead, tracing accuracy respectively.

1. Host-based: DIDS[5] developed at UC Davis is a host-based tracing mechanism that each monitored host in the DIDS domain collects audit trails and sends audit abstracts to a centralized DIDS director for analysis. The Caller Identification System[6] attempts to maintain the integrity of login chain by reviewing information from hosts along the login chain. Caller ID[5] is controversial in that it actually utilizes the same break-in technique used by intruders

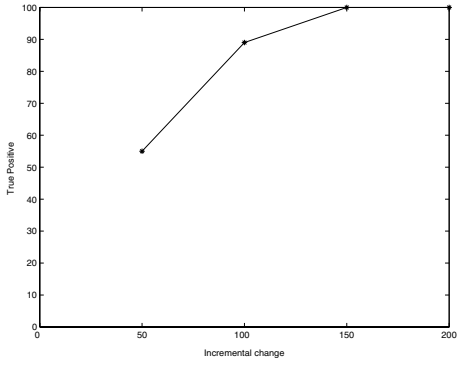


Fig. 3. True positive for m=20

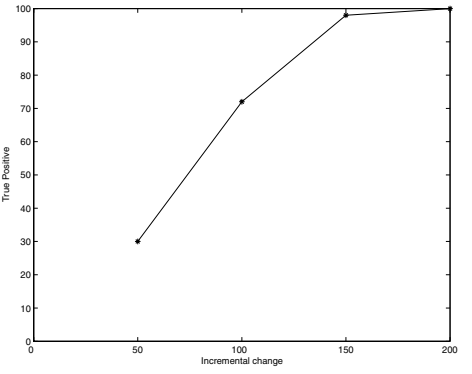


Fig. 4. True positive for m=15

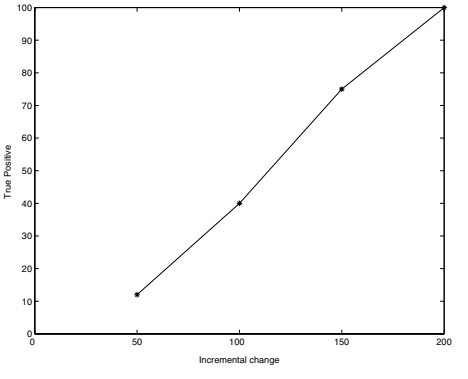


Fig. 5. True positive for m=10

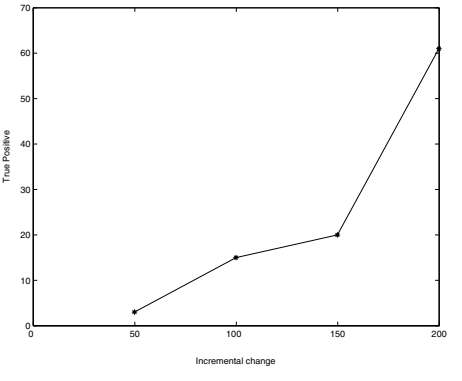


Fig. 6. True positive for m=5

to break into the hosts along the connection chain reversibly. Yung et al[7] proposes a new strategy for detecting suspicious remote sessions, used as part of a long connection chain. Interactive terminal sessions behave differently on long chains than on direct connections. The host-based approaches are restricted the ability of hosts processing because they utilize hosts as information collect point. Too many authentications and communications between the hosts result in more processing time.

2. Network-based: The thumbprint[8] is a pioneering correlation technique that utilizes a small quantity of information to summarize connections. The timing-based scheme[2] by Zhang and Paxson is a novel network-based correlation scheme for detecting stepping stones across the connection chain. The correlation is based on the distinctive timing characteristics of interactive traffic, rather than connection contents. The deviation-based approach[9] by Yoda and Etoh defines the minimum average delay gap between the packet streams of two TCP connections as deviation. The deviation considers both timing characteris-

tics and the TCP sequence number. Wang et al[10] propose a novel correlation scheme based on inter-packet timing characteristics of both encrypted and unencrypted connections that only uses packets in slide windows. One fundamental problem with passive network-based approaches is its computational complexity. Because it passively monitors and compares network traffic, it needs to record all the concurrent incoming and outgoing connections even when there is no intrusion to trace. The irrelevant traffic wastes much computation time and needs long time to collect.

Active approaches differ from passive approaches that they can perturb connection actively and analyze correlations to reduce tracing time and overhead. AN-IDR project[11] proposes to append connection guard to connection content by each active node. Wang et al[12] use active sleepy watermark correlation technology, it injects a watermark into the backward connection of the intrusion, and wake up and collaborate with intermediate routers along the intrusion path. Because it modifies the content of packets, this approach does not adapt for encrypting connection and needs the whole networks to operate cooperatively.

FootFall project[12] proposes a novel watermark-based correlation scheme that the watermark is introduced by slightly adjusting the timing of selected packets of the flow. And the parameters of the watermarking or active delay are not known by the attacker to prevent attacker using perturbation specifically. Pai Peng et al[13]'s algorithms reply on the assumption that packets should not be lost or combined together after passing through a stepping stone. However, packet loss or re-packetization are common when packets arrive too closely or system load is high. In this case, their scheme may not always return the desired result.

7 Conclusions

In this paper, we propose two methods to construct correlations of perturbed connections under packets loss and disorder. Within the attacker's perturbation range, the first method analyzes the activity degree of the correlation windows and monitors increasing characteristic of inter-packets delay. The stepping stone connection in each detecting window can have an increasing average value of the inter-packets delay through changing a part of the packets' arrival delay at the network's ingress. The method can construct correlations in attacking connection chains through detecting these increases at the network egress. The method uses actively perturbed correlation algorithm based on passively monitoring the network egress, it can reduce the complexity of correlation computations and improve the efficiency of detecting stepping stones when the attackers use the encrypting connection and timing perturbation. And the second approach uses the principle of statistics and divides the time into windows and makes them into groups. It then uses pairs of group to take the watermarks. Because the packets are limited in a determinate time segment, it can deal with packet loss and disorder.

References

1. S. C. Lee and C. Shields, "Tracing the Source of Network Attack: A Technical, Legal and Societal Problem", Proceedings of the 2001 IEEE Workshop on Information Assurance and Security, June 2001.
2. Y. Zhang and V. Paxson, "Detecting Stepping Stones", Proceedings of 9th USENIX Security Symposium, August 2000.
3. D. Donoho, A.G. Flesia, U. Shanka, V. Paxson, J. Coit and S. Staniford. "Multiscale Stepping Stone Detection: Detecting Pairs of Jittered Interactive Streams by Exploiting Maximum Tolerable Delay". In Proceedings of the 5th International Symposium on Recent Advances in Intrusion Detection RAID 2002, October, 2002. Springer Verlag Lecture Notes in Computer Science, 2516.
4. NLANR Trace Archive. http://pma.nlanr.net/Traces/long/.
5. X. Wang, D. Reeves, S. F. Wu, and J. Yuill, "Sleepy Watermark Tracing: An Active Network-Based Intrusion Response Framework", Proceedings of IFIP Conference. on Security, Mar. 2001.
6. H. Jung, et al. "Caller Identification System in the Internet Environment", In Proceedings of 4th USENIX Security Symposium, 1993.
7. Kwong, H. Yung. " Detecting Long Connection Chains of Interactive Terminal Sessions". In Proceedings of the RAID 2002 Conference. October 16-18, 2002.
8. S. Staniford-Chen, L. T. Heberlein. "Holding Intruders Accountable on the Internet", In Proceedings of IEEE Symposium on Security and Privacy, 1995.
9. K. Yoda and H. Etoh, "Finding a Connection Chain for Tracing Intruders", In F. Guppens, Y. Deswarte, D. Gollamann, and M. Waidner, editors, 6th European Symposisum on Research in Computer Security - ESORICS 2000 LNCS -1985, Toulouse, France, Oct 2000.
10. X.Wang, D.Reeves, and S.F Wu, "Inter-Packet Delay Based Correlation for Tracing Encrypted Connections Through Stepping Stones", Proc. of European Symposium on Research in Computer Security ESORICS 2002.
11. Active Network Intrusion Detection and Response project, http://www.pgp.com/research/nailabs/adaptive-network/active-networks.asp, 2001.
12. X. Wang and D. S. Reeves. "Robust Correlation of Encrypted Attack Traffic Through Stepping Stones by Manipulation of Interpacket Delays". Proc. of ACM Conference on Computer and Communications Security CCS 2003, October 2003.
13. P. Peng, P. Ning, D. S. Reeves and X. Y. Wang. "Active Timing-Based Correlation of Perturbed Traffic Flows with Chaff Packets". To appear in Proceedings of the The 2nd International Workshop on Security in Distributed Computing Systems (SDCS-2005), January, 2005.

An Enhanced Packet Scheduling Algorithm for QoS Support in IEEE 802.16 Wireless Network[1]

Yanlei Shang and Shiduan Cheng

The State Key Lab of Networking and Switching, Beijing University of Posts and Telecommunications 100876 Beijing, P.R. China
shangyl@bupt.edu.cn

Abstract. In this paper, a novel hierarchical packet scheduling model for IEEE 802.16 uplink is proposed based on J. Bennett & H. Zhang scheduling model [1]. The soft-QoS traffics introduced in this new model together with hard-QoS and best-effort traffics are scheduled by the Base Station (BS). The model can distribute bandwidth reasonably between the QoS and the best-effort traffics. It also guarantees the validness and fairness among the QoS traffics. We give the analytical delay comparison between two models and evaluate the performance by simulations.

1 Introduction

The IEEE 802.16 broadband wireless access standard developed by the IEEE 802.16 working group [2] was recently approved. IEEE 802.16 media access control, which is based on the concepts of connections and service flows, specifies QoS signaling mechanisms (per connection or per station) such as bandwidth requests and bandwidth allocation. However, IEEE 802.16 standard left the QoS based packet scheduling algorithms undefined [3].

J. Bennett and H. Zhang proposed a nice H_WF2Q+ scheduling framework [1](Fig. 1), which distributes weighted bandwidth to different sets of flows grouped according to some criteria. However, the model can not serve the multimedia traffics well because of not taking into account the diverse requirements of multimedia traffics, not addressing the problem of dynamic flow set and not insulating the similar traffics.

In this paper, we propose a hierarchical packet scheduling algorithm that provides QoS support for a wide range of real time applications as defined in IEEE 802.16 based on Bennett_Zhang model. The QoS traffics are divided into hard-QoS and soft-QoS traffics. The new model focuses on distributing network resources to QoS and best-effort traffics according to the available bandwidth efficiently and effectively so as to achieve QoS and fairness in a dynamic network. The proposed solution is practical and compatible with the IEEE 802.16 QoS signaling mechanisms. The simulation results we obtained show that the proposed solution can support diverse traffic classes of traffic with different QoS requirements in terms of bandwidth and maximum delay.

[1] This work is supported by the NFSC (No.90204003, 60472067, 60402012), the National 973 project (No.2003CB314806), the Fund of DPHE (No.20010013003), RFDR (20010013003) and EYTP.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 652 – 661, 2005.

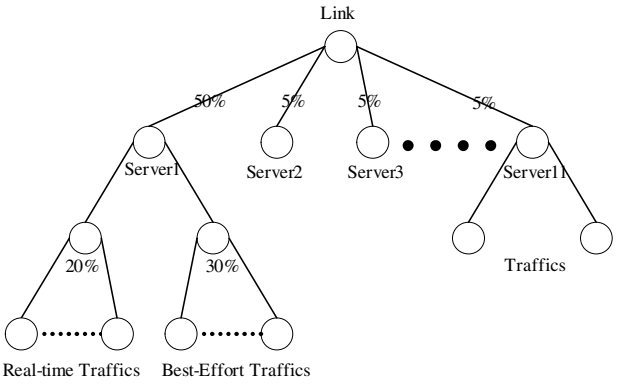


Fig. 1. The Bennett_Zhang Scheduling Model

The remainder of this paper is organized as follows. In Section 2, we briefly introduce the IEEE 802.16 broadband wireless access systems and the existing IEEE 802.16 QoS architecture as well as our proposed QoS architecture. The terminology used in this novel packet scheduling algorithm is provided in Section 3. In Section 4, we describe in details the proposed uplink packet scheduling (UPS) algorithm. We evaluate the performance by simulation methodology in Section 5. Section 6 concludes the paper by summarizing results and outlining the future works.

2 IEEE 802.16 and Its QoS Architecture

IEEE 802.16 architecture consists of two kinds of fixed (non-mobile) stations: subscriber stations (SS) and base station (BS). The BS regulates all the communication in the network, i.e. we would not consider the mesh scenario that peer-to-peer communicate directly between the SSs defined in the IEEE 802.16a. The communication path between SS and BS has two directions: uplink (from SS to BS) and downlink (from BS to SS) [2].

IEEE 802.16 can support multiple communication services (data, voice, video) with different QoS requirements. The media access control (MAC) layer defines QoS signaling mechanisms and functions that can control BS and SS data transmissions. On the downlink (from BS to SS), the transmission is relatively simple because the BS is the only one that transmits the downlink subframe. The data packets are broadcasted to all SSs and an SS only accepts the packets destined to it.

IEEE 802.16 defines four types of service flows, each with different QoS requirements and corresponding uplink scheduler policy:

- Unsolicited grant service (UGS): this service supports constant bit-rate (CBR) or CBR-like flows such as Voice over IP. These applications require constant bandwidth allocation.
- Real-time polling service (rtPS): this service is for real-time VBR-like flows such as MPEG video. These applications have specific bandwidth requirements as well as a deadline (maximum delay).

- Non-real-time polling service (nrtPS): this service is for non-real-time flows which require better than best effort service, e.g. bandwidth intensive file transfer. These applications are time-insensitive and require minimum bandwidth allocation.
- Best effort service (BE): this service is for best effort traffic such as HTTP. There is no QoS guarantee. The applications in this service flow receive the available bandwidth after the bandwidth is allocated to the previous three service flows.

3 Terminology in the New Scheduling Algorithm

A connection in the Internet could be denoted by f. $F(l)$ are a group of traffics with the similar QoS requirements in the link l. $C(l)$ is the bandwidth capacity of link l. Each connection f can be characterized described by the bandwidth requirement sets $< B_{\min}(f), B_{\max}(f) >$. Where $B_{\min}(f)$ is the minimum QoS transmission bandwidth which the traffic f can get and $B_{\max}(f)$ is the highest bit rate or the highest costs the user would like to pay for.

We call it Best effort traffic when $B_{\min}(f) = 0$, such as FTP, E-mail and WWW, and call it QoS traffic if $B_{\min}(f) \neq 0$. In QoS traffics, we say f is soft-QoS traffic while $B_{\min}(f) < B_{\max}(f)$. The router must guarantee the $B_{\min}(f)$ for the soft-QoS traffics. Actually, the real data rate of soft-QoS traffics can range dynamically from $B_{\min}(f)$ to $B_{\max}(f)$. The rtPS and nrtPS traffics in IEEE 802.16 is soft-QoS traffics. We call it the hard-QoS traffic if $B_{\min}(f) = B_{\max}(f)$, such as UGS of IEEE 802.16 whose bit rate is a constant.

$F_{best}(l)$ is the set of the best effort traffics of link l and $F_{QoS}(l)$ the set of QoS traffics. There is the relationship $F_{best}(l) \bigcup F_{QoS}(l) = F(l)$. $F_{best}(l) \bigcap F_{QoS}(l)$ is the soft-QoS traffics set of link l. $C_{QoS}(l)$ and $C_{best}(l)$ are the bandwidth for QoS and best-effort traffics available in link l respectively. Then we get the following:

$$C_{QoS}(l) \geq \sum_{f \in F_{QoS}(l)} B_{\min}(f) \quad C_{QOS}(l) = C(l) - C_{best}(l) \tag{1}$$

In this hierarchical scheduling model, the packet scheduling is performed in the base station uplink l. In the first level, the link capacity is assigned to three logical scheduling servers, i.e., hard-QoS server, soft-QoS server and best effort server. It is demonstrated in Figure 2. The capacity of hard-QoS server is:

$$C_{hard\_QoS}(l) = \sum_{f \in F_{hard\_QoS}(l)} B_{\min}(f) \tag{2}$$

The capacity of the soft-QoS server is:

$$C_{soft\_QoS}(l) = \sum_{f \in F_{soft\_QoS}(l)} B_{\min}(f) \tag{3}$$

It can offer the minimum available bandwidth $B_{\min}(f)$. In the same time, the soft-QoS traffics can also be scheduled by the best-effort server so as to obtain the additional bandwidth. The best-effort server assigns bandwidth for the soft-QoS traffics according to its dynamic capacity $C_{best}(l)$:

$$C_{best}(l) = C(l) - \{C_{hard\_QoS}(l) + C_{soft\_QoS}(l)\} \tag{4}$$

We set a constant $a(<1)$ to limit the QoS server maximum available bandwidth share so that the best-effort server may obtain the reasonable bandwidth, i.e. $C_{QoS}(l) \le a \times C(l)$.

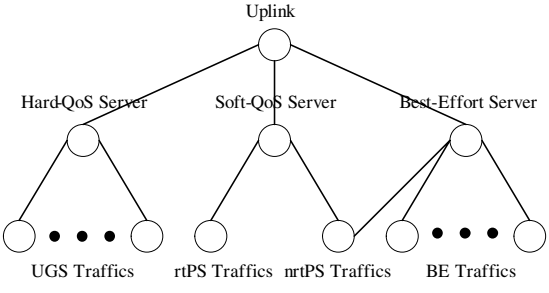


Fig. 2. The Hierarchical Packet Scheduling Model of the Uplink in IEEE 802.16

4 The Novel Packet Scheduling Algorithm

The algorithm comprises the following 4 parts:

- hard-QoS server scheduling;
- soft-QoS server scheduling;
- best-effort server scheduling;
- co-scheduling among the above three servers.

All four servers implement WF^2Q+ [4][5] in their buffer queues. The packet u in $F_{hard\_QoS}(l)$ traffics is scheduled by the hard-QoS server, packet v in $F_{soft\_QoS}(l)$ traffics scheduled by the soft-QoS server and packet w in $F_{best}(l)$ and $F_{soft\_QoS}(l)$ traffics scheduled by the best-effort server. The general server will schedule one from u, v and w. All packets are scheduled according to the Virtual Start Time computed by WF^2Q+.

4.1 The Scheduling of the Hard-QoS Traffics

We set up a queue for $f \in F_{hard\_QoS}(l)$. The newly arrived packet appends to the queue tail. We get the time stamp of the i_th packet by:

$$t^{i}_{hard\_QoS}(f) = \max\{V_{hard\_QoS}, t^{i-1}_{hard\_QoS}(f)\} + \frac{p_i(f)}{B_{\min}(f)} \tag{5}$$

$p_i(f)$ is the size of the i_th packet in f. $B_{\min}(f)$ is the weight of the queue. $t^{i-1}_{hard\_QOS}(f)$ is the time stamp of the $(i-1)\_th$ packet. V_{hard_QOS} is the reference virtual time which is held by the hard-QoS server. It can be used to decide the Virtual Start Time of a newly activated queue. All queues in $F_{hard\_QOS}(l)$ use the same V_{hard_QOS}. $t^{i}_{hard\_QOS}(f)$ is the expected Finish Time of transferring the i_th packet. Once the hard-QoS server is in leisure, the packet with minimum $t^{i}_{hard\_QOS}(f)$ in the non-empty queues will be scheduled.

4.2 The Scheduling of the Best-Effort Traffics

We compute the time stamp of the i_th packet in the best-effort traffic f as:

$$t^{i}_{best}(f) = \max\{V_{best}, t^{i-1}_{best}(f)\} + p_i(f) \tag{6}$$

V_{best} which has the similar meaning with V_{hard_QOS} is used to decide the Start Time for a newly activated queue. All traffics in $F_{best}(l)$ use the same V_{best}. Once the best-effort server is in leisure, the packet with minimum $t^{i}_{best}(f)$ in the non-empty queues will be chosen and transferred. All best-effort traffics in the queue will be assigned the same bandwidth because they have equal weight. We should note that some $F_{soft\_QOS}(l)$ traffics are scheduled in this queue.

4.3 The Scheduling of the Soft-QoS Traffics

The soft-QoS traffics scheduling is more complex than the above two. The soft-QoS traffics is scheduled together by soft-QoS server and best-effort server. We also set up a queue for $f \in F_{soft\_QOS}(f)$. The newly arrived packet appends to the queue. For every packet, we have to compute two time stamps: $t^{i}_{soft\_QOS}(f)$ is used for the soft-QoS server scheduling and $t^{i}_{best}(f)$ for best-effort server. The weight of the packet scheduled by soft-QoS server is $B_{\min}(f)$. The weight of the packet scheduled by best-effort server is 1. The time stamp can be computed respectively.

$$t^{i}_{soft\_QoS}(f) = \max\{V_{soft\_QoS}, t^{i-1}_{soft\_QoS}(f)\} + \frac{p_i(f)}{B_{\min}(f)} \tag{7}$$

$$t^{i}_{best}(f) = \max\{V_{best}, t^{i-1}_{best}(f)\} + P_i(f) \tag{8}$$

If the $(i-1)\_th$ packet is scheduled by soft-QoS server, $t^{i}_{soft\_QOS}(f)$ increases and $t^{i}_{best}(f)$ keeps unchanged. Thus the scheduling priority of the best-effort server will not be affected by the change of soft-QoS server. If it is the best-effort server that schedules the $(i-1)\_th$ packet, $t^{i}_{best}(f)$ increases and $t^{i}_{soft\_QOS}(f)$ keeps unchanged.

So the available bandwidth for soft-QoS traffics is $B_{\min}(f) + B_{best}(f)$. $B_{best}(f)$ is the bandwidth assigned by best-effort server.

4.4 The Co-scheduling of Three Traffics

The three servers discussed above are logical servers in the same physical link. When they have packets to be sent simultaneously, the general server choose only one packet from u, v and w. All three servers are used as logical queues in general scheduling. Their capacities are regarded as the corresponding weights. The co-scheduling among the three servers in the link l implements WF^2Q+ too. [6]

- The weight of hard-QoS server:

$$W_{hard\_QoS} = C_{hard\_QoS}(l) = \sum_{f \in F_{hard\_QOS(l)}} B_{\min}(f) \tag{9}$$

- The weight of soft-QoS server:

$$W_{soft\_QoS} = C_{soft\_QoS}(l) = \sum_{f \in F_{soft\_QOS(l)}} B_{\min}(f) \tag{10}$$

- The weight of best-effort server:

$$W_{best\_effort} = C_{best\_effort}(l) = C(l) - \sum_{f \in F(l)} B_{\min}(f) = C(l) - \{C_{hard\_QoS}(l) + C_{soft\_QoS}(l)\} \tag{11}$$

All of the weights vary with F_{hard_QOS} and F_{soft_QOS}.

The steps of the co-scheduling are as following:

Firstly, the time stamp of the i_th packet chosen by the hard-QoS server is:

$$T^i_{hard\_QOS} = \max\{V_{link}, T^{i-1}_{hard\_QoS}\} + \frac{p_i}{W_{hard\_QoS}} \tag{12}$$

p_i is the size of the packet. $T^{i-1}_{hard\_QOS}$ is the time stamp assigned for the $(i-1)th$ packet by the hard-QoS server. V_{link} is the time stamp of the last packet sent by the physical link.

The time stamp of the i_th packet scheduled by the soft-QoS server is:

$$T^i_{soft\_QOS} = \max\{V_{link}, T^{i-1}_{soft\_QoS}\} + \frac{p_i}{W_{soft\_QoS}} \tag{13}$$

The time stamp of the i_th packet selected by the best-effort server is:

$$T^i_{best} = \max\{V_{link}, T^{i-1}_{best}\} + \frac{p_i}{W_{best}} \tag{14}$$

Secondly, the packet with the minimum time stamp will be scheduled. Then the available bandwidth of the hard-QoS server is:

$$\frac{W_{hard\_QOS}}{W_{hard\_QOS} + W_{soft\_QOS} + W_{best}} \times C(l) = \sum_{f \in F_{hard\_QoS(l)}} B_{\min}(f) \tag{15}$$

The available bandwidth of the soft-QoS server is:

$$\frac{W_{soft\_QOS}}{W_{hardt\_QOS}+W_{soft\_QOS}+W_{best}}\times C(l)=\sum_{f\in F_{soft\_QoS(l)}}B_{\min}(f) \tag{16}$$

The available bandwidth of the best-effort server is:

$$\frac{W_{best}}{W_{hardt\_QOS}+W_{soft\_QOS}+W_{best}}\times C(l)=C(l)-\sum_{f\in F(l)}B_{\min}(f) \tag{17}$$

4.5 The Delay Comparison Between Two Models

In the same environment, i.e., all types of traffics in two models have the same bandwidth requirements, arrival rate and implement WF^2Q+. We study the delay of both models. The technology to analyze the delay properties is borrowed from J. Bennett and H. Zhang [1].

- The delay of hard-QoS traffic:

Two models have the same scheduling algorithm for the hard-QoS traffics. $D_k^{'i}$ and D_k^i are the delay of the i_th packet in the k_th hard-QoS session in Bennett and Zhang model and the hierarchical model respectively: [1]

$$D_k^{'i}-D_k^i=0 \tag{18}$$

- The delay of soft-QoS traffic:

$$(D_k^{'i}-D_k^i)\propto\frac{1}{(N_{soft\_QOS}-N_{best\_effort})\rho_k} \tag{19}$$

$D_k^{'i}$ in Equation 18 is the delay of the i_th packet in the k_th soft-QoS session that is calculated by Bennett and Zhang model. D_i^k is the corresponding delay calculated by hierarchical model. N_{soft_QoS} and N_{best_effort} are the session numbers in soft_QoS and best-effort server respectively.

The delay performance of soft_QoS traffics in this hierarchical model is better than Bennett and Zhang model even in heavy network load.

- The delay of best-effort traffic:

$$(D_k^{'i}-D_k^i)\propto\frac{N_{soft\_QOS}}{N_{best\_effort}(N_{soft\_QOS}-N_{best\_effort})\rho_k} \tag{20}$$

The delay of best-effort traffics increases with the number of soft_QoS sessions. But the increase is not as linear as in Bennett and Zhang model.

5 Simulations and Evaluation

Let us now analyze the delay performance of the proposed packet scheduling model. For this purpose, we use the topology shown in Figure 3, which consists of 10 subscriber stations (SS) indexed from 0 to 9. Station 0 and 1 generate the UGS traffic with the constant rate. Station 2 and 3 generate the rtPS and nrtPS traffic respectively. Station 4 to 9 generate the BE traffic. All these SSes send packets to the core network host which is connected with the BS by the wire line.

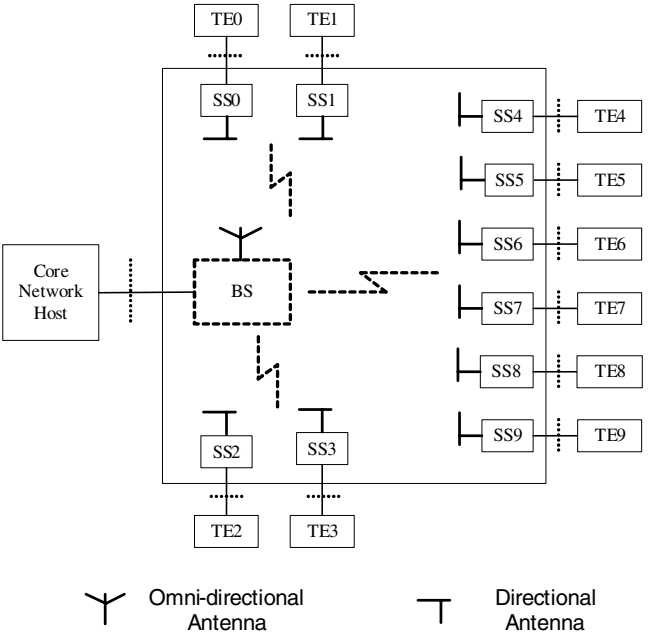


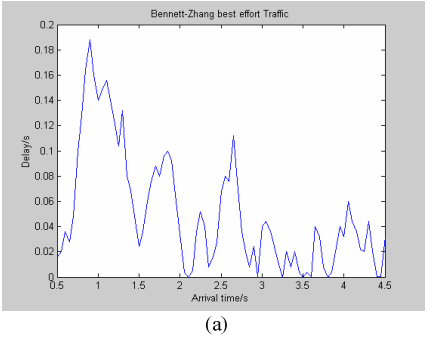
Fig. 3. The Simulation Topology

These traffic flows belonging to the three classes of service: Audio (hard_QoS), Video (soft_QoS), and Background Traffic (Best Effort). We use On-Off exponential distribution sources to simulate BT, video, and audio traffics. The link bandwidth from BS scheduler to server in core network is 20 Mbps.

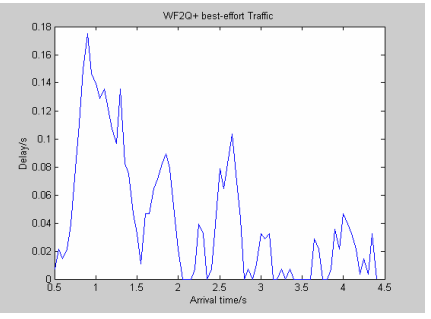
Table 1. Simulation Parameters

Parameters	hard_QoS	soft_QoS	Best Effort
Packet Size(*bytes*)	160	1280	200
Packet Interval(*ms*)	20	10	12.5
Sending rate(Kbit/s)	64	1024	128

The simulation results (Fig. 4, 5 and 6) give the relations between the Delay (Y-axis) and packet Arrival_time (X-axis). In every figure, curves (a) and (b) are simulated in Bennett and Zhang model and the hierarchical model respectively. The results prove that the hierarchical model can guarantee lower delay and delay jitter for traffics of variable bit rate (soft-QoS traffics) than Bennett_Zhang model. In the mean time, the performance for constant bit rate (hard-QoS traffics) and best-effort traffics is equal to or better than that in Bennett-Zhang model.

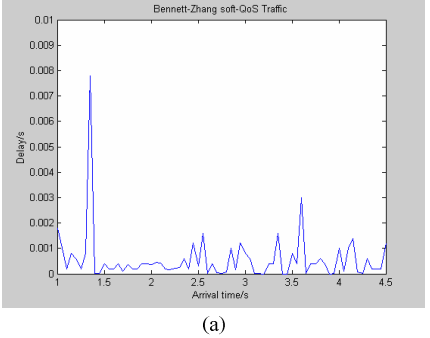


(a)

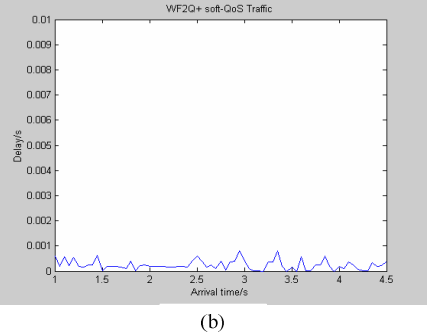


(b)

Fig. 4. The Delay as the Function of Arrival Time of Best Effort Traffic. (a) shows the result of the Bennett_Zhang scheduling model. (b) demonstrates the scenario of proposed packet scheduling model

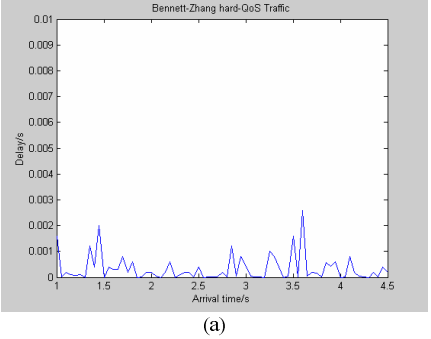


(a)

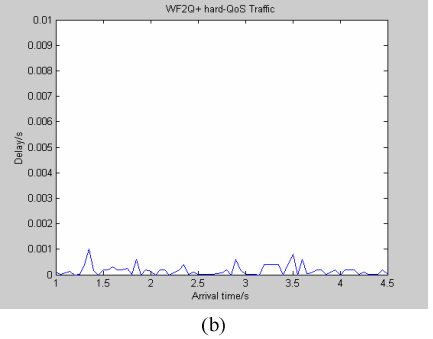


(b)

Fig. 5. The Delay as the Function of Arrival Time of soft-QoS Traffic. (a) shows the result of the Bennett_Zhang scheduling model. (b) demonstrates the scenario of proposed packet scheduling model



(a)



(b)

Fig. 6. The Delay as the Function of Arrival Time of hard-QoS Traffic. (a) shows the result of the Bennett_Zhang scheduling model. (b) demonstrates the scenario of the proposed packet scheduling model

6 Conclusions

In this paper we present a QoS-support scheduling algorithm based on Bennett_Zhang model for IEEE 802.16 wireless network. The proposed solution is practical and compatible to the standard IEEE 802.16. As discussed in this paper, the main difference between two models is the treatment for soft-QoS traffics. The novel model changes the tree-like structure to a two-level hierarchical structure. We demonstrate, both analytically and empirically, the delay performance improvement in the new hierarchical model. Firstly, the soft-QoS traffics defined in the model can get bandwidth as large as $B_{\min}(f) + B_{best}(f)$ according to the network load. So it is suitable for real time traffics with bursts such as the video. Secondly, every server will change their weights for different traffics to adapt to the network dynamics. Thirdly, because it takes the min-bandwidth of QoS traffic as the weight, it can assign the link resources according to the real need. It not only guarantees QoS but also saves the network resources. Lastly, we can offer the reasonable bandwidth for best-effort traffics by setting the available maximum bandwidth for the QoS server. This proposed packet scheduling algorithm is more flexible and lower complexity. The simulation studies show that the proposed solution provides QoS support in terms of bandwidth and delay bounds for all types of traffic classes as defined the IEEE 802.16 standard.

References

1. J. Bennett, H. Zhang: Hierarchical packet fair queueing algorithms. ACM SIGCOMM, (1996)
2. IEEE 802.16 Standard: Local and Metropolitan Area Networks, Part 16
3. Kitti W., Aura G.: Packet scheduling for QoS support in IEEE 802.16 broadband wireless access systems, International Journal of Communication Systems, (2003)
4. L.X. Zhang: Virtual clock:a new traffic control algorithm for packet switching networks, Proceedings of ACM SIGCOMM'90, (1990), 19-29
5. J. Bennett, H. Zhang: WF2Q: worst case fair weighted queuing, IEEE INFOCOM, San Francisco, (1996)

A Novel Core Stateless Virtual Clock Scheduling Algorithm*

Wenyu Gao, Jianxin Wang, and Songqiao Chen

School of Information Science and Engineering, Central South University,
410083, Changsha, Hunan, P.R. China
{gwyy@163.com, jxwang, csq}@mail.csu.edu.cn

Abstract. In this paper, a core-stateless virtual clock-based scheduling algorithm is developed, which combines the simplicity and high performance of FCFS and the fair resource allocation of Fair Queue. The basic idea of this algorithm is using virtual clock to calculate the expected depart time of every packet, and construct a packet queue according to the expected depart time. The algorithm uses only one queue to approximate the multi queue in fair queue algorithm. Because of using the only one packet queue, it is unnecessary for routers to perform per flow state management and the algorithm has good performance in scalability.

1 Introduction

Packet scheduling algorithm is an important part to provide QoS control in the network. The FCFS scheduling algorithm, which is used widely, cannot support QoS because it cannot allocate bandwidth among different flows. The per-flow packet scheduling algorithm such as FQ[1], DRR[2], etc. can realize bandwidth allocation effectively, but their per-flow state management brings serious problem of scalability.

VirtualClock algorithm was proposed in [3], which controls average transmission rate of statistical data flows, enforces each user's average resource usage according to the specified throughput and provides firewall protection among individual flows.

The basic idea of VirtualClock algorithm was borrowed from Time Division Multiplexing (TDM) systems. To make a statistical data flow resemble a TDM channel, imagining that arriving packets from the flow were spaced out by a constant interval in virtual time, so that each packet arrival indicated that one slot time period has passed. So each data flow could be assigned a VirtualClock, which ticks at every packet arrival from that flow. If the tick step was set to the mean inter-packet gap (assuming a constant packet size for the moment), the value of the VirtualClock denoted the expected arrival time of the arrived packet. To imitate the transmission ordering of a TDM system, each switch node stamped packets of each flow with the flow's VirtualClock time and ordered packet transmissions according to the stamp

* This work is supported by the Major Research Plan of National Natural Science Foundation of China, Grant No.90304010.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 662 – 671, 2005.

values, as if the VirtualClock stamp were the real-time slot number in a TDM system. If a flow transmitted according to its specified average rate, its VirtualClock reading should fluctuate about real time.

VirtualClock algorithm can support the diverse performance requirements of various applications by enforcing the resource usage according to the throughput reservation of each individual flow, while preserving the flexibility of statistical multiplexing of packet-switching networks. And it also provided firewall protection among individual data flows, particularly firewalls between datagram traffic and flows that required performance guarantees.

But in fact, VirtualClock algorithm is a stateful algorithm. Each node must maintain a virtual clock for each flow in VirtualClock algorithm, which will bring problem of scalability.

In [4], DPS (Dynamical Packet State) was proposed to relieve core node from per-flow state management. With DPS, each packet carried in its header some state that is initialized by the ingress router. Core routers process each incoming packet based on the state carried in the header of the packet, updating both its internal state and the state in the packet's header before forwarding it to the next hop. By using DPS to coordinate actions of edge and core routers along the path traversed by a flow, distributed algorithms can be designed to approximate the behavior of a broad class of stateful networks by using networks in which core routers do not maintain per flow state.

Also in [4], a core-stateless version of Jitter Virtual Clock (CJVC) scheduling algorithm was proposed. CJVC provides the same delay guarantee as Jitter Virtual Clock (JVC)[5,6], while maintaining and using per-flow state only at the edges of the network. Since CJVC is non-work-conserving and employs a constant bit-rate (CBR) per-flow shaper at every router, queue lengths observed in a network of such servers are generally smaller than in networks of work-conserving schedulers. This further reduces the computation complexity of implementing such a scheduler. Unfortunately, the non-work conserving nature of the CJVC algorithm limits the extent of statistical multiplexing gains that the framework can benefit from. This is because non-work-conserving algorithms shape the traffic to the maximum of the reserved rate and sending rate for that flow; when a flow sends a burst of packets at a rate greater than its reserved rate, extra packets are held until their eligibility time, even if idle bandwidth is available for transmitting these packets. Such an approach may underutilize available network resources. Hence, stateless algorithms that are work conserving are desirable.

In [7], a framework named VTRS was proposed to realize QoS control. The key idea of VTRS is virtual time stamp, which is like DPS. In VTRS, scheduling algorithm based on virtual time stamp can also realize bandwidth allocation. But in VTRS, each flow has to conform to a strict condition, that is:

$$\hat{a}_1^{j,k+1} - \hat{a}_1^{j,k} \geq \frac{L^{j,k+1}}{r^j}$$

Where $\hat{a}_1^{j,k+1}$ is the arrival time of the k+1 packet of flow j at node 1 (i.e., the ingress node), $L^{j,k+1}$ is the length of the k+1 packet of flow j, and r^j is the allocated rate of flow j. In the above formula, it is given a very stringent constraint that the transmission rate of each flow must be less than r^j at each packet level.

So a simple and effective method to construct a core stateless algorithm that can support bandwidth allocation and some degree of burstiness at the same time is still desirable.

In this paper, we proposed a core-stateless virtual clock scheduling algorithm, which combined the simplicity and high performance of FCFS and the fair resource allocation of FQ. The basic idea of this algorithm is using virtual clock to compute the expected depart time of each packet, and construct only one packet queue sorted by expected depart time. In other words, the algorithm uses only one queue to approximate the multi queues in fair queue algorithm. Also because of the only one queue, which makes it unnecessary for routers to perform per flow state management, thus, scalability is got.

The network architecture we consider is similar to the DiffServ[8] architecture, where per-flow functionality is implemented only at the edges of the network, and core routers do not maintain any per-flow state.

In the following section 2, we give a detail description of our algorithm and its properties. Then in section 3 we discuss the implementation of our algorithm. In section 4 we present simulations of our algorithm to verify the effectiveness and performance of our algorithm. Finally, we conclude the paper in section 5.

2 CSVC Algorithm

Inspired by VirtualClock algorithm, we know that we can use only one queue sorted by packet's stamp to realize bandwidth allocation, instead of the multi-queues in DRR or FQ.

But in VirtualClock algorithm, the calculation of each packet's stamp is done through maintaining a virtual clock for each flow at each node. Thus, each node will have to implement per-flow management, which brings serious problem for scalability. In order to overcome this problem, we borrowed DPS from [4].

So, the basic idea of our algorithm is initializing a "virtual clock" for each flow at the ingress node, and making each packet itself carry the "virtual clock" in its header when the packet traverse the following core nodes. And the core nodes are responsible for updating the "virtual clock" carried by packet besides forwarding packet according to packet's "virtual clock" value. In this way, core nodes wouldn't have to maintain "virtual clock" for each flow, thus a core-stateless virtual clock algorithm is got. Because of this algorithm's core-stateless property, it will be more scalable than VirtualClock algorithm.

The other problem is, how to calculate and update the "virtual clock" of each packet, it must provide rate guarantee and support burstiness in some degree at the same time.

2.1 Description of Core-Stateless Virtual Clock Algorithm

First, we consider the transmission of a flow with constant packet length.

In Fig. 1, suppose a CBR flow i's packet length is $\bar{l}_i$, packets of flow i traverse node 1, 2, … j, the propagation delay of node 1 to node 2 is denoted by $\pi_{1,2}$, so the delay between node $j-1$ and node j is $\pi_{j-1,j}$, the allocated rate of flow i is r_i.

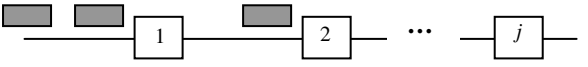


Fig. 1. Transmission of a CBR flow

In this case, suppose the arrive time of the first packet of flow i at node 1 is:

$$a_{i,1}^1 = t_0$$

Then the first packet's depart time at node 1 is:

$$d_{i,1}^1 = t_0 + \frac{\bar{l}_i}{r_i}$$

Then, the first packet's depart time at node j is:

$$d_{i,j}^1 = t_0 + j \times \frac{\bar{l}_i}{r_i} + \pi_{1,2} + \ldots + \pi_{j-1,j}$$

For the same reason, the second packet's depart time at node j is:

$$d_{i,j}^2 = t_0 + (j+1) \times \frac{\bar{l}_i}{r_i} + \pi_{1,2} + \ldots + \pi_{j-1,j}$$

So depart time of the *kth* packet at node j is:

$$\begin{aligned} d_{i,j}^k &= t_0 + (j + (k-1)) \times \frac{\bar{l}_i}{r_i} + \sum_2^j \pi_{m-1,m} \\ &= t_0 + \frac{(j+k-1)}{k \times r_i} \times k\bar{l}_i + \sum_2^j \pi_{m-1,m} \end{aligned} \tag{1}$$

In (1), the item of $k\bar{l}_i$ is the total data send by flow i from t_0, let $L_i^k = k\bar{l}_i$, and then we get:

$$d_{i,j}^k = t_0 + \frac{(j+k-1)}{k \times r_i} \times L_i^k + \sum_2^j \pi_{m-1,m} \tag{2}$$

Now, let's consider the case in real network. The packet length of a flow is variable. When there are several flows that compete for bandwidth, how to allocate bandwidth for different flows is an important problem.

To provide rate guarantee and support burstiness at the same time, we can use $d_{i,j}^{k}$ calculated by equation (2) as the expected depart time at node j of the *kth* packet of flow i. Then we can calculate the expected depart time of each packet of different flows at node j, and forward each packet according to its expected depart time.

Now, let's see how to calculate the value of (2) at core nodes. In (2), t_0, k, r_i, L_i^k are all related to flow i, and these values can be get at the first node. So we can get the four values at the first node (edge node), and insert them into the packet header, the following nodes (core nodes) can read these values from the packet header. The other values j and π can be got at core node. So the value of (2) can be easily calculated by core node.

2.2 The Effectiveness of This Algorithm

1) Ability of Rate Guarantee. Because packets are forwarded according to their expected depart time, and the expected depart time is given by (2):

$$d_{i,j}^{k} = t_0 + \frac{(j+k-1)}{k \times r_i} \times L_i^k + \sum_{2}^{j} \pi_{m-1,m}$$

When the allocated rate of flow i (denoted by r_i) is bigger, the value of expected depart time ($d_{i,j}^{k}$) is smaller, so the flow with a bigger allocated rate will have more data be transmitted.

2) Providing Firewall Protection among Flows. Forwarding packets in the order of their expected depart time assures that each flow will receive the resources that it has reserved. Although an aggressive flow can consume idle resources, it cannot disturb network service to other flows.

If a flow send data at a rate larger than the rate allocated to it, then its packets' expected depart time will get larger and larger, so packets from this flow will be put at the end of service queue or even be discarded.

Now, let's take the opposite case into account. When a flow's rate less than its allocated rate, then its $d_{i,1}^{k}$ will less than $\hat{d}$ ($\hat{d}$ is the real depart time of a packet at the first node, if a flow send data according to its allocated rate, the value of $d_{i,1}^{k}$ will fluctuate around $\hat{d}$), the difference between the two may be considered some sort of "credit" that the flow has built up. If after a slot of time, this flow send data at a rate larger than its allocated rate, than its packets will get a priority until the "credit" reduces to ZERO. In such case, this flow will disturb other flows if its credit is large enough.

So we introduce another item to control the "credit" saved by such flow. We chose an interval T, with T, the credit saved by a flow is effective, but after an interval T, the credit saved by a flow is set to ZERO.

At the first node, after T, a packet's real depart time $\hat{d}$ will be recorded, then $\hat{d}$ and $d_{i,1}^k$ are compared, if $\hat{d} > d_{i,1}^k$, and then in the next cycle, related values are re-initialized to prevent credit saving span T.

With T, we can control flow's burstiness under a given level, but still support burstiness in some degree.

3) Support of Priority. According to (2), our algorithm can provide priority services to a flow simply by letting edge node replace "t_0" by "$t_0 - t$", where t is a chosen value representing the priority. Use of a priority value, however, will not allow priority flows to take unfair advantage of others. If a prioritized flow runs faster than the allocated rate, its "virtual clock" will eventually run ahead of the real time; hence, its packets will lose priority in service.

3 Implementation of CSVC

The implementation of this algorithm requires a network like DiffServ. At the edge node, the packets from different flows will be classified and related flow information is insert into the packet head, and than is forwarded to core node, the core node read the flow information from packet head, then schedule this packet by $d_{i,j}^k$, the core node will not need to implement per-flow state management, thus improve scalability.

According to (2), calculation of $d_{i,j}^k$ requires t_0, k, j, r_i, L_i^k. So we can insert these values into packet head at the edge node, read these values from packet header at core nodes and calculate $d_{i,j}^k$. To reduce the data carried in packet header, we can use following method.

According to (2), we have

$$\begin{aligned} d_{i,j+1}^k &= t_0 + \frac{(j+1+k-1)}{k \times r_i} \times L_i^k + \sum_{2}^{j+1} \pi_{m-1,m} \\ &= t_0 + \frac{(j+k-1)}{kr_i} \times L_i^k + \sum_{2}^{j} \pi_{m-1,m} + \frac{L_i^k}{kr_i} + \pi_{j,j+1} \\ &= d_{i,j}^k + \frac{L_i^k}{kr_i} + \pi_{j,j+1} \end{aligned} \tag{3}$$

So, to a certain packet P^k we insert $d_{i,1}^k$ and L_i^k / kr_i into packet header at the edge node (the first node). When this packet arrives following nodes, $d_{i,j}^k$ is calculated according (3), $\pi_{j,j+1}$ can be pre-stored at core nodes.

Fig. 2 is the pseudo code of this algorithm.

```
on receiving packet P;
if (edge router) {
  i=classify(P);
  if (P is flow i's 1st packet) || ((next cycle T) && ( d̂ > d_{i,j}^k )) {
      t_0 = arrive time of P;
      k = 0;  L_i^k = 0;
  }
  j = 1;  r_i = allocated rate;
  k = k + 1;  L_i^k = L_i^{k-1} + l_i^k;
  use equation (2) to calculate d_{i,1}^k;
  insert d_{i,1}^k, L_i^k / kr_i into P's header;
} else {
  read d_{i,j}^k, L_i^k / kr_i from P's header;
  use equation (3) to calculate d_{i,j+1}^k; // π_{j,j+1} is pre-stored at this nodes
  update packet header with new d_{i,j+1}^k;
}
if (queue is not overflow)
  insert P into queue sorted by d_{i,j}^k;
else
  first drop tail, then insert P into queue;
```

Fig. 2. Pseudo code of CSVC

4 Simulation Results

In this section, we give the simulation result and analysis about CSVC algorithm. To provide some context, we compare the performance of CSVC to that of DRR.

We have examined the behavior of our algorithm under a variety of conditions. We use an assortment of traffic sources and topologies. All simulations were performed in NS2 [9].

Due to space limitations, in this section, we merely highlight a few important points and omit detailed explanations of the dynamics.

4.1 Single Congested Link

We first consider a single congested link shared by three flows, see Fig. 3. There is only one congested link between router R1 and R2, which has a capacity of 3Mbps. We performed three related experiments.

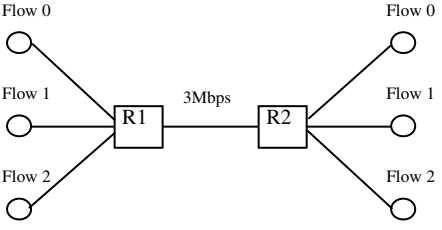
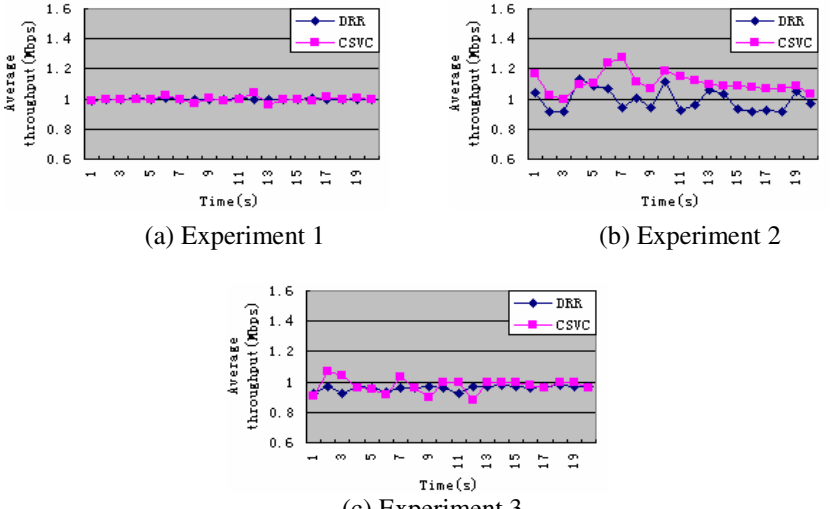


Fig. 3. Network topology for simulation



(a) Experiment 1 (b) Experiment 2

(c) Experiment 3

Fig. 4. Average throughput of flow 0 in single congested link

In the first experiment, we use three UDP flow, denoted by flow 0, flow 1, and flow 2 (left are senders, right are receivers).

Flow 0 is a CBR flow, its rate is 1Mbp; flow 1 is a CBR flow, its rate is 2Mbps; flow 2 is a exponential ON-OFF source, its rate on ON is 3Mbps. Each flow is allocated 1Mbps at the congested link.

Fig. 4(a) shows the average throughput of flow 0 at node R2 in this experiment. From fig. 4(a), we can see that even flow 1 and flow 2 send data at a rate larger than their allocated rate, flow 0, which sends data according to its allocated rate, can get its fair bandwidth. The performance of CSVC likes that of DRR.

In the second experiment, we replace flow 0 with a TCP flow, but flow 1 and flow 2 are the same as them in the first experiment. Each flow is also allocated 1Mbps at the congested link.

Fig. 4(b) shows the result of this experiment. We can also see that flow 0 (TCP flow) can get its fair bandwidth in CSVC while UDP flows violate their allocated bandwidth.

In the third experiment, flow 0, flow 1 and flow 2 are all TCP flow, and each flow is allocated 1Mbps at the congested link.

Fig. 4(c) is the result of this experiment.

From fig. 4(c), we can also see that the performance of CSVC is like that of DRR, CSVC can guarantee each flow's fair bandwidth.

4.2 Multiple Congested Links

So far we have seen the performance of CSVC in a simple network configuration with single congested link. In this section, we study how CSVC performs when there are multiple congested links in the network. A sample network configuration with four routers is constructed as shown in Figure 5. The first link between router R1 and R2 (R1-R2) has a capacity of 5Mbps, the following link R2-R3 has a capacity of 10Mbps. The third link, R3-R4, has a capacity of 5Mbps.

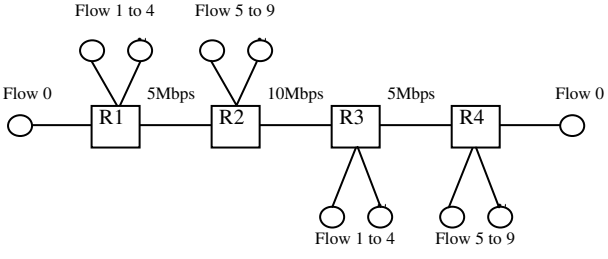


Fig. 5. Topology of multiple links

In the first experiment, flow 0 to flow 9 are all CBR flow (left are senders, right are receivers), flow 0's transmit rate is 1Mbps, flow 1's rate is 1.2Mbps, flow 2's rate is 1.5Mbps, flow 5's rate is 1.5Mbps, flow 6's rate is 2Mbps, the rest flows' rate are 1Mbps, each flow were allocated 1Mbps bandwidth at each congested link.

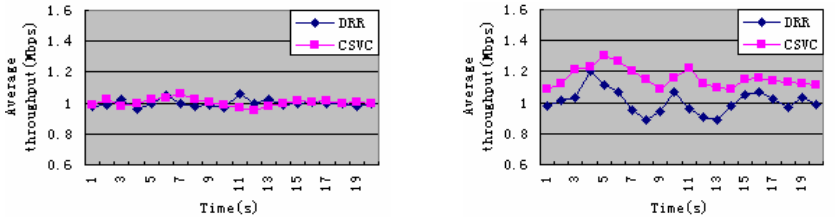
Fig. 6(a) shows the average throughput of flow 0 at node R4 in experiment 1.

In the second experiment, we replace flow 0 with a TCP flow, the rest flows are the same as them in the first experiment. Each flow is also allocated 1Mbps bandwidth at each congested link.

Fig. 6(b) shows the average throughput of flow 0 (TCP flow) at node R4 in experiment 2.

From fig. 6(a) and 6(b), we can see that even flow 1, flow 2, flow 5, and flow 6 send data at a rate larger than their allocated rate, flow 0 can get its fair bandwidth. The performance of CSVC likes that of DRR.

In general, CSVC achieves a reasonable degree of rate guarantee, as well as that of DRR, moreover, its property of core-stateless makes it more scalability than DRR.



(a) Experiment 1 (b) Experiment 2

Fig. 6. Average throughput of flow 0 in multiple congested links

5 Conclusions

In this paper, we proposed a core stateless virtual clock algorithm. This algorithm can implement bandwidth allocation without per-flow state management at core nodes. Simulation also verified this algorithm's effectiveness. This algorithm can be deployed in a network like DiffServ to implement QoS control.

References

1. A. Demers, S. Keshav, S. Shenker. Analysis and simulation of a fair queuing algorithm. In Proceedings of ACM SIGCOMM'89, pages 1-12, Austin, TX, 1989
2. M. Shreedhar, G. Varghese. Efficient fair queuing using deficit round-robin. IEEE/ACM Transactions on Networking. 1996,4(3): 375-385
3. L. Zhang. Virtual Clock: A new traffic control algorithm for packet switching networks, In Proceedings of ACM SIGCOMM'90, page 19-29, Philadelphia, PA, Sept. 1990
4. I. Stoica, and H. Zhang. Providing guaranteed services without per flow management[A]. ACM SIGCOMM'99[C], 1999
5. D. Verma, H. Zhang, D. Ferrari. Guaranteeing delay jitter bounds in packet switching networks. In Proceedings of Tricomm'1991, pages 35-46, Chapel Hill, North Carolina, April 1991
6. H. Zhang, D. Ferrari. Rate-controlled service disciplines. Journal of high speed networks, 3(4):389-412, 1994
7. Z. Zhang, Z. Duan, and Y. Hou. Virtual time reference system: A unifying scheduling framework for scalable support of guaranteed services. IEEE Journal on Selected Areas in Communications, vol. 18, no. 12, pp. 2684-2695, Dec. 2000
8. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated services. IETF, RFC 2475, Dec. 1998
9. Network simulator (NS2-2.27)[EB/OL]. http://www.isi.edu/nsnam/ns, 2004

Proportional Differentiated Services for End-to-End Traffic Control*

Yong Jiang and Jianping Wu

Graduate School at Shenzhen, Tsinghua University,
518055 Shenzhen, P.R. China
jiangy@sz.tsinghua.edu.cn

Abstract. Due to multiple performance objectives of network traffic control, corresponding packet scheduling strategy in next generation broadband service convergence network attract more and more attention. At the same time, the end-to-end Quality of Service (QoS) requirements need to be satisfied and the network resource should be allocated fair and efficiently. In this paper, we provides nPFS, a network proportional fairness scheduling strategy in packet-switched networks from proportional fairness principle [1], and the nPFS integrates several objects, such as the network performance, user's QoS requirement and system fairness. Then it is analyzed and proved in detail. Moreover, the nPFS can be applied to design and improve the scheduling strategy and algorithms in packet-switched networks.

1 Introduction

As the development of network technologies, the forwarding efficiency, bandwidth, delay and loss rate should meet the new requirements brought forward by all kinds of new scheduling strategies in packet-switched networks.

Previously, the research to packet scheduling strategy mostly focuses on one side of the problem, such as the requirement of some performance goal, or the integrated performance research in some specific area. For example, several new traffic models, deterministic [2][3] or stochastic [4][5][6], have been proposed which have made end-to-end network analysis tractable and have yielded some bounds on performance metrics such as delay, throughput, and backlog. In [7], the author compared the influence of throughput and delay jitter to different IP packets, and then put forward an asymmetric best-effort service model providing different throughput and delay jitter to the two kinds of IP packets. The performance differences between the classical constrained optimization and genetic algorithm in throughput, fairness, and time complexity is detailed in [8]. The author brought forward an integrated compromise, but it only focused on the allocation of bandwidth mostly.

In this paper, we proposed a *network* proportional fairness scheduling strategy (nPFS) which integrated network efficiency, user QoS requirement, system fairness, and other multi-target performance requirements effectively.

* This research was sponsored by CNSF (No. 90104002) and GDNSF (No. 034308), and Development Plan of the State Key Fundamental Research (973) (No. 2003CB314805).

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 672 – 681, 2005.

2 Background

2.1 Proportional Fairness Principle

Internet users and applications have diverse service expectations to the networks, making the current *same-service-to-all* model inadequate and limited. In the *relative differentiated services* [9] approach, the network traffic is grouped in a small number of *service classes which are ordered based on their packet forwarding quality*, in terms of per-hop metrics for the queueing delay and packet loss. In [1], we proposed the *proportional fairness principle*. According to this principle, the basic performance measures for packet forwarding locally at each hop are ratioed proportionally to certain *class differentiation parameters* that the network operator chooses, *independent of the class loads*.

Considering queueing delay fairness, we use queueing delay as the proportional fairness principle parameter. Specifically, if $\hat{d}_i$ is the queueing delay of the class-i packets, the proportional fairness principle states that

$$F = \frac{d_i}{\delta_i} = \frac{d_j}{\delta_j} \qquad (i, j = 1...N) \tag{1}$$

for all pairs of classes i and j. The parameters $\{\delta_i\}$ are referred as Delay Fairness Parameters (DFPs), and because higher classes have better service performance, they are ordered as $\delta_1 > \delta_2 > ... > \delta_N > 0$.

2.2 Service Function

The notion of a service function has its roots in the work of Parekh and Gallager[13], who introduced the concept of a universal service function in the context of a specific scheduling algorithm. A key feature of characterizing service for a connection using a service function is that the quality of service guarantees can be expressed as simple functions of the service definitions and the traffic burstiness constraint of the connection. These functions are independent of the service guarantees and burstiness constraints of all other connections. Another key feature of the service curve specification is that it gives greater flexibility to a server in allocating its resources to meet diverse delay and throughput requirements.

3 Proportional Fairness Function

Throughout this paper we assume that time is divided into slots, numbered 0, 1, 2, …. We consider a server which receives (and buffers) packets from M service classes and sends up to c packets per slot. We call c the capacity of the server.

In the following the operation and the traffic flow through the server equipped with buffers are considered. Assume those buffers are partitioned so that each service class has a dedicated space, we focus on a single traffic stream passing through a server.

Let $R_i^{in}[t]$, $R_i^{out}[t]$, $Q_i[t]$ be the total number of packets from the specific service class i arriving at, departing from, stored in and discarded by the server during slot t, where t is a non-negative integer. Without loss of generality, let $R_i^{in}[0]=0$, $R_i^{out}[0]=0$, $Q_i[0]=0$.

Define $R_i^{in}[s,t]$ to be the number of packets arriving in the interval $[s,t]$, i.e. $R_i^{in}[s,t]=\sum_{m=s}^{t} R_i^{in}[m]$. If s >t, define $R_i^{in}[s,t]=0$. Similarly, $R_i^{out}[s,t]$ is defined to be the number of packets leaving the server in the interval $[s,t]$. For simplicity, we will focus our discussion on a given class and omit the subscript i for the rest of the section.

We assume that there are no packets stored in the server at the end of slot zero. Therefore, the number of packets from the class which are stored in the server at the end of slot t, called the backlog for the connection at the end of slot t, is given by

$$Q[t]=R^{in}[1,t]-R^{out}[1,t]\geq 0 \tag{2}$$

The virtual delay, relative to t, suffered by the class is denoted by $d[t]$ and defined to be:

$$d[t]=\min\{\Delta:\Delta\geq 0 \quad and \quad R^{in}[1,t]\leq R^{out}[1,t+\Delta]\} \tag{3}$$

Note that if packets depart the server in the same order they arrive (FIFO), then $d[t]$ is an upper bound of the delay suffered by a packet that arrives in slot t.

We shall consider constraints on the behavior of the network element, as well as constraints on traffic streams. The following definition is useful in this regard.

Definition 1. (Convolution in the min-plus algebra)
Given two functions F and G defined on the non-negative integers, define $F*G(n)$, called the convolution of F and G, as

$$F*G(n)=\min\{F(m)+G(n-m):0\leq m\leq n\},\quad n\geq 0 \tag{4}$$

It is easy to verify that the convolution operator above is associative and commutative. Also note that is F and G are each non-decreasing, then $F*G$ is non-decreasing.

A Service and Arrival Functions

Having defined all the necessary terms, we are now ready to define service functions.

Definition 2. (Burstiness Constraints)
Given a non-decreasing function $b(\cdot)$, called an arrival function, we say that the input traffic R^{in} is b-smooth if $R^{in}*b(n)\geq R^{in}(n)$ for all $n\geq 0$. In the special case where b is affine, i.e. $b(x)=\sigma+\rho x$, we say that R^{in} is (σ,ρ)-smooth.

If $b(0)=0$, the statement that R^{in} conforms to b is equivalent to the equality $R^{in}=R^{in}*b$.

By the delay proportional fairness principle [1], it suffices to show $\frac{d_i}{\delta_i}=\frac{d_j}{\delta_j}=\tilde{d}$, where $\tilde{d}$ is the delay proportional fairness parameter, and its value will be discussed later. Intuitively, if a server guarantees the delay condition $\delta_i\tilde{d}-\Delta_d \le d_i \le \delta_i\tilde{d}+\Delta_d$ for each class i, where Δ_d is the endurance parameter defined by system, then the delay proportional fairness principle (Equation 2) is guaranteed.

Definition 3. (Maximum Delay Proportion Function)

Suppose that the input traffic R^{in} is b-smooth, and the required delay bound for class i is $d_i^{\max}=\delta_i\tilde{d}+\Delta_d$. Let $P^{D}{}_i(\cdot)$ be a non-decreasing function, with

$$P^{D}{}_i(t)=\begin{cases}0 & ,if \quad 0\le t\le d_i^{\max}-1\\ b\left(t-d_i^{\max}\right) & ,if \quad t\ge d_i^{\max}\end{cases} \tag{5}$$

We say that the server guarantees the maximum delay proportion function $P^{D}{}_i(\cdot)$ for the class i if for any t, there holds $R_i^{out}(t)\ge R_i^{in}*P^{D}{}_i(t)$.

Definition 4. (Minimum Delay proportion function)

Suppose that the input traffic R^{in} is b-smooth, and the required delay bound for class i is $d_i^{\min}=\delta_i\tilde{d}-\Delta_d$. Let $\overline{P}^{D}{}_i(\cdot)$ be a non-decreasing function, with

$$\overline{P}^{D}{}_i(t)=\begin{cases}0 & ,if \quad 0\le t\le d_i^{\min}-1\\ b\left(t-d_i^{\min}\right) & ,if \quad t\ge d_i^{\min}\end{cases} \tag{6}$$

We say that the server guarantees the minimum delay proportion function $\overline{P}^{D}{}_i(\cdot)$ for the class i if for any t, there holds $R_i^{out}(t)\le R_i^{in}*\overline{P}^{D}{}_i(t)$.

Intuitively, $P^{D}{}_i(t-s)$ specifies the minimum number of packets from the class that have to depart the server within some specific interval $[s+1,t]$, and $\overline{P}^{D}{}_i(t-s)$ specifies the maximum number of packets from the class that may depart the server within some specific interval $[s+1,t]$, where t is any given slot and s is some slot no later than t, in which the backlog of the class is zero.

Note that $F*\delta_0=F$ for any non-decreasing function F, where we define the function $\delta_d(x)$ as

$$\delta_d(x) = \begin{cases} 0 & , x \le d \\ +\infty & , x > d \end{cases}.$$

Thus, it follows that any server trivially guarantees the minimum delay proportional function $\delta_0(x)$. For convenience, we assume that all the functions are integer valued.

The next Theorem is a simple generalization of a known property of maximum delay proportional functions.

Theorem 1. (Network Servers in Series)

Suppose a stream passes through two servers in series, where the i^{th} server guarantees the maximum delay proportional function $P^D{}_i$ and the minimum delay proportional function $\overline{P}^D{}_i$, i =1,2. Then the entire system guarantees the maximum and minimum delay proportional function $P^D{}_1 * P^D{}_2$ and $\overline{P}^D{}_1 * \overline{P}^D{}_2$, respectively.

Proof: The result follows easily from the associativity of convolution.

B Bounds on Delay and Backlog

Suppose it is known that $R^{in}\left(t-\overline{d}\right) \ge R^{out}(t) \ge R^{in}\left(t-\hat{d}\right)$ for all t, where $\overline{d}$ and $\hat{d}$ are constants. This implies that $d[t] \le \hat{d}$ for all t. Furthermore if there is an arrival at time t, then $R^{in}(t) > R^{in}(t-1) \ge R^{out}\left(t+\overline{d}-1\right)$, which implies that $d[t] \ge \overline{d}$. The following theorem therefore establishes an upper bound on $d[t]$ when there is an arrival at time t. The quantity $\hat{d}-\overline{d}$ is called the *delay jitter*.

Theorem 2 (Delay Jitter Bound)

Consider a server that guarantees the maximum delay proportion function $P^D(\cdot)$ and minimum proportion function $\overline{P}^D(\cdot)$ for the class and suppose that the input traffic R^{in} is b -smooth. Then for every t, there holds

$$R^{in}\left(t-\overline{d}\right) \ge R^{out}(t) \ge R^{in}\left(t-\hat{d}\right) \tag{7}$$

where

$$\hat{d} = \min\left\{\Delta : \Delta \ge 0 \quad and \quad P^D(t) \ge b * \delta_\Delta(t), \quad t \ge 0\right\}$$

and

$$\overline{d} = \max\left\{t : t \ge 0 \quad and \quad \overline{P}^D(t) = 0\right\}$$

Theorem 3. (Upper Bound on Backlog)

Consider a server that guarantees the delay proportion function $P^D(\cdot)$ for the class and suppose that the input traffic R^{in} is b -smooth. Then for every t, the backlog $Q[t]$ is upper bounded by

$$Q[t] \le \max_{s:s\ge 0}\left\{b(s) - P^D(s)\right\} \tag{8}$$

By the proportional fairness principle and definition 2, if a server guarantees the delay proportion function $P^D(\cdot)$ and the input traffic R^{in} is b -smooth for every class, then the server meets delay proportional fairness principle.

We discussed the loss ratio proportional fairness in [11].

C Dampers

A *damper* is a network element which *may* "slow down" a traffic stream passing through it. It may be desirable to pass a packet stream through a damper inside a packet switch, in order to deliver lower delay to other traffic streams. A damper may also provide a traffic shaping function - in fact a regulator is a special case of a damper.

Suppose a_k is the arrival time of the k^{th} packet from the traffic stream incident to a damper, where we assume that $a_k \le a_{k+1}$ for each k. Packet k is assigned an *initial eligibility time* $e_k^{initial}$ and *a terminal eligibility time* $e_k^{ter\min al}$, where $e_k^{initial} \le e_k^{ter\min al}$, and we assume that $e_k^{initial} \le e_{k+1}^{initial}$ and $e_k^{ter\min al} \le e_{k+1}^{ter\min al}$ for each k. If x_k is the departure time of packet k from the damper, then $x_k = a_k$ if $a_k \ge e_k^{ter\min al}$, i.e. packet k departs immediately if it arrives at or after its terminal eligibility time. Otherwise the damper insures that the departure time of packet k satisfies $e_k^{initial} \le x_k \le e_k^{ter\min al}$. In other words, if packet k arrives before its terminal eligibility time, then it will depart no earlier than its initial eligibility time and no later than its terminal eligibility time. We assume that $a_k \le x_k \le x_{k+1}$ for each k, i.e. the damper serves packets in a causal, FIFO manner. The actual departure times for a traffic stream from a damper may be determined by the state of other network elements, but always satisfies the constraints above.

We will make use of the following lemma, which is intuitively obvious.

Lemma 1: Suppose $R^{in}(t)$ is the number of packets arriving to a damper in the interval [1, t], $\overline{Z}(t)$ is the number of packets that are assigned initial eligibility times in [1, t], $\hat{Z}(t)$ is the number of packets that are assigned terminal eligibility times in [1, t], and finally $R^{out}(t)$ is the number of packets departing the damper in [1, t]. If $R^{in}(t) \ge \hat{Z}(t)$ then $\overline{Z}(t) \ge R^{out}(t) \ge \hat{Z}(t)$.

A *null damper* is defined to be a damper which passes each packet with zero delay. In the notation of Lemma 1, this implies $R^{out}(t) = R^{in}(t)$. A damper may operate as a null damper if $\overline{Z}(t) \ge R^{in}(t) \ge \hat{Z}(t)$ for all t. Of course, Lemma 1 is trivially true for a null damper. As a practical matter, a null damper does not need to be actually

implemented. We define a null damper to address the situation where dampers are not used. This is convenient for analysis purposes.

4 Network Proportional Fairness Scheduling (nPFS)

We consider a service class traversing a series of H servers in tandem. Let R_{h-1} describe the traffic entering server h, and suppose the traffic departing server h feeds server $h+1$ for all h satisfying $1 \le h < H$. We also define $R^{in} = R_0$ and $R^{out} = R_H$ throughout this section.

Define $d_h[t]$ to be the virtual delay of the class through the first h servers, i.e. $d_h[t] = \min\{\Delta : \Delta \ge 0 \quad and \quad R_0[1,t] \le R_h[1,t+\Delta]\}$ and define $d_0[t] = 0$. Finally, let $B_h[t]$ be the backlog at the end of slot t at server h, i.e. $B_h[t] = R_{h-1}[1,t] - R_h[1,t]$.

For simplicity, we assume that each server has the same capacity c. Specifically, the maximum number of packets that a server can serve is assumed to be c per slot, for each server in tandem network. We assume that each server serves packets in 'cut-through' manner, meaning that a packet, which arrives in one slot, may depart in the same slot. Let n_i be the number of servers traversed by an arbitrary virtual path i in tandem network. We denote the route of this virtual path by $\{o(i,h)\}_{h=1}^{n_i}$, where $o(i,h)$ maps to an outgoing link of a switch for $h = 1,2,...,n_i$. We represent the "source" of this virtual path by $o(i,0)$. Define, $I_{i,h}$ to be the set of virtual paths that pass through the outgoing link $o(i,h)$, i.e. $I_{i,h} = \{(j,m) : o(j,m) = o(i,h)\}$.

nPFS Algorithm

Each virtual path i in the network is assigned a set of "hop-by-hop" maximum delay proportional functions $P^{D_i^1}, P^{D_i^2}, \ldots, P^{D_i^{n_i}}$. We define the subnet maximum delay proportional function for hop h for virtual path i to be $P^{D_i^{(h)}}$, where $P^{D_i^{(h)}} = P^{D_i^1} * \cdots * P^{D_i^h}$. Virtual path i is also assigned a *damper function* $\overline{P}^{D_i^{(h)}}$ for $h = 1,2,...,n_i+1$, where we assume that $\overline{P}^{D_i^{(h)}}(x) \ge P^{D_i^{(h-1)}}(x)$ for all x. For notational convenience, define $P^{D_i^0} = P^{D_i^{(0)}} = \overline{P}^{D_i^{(0)}}$ for all i.

Deadlines and eligibility times *for each server* are determined by the traffic that *enters* each virtual path. In particular, these deadlines and eligibility times do *not* depend on traffic flow *inside* the route of a virtual path, and can be calculated at the entrance to the virtual path. The eligibility times at each server govern the operation of the corresponding dampers at that server.

A packet is said to become *active* at server $o(i,h)$ as soon as it departs from the corresponding damper at server $o(i,h)$. More specifically, in the network proportional fairness scheduling (nPFS) algorithm, each server in the network serves packets such that in each slot, an *active* packet with the smallest possible deadline *for that server* is served. If there are no active packets stored at a server in a given slot, that server will be idle in that slot. The deterministic performance bounds we derive hold independently of how the dampers operate, as long as the dampers respect the initial and terminal eligibility times of each packet.

Assignment of Deadlines and Eligibility Times

The deadlines and eligibility times for the packets of virtual path i are functions of the arrival process R_i^0, so they can be calculated prior to entering server $o(i,1)$. The following lemma demonstrates that it is possible to compute these deadlines and eligibility times in real-time.

Lemma 2: If the k^{th} packet of virtual path i arrives at time t, then $D_{i,k}^h = \hat{E}_{i,k}^{h+1} = D_{i,k}^h(t)$ and $\hat{E}_{i,k}^h = \hat{E}_{i,k}^h(t)$, where

$$D_{i,k}^h(t) = \min\left\{ \begin{array}{c} u : u \geq t \quad and \\ \min\limits_{s:0\leq s\leq t-1} \left(R_i^0(s) + P^{D_i^{(h)}}(u-s)\right) \geq k \end{array} \right\}$$

and

$$\overline{E}_{i,k}^h(t) = \min\left\{ \begin{array}{c} u : u \geq t \quad and \\ \min\limits_{s:0\leq s\leq t-1} \left(R_i^0(s) + \overline{P}^{D_i^{(h)}}(u-s)\right) \geq k \end{array} \right\}$$

Note that the lemma also implies that each deadline and eligibility time of a packet is never less than the arrival time of the packet to the virtual path.

5 Simulation Result

In order to demonstrate the ability of the nPFS algorithm to efficiently statistically guarantee the proportional fairness principle in multiplex service classes, we ran a simulation on a small network, consisting of three servers. The simulation was based on a continuous time model. Each server had a capacity of C = 155, 520, 000 bps, corresponding to an OC-3 link. There were three different service classes, and they were routed through all three servers. Each service class stream was generated using a Markov modulated Poisson process model, with the average burst length of 10msec, 20msec and 30 msec, such that during a burst data was generated at rate C bps, and the average rate of data generated was $0.3C$ bps. The service class streams were shaped to the envelope $b(t) = \min\{Ct, \sigma + 0.5Ct\}$, where σ was set to $0.015C$,

corresponding to a maximum burst length of 30msec. The end-to-end delay para-meters allocated to the three service classes were δ_1=1.0, δ_2=2.0 and δ_3=3.0. A packet from each service class stream generated at time t was assigned the deadline $t + \Delta^{(i)}$ at server i, where $\Delta^{(i)} = 0.01i\delta_i$. Deadlines for service class traffic were assigned consistent with equation (13), and packets released from dampers before their terminal eligibility time were chosen on the basis of earliest deadline.

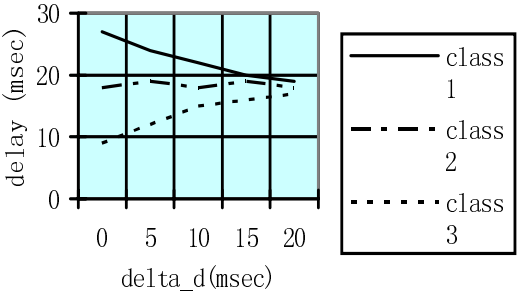


Fig. 1. Comparing delay of three service classes

In Figure 1, the delay comparison of three service class traffics at the tandem network is shown as a function of the endurance parameter Δ_d. These results are based on one simulation run of 10 seconds for each data point. The case Δ_d = 0 corresponds to effectively forcing the jitter of the service class traffic to zero at each hop. As Δ_d increases, the flexibility afforded to the damper increases. As expected, the delay differentiation in the traffics decreases with increasing Δ_d. This is due to the ability of server to delay serving traffic that was to guarantee the minimum delay proportion function.

6 Conclusion

This paper made three contributions. First, we proposed the proportion function from proportional fairness principle [1]. Second, we proposed and analyzed the *network* proportional fairness scheduling strategy (nPFS), which considered QoS requirements, such as packet delay and loss rate, and system fairness simultaneously. At last, the simulation result of nPFS was proposed. Because of the complexity of the research on multiple-object performance, there is still no effective integrated performance scheduling strategy up to now. This paper made a useful theoretical pilot study, and the result can be applied to design, implement and optimize of packet scheduling strategy.

References

1. C. Lin, Y. Jiang, and W. Zhou. Integrated performance evaluating criteria for network traffic control. IEICE Trans. Commun., vol.E85-B, no.11, November 2002. pp2447-2456
2. R. L. Cruz. A calculus for network delay, Part I: Network Elements in Isolation, IEEE Trans. on Information Theory, vol. 37, no. 1, Jan. 1991, pp.114-131.
3. R. L. Cruz. A calculus for network delay, Part II: Network Analysis, IEEE Trans. on Information Theory, vol. 37, no. 1, Jan. 1991, pp. 132-141.
4. J. Kurose. On computing per-session performance bounds in high-speed multi-hop computer networks. In: Proc. of ACM Sigmetrics and Performance '92, New York, 1992. pp. 128-134
5. O. Yaron and M. Sidi. Performance and stability of communication networks via robust exponential bounds. IEEE/ACM Trans. on Networking, vol.1, no.3, June 1993, pp. 372-85.
6. C. S. Chang. Stability, queue length, and delay of deterministic and stochastic queueing networks. IEEE Trans. on Automatic Control, vol.39, no. 5, May 1994, pp. 913-931.
7. Hurley P, Boudec J L. A Proposal for an Asymmetric Best-Effort Service. In: Proceedings of IWQOS'99, London, 1999. pp. 129-132
8. Pitsillides A., Stylianou G., et al. Bandwidth Allocation for Virtual Paths (BAVP): Investigation of Performance of Classical Constrained and Genetic Algorithm Based Optimisation Techniques. In: Proceedings of INFOCOM'2000, Tel Aviv, 2000. pp. 1379-1387
9. Dovrolis C, Stiliadis D. Relative Differentiated Services in the Internet: Issues and Mechanisms. In: Proceedings of ACM SIGMETRICS'99, Atlata , May 1999. pp204-205
10. 10. A.K. Parekh and R.G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: the single-node case. IEEE/ACM Trans. on Networking, vol. 1, no. 3, June 1993, pp. 344-57
11. Y. Jiang, J. Wu, The proportional fairness scheduling algorithm on multi-classes, Science in China Series F, Vol.46 No.3, June 2003, p 161-174

Probability Based Dynamic Load-Balancing Tree Algorithm for Wireless Sensor Networks[1]

Tingxin Yan, Yanzhong Bi, Limin Sun, and Hongsong Zhu

Institute of Software, Chinese Academy of Sciences,
P.O.Box 8718, Beijing, P.R.China 100080
{tingxin03, yanzhong02, sunlimin, hongsong}@ios.cn

Abstract. Load balance can reduce hot spots, maintain network connectivity and prolong lifetime of wireless sensor networks. In large scale and randomly deployed wireless sensor networks, the energy consumption is sharply uneven among sensor nodes. We propose a new routing mechanism to achieve load balance through constructing a dynamic load-balancing tree (DLBT) in wireless sensor networks. The DLBT structure is a tree-like topology with a critical character that one node may have more than one candidates of parent node. DLBT achieves load balance by adjusting the forwarding probability to each parent candidate according to its traffic burden. We also provide a distributed algorithm to construct and maintain the DLBT structure. Simulation results show that our DLBT routing provides much higher load balance than the shortest path tree mechanism.

1 Introduction

Wireless sensor network is a promising research area in recent years. A typical sensor network consists of a large amount of tiny devices which are equipped with sensors, processors, wireless transceivers and power units. These tiny devices are deployed randomly and used to collect surrounding information in most cases by forming an ad-hoc multihop network and transferring data to the base stations.

The network lifetime is a fundamental issue in determining the performance of wireless sensor networks. Wireless sensor network is a self-organizing network without human management where all the nodes are mainly supplied with batteries which cannot be replenished. Sensor nodes run out their energy easily and it will generate holes and even separate the network into several parts. Routing protocol is a critical issue to the lifetime of sensor networks. A well-designed routing protocol should reduce the total energy consumption and make energy consumption evenly to all sensor nodes. Some of the routing mechanisms, such as shortest path first, will find out the shortest path from data source to the destination, but the disadvantage is that it will cause the nodes in the data transferring path to have a heavier forwarding burden than other nodes in the network and separate the network into several parts if the optimal path is used for a long time.

[1] This work is supported by National Natural Science foundation of China under grants No.60434030 and No.60373049.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 682 – 691, 2005.

We focus on the problem of load balance in a data-centric sensor network. A typical data-centric sensor network consists of one or a few base stations (or sink nodes) responsible for data gathering and a large amount of sensor nodes collecting data and transfer them to the base stations. This kind of network has a common sense in wireless sensor network applications [12] [13] such as environmental monitoring, battlefield evaluating, fire monitoring and alarming, human-body healthy care, etc. The base stations are generally supposed to have more energy supply and more powerful processor than sensor nodes. They can even provide connection to the Internet or other traditional networks if necessary.

In wireless sensor networks, the major part of energy consumption is caused by the wireless communication. So we can use the number of packets delivered by a sensor node to measure its traffic load. A load-balancing routing protocol should provide certain mechanisms to manage the data flow and reduce the differences of traffic load among nodes as much as possible. In data-centric sensor networks, as the data flow is convergent to the base station, the sensor nodes close to the base station would afford heavier load because they have to relay much more packets than the nodes far from the base station. So it is not practical to achieve a complete load balance among all the nodes in data centric networks, but a local load balance is feasible through certain mechanisms. In sensor networks, it is common to take hop count as the metric of distance from a sensor node to the base station. We use level to present the hop count from one node to the base station and call them homo-leveled nodes if they have same level values. The aim of this paper is to achieve load balance among the homo-leveled nodes in each level of the network.

The key contributions of this paper are in the following areas. First, we introduce the Dynamic Load-Balancing Tree (DLBT) for wireless sensor networks which can achieve load balance in a randomly deployed sensor network. Second, we use forwarding probability to adjust the load assignment. Third, we give a distributed algorithm for each node to compute its load and the probability to next node.

The rest of this paper is organized as follows. Section two presents the previous works on wireless sensor network routing which are related to our research. Section three presents the Dynamic Load Balancing Routing in detail. Section four gives out the simulation result and section five is the conclusion.

2 Related Works

There have been extensive studies on wireless sensor network routing in recent years. Some recent works have touched on the load balance problem in wireless sensor networks and have provided some solutions.

Hsiao et al [3] introduced two kinds of load balanced trees for wireless access networks which are fully load-balanced tree and top load-balanced tree. These load-balanced trees can be applied in wireless sensor networks as well. They also provided an algorithm to construct the load-balanced tree in the network.

Dai et al [11] studied the load balance problem in a grid topology. They introduced another load-balanced tree called hierarchy-balanced tree besides the two introduced by Hsiao. They also bring forward to use Chebyshev sum as a criteria of load balance and design an algorithm to construct a top-balanced tree in wireless sensor networks.

The two algorithms mentioned above are both centralized, that is to say, they need to be computed on the data center of the network. Besides, they only achieve a top-level balance which is not enough for large scale sensor networks.

Gao et al [5] introduced a load balanced routing in a narrow strip topology with width at most $\sqrt{3}/2$ times the communication radius. They provided three greedy algorithms to implement different degree of load balance. They also designed a distributed implementation of the algorithms. But the strong precondition restricts the application range of this algorithm.

Kakiuchi [6] also designed a routing tree and designed an algorithm to modulate the routing tree when the traffic load of some nodes has exceeded a predefined threshold. But it must inform the root of the tree when adjustment is needed, so the protocol cost may be considerably high.

G.Gupta et al [8] divided the network into clusters and presented an algorithm to achieve the load balance among different cluster heads. This mechanism is suitable for large scale networks, but the load balance is still restricted among cluster headers.

The research on geographic routing and probability based routing are also part of the fundamental of our research. J. Newsome and D. Song introduced Graphic Embedded routing (GEM) [7]. They defined the routing tree as a virtual polar system. The virtual angle range for each node is a great measurement for the size of the network. It can be used in load-balancing routing if the virtual angle range is endowed with the meaning of traffic load.

In some routing protocols, they dynamically choose next hop according to certain forwarding probability. In Parametric Probabilistic Routing [4], the nodes close to the destination would have greater forwarding probability than the nodes further from the destination. In Reliable Information Forwarding (ReInForM) [10], the nodes with better communication quality would have higher forwarding probability. In SPEED [11], the nodes with less latency would have greater probability to be chosen as forwarding nodes. Our mechanism builds a dynamic load-balancing tree which is a hierarchical structure and uses forwarding probability to adjust traffic load among homo-leveled nodes.

3 Dynamic Load-Balancing Routing

3.1 Dynamic Load-Balancing Tree and Routing Algorithm

We propose dynamic load-balancing tree to realize load balancing. In the dynamic load tree, base station is the root of the tree and nodes are organized in a hierarchical manner as in the common tree topology. Every node has a level value which represents the number of hops to the base station. In dynamic load-balancing tree, each node may have more than one upstreaming node as the parent candidates and each candidate has a probability to be chosen as the parent node. The probability is related to the traffic load of each candidate and is also a measurement of load assignment to each candidate. One of these candidates will be chosen as the forwarding node at a time, so the dynamic load tree is totally a tree topology at any snapshot of the network, while the node may switch to different upstreaming node at different times. An example of dynamic load tree is shown in Figure 1.

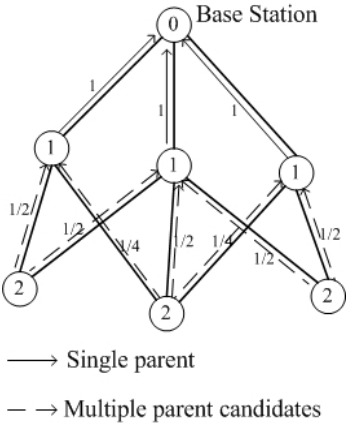


Fig. 1. An example of Dynamic Load-Balancing Tree. Some nodes have multiple parent candidates, and have separate forwarding probability to each candidate

If node n has only one parent candidate m, the node m will be the parent node of node n and the load of n will be distributed to m completely. If node n has several parent candidates, say $M_n = \{m_1, m_2, m_3, \ldots, m_n\}$, then node n will have a forwarding probability to each m_i belongs to M_n, so the forwarding probability is a measurement of the load distribution of node n. The greater probability one candidate has, the more load is assigned to that candidate.

We use Dynamic Load-Balancing Tree to achieve load balance among homo-leveled nodes. Consider node N_d in level k+1, assume it has n parent candidate in level k, we use $N_k = \{N_{k1}, N_{k2}, N_{k3}, \ldots, N_{kn}\}$ to denote the candidate nodes and $L_k = \{L_{k1}, L_{k2}, L_{k3}, \ldots, L_{kn}\}$ to denote the traffic load of each candidate node in level k. In order to make the traffic load even to each candidate, let the forwarding probability be in inverse proportioned to the traffic load, so we have:

$$L_{k,i} P_{k,i} = C_{k+1,d} \tag{1}$$

Here $C_{k+1,d}$ is a constant for node N_d. The sum of forwarding probabilities to all its candidates equals to one, i.e.

$$\sum_{i=1}^{n} P_{k,i} = 1 \tag{2}$$

So we can conclude that:

$$\sum_{i=1}^{n} \frac{1}{L_{k,i}} = \frac{1}{C_{k+1,d}} \tag{3}$$

We can calculate the constant $C_{k+1,d}$ from all the $L_{k,i}$ of the candidates according to equation 3 and then calculate every $P_{k,i}$ according to equation 1.

Figure 2 is an example of the load-balancing algorithm on DLBT. The initial load of node j is 2 and is higher than its homo-leveled node i and k. After the probability assignment, the nodes in level one have the same load value.

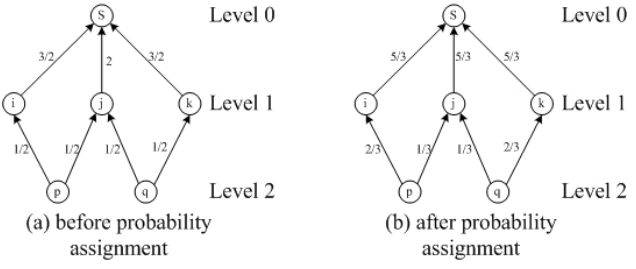


Fig. 2. An example: load balance through probability assignment in DLBT

The routing on the DLBT is relatively simple. For each data packet, node will choose one of its parent candidates as the forwarding node according to the forwarding probability. So data packets may go through different paths from data sources to the base station. This mechanism disperses the traffic load effectively and avoids using a certain path excessively.

3.2 DLBT Construction

The constructing process of dynamic load tree consists of three phases which are tree initializing, load announcing and probability allocating.

Tree-Initializing Phase. The base station first broadcasts a message of type "tree initialization", with a level field set to zero. The nodes within the communication range of base station will become its children with their level value set to one. Then each of these nodes wait for a short random time and broadcasts the constructing message with level field set to one. When node i with level k receives another constructing message whose level value equals to k-1, it marks the node sending this

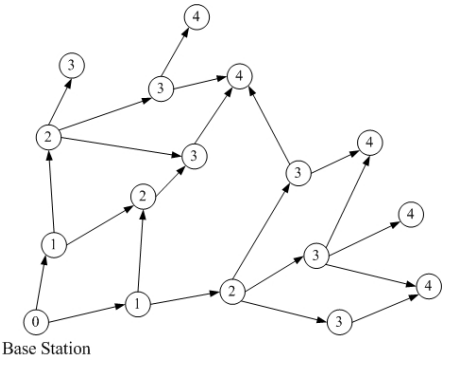


Fig. 3. Tree initialization phase of Dynamic Load-Balancing Tree, started up from base station

message as the parent node instead of discarding the message. This mechanism ensures every node remember all the reachable nodes in the ascending order. The constructing process will continue until all nods in the network have settled their level. This process is illustrated in Figure 3.

Load-Announcing Phase. After initializing the dynamic load-balancing tree, a load-announcing process will be started. This process runs from leaf node to the base station. As the load of each node is not determined at the very beginning, we assume every node generates the same amount of traffic load, and so we can use the number of downstream nodes as a reasonable initial value for traffic load. After the DLBT is established, every node will use its practical load value. The load-announcing process is as follows:

1. The traffic load of each node equals to one;
2. One node will divide its traffic load evenly to all of its upstream node;
3. One node will count its load as the sum of the load generated itself and the load its downstream nodes send to it.

This process will continue until the base station get its traffic load, and this traffic load is the total load of the network.

Probability-Allocating Phase. According to the process described above, the nodes with the same level may afford different traffic load because the number of their downstream nodes is different. So we need a feedback process to adjust the traffic load. One node will tell its traffic load to all of its downstream-neighbors. After receiving the traffic load from all of its upstream nodes, one node will redistribute its traffic load to these nodes according to equation 1 and 3 described in the above section.

In fact, one node can know its parents' load by intercepting the message broadcasted by their parents to the upstreaming nodes in load announcing phase. It can greatly reduce the packets transferred and save energy.

The load announcing phase and probability allocating phase will be executed recursively. As these two phases can improve load balance every time they are executed, it can acquire much higher degree of load balance in a few running times.

3.3 DLBT Maintenance

The sensor network topology may change over time. Nodes may fail, and new nodes may be added to the network. Some adjustments are needed to keep load balance when either of these events happens.

When a new node comes into the network area, it would find the nodes with minimum hop in its communication range and send a "parent choosing" message to inform them. The nodes which are chosen as new parents will recount both its load and the forwarding probability according to equation 3. Then it will reply to its new child with that forwarding probability.

If a node cannot communicate with one of its parent node for a certain period of time, it will mark the parent node invalid. In this case, node will recompute the division of its load to the rest of its parent nodes and tell them the new load. The nodes which receive a new load announcement will recompute its load recursively.

4 Simulation

We evaluate the load balance performance of Dynamic Load-balanced tree and compare it with the shortest path tree (SPT). We use Chebyshev Sum Inequality as the criteria of load balance. This criterion has been introduced and used in [1]. The definition of the Chebyshev Sum Inequality is as follows: for all $a \subseteq C^N$ and $b \subseteq C^N$, where

$$a = \{a_1, a_2, ..., a_n\} \tag{4}$$
$$b = \{b_1, b_2, ..., b_n\}$$

and

$$a_1 \geq a_2 \geq a_3 \geq ... \geq a_n \tag{5}$$
$$b_1 \geq b_2 \geq b_3 \geq ... \geq b_n$$

we have

$$n\sum_{k=1}^{n} a_k b_k \geq (\sum_{k=1}^{n} a_k)(\sum_{k=1}^{n} b_k) \tag{6}$$

Define W_{ki} be the load of the i_{th} node in level k, we can form a vector W_k = $\{W_{k1}, W_{k2}, W_{k3}, \ldots, W_{kn}\}$ to present the load of each node in level k. To evaluate the load balance among different nodes in the same level, let a=b=w, in this case, the inequality will become:

$$n\sum_{k=1}^{n} W_{bk}^2 \geq (\sum_{k=1}^{n} W_{bk})^2 \tag{7}$$

With equality if and only if $W_{b1}=W_{b2}=\ldots=W_{bk}$ Then the balance factor used in the simulation is as follows:

$$\theta = \frac{(\sum_{k=1}^{n} W_{bk})^2}{n\sum_{k=1}^{n} W_{bk}^2} \tag{8}$$

We compare the performance of our algorithm with SPT within the homo-leveled nodes. In SPT, each node chooses one of its neighbors who have the shortest path to the base station. In fact, just like we have mentioned above, there may exist more than one node with shortest path to the base station, we choose one of them randomly to break the tie.

We use a randomly deployed network to evaluate the algorithms. We aim to check the degree of load balance with certain level nodes in the network. The number of nodes is up to 20. Figure 4 and Figure 5 access the balance factor of routing tree as a function of number of nodes within the same level. In the average case shown in Figure 4, our algorithm can achieve a great level of load balance when just execute one time. In the worst case shown in Figure 5, our algorithm is slightly better and smooth than SPT when runtime equals to one, and the degree of load balance increases as the runtime increases.

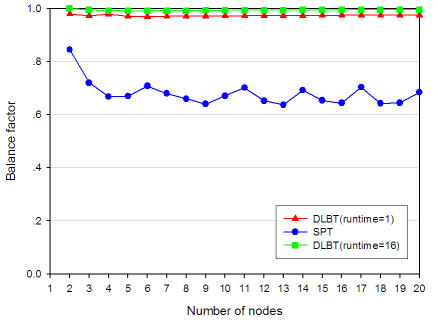


Fig. 4. Average Performance in a randomly generated network

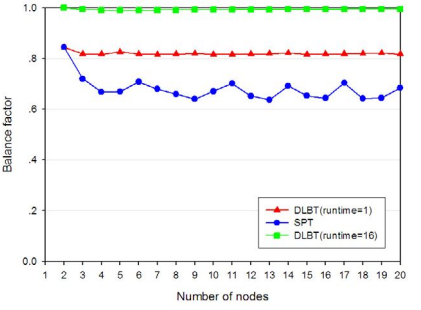


Fig. 5. Worst Performance in a randomly generated networkIn

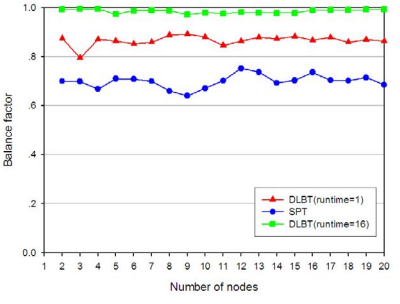


Fig. 6. Average Performance in a sharply uneven-loaded network

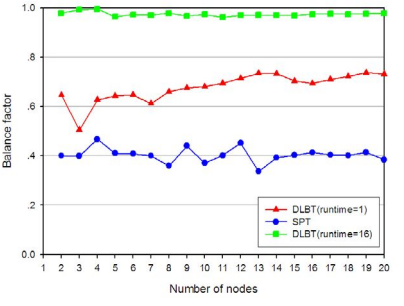


Fig. 7. Worst Performance in a sharply uneven-loaded network

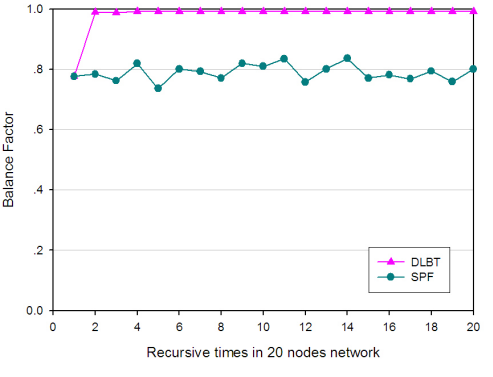


Fig. 8. Improvement of load balance vs. execution times

In Figure 6 and Figure 7, we construct an unevenly-loaded network where we randomly assign 3 nodes with a much heavier load than others. In Figure 6, we can see that our algorithm can exalt load balance even if execute only one time, and can

achieve much higher load balance as runtime increasing. Even in the worst case shown in Figure 7, our algorithm can still achieve load balance after 16 times of execution.

In Figure 8, we measure the relation between the balance factors and execute times in a randomly generated network with 20 nodes. We can see that the load balance will be enhanced greatly when only executed 2 times.

5 Conclusion

In this paper, we provide a load balancing routing mechanism for wireless sensor networks. Our load balancing mechanism is suitable for the network architecture with multiple data source and single base station. By using multiple parent node candidates, our mechanism can distribute load dynamically according to the traffic burden of the parent node candidates. Our mechanism avoids the extension of transfer path while providing load balance among homo-leveled nodes. Our load balancing algorithm can be executed recursively to achieve higher degree of load balance and increase the load balance compared with shortest path tree in a randomly deployed network.

References

1. H. Dai, R. Han, "A Node-Centric Load Balancing Algorithm For Wireless Sensor Networks". IEEE GLOBECOM – Wireless Communications 2003
2. I.Akyildiz, W. Su, Y. Sankarasubramaniam, and E.Cayirci. “Wireless Sensor Networks: A Survey”. Computer Networks, 38(4): 393-422, March 2002.
3. P. H. Hsiao, A. Hwang, H. T. Kung, and D. Vlah. “Load-Balancing Routing for Wireless Access Networks”. IEEE Infocom, April 2001.
4. Christopher L. Barrett, Stephan J. Eidenbenz and Lukas Kroc, "Parametric Probabilistic Sensor Network Routing". ACM WSNA03, September 19, 2003.
5. Jie Gao, Li Zhang, "Load Balanced Short Path Routing in Wireless Networks". IEEE Infocom, March 2004.
6. Hirofumi Kakiuchi. "Dynamic Load Balancing in Sensor Networks". Technical report on Stanford University, June 2004
7. J. Newsome and D. Song. GEM: Graph EMbedding for Routing and Data-Centric Storage in Sensor Networks Without Geographic Information. The First ACM Conference on Embedded Networked Sensor Systems,2003
8. G.Gupta and M. Younis. Performance Evaluation of Load-Balanced Clustering of Wireless Sensor Networks. Telecommunications, 2003. (ICT'03). March 2003.
9. C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networks, August 2000.
10. B. Deb, S. Bhatnagar, and B. Nath. “ReInForM: Reliable Information Forwarding Using Multiple Paths in Sensor Networks”. The 28th Annual IEEE Conference on Local Computer Networks (LCN), October 2003.
11. Tian He, John A. Stankovic, Chenyang Lu, and Tarek F. Abdelzaher, "SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks," International Conference on Distributed Computing Systems (ICDCS 2003), Providence, RI, May 2003.

12. S. Madden, M. Franklin, J. Hellerstein, and W. Hong, “TAG: a tiny aggregation service for ad hoc sensor networks”, in USENIX Symposium on Operating Systems Design and Implementation, 2002.
13. S. R. Madden, R. Szewczyk, M. J. Franklin, and D. Culler, “Supporting Aggregate Queries Over Ad-Hoc Wireless Sensor Networks”, in Workshop on Mobile Computing Systems and Applications, 2002.
14. David B Johnson and David A Maltz, “Dynamic source routing in ad hoc wireless networks,” in Mobile Computing, Imielinski and Korth, Eds., vol. 353, pp. 153–181. Kluwer Academic Publishers, 1996.

A Prediction-Based Location Update Algorithm in Wireless Mobile Ad-Hoc Networks

Jun Shen[1,2], Kun Yang[1], and Shaochun Zhong[3]

[1] University of Essex, Department of Electronic Systems Engineering, UK
{junshen, kunyang}@essex.ac.uk
[2] Beijing Institute of Technology, Dept. of Computer Science, Beijing, P.R. China
[3] Northeast Normal University, School of Software. Changchun, P.R. China
sczhong@yahoo.com

Abstract. Location information in Mobile Ad Hoc Networks (MANETs) provides efficiency to uni-cast and multi-cast algorithms. This paper proposes a new location update algorithm called *PLU*. The essence of PLU lies in the integration of location prediction and one-hop broadcasting of location update packets. The full PLU algorithm and its associated data structure are described in the paper. The simulation results have shown an improved performance of PLU thanks to its newly introduced type of updates where location prediction is applied to reduce the number of packets or bytes transmitted for location update purpose whereas keeping a high accuracy level of location information.

1 Background and Motivation

Mobile ad-hoc networks (MANETs) consist of a set of wireless mobile nodes that cooperatively form a network without a fixed infrastructure. In such a wireless network, a message sent by a node usually reaches all its neighbouring nodes that are located within the transmission radius of the sender. Because of the limited transmission radius, the routes between the original sender node and the intended final receiver node normally consist of several hops. As such, each node in a MANET also serves as a router to route information between neighbours, thus contributing to the end-to-end message delivery across the whole network. Routing plays a critical part in the practical success of any MANET. Many routing protocols have been proposed for MANETs, and a comprehensive review of these protocols can be found in [1].

Recently, more and more researchers in MANET community realized the importance of location information of nodes in MANET routing and some location-aided routing algorithms were put forward, such as Location-Aided Routing (LRA) [2], the Distance Routing Effect Algorithms for Mobility (DREAM) [3], and the Geographical Routing Algorithm (GRA), amongst others. I. Stojmenovic gave a well-summarized survey of most of the typical location-based routing algorithms in ad-hoc networks [4]. It is believed that the advantages of using location information outweigh the additional cost [5]. This is further justified by the increasing availability of small, inexpensive low-power GPS receivers and techniques for finding relative coordinates based on signal strengths. Given the fact that location information of nodes can be obtained using whatever way, the next step is how to utilize them effectively and

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 692 – 701, 2005.

efficiently to benefit the routing in MANETs. This is typically what the location information service (LIS) is supposed to do. Actually knowing other nodes' location information is also useful, and sometimes vital, in some cases other than routing, such as life rescue, movement decision making in war field, etc.

The essence of a location information service is its location update scheme. The ultimate goal of any LIS is to reduce the number of *location update packets* (*LUP*) or bytes (as overhead) transmitted across the MANET for the maintenance of LIS whereas keeping as high as possible the accuracy level of location information. To this end, many location update schemas are proposed in the current literature.

The simplest location update scheme is location information flooding where each node broadcasts its own location information in a flooding way on a periodic basis. A location table, containing nodes' location information received by the node, is maintained in every node in the network. Flooding causes massive transmission of redundant messages and consequently collisions, and usually is not an ideal solution for location information updating. So a number of improved location information service algorithms were proposed.

The location information service utilized in the DREAM routing protocol [3], referred to as DREAM Location Service (DLS) here, takes into consideration the *distance* factor when sending location updating packets. In DLS, if the distance of two nodes is further away then less updates are produced – this is because faraway nodes *appear* to move more slowly than nearby nodes. DLS classifies the whole nodes in a network into two types: nearby nodes and faraway nodes. Each mobile node in the MANET transmits an LUP to nearby nodes at a given rate and to faraway nodes at another lower rate. By reducing the frequency of sending location updating packets to faraway nodes, the overall DLS overhead is reduced.

T. Camp *et al.* discussed in [6] another LIS algorithm named Simple Location Service (SLS). SLS also transmits LUP to its neighbours at a give rate. The difference between SLS and DLS lies in the type of information exchanged and the distance the information is propagated. Specifically, SLS transmits *table* containing multiple nodes' locations to neighbours and then neighbours will carry out some processing before the next hop transmission; whereas DLS transmits only the sending *node*'s location to its neighbours and then immediately to other nodes via neighbour. Our LIS algorithm proposed in this paper, called PLU (short for Prediction-based Location Updating), also utilizes the *table* transmission idea as that in SLS but is also different in terms of the content of the table entry, updating strategy, etc. In the same paper [6], T. Camp *et al.* also presented another LIS algorithm called Reactive Location Service (RLS). RLS is similar to LAR [2] but has a comprehensive mechanism to support location inquiry. In some research work, nodes as location servers were proposed to maintain location information of some other nodes in a MANET. A typical example of this kind is Grid Location Service (GLS).

One common aspect of all these abovementioned algorithms is that none of them beard the idea of *prediction*. A proper prediction of node's current location based on its previous location and movement model has the potential to significantly reduce the number of location update packets (and the computation is far more energy-economic than transmission). [7] realized the importance of location prediction in location information system and routing and applied it for efficient QoS routing in MANETs. Later on, similar location prediction mechanism appeared in Agarwal, *et al*'s work

[8]. However, we think the prediction effort made in [8] was undermined to some extent by its adoption of a simple flooding algorithm for location information updating. Inspired by the prediction effort made in [7] and [8], this paper exploits the integration of location prediction with a one-hop broadcasting algorithm. The proper cooperation and better-off balance between location prediction and location update constitute one of the important investigation of this paper.

Based on the above discussion, Section 2 details our location information service algorithm PLU, which is followed by simulation environment and parameter design in Section 3 and simulation result analysis and discussion in Section 4. The last section, Section 5, concludes the paper.

2 PLU Algorithm

The PLU algorithm proposed in this paper aims to reduce the amount of location update packets by limiting the update transmission to those node entries in location update table that satisfy certain criteria. In the criteria there is a need to know the current location of other nodes, and this is carried out by prediction.

2.1 PLU Location Prediction Scheme

We assume that a uniform velocity linear movement model is adopted for each node in the MANET during the period between two location updates. Then based on the most recent previous location (x_1, y_1) of a node at the time point t_1, the current location (x_2, y_2) of the node at the time point t_2 can be predicted by using the following formulas:

$$x_2 = x_1 + v \cdot (t_2 - t_1) \cdot \cos\theta \tag{1}$$

$$y_2 = y_1 + v \cdot (t_2 - t_1) \cdot \sin\theta \tag{2}$$

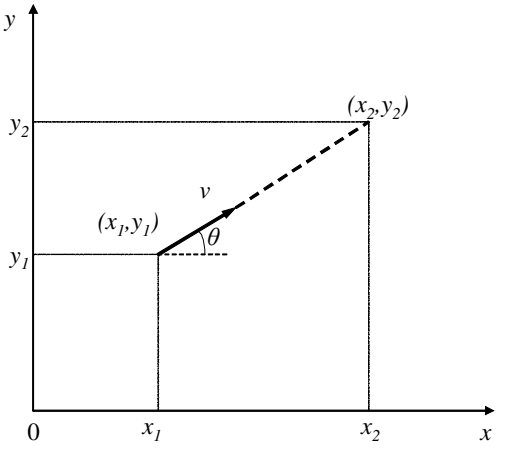


Fig. 1. Node Positions & Movement

As depicted in Fig. 1, it is assumed that the node moves at a velocity of v with movement direction being angle θ to x axis. Velocity v and angle θ can be calculated by using the Pythagoras' theorem via the following formulas:

$$v = \frac{\sqrt{(x-x')^2 + (y-y')^2}}{t-t'} \tag{3}$$

$$\theta = \begin{cases} ar\cos\dfrac{x-x'}{\sqrt{(x-x')^2+(y-y')^2}} & y-y' \geq 0; \\ 2\pi - ar\cos\dfrac{x-x'}{\sqrt{(x-x')^2+(y-y')^2}} & y-y' < 0. \end{cases} \tag{4}$$

where (x, y) and (x', y') are the locations of the node at the time point t and t' respectively.

2.2 Data Structure Used in the PLU Algorithm

The PLU algorithm is a kind of location update scheme where each node in the network sends out Location Update Packets (LUP) to its neighbours (i.e., nodes within its transmission radius) on a periodical basis. The format of LUP is depicted in Fig. 2. Here a *Resource_Information* field is reserved for future use in cases where resource information is needed. For instance, this field could contain the current power level of the node identified by *NodeID*, which are useful in power-sensitive algorithms. To reduce the performance compromise to be potentially introduced by this field, a minimum one byte length is assigned to this field whose value is fixed to "nil" in the current PLU algorithm. It could be easily relaxed to contain any resource-related information of variable length in the future.

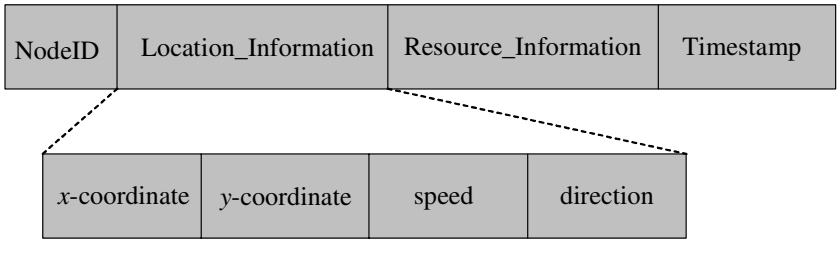


Fig. 2. Location Update Packet Format

As shown in Fig. 2, LUP is employed to propagate node (as identified by *NodeID* field)'s location information at the time point indicated by *Timestamp* field. In a LUP packet, the location information includes not only the geometric coordinates (*x-coordinate* and *y-coordinate*) but also the *speed* and the *direction* of the movement. Be noted that the speed and the moving direction of any individual node may change

as time goes along, so the speed and direction values in LUP reflect only the status at a given time (i.e. *Timestamp*). Both speed and direction are calculated by the sender itself, who is capable of knowing its coordinates at any time point, by using the formula (3) and (4) respectively.

The presence of *speed* and *direction* fields in LUPs enables the PLU to predict the current location of other nodes using formula (1) and (2). It enables the PLU to decide if the difference of location information which will be described in later satisfies certain criteria.

In order to implement the PLU algorithm, the following two tables are maintained at each node: location update table and transmitted information table.

Location Update Table (LUT) contains all the fields packed in LUPs, i.e., *NodeID*, *x-coordinate*, *y-coordinate*, *speed*, direction, empty *Resource_Information*, *Timestamp*, plus a new status field called *New* indicating the readiness of this piece of location information to be checked for transmission to other nodes (1 means ready to be checked for possible sending and 0 means the opposite). Upon the receipt of a LUP, the receiving node invokes an procedure called *locationTableUpdate*() to update its local LUT.

```
procedure locationTableUpdate (LUP lup) {
  while (lup has more node information) do {
    nodeNewInfo = getNextNodeInfo(lup);
    if (nodeNewInfo.NodeID ∉ LUT.nodeList) then {
      newNode = LUT.addRow(nodeNewInfo);
      newNode.New = 1;
    } else {
      existingNode = LUT.getRow(nodeNewInfo.NodeID);
      if (existingNode.TimeStamp < nodeNewInfo.TimeStamp) then {
        updatedNode = LUT.updateRow(nodeNewInfo);
        updateNode.New = 1; // newer info needs to be propagated } } } }
```

Every node in the network has its own location information written in its LUT as well, which is updated periodically by the node itself. If the difference between the new location information and the previous one is different sufficiently enough, the *New* field of this local node is set as well.

Transmitted Information Table (TrIT) contains exactly the same fields as LUPs, i.e., *NodeID*, *x-coordinate*, *y-coordinate*, *speed*, *dire*1*ction*, *Resource_Information*, *Timestamp*. TrIT in a node is updated every time a new LUP is broadcasted by this node. Basically it contains all the node location information that has been broadcasted to other nodes. In other words, the neighbouring nodes should at least know the information stored in the TrIT. TrIT represents the view of other nodes to the locations of these nodes in the TrIT table as this location information is the information received and kept by other nodes.

As such, two views, as represented by two tables respectively, are maintained in each node *a*: the local node *a*'s view to other nodes' location information as represented by LUT (here we call it *myView*) and the view of other nodes to the

location of the same set of nodes as represented by TrIT (here we called it *othersView*). These two views are used by the location update algorithm PLU to check if there is a node location difference between these two views. If the difference is significant enough, a location update is generated by PLU. Intuitively, this mechanism reduces the number of LUPs in PLU by imposing a more constrained location updating criterion.

2.3 PLU Algorithm Description

PLU is a location updating algorithm that involves the following two stages.

Stage-1 is to carry out normal location update. The updates at this stage are generated by PLU on a periodical basis with a variable interval $Interval_1$. Obviously, it is hoped that the bigger the node's transmission range ($Range_{trans}$) is the longer the $Interval_1$ is and at the same time the faster the node is moving the shorter this interval is. As such, a similar approach to that employed in [6] is adopted to decide the $Interval_1$, as being calculated using the following formula:

$$Interval_1 = \frac{Range_{trans}}{\alpha \times v_{avg}} \tag{5}$$

where v_{avg} is the average speed of the node and α is a scaling factor. In this stage, all the entries whose *New* field is set are packed into LUP packet(s) and broadcasted but at a relatively longer interval. This longer interval, while reducing the number of LUP packets, may lead to a situation where large changes in node location might not be able to be propagated quickly enough. To avoid this situation, PLU involves in another stage – *stage-2*.

In Stage-2, location update is triggered when there is a considerable change in the node's location (here location means coordinates) but the time for triggering stage-1 update has not arrived yet. Stage-2 update periodically predicts, by using the prediction formulas (1) and (2) given above, the current location of each node existing in the LUT table by calculating separately its entry in both LUT table and TrIT table. If the difference of the two calculations is greater than a given threshold, which means a quite different understanding of a certain node's location between the node and other nodes, then the node will broadcast this new change (as stored in the node's LUT table) to its neighbours immediately via LUP packet.

Stage-2 updates are triggered periodically at an interval ($Interval_2$) shorter than $Interval_1$ so as to propagate bigger changes in a quicker manner than the normal location updates (i.e. stage-1 updates). Typically $Interval_2 = Interval_1/3$. The introduction of stage-2 updates contributes to the high performance of PLU while keeping a fairly low average location error, as to be shown in the simulation results later.

In PLU, all updates are transmitted using LUP(s) of variable length. The more the number of pieces of node information to be broadcasted the longer the LUP length. Given the data structure of LUP packet, the structure of LUT table and the structure of TrIT table, and the two stages of location updates, the PLU algorithm is described as follows:

```
procedure PLU (LUT lut, TrIT trit) {
  specify the value of Interval1, Interval2;
  if (Interval1 timeouts) then { //stage 1
    create an instance of LUP called lup1;
    for (each entry entry_lut in lut whose New field is 1) do {
      append the content of entry_lut into lup1;
      update the peer entry in trit using the content of
entry_lut;
      reset the New field of entry_lut; }
    broadcast(lup1); }
  if (Interval2 timeouts) then {  // stage 2
    create an instance of LUP called lup2;
    for (each entry entry_lut in lut ) do {
     entry_trit = the entry of the same node as entry_lut in
trit;
     x_predicted_lut = formula1(entry_lut.x);
     y_predicted_lut = formula2(entry_lut.y);
     x_predicted_trit = formula1(entry_trit.x);
     y_predicted_trit = formula2(entry_trit.y);
     if (|x_predicted_lut – x_predicted_trit| >
X_CHANGE_THRESHOLD) or (|y_predicted_lut – y_predicted_trit
|>Y_CHANGE_THRESHOLD) then {
       append the content of entry_lut into lup2;
       update entry_trit using entry_lut;
       reset the New field of entry_lut; } }
    broadcast(lup2); } }  // |x| means the non-negative value of x.
```

What the PLU procedure does is to trigger different type of update according to the type of interval that timeouts.

3 Simulation Environment and Algorithm Parameters

To maintain the compatibility of different location updating algorithms, a similar simulation environment to that employed in [6] is adopted in this paper. The performance of each algorithm was tested using network simulator ns-2 (extended by the wireless protocol stack model developed at CMU [9]) in a network of 50 mobile nodes, each having a transmission range of 100m and 2Mbps IEEE802.11 as MAC layer. The simulation area is 300m*600m. Random waypoint is utilized to model node mobility, where a constant pause time of 10s is employed. Movement scenarios were generated by using movement-connection program *./setdest* provided by ns-2. For each mobility speed, 10 simulation trials were generated and the average was used as the final result.

In our simulation test, LUP packet generation was started from 1000th second onwards rather than from the start of the simulation process. This is to avoid the big vibration in average number of neighbours that occurs usually during the first 600 seconds of the simulation process as pointed out by [10]. To avoid neighbour nodes sending LUPs at same time (and thus causing data collision), a random jitter was added before every transmission.

The following location information service algorithms were implemented for comparison: simple flooding, DLS, SLS, PLU, each representing a type of LIS as discussed in Section 1. This section discusses about the specific value for each parameter used in each algorithm.

In simple flooding, every node generates a location update packet which is flooded into the network after a constant interval. In our simulation, this interval was set to 13 seconds. And the location update packet here includes only the coordinates and the corresponding sampling time of one node, i.e., the send node itself.

In DLS, the transmission interval for nearby LUPs was set to be calculated by using formula (5) in Section 2 where α was set to 4, and $Range_{trans}$ was 100m. Nearby nodes were specified as one-hop neighbours. The transmission interval for faraway LUPs is 13 times the nearby interval, i.e., one faraway LUP was broadcasted after 13 nearby LUPs broadcasts, but no longer than 23s. Similar to the simple flooding, the LUP here includes only the coordinates and the corresponding timestamp of one single node – sending node.

In SLS, the transmission interval was calculated also using formula (5) where α was 4 and $Range_{trans}$ was100m. Nodes broadcasted LUPs in every calculated interval if the interval value was smaller than 13s or in every 13s if the interval was longer than 13s. LUP in SLS includes the coordinates and their corresponding sampling time of more than one node.

As to PLU, $Interval_1$ was calculated using formula (5) where $Range_{trans}$ stayed the same (i.e., 100m) but α was set to 8/3. That is to say, the stage-1 update interval is bigger than the interval employed by SLS. As mentioned in PLU algorithm description in Section 3, $Interval_2 = Interval_1/3$. The other threshold values used in PLU were as follows: `X_CHANGE_THRESHOLD` = `Y_CHANGE_THRESHOLD` = 6m. In order to test the average location error, the real locations of nodes (as generated by ns-2) and the location information predicted were recorded every 2 seconds. The differences between these two sets of values indicate the degree of location errors.

4 Simulation Results and Analysis

The above five LIS algorithms were evaluated in two aspects: 1) performance in terms of the average location errors caused by the node mobility, and 2) overhead in terms of both the number of location updating *packets* broadcasted and the number of *byte* broadcasted for LIS purpose during the simulation period.

Fig. 3 shows the average location error (in meters) of the protocols versus node's speed. It was observed that SLS introduced the largest location errors whereas in most cases (apart from the very beginning) PLU had the smallest average location errors. The average location error increased linearly in flooding and DLS because nodes generated updates (to DLS this means faraway update) at a constant interval. The accuracy of node's location information in DLS and SLS was worse than flooding; this is because most of their LUPs only transmitted to their neighbours. As speed increased, the update interval became shorter in SLS and PLU, which means more LUPs being transmitted and consequently increased location accuracy. Thanks to the prediction mechanism, PLU showed positive evidence in better location accuracy in

comparison with the other protocols. When nodes move in low speed (lower than 5 m/s) update interval in PLU is larger. The less frequency in location updating leads to a bigger average location error as shown in Fig. 3. Fig. 3 demonstrated that PLU can provide constantly good location accuracy when the nodes are moving at a higher speed (regardless how high the speed is).

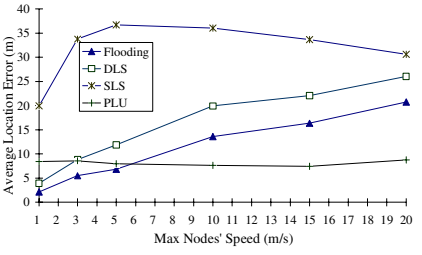


Fig. 3. Average Location Error vs. Node Speed

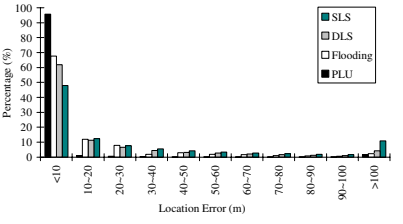


Fig. 4. Histogram of Location Error Distribution

A close evaluation of the location error for each protocol is depicted in Fig. 4, which shows a histogram of the location error introduced by each protocol when maximum average speed is 10 m/sec. It is observed that in PLU, the percentages of location errors less than 10m are higher than 95% and this is much larger than those of the other location services. This picture also illustrates that almost all the location error of PLU is within 30m.

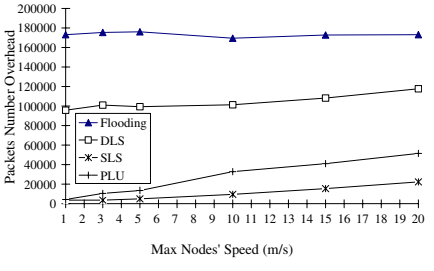


Fig. 5. LUP Packet Overhead vs. Node Speed

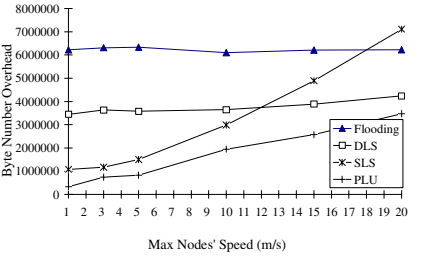


Fig. 6. LUP Byte Overhead vs. Node Speed

Fig. 5 illustrated the packet overhead versus node speed for each algorithm. Theoretically, in flooding the number of LUP packets broadcasted during every update interval equals to n^2 (n is the number of nodes in the network). As shown in Fig. 5, flooding uses the most significant amount of packets. The faraway LUPs in DLS also use flooding scheme, so its packets number is just next less than flooding, but still more than the others. SLS and PLU consume fewer packets due to their one-hop broadcast feature. Given the same location information update interval, PLU uses more LUP packets than SLS because they introduce an extra kind of update.

Fig. 6 describes the byte overhead versus node speed. The byte overhead represents the network bandwidth requirement. The simulation result shown here counted only

these bytes transmitted by routing layer. Actually, an increase in transmitted packet number at network layer means a corresponding increase in number of bytes transmitted at link layer. In flooding and DLS, LUP contains only coordinates and timestamp of *one* single node (i.e., the sending node) so the length of LUP is constant. As a result, the byte number keeps scale with the transmitted packet number. PLU consumes less byte than SLS does. The byte number overhead for SLS and PLU increases when mobile node moves more quickly; this is because these protocols transmit LUPs more frequently when the average speed gets bigger.

5 Conclusion

This paper proposed and evaluated (via simulation) a new location update algorithm called PLU for mobile ad hoc networks. Its high performance comes from the introduction of location prediction and a more constrained one-hop broadcast transmission strategy. Stage-2 updates were introduced into PLU. As a complement to the main location update scheme (stage-1 update), this stage increases PLU's adaptability to the big and quick change of node locations.

Overall speaking, PLU provides improved overall performance (esp. in terms of the average location errors) than other typical protocols evaluated. Our future work is to apply PLU on location aided routing protocols and based on the feedback to carry out further evaluation and improvement.

References

1. L. M. Feeney, "A Taxonomy for Routing Protocols in Mobile Ad Hoc Networks," Swedish Institute of Computer Science Technical Report T99/07, October 1999.
2. Y-B.Ko, N.H.Vaidya. "Location- aided routing (LAR) in mobile ad hoc networks". The ACM/ IEEE Int' l Conf on Mobile Computing and Networking (MOBICOM), Dallas, 1998
3. S. Basagni, I. Chlamtac, V. R. Syrotiuk etal. "A distance routing effect algorithm for mobility (DREAM)." The ACM/IEEE Int' l Conf on Mobile Computing and Networking (MOBICOM), Dallas, 1998
4. Stojmenovic, "Position based routing in ad hoc networks", *IEEE Commmunications Magazine*, Vol. 40, No. 7, July 2002, 128-134.
5. Stojmenovic, "Location Updates for Efficient Routing in Ad Hoc Networks," Handbook of Wireless Networks and Mobile Computing, Wiley, 2002, http://www.site.uottawa.ca/~ivan, pp. 451–71
6. T. Camp, J. Boleng, and L. Wilcox. "Location information services in mobile ad hoc networks". In Proceedings of ICC, 2001.
7. S. H. Shah, and K. Nahrstedt. "Predictive Location-Based QoS Routing in Mobile Ad Hoc Networks". In Proceedings of ICC2002, New York, NY, April 2002.
8. Agarwal and S. R. Das, "Dead Reckoning in Mobile Ad Hoc Networks", Proc. of the 2003 IEEE Wireless Communications and Networking Conference (WCNC 2003), New Orleans, March 2003
9. The CMU Monarch Project. http://www.monarch.cs.cmu.edu/cmu-ns.html
10. T. Camp, J. Boleng, B. Williams, *et al.* "Performance Comparison of Two Location Based Routing Protocols for Ad Hoc Networks". Proc. of InfoCom 2002 pp. 1678-1687.

Combining Power Management and Power Control in Multihop IEEE 802.11 Ad Hoc Networks

Ming Liu[1], Ming T. Liu[1], and David Q. Liu[2]

[1] Department of Computing and Information Science, The Ohio State University, Columbus, OH 43210 USA
{mliu, liu}@cse.ohio-state.edu
[2] Department of Computer Science, Indiana University - Purdue University Fort Wayne, Fort Wayne, IN 46805 USA
liud@ipfw.edu

Abstract. Power-saving is a major topic in mobile network research. MAC and network level power-saving schemes can be divided into two categories. One is power management, which tries to put as many idle stations into sleep mode as possible. The other is power control, whose goal is to minimize the transmission power by reducing the transmitting range. However, the two catagories of schemes do not take advantage of each other. In this paper, we propose an algorithm which integrates and balances both of the two approaches to get maximal power conservation, and closely combines power-saving and routing in the wireless ad hoc networks. Using power consumption figures measured and reported by other researchers, a simulation of power control, power management and the proposed combined algorithms has been set up and results are presented.

1 Introduction

With the popularization of small computing devices such as PDA and laptop computers, mobile computing has gained a lot of interests since the turn of the century. Mobile networks present many new challenges, such as security concerns, higher error rate due to open medium which is more prone to interferences, and energy conservation issues... Among these challenges, power-saving is one of the most acute issues. It has long been proved that the transceiver takes at least half of the power consumed by the whole IEEE 802.11 network interface [1,2]. However, Because working in a totally distributed manner, power conservation proves to be a very tough problem in mobile ad hoc networks (MANET). Besides improving technologies on electronic design and implementation of the RF amplifier, a lot of research has been done in devising algorithms to control transceivers more efficiently to conserve energy. Those algorithms can be divided into two categories. The first category is called power management, which is to shut down the node's transceiver and put it into sleep mode for some period when it appears to be idle. The major challenge for power management is how

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 702–711, 2005.

to keep a station in sleep mode as long as possible while wake it up as soon as it has incoming traffic. IEEE 802.11 standard adopts power management for energy conservation[3]. Many other power-saving schemes attacking the problem using power control belong to the second category. The idea behind it is to adjust transmission power dynamically so that only the minimal power required is used by the RF amplifier. Because of the dynamic of the network topology, how to efficiently find that minimal power level becomes the main problem here.

It is well-known that the transmission power needed to deliver a packet from node A to node B, E_{AB}, is an exponential function of the distance between the two nodes, $dist_{AB}$, i.e. $E_{AB} = \beta \times dist_{AB}^{\gamma}$ with $\gamma > 1$ as the path-loss exponent, which depends on the RF environment, and is generally between 2 and 4 for indoor situations [4]. This usually causes a power control algorithm to pick a path with multiple short hops over a long transmission for better energy conservation. However, both lab test [5] and manufacturer's specifications [6,7] show that receiving and idle states consume considerable amount of energy as well. In Table 1, we compile the energy specifications from a couple typical market leaders' products (Short antenna type II extended PCMCIA IEEE 802.11b wireless LAN card by Agere Systems and Cisco Aironet 350 series wireless LAN adapter by Cisco Systems), plus the lab measurement reported in [5] (a Lucent WaveLAN card was measured). We can see that even receiving and idle still take about 55% to 70% of the energy used during transmitting. So even when transmission distance, $dist_{AB}$, approaches zero, the energy consumption, E_{AB}, should not approaches zero as indicated by function above. For senders, a better approximation on the power consumption at the network interface for transferring a packet from node A to node B should be $E_{AB} = P_{Idle} + \beta \times dist_{AB}^{\gamma}$ (since receiving power does not change with the transmission distance, the receiver's power consumption remains at P_{Recv} and is not the focus of this paper). On the other hand, the power consumption in the sleep mode is in a different order of magnitude as in active mode. It only needs 1/30 of the peak power usage. So it is desirable to combine power control schemes with power management to take the power consumed in receiving and waiting by relaying stations into consideration when choosing a path with optimal transmit power.

This paper focuses on closely integrating power control with IEEE 802.11 power management to maximize energy conservation in a realistic setting. Power management and power control deal with two different aspects of power-saving,

Table 1. WLAN card power consumption comparison

	Specification		Measurement
	Agere	Cisco	Lucent
Sleep mode	9mA	15mA	10mA
Idle	n/a	n/a	156mA
Receive	185mA	270mA	190mA
Transmit	285mA	450mA	284mA
Input Voltage	5V	5V	4.74V

and in most of situations, the optimal conditions which they are seeking conflict with each other. For example, to set up a path from node A to node E, a power management algorithm picks the simple path ACE to get as fewer stations involved as possible. At the same time, a power control algorithm selects nodes B, C and D as intermediate nodes so that short transmissions can be used. The shorter the transmission along the path, the more stations get involved in relaying the traffic. So the two power-saving schemes will not integrate naturally. This paper proposes an algorithm combining and balancing the two different approaches to take advantages from both. From the simulation results, we can see the proposed algorithm successfully achieves that goal.

After this introduction, we will discuss related works in Sect. 2 before presenting the proposed algorithm in Sect. 3. Then, in Sect. 4, simulation results are presented. Finally, we conclude the paper with a short summary and some discussions in Sect. 5.

2 Related Work

Many works have been done on power management for wireless ad hoc networks. First of all, IEEE 802.11 standard adopts a power management scheme for its ad hoc mode [3]. Time is divided into slots. At the beginning of each slot, there is a special period called Announcement Traffic Indication Message (ATIM) window when all stations stay awake to decide which have data to exchange in the upcoming slot and which don't. Those who do not involve in data traffic go to sleep or power-saving (PS) mode to conserve energy. In the sleep mode, most components of the WLAN interface including transceiver are turned off, so the power consumption of this mode is extremely low. However, as pointed out in [8], IEEE 802.11 standard doesn't take multihop scenarios into consideration. So most of the power management research try to solve the problems arisen in a multihop environment (mainly synchronization problem) [9,10], and they all assume stations throughout the system always use the same or full power to transmit.

A routing algorithm with a goal of energy conservation using power control is given in [11]. For each pair of neighbors, transmission power required for a successful delivery is tracked based on the distance between them and a two-ray propagation model. This power is then used as the weight of the link to participate in path selection and redirection. Stations overhear ongoing transmissions and calculate if it is better to redirect some transmission through themselves. If it is better to redirect, this station sends a redirecting message, and in the next slot, the path going through it will replace the old one. However, the authors have not considered any power management procedure. The overhearing and redirecting result multiple short hops to replace a long path in which more intermediate stations are involved in relaying packets.

In [1], the authors study the effects of transmission power/range on the energy per bit successfully transmitted and on the network capacity. The simulations show energy conservation improving as transmission range reducing. However,

those simulations are done without any power management implemented. The authors also find that reducing transmission power has a reverse effect on network capacity, even though smaller transmission range means less multiple access collision which should introduce a capacity gain. It turns out that this capacity reduction is caused by the increased number of hops/transmissions per packet.

Another group of power-aware routing schemes use cluster-based approach [12,13]. These schemes create a hierarchical structure in the system, and use only a subset of the stations to handle the relaying of the traffic. They work well in a network where some nodes are more resourceful than others in terms of computing and power. However, since we consider a general wireless ad hoc network in this paper, even though stations could be heterogeneous, the randomness usually inherited in this kind of networks still leads us to take a non-clustered, pure ad hoc approach.

Many proposed power-aware routing algorithms also take advantage of the information about station's battery capacity and/or remaining power [14,15,16]. In this paper, we do not take those information into account. However, as discussed in Sect. 5, we can easily adopt it into the proposed algorithm.

3 Combining Power Control and Power Management in Multi-hop Ad Hoc Networks

In this section, we are going to propose a power-saving scheme which combines both IEEE 802.11 power management and dynamic transmission power control.

In IEEE 802.11 standard, at the beginning of each slots, all stations must stay awake to exchange control packets announcing pending data packets. Those control packets work as a hand-shaking mechanism for the senders and receivers. Those stations which succeed in their hand-shaking process remain awake for data transmission, while the rest go to PS mode until the next slot. From Table 1, we can see that PS mode saves power much more significantly than reducing transmission power. So in the proposed algorithm, the priority is given to power management procedure, i.e., only those stations which stay active according to the power management procedure will try to further conserve energy using power control. Also, for example, a transmission from O to C which is in the transmission range of O, it will be a direct one-hop transfer. Although there is another node A inside O's range, and the fact that $E_{OA} + E_{AC} < E_{OC}$ makes point A a good relaying candidate according to power control algorithms, to avoid delay and congestion inspired by relaying, a direct link is preferred over multi-hop link when available.

When a direct link is unavailable, i.e., a multihop transmission is inevitable, the source node will try to find the next-hop node according to the local information that it has at that time. We do not use the overhearing-redirecting scheme presented in [11] for several reasons:

– Overhearing-redirecting scheme is based on a system without power management. However, in a system with power management, the active neighbor

nodes are different and unpredictable from slot to slot. So the overhearing node might not participate in transmission at all (in PS) in the next slot.
- Overhearing-redirecting scheme use many packet exchanges for updating routing information. As pointed out in [1,5], these extra control packet exchanges are not good for conserving energy. So in the proposed scheme, it is sender's responsibility to pick the next-hop nodes among its neighbors.
- It takes multiple rounds to converge to an energy efficient route in [11]. This not only takes time but also requires each node to maintain a routing table. In an environment where node mobility is high or packet exchanges are short and sporadic, this convergence time may become too much a price to pay.

For any active neighbor N of sender O, the one can reach destination E with the minimal transmission power shall be selected as the next-hop node from O to E, i.e., find N which satisfies

$$min\left\{E_{ON} + E_{NN_1} + \sum_{i=1}^{h-1} E_{N_i N_{i+1}} + E_{N_h E} | N \in \mathcal{N}_A^s(O)\right\}$$

where E_{ON} is the transmission power needed for transferring a packet from O to $N, N_1, N_2, N_3, \ldots, N_h$ represent intermediate nodes in the path from the next-hop node N to final destination E, and $\mathcal{N}_A^s(O)$ denotes the active neighbor set of node O in slot s. However, the active neighbor set of a node changes in each slot, it is impossible to find a pre-defined route from O to E. A route has to be decided hop by hop on the fly. This has two implications on route selection. First, the route is not strictly optimal based on energy consumption, since it is impossible to find a shortest path in a graph which keeps changing. Second, the selection of next-hop N has to be based on estimation. The power needed for a direct transmission from N to E[1], E_{NE}, is used in place of the transmission energy of a path from N to E ($E_{NN_1} + \sum_{i=1}^{h-1} E_{N_i N_{i+1}} + E_{N_h E}$). This is a tradeoff to reduce the complexity of the algorithm and make it rely only on local information.

As mentioned above, in order to avoid wasting energy on small control packets like route update packets, next-hop node is chosen by the sender using local information only. This might represent an extra computational burden to the sender, especially in a dense system. So we try to reduce the number of neighbors the sender has to consider when finding the optimal next-hop node. Go back to the relationship of transmission power and range given in Sect. 1, for $\gamma = 2$, the next-hop nodes should be those satisfy $dist_{ON}^2 + dist_{NE}^2 < dist_{OE}^2$. From basic geometric knowledge, those candidate nodes are in the intersection of the transmission range of O and a circle centered at E with $dist_{EO}$ as radius. By only considering the neighbors falling in this region for the next-hop candidate, computations of the senders can be reduced. For higher γ value, the candidate region should be bigger than this, but it won't be significantly larger.

Summarize the points we have talked above, the proposed algorithm is described in Fig. 1.

[1] Even if transmission range prevents such a direct transfer between N and E, we approximate it by still using the function of power and transmission distance with $dist_{NE}$ as if it were a direct transmission.

```
for each slot
  for each node O which has pending traffic
    Mark current node as active node
    if the destination D is one of the neighbors then
      Choose destination as next-hop node, mark it active and send TIM packet
    else
      for each active node N in the neighbor
        if d²_ON + d²_ND < d²_OD then
          Find the N which gives the smallest d²_ON + d²_ND and pick it as next-hop node, mark
active, send TIM packet
      if couldn't find next-hop node among active neighbors then
        for the rest of the neighbor N
          if d²_ON + d²_ND < d²_OD then
            Find the N which gives the smallest d²_ON + d²_ND and pick it as next-hop node, mark
active, send TIM packet
```

Fig. 1. Proposed power-saving algorithm

4 Simulations and Results

In this section, simulations[2] are done to demonstrate the efficiency of the proposed algorithm compared with power management algorithm by IEEE 802.11 and a power control algorithm. The simulations show the algorithm works well and take the advantages of both power management and power control. Let's take a detailed look of the simulations and have a discussion about the results.

In each simulation, 200 stations with a transmitting range of 30 are evenly distributed in a square area of 100 by 100. Node movement is not considered in this paper since we think how the scheme handles the mobility relies on the routing scheme rather than the power-saving functions. Random traffic composed of one packet or a burst of packets that enough to be filled in a slot per transmission is used, and all algorithms use the same traffic in order to make comparisons more precise. Also, to highlight the effect of different power-saving schemes and make result analysis easier, the simulation does not include MAC functions like contention avoidance and error control. It is assumed that all packets stored in each station will be delivered successfully to the next-hop stations by the end of each slot. As described in Sect. 2, we use a special ATIM window in the power management algorithm for transferring paging packets to let the receiver know it has pending packets, and the nodes which have not received any such packet, nor have any outgoing packets will go to sleep mode until the next beacon interval. For multihop routing, a globally optimized hop-count-based SPF routing is used. The power control algorithm routes with a globally SPF routing using the square of the distance of the link as link weight, and use a reduced transmission power according to the how far the receiver is. Even in this power control algorithm, we put nodes which are not participating sending, receiving, or relaying packets into sleep mode to save energy. The combined algorithm follows the description

[2] The current version of popular network simulator, *ns-2*, supports neither power management nor power control, so we decide to write our own simulation program rather than modifying *ns-2*.

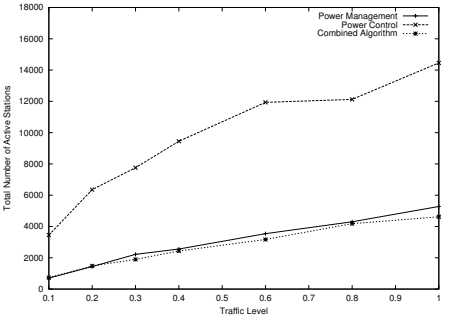


Fig. 2. Comparison of the number of active stations involved in traffic

in Sect. 3 using a reduced transmission power and a local routing scheme taking power management into consideration.

Because power control scheme prefers multiple short hops over single long one, it will have more stations involved in transmitting/forwarding than the power management, and we can see this clearly in Fig. 2 which counts the number of nodes involved in traffic for different schemes. The numbers shown in this figure is the sum of active stations count in each slot throughout our simulation interval. Since the combined algorithm always tries to route packets to an already active node, it performs even a little bit better than power management algorithm, especially when the traffic is heavy. However, since power management scheme uses a globally optimized hop-based SPF routing, the improvement brought by the combined algorithm is very limited. In the real world, because globally optimized routing is hardly achievable, the power management algorithms will have worse performance than presented here.

Another advantage of power management scheme over power control is less hops for each packet to get to the destination. Figure 3 shows the average hops per packet. We can see the proposed scheme shows a performance close to power management scheme which represents the lower bound on the hop count. Less hops means shorter delay which could be very important for some applications.

The goal of power control algorithms is to reduce the transmission range. We can see in Fig. 4, it has a very low numbers, and power management algorithm stays in a straight line since it always uses the maximum power (Please note that in this figure, the y axis represents the *square* of transmission distance, since the transmission power is a linear function of the square of that distance. So though maximum power in our simulation is 30, it shows in figure as 900). Since all three algorithms handle the exactly same traffic, the average transmitting distance of each transmission is closely related with the number of hops it takes to reach the final destination, so the combined algorithm runs between the power management and the power control.

Based on the relationship between transmission power consumed and corresponding transmission range and WLAN card power consumption specifications given in Sect. 1, by plugging in the measured number of Lucent's WaveLAN in

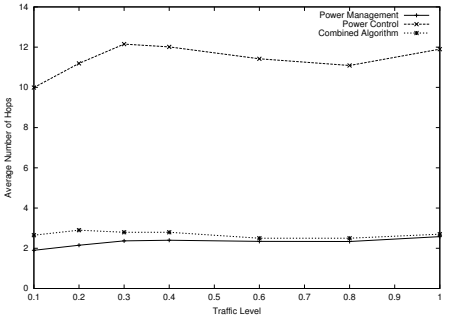


Fig. 3. Comparison of average number of hops needed for a packet to be delivered to its final destination

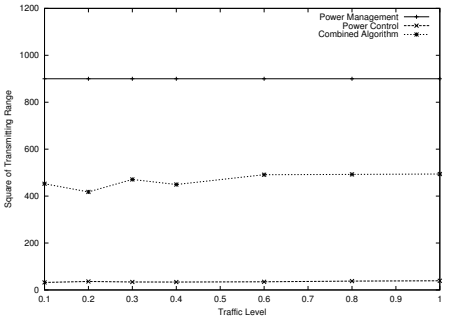


Fig. 4. Comparison of the average of the square of transmitting distance of each transmission

Table 1 to get the coefficient β, we devise this simplified function for the total power consumption of a WLAN card at the transmissions and the resulting transmitting range

$$Power\_Tx = 156 \times T_{sending} + 0.14222 \times d_{tran}^2$$

and the corresponding power consumption at the receiving stage is

$$Power\_Rc = 190 \times T_{receiving}$$

where $T_{sending}$ and $T_{receiving}$ are the time spend in sending and receiving stages respectively. Using this function as a guideline, the power consumed by the LAN card is shown in Fig. 5 for different schemes based on simulation results. Because the power control can only control 45% of maximum power, power management delivers a better performance by putting as many stations into sleep mode as possible, especially when the traffic is low. However, the higher traffic level is, the more likely the relaying traffic can tag along the transmitting stations' own transmissions when the power management looses its advantage although it still

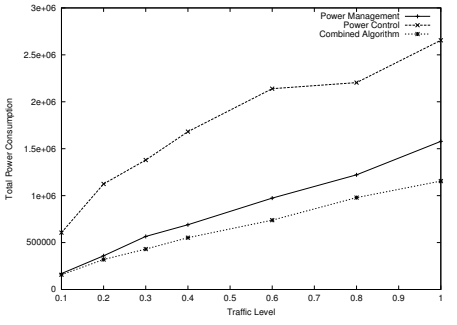


Fig. 5. Comparison of total power consumption of the WLAN card

bests the power control scheme. For the combined algorithm, just like in Fig. 2, the heavier the traffic is, the clearer it is the best one among all three algorithms. Although it is not shown here the situation for different node densities, we can expect that with a higher node density, the combined algorithm will also show a bigger improvement over power management scheme.

5 Conclusions

In this paper, a power-saving scheme which combines the advantages from both power management and power control is presented. It elegantly integrates the two main approaches for power-saving, incorporates both reducing transmission power and putting idle stations into sleep mode to achieve the maximum power-saving, and closely combines power-saving and routing in a ad hoc network. Our simulation shows that the proposed algorithm has a better performance over both power control and power management algorithms in a more realistic environment. As we point out in Sect. 2, several researchers present works which take the remaining battery power into consideration when picking relaying stations. Although we do not include that information in the proposed algorithm, it can be added easily provided that such information is made available by the system physical hardware. Just like power control, this should be considered after power management has been done.

References

1. Monks, J.P., Ebert, J.P., Wolisz, A., Hwu, W.W.: A study of the energy saving and capacity improvement potential of power control in multi-hop wireless networks. In: Proceedings of the 26th Annual IEEE Conference on Local Computer Networks (LCN'01). (2001) 550–559
2. Ebert, J.P., Burns, B., Wolisz, A.: A trace-based approach for determining the energy consumption of a wlan network interface. In: Proceedings of European Wireless 2002. (2002) 230–236

3. IEEE 802.11 Working Group: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. (1999)
4. Rabaey, J.M., Ammer, M.J., da Silva Jr., J.L., Patel, D., Roundy, S.: Picoradio supports ad hoc ultra-low power wireless networking. IEEE Computer **33** (2000) 42–48
5. Feeney, L.M., Nilsson, M.: Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In: Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies 2001 (INFOCOM'01). (2001) 219–228
6. Agere Systems Inc.: Agere Systems Wireless LAN World PC Card (Extended) Data Sheet. (2002) http://www.agere.com/client/docs/DS02115-1.pdf.
7. Cisco Systems Inc.: Cisco Aironet 350 Series Client Adapters Data Sheet. (2003) http://www.cisco.com/warp/public/cc/pd/witc/ao350ap/prodlit/a350c_ds.pdf.
8. Liu, M., Liu, M.T.: A power-saving scheduling for ieee 802.11 mobile ad hoc network. In: Proceedings of 2003 International Conference on Computer Networks and Mobile Computing (ICCNMC 2003). (2003) 238–245
9. Jiang, J.R., Tseng, Y.C., Hsu, C.S., Lai, T.H.: Quorum-based asynchronous power-saving protocols for ieee 802.11 ad hoc networks. In: Proceedings of 2003 International Conference on Parallel Processing (ICPP 2003). (2003) 257–264
10. Ye, W., Heideman, J., Estrin, D.: An energy-efficient mac protocol for wireless sensor networks. In: Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies 2002 (INFOCOM 2002). Volume 1. (2002) 200–209
11. Gomez, J., Campbell, A.T., Naghshineh, M., Bisdikian, C.: Conserving transmission power in wireless ad hoc networks. In: Proceedings of the Ninth International Conference on Network Protocols (ICNP 2001). (2001) 24–34
12. Sinha, P., Sivakumar, R., Bharghavan, V.: Cedar: a core-extraction distributed ad hoc routing algorithm. In: Proceeding of the 18th Annual Joint Conference of the IEEE Computer and Communications Societies 1999 (INFOCOM'99). (1999) 202–209
13. Chen, B., Jamieson, K., Balakrishnan, H., Morris, R.: Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In: Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking (MobiCom'01). (2001) 85–96
14. Tragoudas, S., Dimitrova, S.: Routing with engergy considerations in mobile ad-hoc networks. In: Proceedings of 2000 IEEE Wireless Communications and Networking Conference (WCNC 2000). Volume 3. (2000) 1258–1261
15. Michail, A., Ephremides, A.: Energy efficient routing for connection-oriented traffic in ad-hoc wireless networks. In: Proceedings of the 11th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2000). Volume 2. (2000) 762–766
16. Kim, D., Garcia-Luna-Aceves, J., Obraczka, K., Cano, J.C., Manzoni, P.: Power-aware routing based on the energy drain rate for mobile ad hoc networks. In: Proceedings of the Eleventh International Conference on Computer Communications and Networks (ICCCN 2002). (2002) 565–569

Minimum Disc Cover Set Construction in Mobile Ad Hoc Networks

Min-Te Sun[1], Xiaoli Ma[1], Chih-Wei Yi[2], Chuan-Kai Yang[3], and Ten H. Lai[4]

[1] Wireless Engineering Research and Education Center, Auburn University
{sunmint, maxiaol}@eng.auburn.edu
[2] Department of Computer Information Science, National Chiao Tung University
cwyi@cis.nctu.edu.tw
[3] Department of Information Management,
National Taiwan University of Science and Technology
ckyang@cs.ntust.edu.tw
[4] Department of Computer Science and Engineering, Ohio State University
lai@cse.ohio-state.edu

Abstract. The *minimum disc cover set* can be used to construct the dominating set on the fly for energy-efficient communications in mobile ad hoc networks. The approach used to compute the minimum disc cover set proposed in previous study has been considered too expensive. In this paper, we show that the disc cover set problem is in fact a special case of the general α-hull problem. In addition, we prove that the disc cover set problem is not any easier than the α-hull problem by linearly reducing the element uniqueness problem to the disc cover set problem. In addition to the known α-hull approach, we provide an optimal divide-and-conquer algorithm that constructs the minimum disc cover set for arbitrary cases, including the degenerate ones where the traditional α-hull algorithm incapable of handling.

1 Introduction

Finding a minimum cover is an interesting combinatorial problem. Many well-known problems, such as *Vertex Cover*, *Dominating Set*, *Covering by Cliques*, and *Set Minimum Cover* [1], can be viewed as such a problem. In this paper, we are interested in a specific minimum cover problem which has been formulated in [2].

Let $\Delta = \{D_0, D_1, D_2, \ldots, D_n\}$ be a set of discs of radius R with all their origins located inside D_0. Given Δ, the *disc cover set problem* is to find a minimum subset of Δ, say Δ', such that the union of the discs in Δ' is equal to the union of the discs in Δ. The minimum disc cover set problem has applications in inter-vehicle communications [3], broadcast in location-based mobile ad hoc networks [4] [5] [6] [7], and multicast medium access control [8]. For instance, assume all mobile stations transmit data wirelessly at the same distance. If we denote a mobile station as s and its immediate neighbors as $N[s]$ with s itself included, the effect of having all neighbors broadcast a message is theoretically the same

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 712–721, 2005.

(i.e., cover the same area) as having only the members in the minimum cover set of the coverage areas (discs) associated with $N[s]$ broadcast the message. If the minimum cover set can be constructed efficiently, a node can simply instruct only neighbors in the minimum disc cover set retransmit the broadcast message. This simple broadcast protocol effectively reduces the number of retransmissions and alleviates the channel contention and message collisions caused by massive broadcast retransmissions.

Contrary to the other minimum cover combinatorial problems mentioned above, the disc cover set problem *can* be solved in polynomial time. In [2], a Graham-scan based algorithm is provided that constructs the minimum disc cover set in $\mathcal{O}(n^{4/3})$ time complexity, where n is the number of discs (or participating neighbors). As mentioned earlier, most of the known applications of the minimum disc cover set problem lie in the area of mobile ad hoc networks, where the computing power and battery of each node are limited. Hence, the $\mathcal{O}(n^{4/3})$ time complexity is still considered expensive and inefficient.

In this paper, we propose two methods for the minimum disc cover set problem. The first method involves the reduction of our covering problem into another problem — the *α-hull problem* [9] — and then solves the latter using an existing algorithm. This method needs to assume that no more than three disc origins fall at the circumference of a disc. The second method solves the minimum disc cover set problem directly using a simple divide-and-conquer strategy. It works conveniently without the need of the previous assumption. Both methods have a time complexity of $\mathcal{O}(n \log n)$. As the last contribution of this paper, we show that any algorithm that solves the minimum disc cover set problem needs at least $\mathcal{O}(n \log n)$ time in the worst case, thereby establishing the optimality of both our methods.

2 Definition of Minimum Disc Cover Set and Its Connection to α-Hull

The disc cover set problem is originated from mobile ad hoc networks. In such networks, each node (i.e., mobile station) is usually assumed to have the same transmission power, thus assumed the same transmission radius. If the transmission radius is denoted as R, the coverage area of a node i can be modeled as a disc D_i with radius R centered at the location of the node. If the location of node i is known as p_i, the corresponding disc D_i can also be referred as $D(p_i)$. If the locations of node i and j are p_i and p_j, the Euclidean distance of i and j is denoted as either $dist(i, j)$ or $dist(p_i, p_j)$. Using the above denotations, the minimum disc cover problem can be formally defined as follows.

Definition 1. Minimum Disc Cover Set Problem
INSTANCE: *A set of disc* $\Delta = \{D_0, D_1, D_2, \ldots, D_n\}$ *such that* $\forall i$ $dist(0, i) < R$.
QUESTION: *Find a subset* $\Delta' \subseteq \Delta$ *such that* $\|\Delta'\|$ *is minimum and*

$$\bigcup_{D_i \in \Delta} D_i = \bigcup_{D_j \in \Delta'} D_j$$

Notice that D_0 may also be included in the minimum disc cover set.

On the other hand, the α-hull (for negative α value) for an arbitrary set of points S on the two-dimensional plane is defined as the intersection of all closed complements of discs (with radius $-1/\alpha$) that contain all the points in S. The shape of the α-hull for S is like an inward curved convex hull, where each edge of the hull is an arc (of a circle with radius $-1/\alpha$). A good way to visualize the shape of such a α-hull is to imagine the points in S as nails fixed on the two-dimensional plane and a steel circle with radius $-1/\alpha$ rolled around the nails to obtain the outer contour of S. In Figure 1, the shape of the α-hulls for points in the figure is denoted by the dash line.

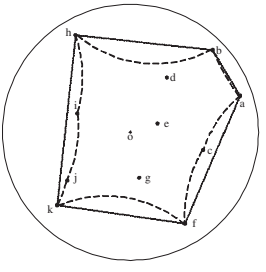


Fig. 1. α-hull (in dash line) for a given set of points

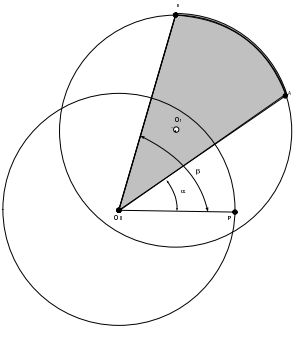


Fig. 2. The shaded area is covered by $D(O_1)$

The following theorem shows a simple connection between the minimum disc cover set and the α-hull.

Theorem 1. *Let S be the set of disc origins for a given disc set problem, the collection of vertices in the α-hull for S, where $\alpha = -1/R$, forms the minimum disc cover set.*

Due to the page limit, we are unable to give proof for Theorem 1 in the paper. However, interested readers could refer to the complete version of this paper in [10].

3 α-Hull Algorithm for the Minimum Disc Cover Set

In Section 2, we have established the connection between the minimum disc cover problem and the α-hull problem. In [9], the following algorithm is provided to construct the α-hull in optimal time complexity $\mathcal{O}(n \log n)$.

Algorithm. α-hull($S = \{O_0, O_1, O_2, \ldots, O_n\}$)
Construct Delaunay Triangulations
Determine the α-extreme points of S
Determine the α-neighbors of S
Output the α-hull

Because the Delaunay Triangulations contain all the information necessary for the second and third statements, this algorithm efficiently computes the vertices of the α-hull and at the same time the minimum disc cover set is obtained. The correctness and analysis of this algorithm can be found in [9]. However, there are two pieces of information missing here. First, since the very first step of the above algorithm is to compute the Delaunay Triangulations, it fails on cases when there are more than 3 disc origins falling on the circumference of a circle. Second, while this algorithm has been shown to be optimal, it does not tell us if the algorithm is still optimal when all points in S fall in one single disc (i.e., $D(O_0)$ in our case). In other words, we do not know if there is a better α-hull algorithm with time complexity lower than $\mathcal{O}(n \log n)$ when our extra constraint is present. In Section 4, we will first show the relationship between the minimum disc cover and the boundary of disc union. This relationship can then be used to design an intuitive divide-and-conquer algorithm that works for all cases, including the degenerate ones. In Section 5, we will show that there is no algorithm capable of computing minimum disc cover set in time complexity lower than $\mathcal{O}(n \log n)$.

4 Alternative Algorithm for the Minimum Disc Cover Set

4.1 Theoretical Basis of Our Algorithm

We state the following simple, yet useful fact as lemma for ease of reference. Figure 2 illustrates the situation.

Lemma 1. *Let O_0 and O_1 be a pair of points with $dist(O_0, O_1) < R$, and let $\overset{\frown}{AB}$ be an arc on the boundary of $D(O_1)$ which is outside the area $D(O_0)$. The area surrounded by $\overset{\frown}{AB}$, $\overline{AO_0}$, and $\overline{BO_0}$ is completely covered by $D(O_1)$.*

Now, consider the union of discs $\bigcup_{O_i \in S} D(O_i)$. As illustrated in Figure 3, the area $\bigcup_{O_i \in S} D(O_i)$ is bounded by a series of arcs, say ($\overset{\frown}{A_1A_2}$, $\overset{\frown}{A_2A_3}$, ..., $\overset{\frown}{A_{k-1}A_k}$, $\overset{\frown}{A_kA_{k+1}}$), where $A_1 = A_{k+1}$. Then, let Sector($\overset{\frown}{PQ}$, $\overline{PO}$, $\overline{QO}$) be the region bounded by $\overset{\frown}{PQ}$, $\overline{PO}$, and $\overline{QO}$, $\bigcup_{O_i \in S} D(O_i)$ can be viewed as the union of Sector($\overset{\frown}{A_iA_{i+1}}$, $\overline{A_iO_0}$, $\overline{A_{i+1}O_0}$), $1 \leq i \leq k$. In the following lemma, we show that the discs that contribute arcs $\overset{\frown}{A_iA_{i+1}}$ ($1 \leq i \leq k$) form a minimum disc cover set.

Lemma 2. *Let* $\overset{\frown}{A_1A_2}$, $\overset{\frown}{A_2A_3}$, ..., $\overset{\frown}{A_{k-1}A_k}$, $\overset{\frown}{A_kA_{k+1}}$ *be the arcs surrounding* $\bigcup_{O_i \in S} D(O_i)$, *where* $A_1 = A_{k+1}$; *and for* $1 \leq i \leq k$, *let* $D(O_{t_i})$ *be the disc that contributes* $\overset{\frown}{A_iA_{i+1}}$. *The discs,* $\{D(O_{t_1}), D(O_{t_2}), \ldots, D(O_{t_k})\}$, *form the minimum disc cover set of discs with origins in* $S = \{O_0, O_1, O_2, \ldots, O_n\}$.

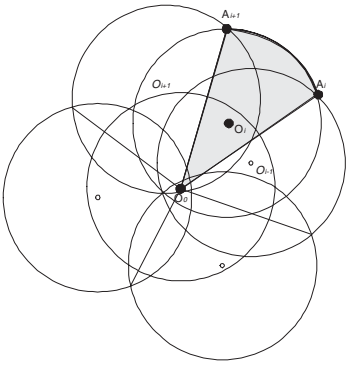


Fig. 3. The boundary of $\bigcup_{O_i \in S} D(O_i)$ is formed by a series of arcs $<\overset{\frown}{A_iA_{i+1}}>$

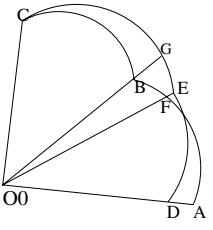


Fig. 4. Splitting $\overset{\frown}{AB}$ into $\overset{\frown}{AF}$ and $\overset{\frown}{FB}$, and splitting $\overset{\frown}{EC}$ into $\overset{\frown}{EG}$ and $\overset{\frown}{GC}$

Notice that a disc $D(O_{t_i})$ may contribute more than one arc to the boundary of $\bigcup_{O_i \in S} D(O_i)$. That implies that the same disc $D(O_{t_i})$ may appear more than once in the disc cover set $\{D(O_{t_1}), D(O_{t_2}), \ldots, D(O_{t_k})\}$. (In fact, a disc may contribute either zero, one, or two arcs to the boundary of $\bigcup_{O_i \in S} D(O_i)$.) The possibility of such multiple appearances is the main reason why the minimum disc cover set is nontrivial. This will be further discussed in Section 4.5.

4.2 Abstract of Algorithm

According to Lemma 2, finding the minimum disc cover set for discs with origins in $S = \{O_0, O_1, \ldots, O_n\}$ is equivalent to finding the arcs that make up the boundary of $\bigcup_{O_i \in S} D(O_i)$. We develop an efficient algorithm that identifies these arcs.

Since the boundary of the union of a collection of discs is formed by arcs, we represent the boundary by a list of arcs. Given two boundaries (i.e., two arc lists), if we are able to efficiently find the boundary of the union, which is again a list of arcs, we can come up with a divide-and-conquer algorithm for the computation of the minima disc cover set. The abstract of the algorithm is as follows.

Algorithm. Boundary($\{O_0, O_1, O_2, \ldots, O_n\}; i, j$)
  if $i = j$
    return the boundary of $D(O_0) \cup D(O_i)$
  else
    $m := \lfloor (i + j)/2 \rfloor$)
    $ArcList_1$ = Boundary($\{O_0, O_1, O_2, \ldots, O_n\}; i, m$)
    $ArcList_2$ = Boundary($\{O_0, O_1, O_2, \ldots, O_n\}; m + 1, j$)
    return ArcMerge($ArcList_1$, $ArcList_2$)

Basically **Boundary**($\{O_0, O_1, O_2, \ldots, O_n\}; i, j$) returns the boundary (represented by a list of arcs) of $D(O_0) \cup \bigcup_{i \le t \le j} D(O_t)$. Note that $D(O_0)$ is always present and $1 \le i \le j \le n$. In the rest of the paper, we refer $D(O_0) \cup \bigcup_{i \le t \le j} D(O_t)$ as super disc $U_D(i, j)$. To compute the minimum disc cover set for all discs with origins in S, the algorithm is invoked by **Boundary**($\{O_0, O_1, O_2, \ldots, O_n\}; 1, n$) (i.e., to compute super disc $U_D(1, n)$).

It is a divide-and-conquer algorithm. In order to find the boundary of $Superdisc(i, j) = D(O_0) \cup D(O_i) \cup D(O_{i+1}) \cup \cdots \cup D(O_j)$, we find the boundary of super disc $U_D(i, m) = D(O_0) \cup D(O_i) \cup D(O_{i+1}) \cup \cdots \cup D(O_m)$ and that of super disc $U_D(m+1, j) = D(O_0) \cup D(O_{m+1}) \cup D(O_{m+2}) \cup \cdots \cup D(O_j)$, and then combine the two boundaries, where each boundary is a list of arcs. Evidently, the core of the algorithm is the merger of two arc lists.

4.3 Representation of Arc Lists

An arc can be characterized by three parameters, (O_t, α, β), where O_t is the origin of the disc that contributes the arc, and α and β are the degrees of two angles, which are depicted in Figure 2) and formally defined as follows:

$$\alpha = \begin{cases} cos^{-1}\left(\frac{\overrightarrow{O_0A} \cdot (1,0)}{\|\overrightarrow{O_0A}\|}\right), \text{ if } (1,0) \times \overrightarrow{O_0A} \ge 0 \\ cos^{-1}\left(\frac{\overrightarrow{O_0A} \cdot (1,0)}{\|\overrightarrow{O_0A}\|}\right) + \pi, \text{ otherwise} \end{cases} \quad (1)$$

and

$$\beta = \begin{cases} cos^{-1}\left(\frac{\overrightarrow{O_0B} \cdot (1,0)}{\|\overrightarrow{O_0B}\|}\right), \text{ if } (1,0) \times \overrightarrow{O_0B} \ge 0 \\ cos^{-1}\left(\frac{\overrightarrow{O_0B} \cdot (1,0)}{\|\overrightarrow{O_0B}\|}\right) + \pi, \text{ otherwise} \end{cases} \quad (2)$$

Notice that the angles of the arc are obtained by using O_0 as a reference point, instead of the origin of the disc contributing the arc. In Figure 2, $\overline{O_0P}$ is parallel to the x-axis.

The boundary of a super disc is thus a list of $Arc(O_{s_i}, \alpha_{s_i}, \beta_{s_i})$, $0 \le i \le$ max. For ease of discussion, if there is an arc $Arc(O_k, \alpha_k, \beta_k)$ in the list that spans across 360° (i.e., $\alpha_k < 360°$ and $\beta_k > 360°$), we split it into two arcs $Arc(O_k, 0°, \beta_k)$ and $Arc(O_k, \alpha_k, 360°)$. If the list is sorted based on the value of

the first angle (i.e., α_i) in ascending order, then $\beta_{s_i} = \alpha_{s_{i+1}}$, $0 \le i \le \max$, with the understanding that $i+1$ is computed module max. Thus, the boundary of a super disc can be represented simply as $(O_{s_0}, \alpha_{s_0}, O_{s_1}, \alpha_{s_1}, \ldots, O_{s_{\max}}, \alpha_{s_{\max}})$, where $0° = \alpha_{s_0} < \cdots < \alpha_{s_{\max}} < 360°$.

4.4 Merging Two Arc Lists

Now suppose we are given two sorted arc lists

$$\begin{cases} ArcList_1 = (O_{s_0}, \alpha_{s_0}, O_{s_1}, \alpha_{s_1}, \ldots, O_{s_{\max}}, \alpha_{s_{\max}}) \\ ArcList_2 = (O_{t_0}, \alpha_{t_0}, O_{t_1}, \alpha_{t_1}, \ldots, O_{t_{\max}}, \alpha_{t_{\max}}), \end{cases}$$

which represent the boundaries of super disc $U_D(i,m)$ and $U_D(m+1,j)$, respectively. We want to *merge* the two arc lists so that the resulting list represents the boundary of $U_D(i,j) = U_D(i,m) \cup U_D(m+1,j)$.

The first step is to *split* the arcs in each list into smaller arcs, if necessary, so that the two refined arc lists share the same sequence of angles (i.e., α's). As illustrated in Figure 4, $\overset{\frown}{AB}$ is split into $\overset{\frown}{AF}$ and $\overset{\frown}{FB}$, and $\overset{\frown}{EC}$ is split into $\overset{\frown}{EG}$ and $\overset{\frown}{GC}$. After these two splits, the corresponding pair arcs of arc list $\{\overset{\frown}{AF}, \overset{\frown}{FB}, \overset{\frown}{BC}\}$ and arc list $\{\overset{\frown}{DE}, \overset{\frown}{EG}, \overset{\frown}{GC}\}$ share the same angle span with respect to the reference point O_0. Additionally, for instance, if

$$\begin{cases} ArcList_1 = (O_{s_0}, 0°, O_{s_1}, 30°, O_{s_2}, 140°, O_{s_3}, 240°) \\ ArcList_2 = (O_{t_0}, 0°, O_{t_1}, 120°, O_{t_2}, 240°), \end{cases}$$

they will be refined to

$$\begin{cases} ArcList'_1 = (O_{s_0}, 0°, O_{s_1}, 30°, O_{s_1}, 120°, O_{s_2}, 140°, O_{s_3}, 240°) \\ ArcList'_1 = (O_{t_0}, 0°, O_{t_0}, 30°, O_{t_1}, 120°, O_{t_1}, 140°, O_{t_2}, 240°). \end{cases}$$

Now, returning to the general case, suppose we are given two arc lists. After splitting, both lists should contain the same number of arcs:

$$\begin{cases} ArcList_1 = (O_{s_0}, \alpha_{s_0}, O_{s_1}, \alpha_{s_1}, \ldots, O_{s_k}, \alpha_{s_k}) \\ ArcList_2 = (O_{t_0}, \alpha_{s_0}, O_{t_1}, \alpha_{s_1}, \ldots, O_{t_k}, \alpha_{s_k}), \end{cases}$$

By *merging* the two lists, we simply *merge* each individual pair of corresponding arcs, $Arc(O_{s_i}, \alpha_i, \alpha_{i+1})$ and $Arc(O_{t_i}, \alpha_i, \alpha_{i+1})$ as follows.

Case 1: $Arc(O_{s_i}, \alpha_i, \alpha_{i+1})$ and $Arc(O_{t_i}, \alpha_i, \alpha_{i+1})$ have no intersection. As illustrated in the left side of Figure 5, one arc is closer to O_0 than the other. By merging the two arcs, we means dropping the arc close to O_0, keeping only the farther one. In Figure 5, merging the two arcs in the left diagram will result in arc $\overset{\frown}{AB}$.

case 2: $Arc(O_{s_i}, \alpha_i, \alpha_{i+1})$ and $Arc(O_{t_i}, \alpha_i, \alpha_{i+1})$ intersect at a point, say E. (See Figure 5 for illustration.) In this case, we drop the two inner sub-arcs and keep the two external ones. Thus, using the example of the right side of Figure 5, merging arc $\overset{\frown}{AB}$ and $\overset{\frown}{CD}$ will result in arc $\overset{\frown}{CE}$ and $\overset{\frown}{EB}$.

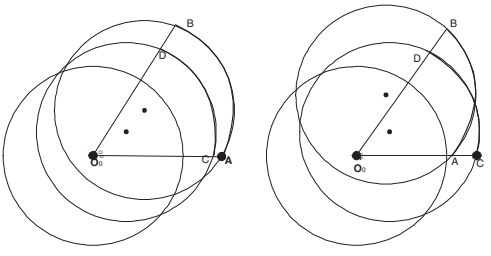


Fig. 5. Case 1 and case 2 when two arcs with the same angle span are merged

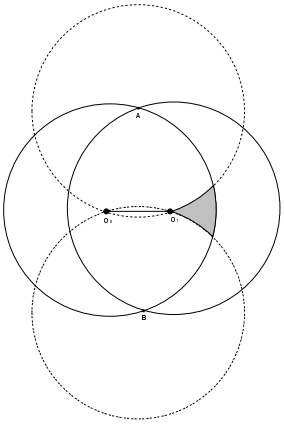


Fig. 6. The possible area for the origin of a third disc that will break $\overset{\frown}{AB}$ into two

4.5 Time Complexity of the Proposed Algorithm

In the algorithm, the base case is to compute the arcs for two intersected discs. Since the location of the disc origins and radius R are known, the computation and arc sorting can be accomplished in constant time. In addition, when merging two arcs with the same angle span, any case mentioned in Section 4.4 takes only constant time.

We claim that the divide-and-conquer algorithm we proposed is of time complexity $\mathcal{OO}(n \log n)$. Before we prove it, we first introduce the following lemma.

Lemma 3. *A disc $D(O_i)$ can contribute at most two arcs to the boundary of $\bigcup_{O_i \in S} D(O_i)$.*

Proof. First, we know that the reference point O_0 is fixed and all the other disc origins are less than R away from O_0. Hence, if we consider the boundary of the union of disc $D(O_0)$ and any other disc, say $D(O_1)$, the arc contributed from $D(O_1)$ must have central angle (with respect to O_1) less than 240°. Let the arc be $\overset{\frown}{AB}$. Now if we want to place a third disc so that it "breaks" the arc $\overset{\frown}{AB}$ into two pieces, the origin of that third disc must be 1 away from both A and B by more than R and be less than R away from some part of $\overset{\frown}{AB}$. It is not hard to see that after adding a third disc with origin inside the shaded area, the remains of the arc $\overset{\frown}{AB}$, say $\overset{\frown}{AP}$ and $\overset{\frown}{AQ}$, must be more than 120^o central angle away from each other (with respect to O_1). In other words, each time an arc is cut into two pieces by another disc, the resulting smaller pieces must be more than 120° central angle (with respect to the origin of the disc contributing the arc) away from each other.

If a disc is able to contribute three arcs to the boundary of $\bigcup_{O_i \in S} D(O_i)$, its circumference is cut into arcs by more than three other discs. Based on the above argument, each pair of these three arcs is at least 120^o central angle away from each other. By adding up the degree of central angle for the gaps and the degree of the central angle for the arcs themselves, the total would exceed 360^o, which is not possible. Therefore, a disc can only contribute at most two arcs to the boundary of $\bigcup_{O_i \in S} D(O_i)$. Q.E.D.

Figure 6 shows the arc $\overset{\frown}{AB}$ contributed from disc $D(O_1)$. If a third disc is added to cut $\overset{\frown}{AB}$ into two arcs, the origin of that third disc must appear inside the shaded region.

Now we are ready to show that our algorithm has time complexity $\mathcal{O}(n \log n)$.

Theorem 2. *The divide-and-conquer algorithm described in Section 4.2 computes minimum disc cover set has time complexity $\mathcal{O}(n \log n)$, where n is the number of discs in Δ.*

Proof. A simple observation of the algorithm skeleton gives us the following formula for the time complexity of our algorithm.

$$T(n) = \begin{cases} 2T(\lfloor \frac{n}{2} \rfloor) + T(ArcMerge) & \text{if } n > 1 \\ O(1) & \text{if } |N| = 1 \end{cases}$$

Now if we are able to show that the time complexity of procedure $ArcMerge$ is $\mathcal{O}(n)$, $T(n)$ would be $\mathcal{O}(n \log n)$. It is clear that the time complexity to merge two arcs with the same angle span (with respect to O_0) is $O(1)$. Since each disc can only contribute at most two arcs, the total number of arcs is bounded by $O(n)$. This means that the time complexity of the procedure *ArcMerge* is bounded by a constant factor of $\mathcal{O}(n)$, which remains to be $\mathcal{O}(n)$, where n is the number of discs in Δ. Q.E.D.

5 Optimal Time Complexity of Minimum Disc Cover Set Computation

In [9], it has been shown that the optimal algorithm that constructs the α-hull for an arbitrary set of points S has time complexity $\mathcal{O}(n \log n)$, where $n = \|S\|$. However, since in the minimum disc cover set problem, the points in S (i.e., disc origins) are confined in a circle of radius R, it is unknown whether or not there exists an algorithm capable of constructing the minimum disc cover set in time complexity lower than $\mathcal{O}(n \log n)$. In this section, we prove that there is no such algorithm. The idea is to show that the element uniqueness problem, which has a lower bound of $\mathcal{O}(n \log n)$ [11], can be reduced in linear time to the disc cover set problem.

Definition 2. Element Uniqueness Problem
INSTANCE: *A multiset of non-negative integers $S = \{a_1, a_2, \ldots, a_n\}$.*
QUESTION: *Are there elements a_i and a_j in S, where $i \neq j$, such that $a_i = a_j$?*

Theorem 3. *The solution to the minimum disc cover set problem has lower bound* $\mathcal{O}(n \log n)$.

Again, due to the page limit, the proof of Theorem 3 is omitted in the paper. Interested readers please refer to [10] for the detail of the problem reduction.

6 Conclusion

In this paper, we have shown the minimum disc cover set problem as a special case of the α-hull problem. Furthermore, we demonstrate that the extra restriction imposed by our minimum disc cover set problem (i.e., all disc origins are inside a circle with the same radius) does not make the problem any easier by proving that no algorithm with lower time complexity than $\mathcal{O}(n \log n)$ can compute the minimum disc cover set. In addition, we provide two different optimal algorithms for the minimum disc cover set construction. The first one is based on the concept of the α-hull. However, since the first step of this algorithm is to construct Delaunay Triangulations, it fails on the degenerate cases when more than three disc origins fall on the circumference of a circle. On the other hand, the second divide-and-conquer algorithm we proposed is capable of finding the minimum disc cover efficiently for all circumstances.

References

1. M. L. Garey and D. S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness," San Francisco: W. H. Freeman, 1979.
2. M. Sun and T. H. Lai, "Computing Optimal Local Cover Set for Broadcast in Ad Hoc Networks," Proc. IEEE ICC pp. 3291-3295, Apr. 2002.
3. M. Sun, et al, "GPS-Based Message Broadcasting for Inter-vehicle Communications," Proc. IEEE ICPP, pp. 279-286, Aug. 2000.
4. C. Intanagonwiwat, et al, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," ACM MOBICOM, pp. 56-67, Aug. 2000.
5. Y. B. Ko and N. H. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks," ACM MOBICOM, pp. 66-75, Aug. 1998.
6. S. Y. Ni et al, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," ACM MOBICOM, pp. 151-162, Aug. 1999.
7. M. Sun and T. H. Lai, "Location Aided Broadcast in Wireless Ad Hoc Network Systems," Proc. IEEE WCNC, pp. 597-602, Mar. 2002.
8. M. Sun, et al, "Reliable MAC Layer Multicast in IEEE 802.11 Wireless Networks," Proc. IEEE ICPP, pp. 527-536, Aug. 2002.
9. H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel, "On the Shape of a Set of Points in the Plane," IEEE Transactions on Information Theory, vol. IT-29, pp. 551-559, 1983.
10. M. Sun, et al, "Minimum Disc Cover Set Construction in Mobile Ad Hoc Networks," Auburn University Technical Report, CSSE05-06, May 2005.
11. D. P. Dobkin and R. J. Lipton, "On the Complexity of Computations Under Varying Sets of Primitives," Journal of Computer and System Sciences, vol. 18 , pp. 86-91, 1979.

A Study on Dynamic Load Balanced Routing Techniques in Time-Slotted Optical Burst Switched Networks

Liang Ou, Xiansi Tan, Huaxiong Yao, and Wenqing Cheng

Department of Electronics and Information Engineering,
Huazhong University of Science and Technology, Wuhan, Hubei, China 430074
ouliang@public.wh.hb.cn

Abstract. Time-slotted optical burst switched network is a new paradigm to support IP over WDM by combining Time Division Multiplexing and Optical Burst switching technology. This paper presents the features of time-slotted optical burst switched network and addresses the related routing, wavelength and time slot assignment techniques to avoid contentions in the network. Cooperated with fiber selection algorithms, the proposed dynamic load balancing routing techniques use shortest path algorithm based on multi-wavelength virtual topologies to avoid congestion for burst requests. Simulation results show that the proposed load balancing routing techniques improve the wavelength utilization and reduce the burst loss rate.

1 Introduction

Exploring new all-optical switching technology to support IP over WDM will meet the fast growing demands on future Internet services. The occurrence of Conventional Optical Burst Switching (C-OBS) can be regarded as the combination of Optical Wavelength Switching (OWS) and Optical Packet Switching (OPS) [1] [2] [3]. With the one-way reservation protocol such as Just-enough-time (JET) or Just-in-time (JIT), C-OBS technique reduces connection setup time in OWS and avoids packet header identification problem in OPS whereas achieves moderate switching granularity. In order to increase the statistical multiplexing performance of wavelength in C-OBS, one of the solutions is to use wavelength converter (WC) components and large number of fiber delay lines as optical buffer. However, WC is neither cost efficient nor time efficient, which blocks the C-OBS to be deployed in large scale of commercial application. Moreover, the burst loss rate (BLR) of C-OBS becomes very high when the network suffers a heavy traffic because there is no enough wavelengths resource in the congestion status. Therefore, the question is whether there is a switching paradigm without WC while no loss the wavelength bandwidth utilization.

Recent advances on optical time division multiplexing (OTDM) wavelength-routed networks [4] [5], by partitioning wavelength bandwidth into partial time slot in a frame and using multi-fiber channels, shows that it is a potential solution to achieve high performance on both blocking and wavelength bandwidth

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 722–731, 2005.

utilization for long-term (static) traffic without any WC component. While In the short-term (dynamics) traffic case, the bandwidth utilization will be very poor by means of end-to-end two-way reservation approach, and nodes processing speed cannot meet the requirement of fast connection setup. Thus, JET or JIT protocol used in C-OBS can be introduced to solve issue for fast connection setup.

TS-OBS is a combination of C-OBS and OTDM paradigm. Replacing the WC device with Time Slot Interchanger (TSI), TS-OBS further improves wavelength utilization by partitioning wavelength into integer granularity time slots that are integrated in a frame. In [6] and [7], it is named as time-sliced OBS (TS-OBS) and optical cell switching (OCS) respectively. A repeating time-slot channel networks concept and a blocking TSI structure is proposed in [6]. And [7] presents the OCS networks design parameters and FDL assignment algorithm.

One of essential objectives of the design of TS-OBS networks is to minimize burst loss rate. Because of the wavelength continuity constrain, data burst loss occurs in core node once multiple bursts attempt to occupy slot channels at the same output wavelength on different output fibers of a link simultaneously. There are several approaches to resolve the contention for OBS networks: deflection routing, buffering, and time-space domain scheduling [3] [8]. All above approaches are based on local node information to resolve contention rather than to avoid it before it happens, except [2]. In order to avoid congested links (or fibers and wavelengths) in TS-OBS networks, routing techniques based on global network status should be developed to balance the traffic loads. In this paper, we propose the contention avoidance routing technique based on network level information for TS-OBS networks.

This paper is organized as follows: Section 2 presents the architecture and features of TS-OBS networks. Routing and wavelength assignment issue for TS-OBS and load balance routing techniques are studied in Section 3. Simulation result and performance studies are presented in Section 4, and this paper is summarized in Section 5.

2 Network Architecture

The TS-OBS network architecture shown in Fig. 1 is a bi-direction multi-fiber multi-wavelength system without WC. It consists of edge and core routers and every node is connected with multi-fiber multi-wavelength optical link. Other features inherited from C-OBS include: (1) physical or logical separated control and data channel, generally, each fiber includes at least one wavelength for control packets. (2) Only control packets are processed by electronic layer in each node. Data bursts keep in optical domain passing by every intermediate node, only at the cost of suffering an acceptable delay coursed by keeping frame synchronized. (3) Upper layer packets are assembled into a big burst in ingress edge and disassembled in egress node. Core nodes just forward bursts in all-optical format. (4) There is an offset time between control header and data burst according to JET or JIT protocol.

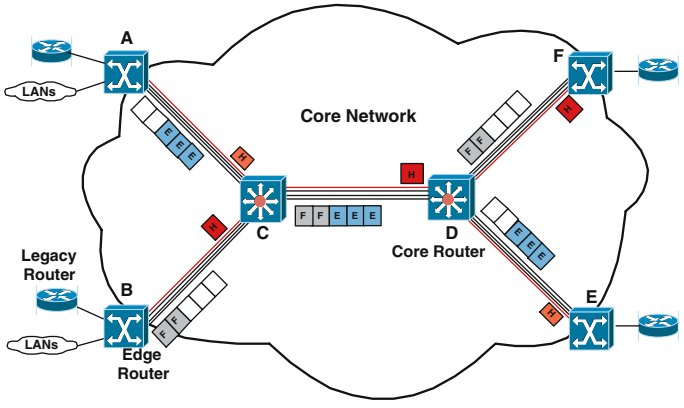


Fig. 1. TS-OBS network architecture

As a TDM network, TS-OBS also includes other specific features. Wavelength band-width in TS-OBS consists of cyclic fixed-length frame structure, and every frame is subdivided into fixed-length time-slots. Each burst can occupy one or more time-slot depending on burst length distribution. And there are two types of traffics existing in the TS-OBS network: constant long-term traffic that occupies a set of the fixed-position time-slots from the source to destination in TDM manner, and dynamic best-effort traffic that occupies time-slots channel hop by hop in uncertain manner depending on slot scheduling algorithm. The constant traffics have to setup the connection by REQUEST and ACK signaling, and are granted with higher priority when contending with best effort traffic for time-slot channel request.

TS-OBS inherits the merits of both TDM-based wavelength-routed network that provides moderate switching granularity and C-OBS network that supports fast connection setup mechanism, while remove their drawbacks, e.g. without use of cost inefficient wavelength converter for C-OBS network and long connection setup time for dynamic traffic for TDM-based WR network.

One of challenges in realizing TS-OBS network lies in the fast configuration capability of core nodes for high-speed dynamic traffics.

3 Dynamic Load Balanced Routing Techniques

For TS-OBS networks, the objective of the dynamic routing technique is to select the route, wavelength or fiber and slot channel for burst requests arriving stochastically and reduce the contentions on the bottleneck links in the network. The contention occurs when all slots are occupied by burst data on the same wavelength of all fibers in the congested links. Contention avoidance can be achieved by utilizing static or dynamic load balanced routing [2].

In static routing case, the route calculation is ahead of time by using some static metrics, such as number of hops or physical distance. A list of path and

fixed alternate paths is maintained by nodes after execute a shortest-path algorithm using chosen metrics. An alternate path is chosen only if the primary path is reported to be congested. In the TS-OBS framework, since the duration of burst data transferred is very short, the traffic is fluctuating over different frame transfer time. Thereby, we choose the dynamic load balanced routing technique.

There are many existing dynamic routing techniques for C-OBS and WR-TDM net-works, including virtual-wavelength-path approach [9]. However, those approaches are different from our proposed routing technique. First, in WR-TDM networks, a route is determined only at the source node due to the wavelength continuity constrain. Routing processing means that the source node has to find not only a route but also an unused wavelength and allocate slot channel for the coming session. This is defined as Routing, wavelength and slot assignment (RWTA) problem in [4], [10]. However, in the TS-OBS network, the burst can be forwarded to any other core node hop-by-hop based on routing table. Second, in the C-OBS networks, due to using WC component, bursts can be forwarded to any wavelength channel on the output link. Because of wavelength continuity constrains and TDM channel, source nodes in TS-OBS network must assign a wavelength for bursts, and bursts can only be forwarded to the identical wavelength channel on the output link at each intermediate core nodes. Moreover, in the multi-fiber network, fiber selection operation should be executed on the new out-put link for slot assignment to avoid contention. The remainder of this section discusses the proposed routing techniques in-depth.

3.1 Multi-wavelength Virtual Topology Routing Techniques

Consider a N nodes TS-OBS network with F fibers in each link and W wavelength channels in each fiber. Successive frames, each consists of T slots, are transmitted in the wavelength channel. In this paper, we assume that the contiguous slot assignment is used to schedule each burst, and only best-effort traffic is taken into consideration. At the edge nodes, we regard every output wavelength as a virtual topology, i.e., a copy of physical topology. Thus, each wavelength channel includes FT slot channels, W wavelengths consist of W candidate routes (Fig. 2). We call these topologies as Multi-Wavelength Virtual Topologies (MWVT). In order to accomplish proposed technique, each node in network maintains following link status information:

$l(i, j)$: Link between node i and node j

Δ: Fixed interval of routing table refresh, which is in the form of multiple frame length T

$T^{\lambda,f}$: Number of time-slots (bursts) have transferred successfully on the wavelength λ of the fiber f during the interval δ, where $1 \leq f \leq F$, and $1 \leq \lambda \leq W$.

L_{node}: The table about load information of all wavelengths on each output link of this node

Additionally, L_{net}, the table of load information of all wavelengths on each link in the network, is maintained at edge nodes.

Route selection is done using an OSPF-style link state algorithm with periodic link state updates, followed by computation of shortest-cost routes using

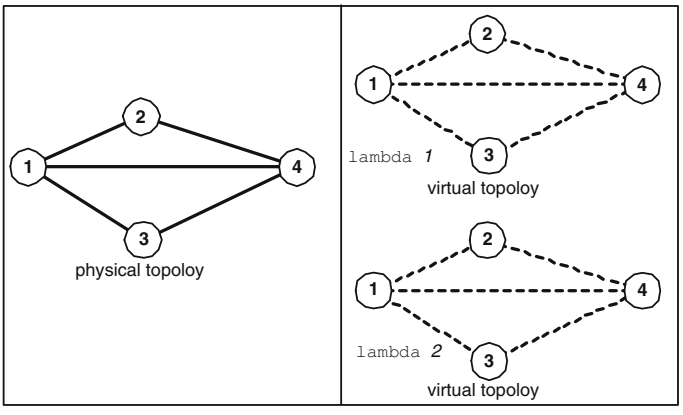


Fig. 2. Example of Multi-Wavelength Virtual Topologies ($W = 2$)

Dijkstra's algorithm. The shortest-cost path calculation is implemented using link weight functions that depend upon current specific network load information on each wavelength. After each routing table update, the shortest path (including perhaps an alternate path) is computed in advance and stored in the routing tables at each node.

Specifically, because there are W MWVTs, the shortest path algorithm is executed W times for all MWVTs to find a suitable route, i.e. a wavelength path. When a new burst arrives, the node uses the routing table to determine the next hop. Given the route/wavelength, the fiber selection algorithm is performed to select a wavelength on this fiber. Once the fiber is determined, then the time-slots required are allocated at the source nodes based upon the burst request. At the intermediate core nodes, the fiber selection algorithm is also executed to avoid contention on the output fibers. Since the TSI component is used, in order to avoid slot channel contention on the same wavelength, the slot scheduling algorithm is performed in the internal time-space switching unit of switching node, which is out of the scope of this paper.

The proposed routing algorithm performs as follows:

Given that the network load information L_{net}, the source S and destination D, and the required slots of a burst, t.

(1) Set link weight $W_{(i,j)}^{(\lambda,f)}$, based upon the weight function, of links $l_{i,j}$ on each MWVT in terms of the link status information, $1 \leq f \leq F$, $1 \leq \lambda \leq W$.

(2) Execute the modified Dijkstra algorithm on each MWVT to obtain cost, $C_{S,D}^{\lambda}$, from S to D, until all W MWVTs are executed, $1 \leq \lambda \leq W$.

(3) Using the function, $\min\limits_{1 \leq \lambda \leq W} C_{S,D}^{\lambda}$, to find a minimum cost wavelength.

(4) If paths that can provide t slots are found, select a path randomly and continue to perform follows:

i. Select this wavelength/route

ii. Execute fiber selection algorithm on selected path

Once the routing algorithm terminates, the slot scheduling algorithm is executed im-mediately. The time complexity of proposed routing algorithm is $O(|N|^2W + W)$.

3.2 Weight Function Calculation

Considering overall network load information, the shortest path algorithm uses adaptive link weight function to improve route selection in TS-OBS networks. The routes are recomputed every units of time based on congestion, physical distance and capability on the links.

The weight, $W_{(i,j)}^{(\lambda,f)}$ is based on single or combined metrics. We define a weighted function based on congestion on the link as well as hop distance:

$$W_{(i,j)}^{\lambda} = \frac{\alpha \sum_{f=1}^{F} T^{(\lambda,f)}}{\Delta FT} + \beta \tag{1}$$

where $0 \leq \alpha, \beta \leq 1$, among which α is the factor of offered load (capacity), the first part of (1), on the MWVT links, and β is the factor of hop distance. For $\alpha = 0$, it is the hop-only routing metrics. Whereas, if $\beta = 0$, only the offered load on the links is taken into consideration for routing policy. While sending the burst on the least congested route results in low burst loss rate at lower loads, under heavy loads, longer paths result in higher overall network loads, thereby increasing the probability of contention. In order to avoid this situation, we can choose a rational rate between and to balance network load conveniently according to link status.

3.3 Fiber Selection Algorithms

In multi-fiber TS-OBS networks, once a wavelength/route is confirmed, fiber selection algorithm is executed to find a suitable output fiber from all fibers on the output links for the coming burst request. It is performed during a frame duration for every burst request within this frame, and is included in both edge nodes and core nodes.

First Fit Fiber Selection (FFF)
Given burst request t, FFF searches wavelength capacity from lower-numbered fiber to the higher numbered ones from fibers bundle of link. It finds the first wave-length capacity in a frame that satisfies $A_{(i,j)}^{(\lambda,f)}$ on the fiber f, and allocates both the potential fiber, f, and the wavelength selected in routing algorithm for the coming burst as output channel. Thus, FFF has low computation complexity since local capacity information at this node is not required. *Least*

Load Fiber Selection (LLF)
This algorithm attempts to find a wavelength capacity of fiber f with the least load (occupied slots) in a frame for the coming burst, that achieves:

$$\min_{1 \leq f \leq F} (A_{(i,j)}^{(\lambda,f)}), \qquad A_{(i,j)}^{(\lambda,f)} \geq t \tag{2}$$

Thus, given the wavelength selected in routing algorithm, the fiber with largest avail-able slots in a frame is selected to be output channel.

Best Fit Fiber Selection (BFF)
Given burst request t, BFF selects the fiber f with the smallest available capacity after remove t slots this burst occupied, that satisfies:

$$\min_{1\le f\le F}(A_{(i,j)}^{(\lambda,f)} - t), \qquad A_{(i,j)}^{(\lambda,f)} \ge t \tag{3}$$

This algorithm attempts to select used fibers as much as possible.

4 Performance Evaluation

We develop a simulation model with NS-2 to evaluate performance of the proposed load balanced routing algorithm as well as fiber selection policy. The network topologies considered in the simulation are the modified 13-node NSFNET backbone net-work [1] with 21 bi-direction links, an average nodal degree 3.15 (Fig. 3). Link transmission rate is 10Gbps. Each frame consists of $T = 100$ time slots, and lasting duration of each slot is 1μs. Bursts arrive at a node according to Poisson process with rate A_r, and uniformly random destination node. The burst length is an exponential distributed duration in the form of number of slots within the set $\{1, \ldots, 100\}$, with a mean of $1/\mu$. We assume that the maximum burst length is equal to frame size T. Define the normalized offered load as follows:

$$A_\lambda = \frac{A_\gamma \mu}{\mathrm{FWT}} \tag{4}$$

We use burst loss rate to be performance metrics to evaluate the routing and fiber selection algorithms, and variable parameters are offered load (satisfies Equation (4)), number of wavelength per fiber and fibers per link and burst length. The choice of weight factors, α and β, in proposed routing weight functions have great effect on load balancing strategy. In simulation, we consider three policies: ($\alpha = 0$, $\beta = 1$), ($\alpha = 1$, $\beta = 0$), and ($\alpha = 1$, $\beta = 1$), corresponding to the Hop-only routing (HR), Capacity-only routing (CR), and Hop and Capacity routing (HCR) respectively. Blocking performances of above routing policies are evaluated in the same simulation environment.

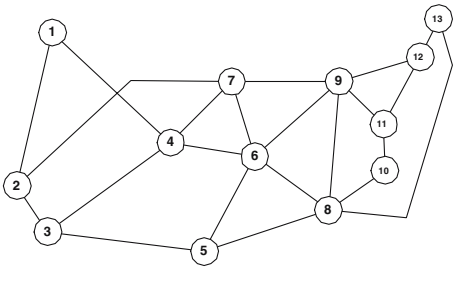


Fig. 3. 21 links modified NSFNET backbone network

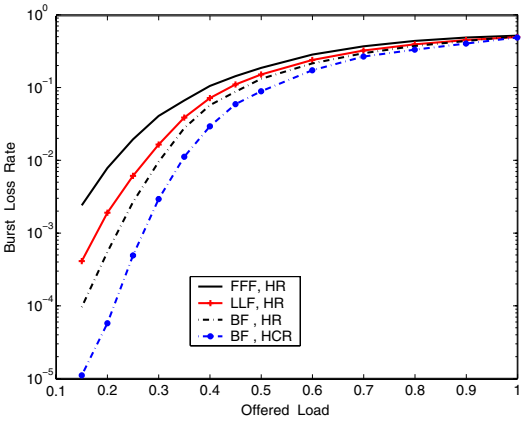


Fig. 4. Comparison of fiber selection algorithms using HR ($\alpha = 0, \beta = 1$) and HCR ($\alpha = 1, \beta = 1$), $F = 2, W = 8, T = 100, 1/\mu = 20$

We begin with studying the blocking performance of fiber selection algorithms. Fig. 4 presents the comparison of FFF, LLF and BFF with HR routing algorithm ($\alpha = 0$, $\beta = 1$). Choosing HR is because number of hop is fixed in a physical topology. It is easy to compare the performance of fiber selection algorithms with a stable weight calculation approach which has constant weight. We find BFF outperforms other two algorithms in either low or moderate high traffics. The reason is that BFF chooses those used slot channels, thus, there are more contiguous free slots that are reserved for next coming bursts. All algorithms tend to suffer a high blocking at the same level once the offered load becomes very high. Fig. 4 also presents the result of BFF with HCR ($\alpha = 1$, $\beta = 1$). It shows that an obvious improvement is achieved by using com-bination routing policy for any fiber selection algorithm.

In Fig. 5, performances are compared for all routing algorithms using BFF fiber selection algorithm. HR performs better than CR only if offered load is low, say, below 0.15. Once the network becomes congested, offered load over 0.2, the CR starts to perform better. This is because CR algorithm can avoid those congested routes by recomputed weight function following the fluctuation of link capacity.

Combined with hop and link capacity factors, the overall burst loss rate of HCR keep lower than that of others except that the offered load become very high, i.e., offered load over 0.7. This reason is that HCR, compared with CR, are more likely to choose congested route at high offered load. However, once the offered load becomes ex-treme high (not shown in this paper), all algorithms tend to reach to the same high level BLR due to no enough free slots channel in all wavelengths/routes in the net-work.

We also study if changing number of fiber per link effects on the performance of HCR policy. In Fig. 6, with increasing fibers per link from 2, 4 to 16, the BLR using HCR with BFF dramatically drop to as low as 10^{-6}. And similar results

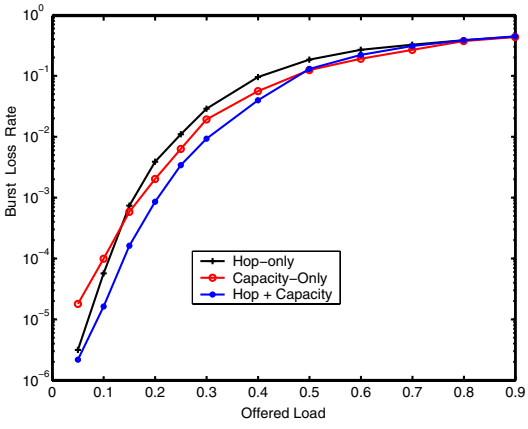


Fig. 5. Comparison of Routing Algorithms (HR, CR and HCR), with BFF, $F = 16, W = 8, T = 100, 1/\mu = 20$

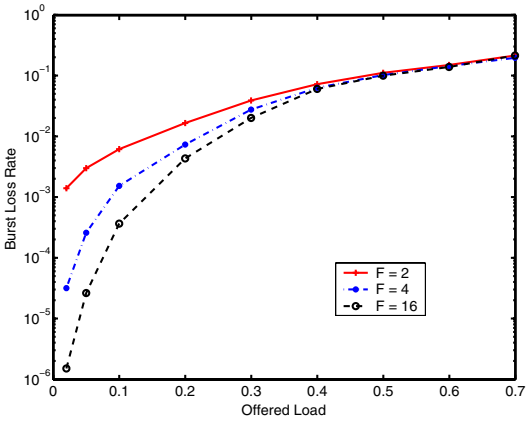


Fig. 6. Effect of Number Fiber per link (HCR with BFF), fiber number increased from 2 to 16, $W = 8, T = 100, 1/\mu = 10, \Delta = 100$

can be obtained from HR and CR. This accords with most conclusions that the role of multi-fiber is similar to the use of wavelength converter in the network.

5 Conclusion

This paper studies routing techniques of TS-OBS network to support dynamic load balancing. The proposed routing techniques include two parts: MWVT routing and fiber selection algorithm. By considering link capacity as well as hops, the proposed routing techniques can adapt to overall network status to avoid

congestion. Simulation shows that the combination approach, HCR with BFF, provides the best overall performance for most offered loads and configurations.

Future work is to enhance the robustness of proposed routing technique. How the refresh time of network information effects on the performance should be investigated. Also, if we can further improve the blocking performance by revising weight factors, α and β, dynamically in run-time is worth studying.

References

1. Rosberg, Z., Vu, H.L., Zukerman, M.: Performance analyses of optical burst switching networks. IEEE J. Select. Areas Commun. **21** (2003) 1187–1197
2. Thodime, G., Vokkarane, V.M., Jue, J.P.: Congestion-based load bal-anced routing in optical burst-switched networks. In: Proc. IEEE GLOBECOM 2003. Volume 5., San Francisco, USA (2003) 2694–2698
3. Hsu, C.F., Liu, T.L., Huang, N.F.: Performance analysis of deflection routing in optical burst-switched networks. In: Proc. IEEE INFOCOM 2002. Volume 1., New York, USA (2002) 66–73
4. Wen, B., Sivalingam, K.M.: Wavelength and time-slot assignment in time divi-sion multiplexed wavelength-routed optical wdm networks. In: Proc. IEEE INFOCOM 2002. Volume 3., New York, USA (2002) 1442–1450
5. Subramaniam, S., Harder, E.J., Choi, H.A.: Scheduling multi-rate sessions in time division multiplexed wavelength-routing networks. IEEE J. Select. Areas Commun. **18** (2000) 2105–2110
6. Ramamirtham, J., Turner, J.: Time-sliced optical burst switching. In: Proc. IEEE INFOCOM 2003, San Francisco, USA (2003) 2030–2038
7. Chao, H.J., Liew, S.Y.: A new optical cell switching paradigm. In: 1st International Workshop on Optical Burst Switching, Co-located with SPIE Opticomm'03, Dallas, USA (2003) 120–132
8. Wang, X., Morikawa, H., Aoyama, T.: Priority-based wavelength assignment algorithm for burst switched photonic networks. In: Proc. IEEE/OSA OFC 2002, Anaheim, CA (2002) 765–767
9. Madhyastha, H., Balakrishnan, N.: An efficient algorithm for virtual-wavelength-path routing minimizing average number of hops. IEEE J. Select. Areas Commun. **21** (2003) 1433–1440
10. Huang, N.F., Liaw, G.H., Wang, C.P.: A novel all-optical transport net-work with time-shared wavelength channels. IEEE J. Select. Areas Commun. **18** (2000) 1863–1875

A Novel Multi-path Routing Protocol

Xiaole Bai[1], Marcin Matuszewski[2], Liu Shuping[2], and Raimo Kantola[2]

[1] Computer Science and Engineering Department, Ohio State University, Columbus, USA
baixia@cse.ohio-state.edu
[2] Networking Laboratory, Helsinki University of Technology, Otakaari 5, Espoo, 02150, Finland
{marcin, shuping, kantola}@netlab.hut.fi

Abstract. The paper advocates a different view of routing protocols according to technology trends nowadays. A Link State Multi-Path routing protocol, LSMP, is thereby introduced. This new hop-count based multi-path routing protocol has novel failure recovery mechanism, which is simple and effective. Through analysis and simulation, our protocol shows much faster convergence and better traffic balancing than traditional routing protocols. We propose LSMP to create an efficient routing architecture supporting Traffic Engineering (TE), while at the same time reducing complexity of a routing system.

1 Introduction

The algorithms to find the shortest path from a source to a destination with high efficiency have being explored thoroughly for many years. These algorithms are all derived from the traditional graph theory that views nodes simply as transparent joints of links. Their underlying assumption is that links are more important than nodes and link parameters determine path quality. However, looking at the trends in link and router capacity growth presented in [1], we note that bottlenecks in networks appear in nodes rather than links. We should also remember that in real networks:

- the propagation delay that is directly proportional to link length, is actually very small,
- compared with the link propagation delay, delays experienced in nodes may be very large,
- the number of parameters associated with a link is small,
- many function blocks are added to nodes, rather than links. We can admit that some of the nodal parameters can be reassigned to links but only in case no shared resources are used.

Quality of Service (QoS) and Traffic Engineering (TE) have been in focus over the recent years. Many inspired ideas for these concepts are based on adding functional blocks into the network architecture, e.g. [4-7], or making extensions to traditional routing protocols, e.g. [8-10]. An example is the QoS extensions to OSPF [11]. Such an approach, however, suffers from slow convergence typical of OSPF and poor traffic balancing. In carrier QoS IP networks, convergence time should be of the same

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 732 – 741, 2005.

order that is achieved by SDH networks. Obviously, not much can be gained by distributing detailed link load information for the purpose of QoS routing if the network needs many seconds to converge using new link state information. Also, the complexity brought by some of the approaches, like those in [12-15], is breaking the simplicity of the original network. We propose Link State Multi-Path (LSMP) routing protocol to create an efficient routing architecture supporting QoS and TE, while at the same time keeping things as simple as possible. Our protocol advocates the opposite view to the traditional graph theory. It assumes that nodes are more important, than links. An example in Figure 2 clarifies the difference. In Figure 2, path A-B-C-F is not as good as path A-D-F and A-E-F from this point of view. We consider our work a first try to find some new protocol based on the above idea. Although we introduce and analyze the LSMP protocol in the next sections, we would like to point out the significance of our novel view of the traditional routing problem.

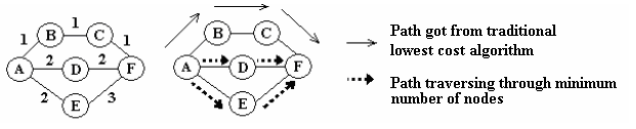


Fig. 2. The numbers in the left denote link costs. We consider the path going through a lesser number of nodes the better, no matter what the link parameters are

The rest of the paper is organized as follows: Section 2 gives a brief description of the Link State Multi-Path routing protocol. An initial performance evaluation is given in Section 3. Finally, we draw some conclusions in Section 4.

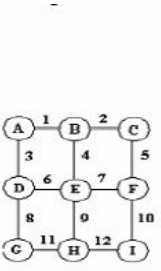
2 Link State Multi-path Routing Protocol

Like the popular OSPF, our multi-path routing protocol is based on a link state database. All nodes have a copy of the network map that is regularly updated. The figures 3(b) and 3(c) represents the copy of the network map stored in router databases. Figure 3(b) describes a traditional data structure used e.g. by OSPF, while Figure 3(c) represents a data structure used in our proposal. As we can see from the example, the traditional data structure requires much more memory than the data structure used by LSMP.

In contrast to OSPF, in LSMP, each node has a position code for each destination. The position code is the same as the shortest distance in terms of number of nodes on a path to a destination. Assuming that node B is the destination, Figure 4(a) presents the position codes to B stored in the other nodes in the network. In the same way Figure 4(b) illustrates the table kept in node B, in which the middle column presents the position code from B to every other node and the last column shows the corresponding interfaces.

When node B, receives a packet whose destination is F, it forwards it to its neighbors with the smallest position code to the destination. That is, B would forward the packet to the interface connecting it to node E or node C. We can easily produce the routing table in every node for each destination from the link state database

presented in Figure 3(c) using the algorithm in appendix A. For example, node B would generate and keep a routing table as shown in Figure 4(b). The core of the algorithm is to find neighbors for each node. We have to stress that our algorithm in its current form cannot deal with different link parameters.



(a)

From	To	Link	LinkCost
A	B	1	1
A	D	3	1
B	A	1	1
B	C	2	1
B	E	4	1
C	B	2	1
⋮	⋮	⋮	⋮
H	I	12	1
I	F	10	1
I	H	12	1

(b)

Node	Neighbors
A	B D
B	A E C
C	B F
D	A E G
E	B D F H
F	C E I
G	D H
H	G E I
I	F H

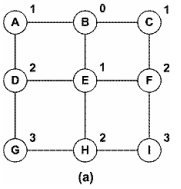
(c)

Fig. 3. Example network and its topology representation. The numbers in (a) represent link indexes, while the link costs are assumed to be equal one

The algorithm progresses as a breadth first search through the network rooted on the source node. Its complexity is $O(M)$, while the traditional Dijkstra algorithm has the complexity of $O(N^2)$, where M is the number of links and N is the number of nodes in the network. Because $M \leq 0.5N(N-1)$, the complexity of our algorithm is related to average degree when the number of nodes in the network is fixed. Our algorithm achieves a similar complexity as SPF when the network is nearly fully meshed, which however is very rare in the real world. For large real networks, there can be a sizable difference in favor of our algorithm. A major drawback of the Dijkstra algorithm is its low efficiency in sparse networks. Our algorithm shows a clear advantage in this kind of networks.

2.1 Load Balancing

As mentioned before, a node that receives packets with the same destination forwards them to one of its neighbors that have the smallest position code corresponding to the destination. The basic forwarding mechanism assumes that interfaces, if not only one, share the traffic to one destination with the same probability. Let λD denote the total traffic to the destination D, and let n denote the number of interfaces to forward the packets. The basic mechanism means the traffic is $\lambda D/n$ for each interface. Obviously, the basic forwarding mechanism does not take into account the link bandwidth for each interface. Local optimization can be done naturally according to different link bandwidths. That is, if $\{B1,B2...,BN\}$ denotes the bandwidth for each link, the improved LSMP (ILSMP) lets interface i receive traffic $\lambda DBi/\Sigma Bi$. We understand limitation of random load balancing scheme especially when TCP protocol is used. Nonetheless, we believe jitter can be effectively reduced by end applications using techniques like redundant data and buffer technology. Besides our protocol can also be combined with other widely used load balancing schemes e.g. hash function, however the analysis of their performance is outside the scope of the document.



Destination	Position	Interface
A	1	A
B	0	-
C	1	C
D	2	A E
E	1	E
F	2	C E
G	3	A E
H	2	E
I	3	C E

(b)

Fig. 4. (a) Position codes from other nodes to node B; (b) The table kept in node B

2.2 Failure and Recovery

First we consider a link failure scenario presented in Figure 5. If OSPF is used, there are three possible paths from D to A, namely D-G-F-A, D-C-B-A, and D-C-E-A. If link A-B fails, an update message is generated and flooded through a network. When D receives the update message it may change its routing table and start forwarding packets with destination A to node G. Here, we can see the typical reason why every node in a network using a traditional routing protocol, like OSPF, should receive an update message as soon as possible. The reason is that new routing decision may need to be made by a far node when the topology changes.

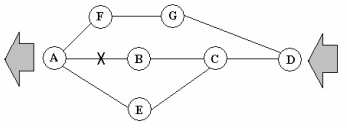


Fig. 5. A link failure example

In contrast to OSPF, performance of LSMP and ILSMP protocols does not depend so strongly on a fast flooding mechanism. Their architecture does not require to flood the update information to every node as soon as possible, although each node holds the map of a whole network as in OSPF. Moreover, some nodes do not have to change their routing table although some changes in the network have happened. In the above example, when link A-B is broken, the position codes for C and G corresponding to destination A do not change, i.e. are still equal to 2. In such a situation node D still forwards packets to C and G with the probability as before no matter whether it has received an update message or not.

In our routing architecture we can distinguish two different update scenarios undertaken by network nodes namely a partial and a full update. The partial update takes place if at least one position code for one destination has changed after the update. If position code has not changed and only some interfaces are added or deleted in the routing table, we call this kind of update a partial update. The partial update is much faster than a full update because no link state based computation is needed.

As we can see in Figure 6, it is not necessary for all nodes to make any changes in their routing tables. When link H-M goes down, nodes C, H, M and R modify some entries in their routing tables. They calculate new routes based on updated information in the link state database. In contrast, nodes, marked using light grey color, only have to perform some comparison and than modify their routing tables directly.

Figure 7 presents two examples related to link H-M failure. As we can notice, the position codes in node C have changed due to link failure, whereas the position codes in node S have not been modified. It is not necessary to re-compute the routing table in node S, since position codes for C and H not only depends on node R but also on node N. Node S just have to remove an interface connecting it to node R from its routing table.

The area in which every node has to change its routing table when link failure happens is called the main update area. The partial update nodes are always on the border of the main update area, while those nodes that take a full update are in the interior places. The flooding mechanisms are different inside and outside the main update area. In these nodes that take the full update, update message forwarding takes place after updating and computing. On the contrary, the partial update nodes forward the update messages immediately after some simple check for flag value. In these nodes, routing table changing and link state database updating may happen at the same time. In nodes that are beyond the main update area, database updating and update messages forwarding may take place in parallel, since the update messages are forwarded immediately.

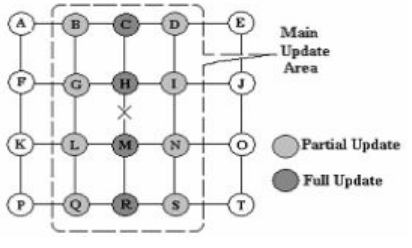


Fig. 6. Update area

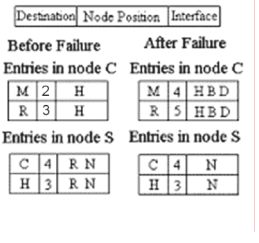


Fig. 7. Some entries in two nodes

In case of link recovery, there are also two types of updates, a full update and a partial update. The difference depends on whether recalculation is needed for the node position code or not. Similarly nodes out of the main update area forward update messages immediately, therefore updating their database and forwarding may take place in parallel.

Node failure causes many links to fail simultaneously. In case of node failure the update procedure follows the same procedure as in case of link failure.

Node recovery procedure has two steps. First, a node has to collect necessary information from neighbor nodes. Neighbors discovery can be done twofold by Hello protocol or by physical layer. After the discovery process, the node calculates its routing table. The second step is to generate and forward update messages. In this step nodes perform the same tasks as in link recovery procedure.

3 Evaluation

This section presents results of simulation and analytical analysis of OSPF and LSMP routing protocols.

The convergence time is a period when routes become stable again after some changes in a network topology. It can be seen that convergence time consists of three parts: T_d, the time neighboring nodes spend on detecting that something has changed e.g. adjacent link or neighbor node is down/up; T_1, the time a update message is propagated over one link, and T_2, the time a router spends on processing incoming update messages and updating its link state information database and its routing table. The three time components are shown in Figure 8(a). T_d depends on the detection mechanism that can be based on hardware, in layer two e.g. SONET, or on layer three protocol e.g. the hello protocol used in the traditional OSPF. T_1 is usually small, however the sum of T_1s along the path, the update message traverse before reaching destination, gives the lower bound for the convergence time. T_2 may be affected by many factors, i.e. router architecture, message size, total number of nodes in the network, data structures and algorithms.

The following evaluation of the convergence time compares the traditional OSPF and LSMP protocols. A topology generator has been used to automatically generate routable topologies with 50 and 100 nodes. The average degree D of the topology changes from 5 to 30. Each type of topology, which means the topologies with the same number of nodes and average degree, is generated 100 times. The simulation scenario assumes that each link has the same failure probability. The result of the simulation can be seen in Table 1, in which T_1 and T_2 are for OSPF and T_1', T_2' are for LSMP. In our simulation T_2' only denotes the times for the full update in LSMP.

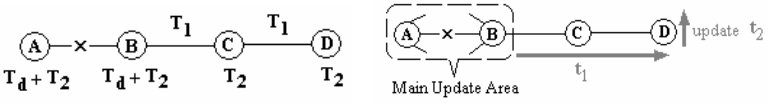


Fig. 8. (a) Convergence time components; (b) Update area

The result shown in Table 1 can be interpreted as follows: when the number of nodes is 100 and average degree is 20, the average convergence time for OSPF is

$$2.90 \times T_1 + 3.90 \times T_2 + T_d, \tag{1}$$

while the average convergence time for LSMP is:

$$0.60 \times T_1' + 0.60 \times T_2' + T_d. \tag{2}$$

Furthermore, T_2' approaches 0 when the average degree is 30 meaning that under this circumstance nearly all updates are partial and the main update area contains just the two ends of the failed link.

The convergence time in LSMP is obviously much shorter than that in conventional OSPF, especially when the average node degree is large. Although the conclusion we get is for link failures, a similar conclusion can be expected when a node failure happens, since in both situations nearly the same update procedure is used.

In addition to convergence time, we may be also interested in answering the question: how long the nodes in the network will have the same link state database. Since the update message does not change the routing table in those nodes that are out of the

Table 1. Convergence time comparison for OSPF and LSMP

	Number of Nodes : 50					
D	5	10	15	20	25	30
T_1	3.91	3.00	2.71	2.18	2.09	2.00
T_2	4.91	4.00	3.71	3.18	3.09	3.00
T_1'	1.68	0.75	0.43	0.11	0.03	0
T_2'	1.68	0.75	0.43	0.11	0.03	0
	Number of Nodes : 100					
D	5	10	15	20	25	30
T_1	4.20	3.24	3.00	2.90	2.81	2.30
T_2	5.20	4.24	4.00	3.90	3.81	3.30
T_1'	2.40	1.00	1.03	0.60	0.30	0
T_2'	2.40	1.00	1.03	0.60	0.30	0

main update area, the smaller the main area is, the more stable routes are. Analyzing T_1' values in Table 1 we can conclude the main update area in LSMP decreases dramatically when the average degree of nodes in the network is growing. Besides the update messages are forwarded immediately in the nodes outside the main area. On contrary, in OSPF, all of the nodes forward the update message after node finishes processing and updating its database and routing table. Suppose t_1 is the time the message takes to travel from a border node to the farthest node, and t_2 is the time each node uses to update its link state database as shown in Figure 8(b). The whole time period for nodes out of the main update area is

$$t_1+t_2\ (t_1 \geq t_2) \text{ or } 2\times t_2\ (t_1 < t_2), \tag{3}$$

which is much less than that in the traditional OSPF.

When evaluating different routing schemes, we try to alleviate the effect from traffic variance. We assume that between each pair of nodes there is a pair of traffic flows with the same rate. Let us denote the bandwidth of link k by w_k and the amount of traffic going through the link by λ_k. We can observe that global traffic balancing is achieved when the larger w_k, the larger is the corresponding λ_k; and vice versa. It means that, the links with larger bandwidth should play the more important roles in the network, otherwise resources are wasted, and those links with smaller bandwidth should play lighter roles, otherwise congestion will happen. Hence, the variance w_k/λ_k should be as small as possible and the mean of w_k/λ_k should be as large as possible for global optimization. We define the standard variance and the mean of w_k/λ_k as follows: $V=var(w_k/\lambda_k)$ and $A = mean(w_k/\lambda_k)$, where $k \in E$. Finally, we define a new metric called a routing balancing factor:

$$\beta = A/V \tag{4}$$

Given a stable network and link bandwidths, some obvious observations can be made. First, in a stable network, β is fixed for a loop-free routing protocol. Second the larger β, the better the routing protocol from the traffic balancing perspective.

In this paper, we explore four loop-free routing schemes namely: Shortest Path with the hop count metric (SP), Bandwidth-inversion Shortest Path (BSP), Link State Multi-Path (LSMP), and Improved Link State Multi-Path (ILSMP). We still use our

Table 2. β for different topologies

N	D	S	SP	BSP	LSMP	ILSMP
5	2	2	2.558451	2.758385	3.224484	3.311428
		4	2.254542	2.768630	2.604955	2.697214
		6	2.024941	2.719088	2.286137	2.367076
		8	1.879220	2.600865	2.124170	2.199236
		10	1.830207	2.539580	2.032487	2.097065
	4	2	2.556193	2.920152	3.612424	3.719059
		4	2.106576	2.718180	2.576678	2.696821
		6	1.910423	2.674422	2.229637	2.319455
		8	1.829896	2.617671	2.139179	2.213716
		10	1.754896	2.591378	2.034266	2.121841
15	6	2	1.372062	1.789455	2.834952	2.989822
		4	1.213127	1.633862	2.119778	2.307371
		6	1.261938	1.907388	1.946040	2.178800
		8	1.116594	1.552662	1.833936	1.989530
		10	1.064496	1.622508	1.781681	1.954117
	8	2	1.444550	1.823206	2.887981	3.078970
		4	1.311741	1.757364	2.198201	2.416277
		6	1.280783	1.787689	1.848417	2.082069
		8	1.218484	1.815218	1.924643	2.143437
		10	1.156890	1.639147	1.891359	2.087615
	10	2	1.558888	1.924333	3.000156	3.178389
		4	1.425956	1.903149	2.360969	2.669537
		6	1.324915	1.858417	2.116090	2.402482
		8	1.284578	1.651527	1.933748	2.173976
		10	1.283758	1.755940	1.892332	2.142496
35	5	2	1.393066	1.727796	2.843095	2.953676
		4	1.170473	1.156601	2.097564	2.293631
		6	1.123194	1.192047	1.914990	2.120861
		8	1.113675	1.118143	1.811987	1.993174
		10	1.107602	1.078012	1.731546	1.890742
	20	2	1.521179	1.738854	3.696585	4.214760
		4	1.279114	1.507586	2.432551	2.947664
		6	1.246285	1.465092	2.185768	2.660619
		8	1.210607	1.451196	2.008804	2.445016
		10	1.206201	1.431959	1.934522	2.346259
45	7	2	1.247312	1.388031	2.949592	3.153439
		4	1.124536	1.250965	2.311052	2.584451
		6	1.113788	1.169503	2.005577	2.262970
		8	1.041946	1.137780	1.849122	2.080396
		10	1.078287	1.126794	1.838146	2.073295
	30	2	1.697510	1.829893	3.777833	4.345482
		4	1.550049	1.572377	2.573549	3.192790
		6	1.382910	1.454816	2.171009	2.664490
		8	1.358083	1.454770	2.117996	2.612138
		10	1.297624	1.412484	1.987195	2.452502

topology generator to automatically generate routable topologies. The generator has similar parameters to ones specified in [16] namely: number of nodes *N*, variance index for the degree of each node *D*, and variance index of link capacities *S*. Given a specific value of *D*, the node's degree is generated randomly within [1,*D*]. Accordingly given a specific value of *S*, each link has a random capacity within [100,100×*S*]. Each set of settings is used to generate 1000 different topologies. The values of β for different topologies are shown in Table 2.

Based on the above results, we can point out some of the following observations. First as we could expect Improved Link State Multi-Path routing protocol is better than Link State Multi-Path protocol in terms of traffic balancing. In all topology settings, ILSMP achieves better β than LSMP. Second, LSMP and ILSMP clearly show better traffic balancing than SP and BSP when the number of nodes is not small, e.g. when there are 15, 35 or 45 nodes in the network. When the number of nodes is 5, under most circumstances, SP and BSP can achieve a better route balancing factor. However, LSMP and ILSMP show clear advantage over those protocols when link bandwidth variance is small. Third, when the number of nodes and the average degree are fixed, β decreases when link bandwidths are increasing. This trend can be clearly observed in Figure 9(a) that presents values of β as a function of *S* when the number of nodes in the network is 25 and the average degree is 12. Last, LSMP and ISMP perform well for larger average degrees when the number of nodes and the link capacity variance are fixed.

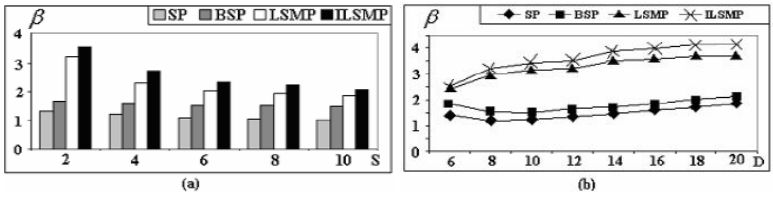


Fig. 9. (a) β for different link capacity ranges; (b) β as a function of node degree

4 Conclusion

In this paper, we have advocated a novel view of routing protocols that take into consideration current trends and forecasts concerning development in ICT industry. Our conclusion is that nodes are more important than links when deciding routes. Taking those findings into consideration we have proposed a Link State Multi-Path (LSMP) routing protocol that shows many advantages over traditional link state routing protocols such as OSPF. LSMP has very simple algorithms. It easily supports local optimization (ILSMP) and can achieve much better traffic balancing in large and complex networks. Due to the fast convergence, LSMP is a good starting point for better QoS in carrier grade IP networks.

References

1. Molinero-Fernández, P.: Circuit Switching In The Internet, Dissertation, Stanford University (2003)
2. Hedrick, C.: Routing Information Protocol, RFC 1058, IETF (1988)
3. Moy, J.: OSPF Version 2, RFC1247, IETF (1991)
4. Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., Weiss, W.: An architecture for Differentiated Service, RFC2475, IETF (1998)
5. Baker, F., Chan, K., Smith, A.: Management Information Base for the Differentiated Service Architecture, RFC3289, IETF (2002)
6. Bernet, Y., Blake, S., Grossman, D., Smith A.: An Informal Management model for Diffserv Routers, RFC3290, IETF (2002)
7. Braden, R., Clark, D., Shenker, S.: Integrated Service in the internet Architecture: An overview, RFC1633, IETF (1994)
8. Guerin, R.A., Orda, A., Williams, D.: QoS routing mechanisms and OSPF Extensions, GLOBECOM (1997)
9. Apostolopoulos, G.; Guerin, R.; Kamat, S.: Implementation and performance measurements of QoS routing extensions to OSPF, INFOCOM (1999)
10. Li, X., King-Shan, L., Jun W., Nahrstedt, K.: QoS extension to BGP, 10th IEEE International Conference (2002)
11. Apostolopoulos, G., Williams, D., Kamat, S., Guerin, R., Orda, A., Przygienda, T.: QoS Routing Mechanisms and OSPF Extensions, RFC 2676, IETF (1999)
12. Jong-Joon, H.; Seung-Hoon, K.; Kyoon-Ha, L.: QoS routing schemes for supporting load balancing, IEEE HSNMC (2002)

13. Yu-Kung, K., Copeland, J.A: Scalability of Routing Advertisement for QoS Routing in an IP Network with Guaranteed QoS, GLOBECOM (2000)
14. Harmatos, J.: A heuristic algorithm for solving the static weight optimisation problem in OSPF networks, GLOBECOM (2001)
15. Sahoo, A.: An OSPF based load sensitive QoS routing algorithms using alternate path, Computer Comunications and Networks (2002)
16. Zhang, P., Bai, X., Kantola, R.: A routing scheme for optimizing multiple classes in a DiffServ network, HPSR (2004)

Appendix A: Routing Algorithm Based on Breadth First Search

```
N: Total number of nodes in the network;
S: Set of nodes already calculated;
LastS: Set of nodes added into S last time;
CurrS: Set of nodes being dealt with currently;
pc: Position Code;
IntfS: Set for interfaces;

For each node { Node.pc = 0;
                Node.IntfS = ∅; }
S = source_node;
IntfS = GetNeighbor(source_node, S);
LastS = IntfS;
S = S∪IntfS;
Source_node.IntfS = NULL;
For (node∈IntfS) { Node.pc = 1;
                   Node.IntfS =nodes; }
while (length(S)) ≠ N) {
     tempS = ∅;
     for (node1∈lastS) {
        CurrS = GetNeighbor(node1,S);
        for (node2∈CurrS) {
           if (node2∉tempS)
             node2.pc = node.pc + 1;
             node2.IntfS = node2.IntfS∪node1.IntfS;}
        tempS = tempS∪CurrS;}
     LastS = tempS;
     S=S∪LastS;}
```

A Simplified Routing and Simulating Scheme for the LEO/MEO Two-Layered Satellite Network

Zhe Yuan, Jun Zhang, and Zhongkan Liu

School of Electronic and Information Engineering,
Beijing University of Aeronautics and Astronautics,
Beijing 100083, P.R. China
yuan_zhe1977@yahoo.com.cn

Abstract. The demand for global mobile communications calls forth the research and development of the satellite networks using non- geostationary orbit satellites. In this paper, a LEO/MEO two-layered satellite network architecture is presented. The routing scheme based on an eclectic inter-satellite link handover strategy for the satellite network is proposed then. And a simplified simulation scheme is also introduced in this paper. Finally the performance of the satellite network and routing strategy is evaluated through the simulation scheme.

1 Introduction

Satellite networks can provide global coverage for personal communications and will become the effective extension of the ground-based mobile communication networks. Multi-layered satellite networks with LEO, MEO and GEO satellites have flexible architecture and high performance in long distance broadband data transmission. Several schemes for multi-layered satellite networks have been proposed. The *Celestri* system [1] consists of 9 GEO satellites and 63 LEO satellites. The *Spaceway* system [2] consists of 16 GEO satellites and 20 LEO satellites. [3] proposes a multi-layered satellite network architecture constructed by satellites in three layers. A satellite-over-satellite (SOS) network is described in [4] with satellites in a multi-layered architecture.

In this paper, we propose a two-layered satellite network consisting of a LEO layer and a MEO layer. The LEO and MEO satellites have much shorter propagation delay than the GEO satellites. The satellites in LEO layer are mainly responsible for the user accesses, and the satellites in MEO layer take charge of routing and data transmission. A simple and efficient routing strategy is also proposed to accommodate with the network architecture.

This paper is organized as follows. In Section 2, the two-layered satellite network architecture is described. In Section 3, the routing strategy for this satellite network is presented in detail. A simplified simulation scheme for satellite networks is given in Section 4. And Section 5 presents the simulation result. Finally, this paper is concluded in Section 6.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 742 – 751, 2005.

2 Network Architecture

The LEO/MEO two-layered satellite network consists of 66 LEO satellites and 8 MEO satellites. The 66 LEO satellites are uniformly distributed in six LEO polar orbits with the altitude of 780 kilometers as the Iridium system [5]. Each orbit has 11 LEO satellites. And the 8 MEO satellites are uniformly distributed in two MEO polar orbits with the altitude of 10350 kilometers. Each orbit has four MEO satellites. The architecture is shown as Figure 1.

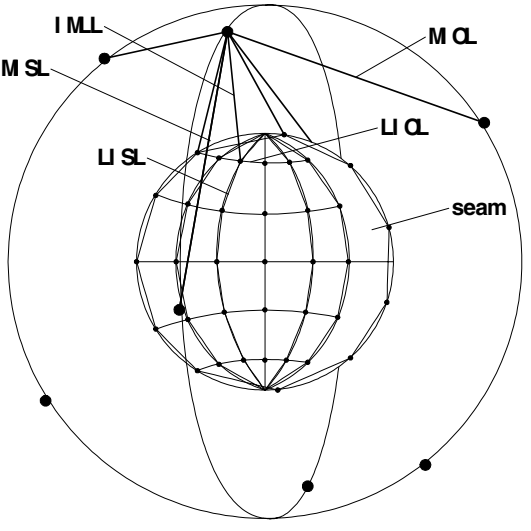


Fig. 1. Architecture of the satellite network

In the LEO layer, each LEO satellite has four inter-satellite links. Two are links connecting with the two adjacent satellites in the same orbit, called the LEO Intra-orbit Links (LISLs). The other two are links connecting with the two adjacent satellites in the left and right neighbor orbits, called the LEO Inter-Orbit Links (LIOLs). In the LEO layer, there are two regions in which the satellites in neighbor orbits are counter-rotating. The two regions are called the "seam". The counter-rotating in the seam causes the frequent handover of the LIOLs crossing the seam. So if the LEO satellite is just at the edge of the seam, it does not have the left or right LIOL crossing the seam. It only has three inter-satellite links. We also do not consider the situation where the LEO satellite switch off their inter-satellite links in the vicinity of the poles.

In the MEO layer, each MEO satellite also has four inter-satellite links. Two are links connecting with the two adjacent satellites in the same orbit, called the MEO Intra-orbit Links (MISLs). The other two are links connecting with the two adjacent satellites in the left and right neighbor orbits, called the MEO Inter-Orbit Links (MIOLs). The MEO layer also has the seam. But the MIOLs crossing the seam can last a much longer time than LIOLs, so the MEO satellite at the edge of the seam will maintain this type of MIOL. Thus, all MEO satellites have four inter-satellite links.

And we also do not consider the situation that MEO satellite should switches off its inter-satellite links in the vicinity of the poles.

Moreover, between the LEO layer and the MEO layer, there exist inter-layer links (IMLLs). Each LEO satellite selects a MEO satellite to establish a IMLL. The rule of the IMLL establishment will be introduced in the next section.

3 Routing Strategy

The frequent handover of inter-satellite links is the main reason for the dynamic topology change of the satellite networks. Each type of inter-satellite links has its own handover characteristic. In this two-layered satellite network, LIOLs, LISLs and MISLs will not be handed over. MIOLs also will not be handed over unless it resides in seam. The IMLLs will frequently be handed over because of the difference of the motion direction and speed between the MEO satellites and LEO satellites. It causes the topology of the two-layered satellite network changes intensely and lowers the performance of routing protocol. A method to control the handover of the IMLLs is necessary to the satellite network. Usually the number of satellite nodes in the satellite network is not too many, so we think the routing problem in the satellite network will simply be solved by the modified OSPF or other dynamic routing protocols.

In this two-layered satellite network, the LEO satellites reside at the different altitude from the MEO satellites. According to the Kepler's law, the LEO and MEO satellites have different rate of rotating. When a MEO satellite run into the visible range of the LEO satellite, an IMLL can be established between them. And when the MEO satellite runs out of the range, the IMLL between them will be cut off by the earth's surface. The ON/OFF switches of IMLL happen frequently during the system period, shown as Figure 2.

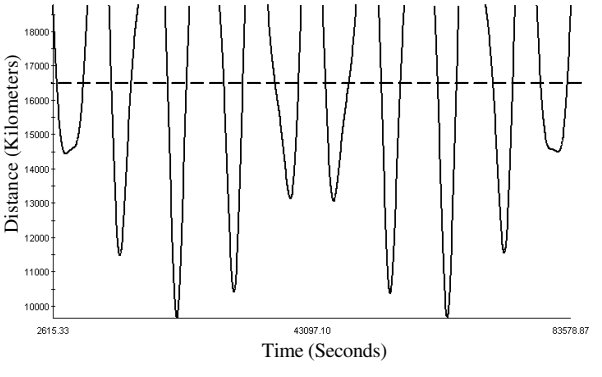


Fig. 2. Dynamic change of the IMLLs

Definition 1 (Inter-satellite Visibility): The MEO satellite M_j is visible to the LEO satellite L_i is that a line joining the satellite L_i and the satellite M_j has no intersection with the earth's surface. So the Inter-satellite Visibility $V_{M_j \to L_i}$ is define as:

$$V_{M_j \to L_i} = \begin{cases} 0 & \text{The intersection exists.} \\ 1 & \text{The intersection does not exist.} \end{cases} \tag{1}$$

Definition 2 (IMLL's Lifetime): At time $t_{ON,k}\,(k = 1,2,3,\cdots)$, a MEO satellite M_j becomes visible to the LEO satellite L_i, then a IMLL $IIMLL_{L_i \to M_j}$ can be established. And at time $t_{OFF,k}\,(k = 1,2,3,\cdots)$, the MEO satellite M_j will become invisible to the LEO satellite L_i, then the IMLL $IMLL_{L_i \to M_j}$ must be cut off. The lifetime $T_k(M_j \to L_i)$ of $IMLL_{L_i \to M_j}$ is the lasting time from $t_{ON,k}$ to $t_{OFF,k}$, i.e.

$$\begin{cases} T_k(M_j \to L_i) = t_{OFF,k} - t_{ON,k} \\ t_{ON,k} : V_{M_j \to L_i} \uparrow \\ t_{OFF,k} : V_{M_j \to L_i} \downarrow \end{cases} \tag{2}$$

where $k = 1,2,3,\ldots$

The values of the IMLL's lifetime are also dynamically changing during the system period. The maximal value is defined as the Maximal IMLL's Lifetime $T_{\max}$.

Definition 3 (IMLL's Residual Lifetime): At time t, the residual lifetime of $IIMLL_{L_i \to M_j}$ is defined as:

$$T(M_j \to L_i, t) = t_{OFF,k} - t,\ t_{ON,k} \le t \le t_{OFF,k},\ (k = 1,2,3,\ldots) \tag{3}$$

Definition 4 (IMLL's Propagation Delay): At time t, the propagation delay of $IIMLL_{L_i \to M_j}$ is defined as:

$$D(M_j \to L_i, t) = r(M_j \to L_i, t) \Big/ C \tag{4}$$

where $r(M_j \to L_i, t)$ is the distance from the MEO satellite M_j to the LEO satellite L_i at time t, and C is the velocity of light.

The minimal propagation delay of IMLLs is

$$D_{\min} = r_{\min} \Big/ C = (H_m - H_l) \Big/ C \tag{5}$$

where $r_{\min}$ is the minimal distance from LEO satellites to MEO satellites, H_m is the altitude of MEO satellites, H_l is the altitude of LEO satellites. Actually $r_{\min}$ is equal to the difference value of H_m and H_l.

Definition 5 (IMLL's Cost Function): At time t, the cost function of $IIMLL_{L_i \to M_j}$ is defined as

$$P_{L_i, M_j}(t) = \alpha \cdot \left[\frac{D(M_j \to L_i, t)}{D_{\min}} \right] + \beta \cdot \left[\frac{T(M_j \to L_i, t)}{T_{\max}} \right]^{-1} \tag{6}$$

where α is the weight coefficient of the IMLL's propagation delay, β is the weight coefficient of the IMLL's residual lifetime.

Suppose that the value of $V_{M_j \to L_i}$ changes from 1 to 0 at time t. The LEO satellite L_i will cut off $IIMLL_{L_i \to M_j}$ connecting with the MEO satellite M_j and find another MEO satellite M_k from $\{M_l \mid l = 1, \cdots, 8 \text{ and } V_{M_k \to L_l} = 1\}$ to establish a new IMLL $IIMLL_{L_i \to M_k}$ to maintain its connection with the MEO layer. The new IMLL $IIMLL_{L_i \to M_k}$ must meet the conditions that the cost function value of $IIMLL_{L_i \to M_k}$ is the minimal of all, i.e. that:

$$P_{\min_{L_i}} = \min(\{P_{L_i M_l} \cdot V_{M_l \to L_I} \mid l = 1, \cdots, 8 \text{ and } V_{M_l \to L_i} = 1\}) = P_{L_i M_k} \tag{7}$$

Once the LEO satellite L_i finishes its IMLL handover, it multicasts its topology change information to all other satellites. And then all satellites in the network refresh their own routing tables. The routing table calculations are performed using Dijkstra's shortest path first algorithm.

4 Simulation Scheme

Satellite network is a dynamic network. Its topology and delay change all the time. To simulate a satellite network, we need to model the radio channels and design the whole IP or ATM protocol stack. So the simulation of constellation satellite networks is always a hard work.

This section presents a simplified simulation scheme for the satellite network. It uses a ground fixed network based on standard TCP/IP protocols stack. We introduce the dynamic changes models of the satellite network into this ground fixed network. Then the ground fixed network will also have dynamic topology and delay characteristic as the satellite networks.

4.1 Dynamic Topology Simulation

Network Topology is the link connectivity of all nodes in the network. The Dynamic topology characteristic of satellite network is the ON/OFF changes of the inter-satellite links.

To simplify the problem resulted from the dynamic topology of satellite network, we divide the whole system period into N time slots. In each slot, although the positions of satellites are keeping changing all the time, the connectivity of links stays unchangeable, i.e. the topology of the satellite network is fixed during a slot. So the dynamic topology of satellite network can be described as a series of fixed topology. This method is much similar to FSA or Snapshot [6,7].

To simulate the LEO/MEO two-layered satellite network, firstly we model it in OPNET using a ground fixed network, shown as Figure 3. The ground fixed network consists of 66 LEO routers (as 66 LEO satellites) and 8 MEO routers (as 8 MEO satellites). Each MEO router has 4 links connecting with its neighbor MEO routers

and 66 links connecting with all LEO routers. Each LEO router has 4 links connecting with its neighbor LEO routers and 8 links connecting with all MEO satellites.

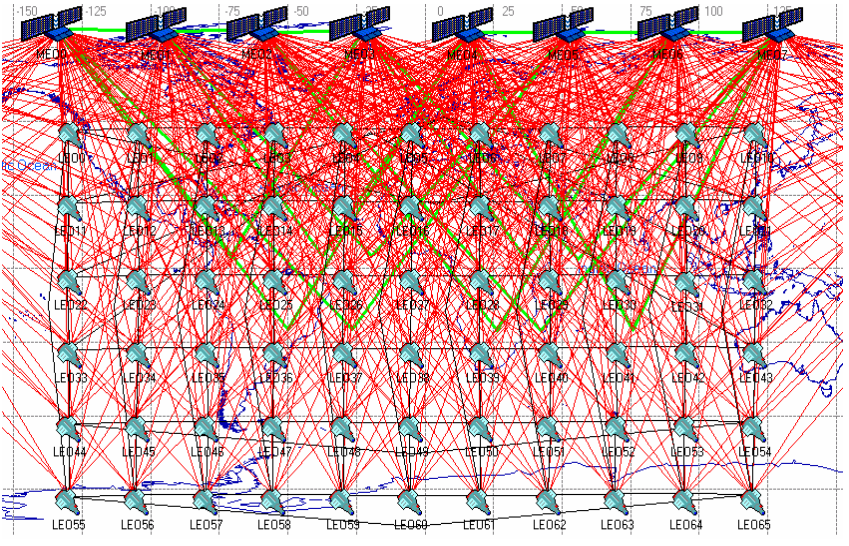


Fig. 3. Topology of the ground fixed network

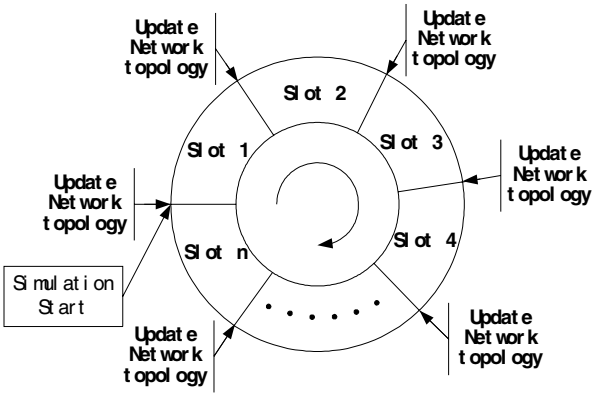


Fig. 4. Dynamic topology simulation

In the simulation, a series of time slots is defined. The value of the slot's length is set to 60 seconds. At the beginning of each time slot, the topology of this ground fixed network will be calculated according to the actual changes of the satellite network, shown as Figure 4.

At the beginning of slot *i*, the position of each satellite can be calculated. Then each LEO satellite can select a MEO satellite to establish IMLL with it according to the handover controlling principle introduced in Section 3. Accordingly, the links in the ground fixed network must be changed too.

For example, in slot *i*-1, LEO satellite L_i has a IMLL $IIMLL_{L_i \to M_j}$ with the MEO satellite M_j . But because of the relative change of their positions, the IMLL $IIMLL_{L_i \to M_j}$ can not be maintained in slot *i*. So LEO satellite L_i establishes a new IMLL $IIMLL_{L_i \to M_k}$ with MEO satellite M_k . Accordingly in the simulation network, at the beginning of slot *i*, the LEO router *i* turn off its link $Link_{ij}$ connecting with the MEO router *j* and turn on its link $Link_{ik}$ connecting with the MEO router *k*.

4.2 Dynamic Delay Simulation

It is assumed that in each time slot, the position change of satellites can be ignored. Then the topology and delay of the satellite network are both fixed during a slot. So the dynamic delay change of satellite networks can also be simulated by the same way as the dynamic topology, shown as Figure 5.

By this way, the ground fixed network modeled by the OPNET also has the dynamic topology and delay characteristic as the satellite networks.

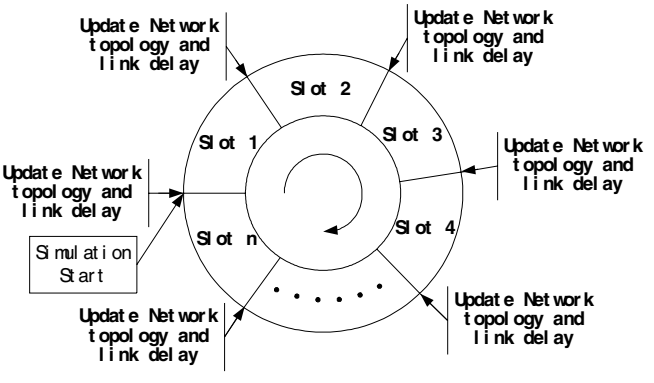


Fig. 5. Dynamic topology and delay simulation

5 Simulation Results

The OPNET is selected as the software tool for the simulation. The simulation network is modeled as Section 4. The OSPF is selected as the dynamic routing protocol of the satellite network. And it can be replaced by other routing protocols. Two services are defined in the simulation. One is the service of voice over IP, the other is a video conference.

In the simulation, we define three different values of (α, β) in (5). When $(\alpha, \beta) = (0,1)$, the selected IMLL is a inter-layer link with the minimal distance from the MEO satellite to the LEO satellite. When $(\alpha, \beta) = (1,0)$, the selected IMLL is a link with the maximal coverage time of the MEO satellite to the LEO satellite. When

$(\alpha, \beta) = (1,1)$, the selected IMLL is a link with the compromise between minimal distance and maximal coverage time.

Figure 6 and 7 shows the delay performance of the simulated satellite network. For the video conference service and the voice over IP service, the simulation compares the end-to-end delay among the minimal distance strategy, the maximal coverage time strategy and the eclectic strategy. Because the minimal distance strategy always selects a shortest transmission path, it has the better delay performance than the other two. Accordingly, the maximal coverage time strategy has the worst performance. The eclectic strategy intervenes between them and is much close to the minimal distance strategy.

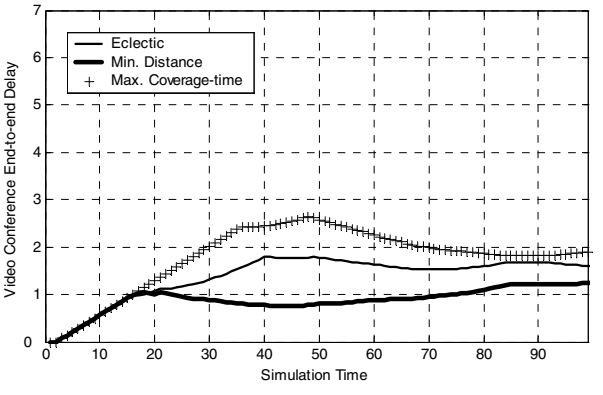


Fig. 6. Delay performance of the video conference service

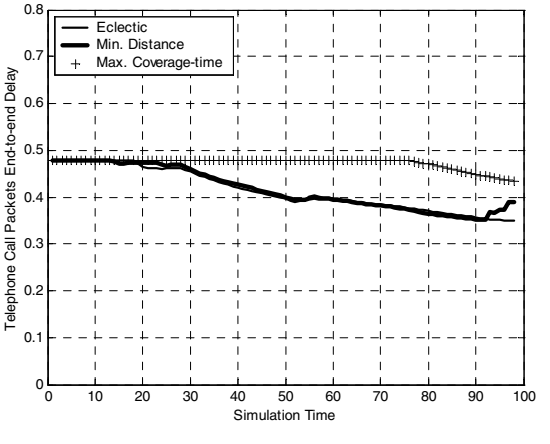


Fig. 7. Delay performance of voice over IP service

In Figure 8 and 9, the IP traffic dropped and dynamic routing protocol traffic performance comparison is depicted. The satellite network with the minimal distance strategy has more attempts to hand over its inter-layer links. So the number of network re-routing is more than the others'. When inter-layer link's handover occurs,

the IP traffic dropped and OSPF protocol traffic send will increase obviously, and then the performance of the satellite network is degraded. The maximal coverage time strategy appears better performance in these two aspects. And the eclectic strategy also intervenes between them.

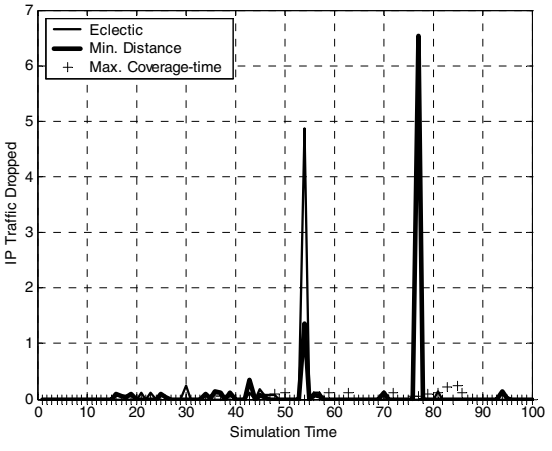


Fig. 8. Protocol traffic performance of IP traffic dropped

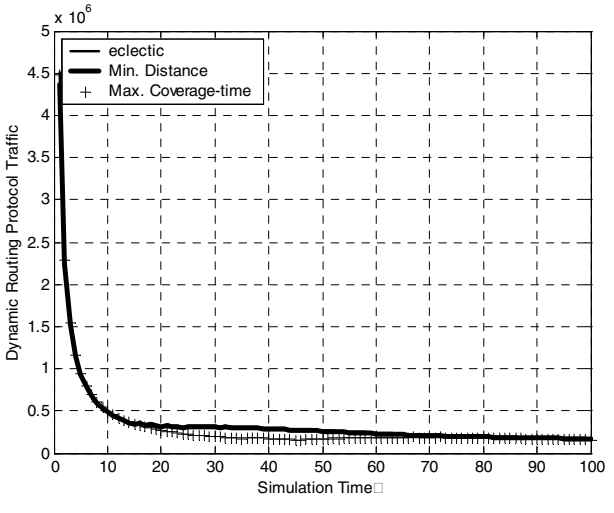


Fig. 9. Protocol traffic performances of OSPF protocol traffic send.

According to the results of simulation, the minimal distance strategy is applicable to common data traffic service such as database, web and email. The eclectic strategy has better performance in the end-to-end delay and traffic dropped. So it is acceptable for the traffic of real-time multimedia service. And it is much adaptable to the broadband satellite network.

6 Conclusion

In this paper, we construct a two-layered satellite network with LEO satellites and MEO satellites. A link handover strategy based on link transmission delay and coverage time is proposed to manage the inter-layer link's handover. Accordingly the routing in this satellite network can simply be solved by Dijkstra's shortest path first algorithm. To evaluate the network performance, a simplified simulation scheme is also introduced in this work. It uses the ground fixed network with the dynamic topology and delay models to emulate the satellite network. The simulation result shows that the eclectic link handover strategy has a better performance than the minimal distance and the maximal coverage time strategy.

References

1. Evans, J.V.: Proposed U.S. Global satellite systems operating at Ka-Band. Proc. IEEE Aerospace Conference, Vol. 4 (1998) 525-537
2. Farserotu, J., Prasad, R.: A Survey of Future Broadband Multimedia Satellite Systems, Issues and Trends. IEEE Commu. Mag., Vol. 38 (2000) 128-133
3. Akyildiz, I.F., Ekici, E., Bender, M.D.: MLSR: A Novel Routing Algorithm for Multilayered Satellite IP Networks. IEEE ACM Transactions On Networking, Vol. 10 (2002) 411-424
4. Lee, J., Kang, S.: Satellite over Satellite (SOS) Network: A Novel Architecture for Satellite Network. INFOCOM'2000, Vol. 1 (2000) 315-321
5. Keller, H., Salzwedel, H.: Link Strategy for the Mobile Satellite System Iridium, IEEE 46th Vehicular Technology Conference, Vol. 2 (1996) 1220-1224
6. Chang, H.S., Kim, B.W., Lee, C.G., Min, S.L., Choi, Y., Yang, H.S., Kim, D.N., Kim, C.S.: FSA-Based Link Assignment and Routing in Low-Earth Orbit Satellite Networks, IEEE Trans. Veh. Tech., Vol. 47 (1998) 1037-1048
7. Gounder, V.V., Prakash, R., Abu-Amara, H.: Routing in LEO-based satellite networks, IEEE Wireless Communications and Systems, Vol. 22 (1999) 1-6

ARS: An Synchronization Algorithm Maintaining Single Image Among Nodes' Forwarding Tables of Clustered Router*

Xiaozhe Zhang, Wei Peng, and Peidong Zhu

School of Computer Science, National University of Defense Technology, 410073 Changsha, China
nudtzhangxz@263.net

Abstract. With the rapid development of Internet, traditional centralized routers can not meet the requirements of next generation Internet for reliability, performance scalability and service scalability. Clustered routers will be the most important components of future Internet. It is very important for clustered routers to maintain the same forwarding table images among clustered router nodes. Different synchronization mechanisms have variant performance to control plane and packet forwarding plane. This paper proposes an asymmetrical forwarding table synchronization algorithm - ARS (Asymmetrical Route Synchronization). It fits the requirements of massively parallel clustered router architecture perfectly. Continuous route flapping of backbone network burdens control plane of core router and causes lots of synchronization costs for clustered router. ARS uses route cache to predict new best route when original best route is deleted and decreases synchronization costs greatly.

1 Introduction

In order to match the rapid development of the Internet, network devices must support more powerful computing ability, packet forwarding ability and huge density of physical interfaces. It makes traditional centralized control plane of network devices unable to meet the requirements of next generation Internet in reliability, performance scalability and service scalability. Clustered router is a promising and attractive architecture in future. Several gigabit routers and future terabit routers support multi-chassis interconnection and backplane extension technology, e.g. Cisco CRS[1] Juniper T640[2] and Anvici TSR[3]. A number of PCs or low-cost routers can also be interconnected and work as a massively parallel router, e.g. Pluris Massively Parallel Router[4], Suez[5]. The massive scalability of clustered router challenges traditional router software architecture, operating system supports and routing protocol implementation model.

* The work has been co-supported by National Natural Science Foundation of China(NSFC), under agreement no. 90412011, 90104001 and 90204005, National Basic Research Priorities Programme of China, under agreement no.2003CB314802.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 752–761, 2005.

Clustered router is a single image[6] system composed of interconnected routing nodes, which can be commercial routers or PCs. "single image" means that interconnected routing nodes are not network but a single router in the view of adjacent routers and Internet. In the packet forwarding plane, the prerequisite of being a single router is that routing nodes of clustered router apply same routing policy on the packets with same destination IP address. It requires that all routing nodes have the identical packet forwarding table image. How to keep the single image of routing nodes' forwarding tables efficiently is very important and critical to the performance of clustered router's control plane and forward plane.

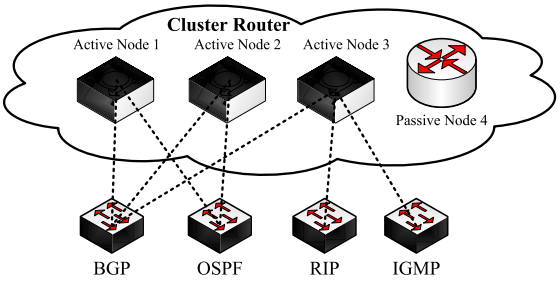


Fig. 1. The asymmetric distribution of routing protocols

Routing protocols store routing prefixes in local prefix databases and executes routing decision procedure to generate packet forwarding tables. The distribution of routing protocols among routing nodes, the frequencies and features of prefixes updating determine the algorithm and mechanism maintaining the single image among routing nodes' packet forwarding tables.

Fig 1 shows an asymmetric distribution of routing protocols among clustered router. routing node 1, 2 and 3 not only forward IP packets but also execute routing protocol to exchange routing prefixes. Routing node 4 is only a packet forwarder and do not execute any protocol. Distributed BGP process resides on node 1, 2 and 3. Distributed OSPF process resides on node 1 and 2; Centralized RIP and IGMP processes reside only on node 3. the detailed discussions[7,8,9] on distributed implementation mechanisms of routing protocols are beyond our caring. Most distributions are in a asymmetric way, for many routing protocols do not support distributed implementation mechanism currently. We call the routing node with routing protocols processes resided *active node*, e.g. routing node 1, 2 and 3, call packet forwarding node without any routing protocols processes resided *passive node*, e.g. routing node 4. We call the set of routing prefixes rooted from local node or learned by routing protocol resided *local protocol database*. *Local forwarding table* is generated by routing decision procedure from local protocol database. *Global forwarding table* is a globally consistent forwarding table generated by route synchronization algorithms among local forwarding

tables of active nodes. Now clustered router is a nodes set including two node types: active node and passive node.

Frequent prefixe burst updates and link-fails[10,11,12] impose huge route change operations on local forwarding table. 40 to 55 percent[13] of BGP prefixes updates flapping also burden local forwarding tables. Lower synchronization cost and more efficient synchronization algorithm are critical to the performance of packets forwarding plane and control plane and helpful to improve high layer routing protocols.

Based on more generalized clustered router composed of active nodes and passive nodes, this paper proposes a route synchronization algorithm—ARS (Asymmetrical Routes Synchronization) algorithm. ARS requires that active nodes store routing protocols' prefixes and local forwarding tables locally and only global forwarding table is replicated on each routing node. It does not require same storage ability for all routing nodes and very suitable for this asymmetric clustered router.

2 Traditional Route Synchronization Algorithms

Most of early clustered routers use CBSA (Centralized Broadcast Synchronization Algorithm) to maintain single image among routing nodes, e.g. pluris massively parallel router[4]. A powerful routing node is designated as route server. Route server collects route information from others active nodes or executes all routing protocols and control functions locally. Global forwarding table is generated on the basis of route server's protocol database and broadcasted to all other routing nodes when forwarding routes are changed. CBSA is a centralized algorithm and used widely in route synchronization of clustered routers and route distribution between control processor and NPs of line cards. But it decreases the reliability of clustered router greatly and makes control plane be performance bottleneck of whole system. Most commercial clustered routers do not use it for reliability requirements of backbone network. CBSA broadcasts route change messages, only global forwarding table is changed. It is the most efficient algorithm and can be used to evaluate the performances of new route synchronization algorithms.

Commercial routers supporting cluster technology use RRA (Redundant Replication Algorithm) to achieve high availability, distribut control plane functions among routing nodes and improve critical routing protocols' performance currently. IOS XR of Cisco CRS[1] is a typical distributed modular router software based on symmetric active nodes. CRS shelf is an active nodes with powerful computing ability and storage ability. Every CRS shelf can execute routing protocols and control functions as an independent router. When multi-CRS shelves are interconnected, routing protocols, UI and management modules can be assigned to any shelf on user configuration. RRA is used to maintain the single image among CRS shelves' forwarding tables. Each shelf includes two protocol databases: global protocol database and local protocol database. Global protocol database consists of local protocol database and other shelves' local proto-

col databases. CRS shelf broadcasts route change messages when local protocol database is changed. Global forwarding table is generated on the basis of each shelf's global protocol database independently. Since each routing node need contain huge global protocol database, RRA can only be used in symmetric clustered router architecture and does not support generalized clustered router composed of active nodes and passive nodes.

3 ARS Synchronization Algorithm

The software forwarding table of traditional centralized router is organized in tree. Routing protocols exchange route prefixes with adjacent routers and store them in software forwarding table. The routes with same network address are stored in same leaf node's route set R_{list} of forwarding table tree. The best route of route set R_{list} is identified with best route pointer r_b. We propose a new route synchronization algorithm—ARS (Asymmetrical Route Synchronization) on the basis of traditional software forwarding table.

In order to decrease the forwarding table size of each routing nodes, the routes are stored distributively. Not all routes but global best routes are replicated on each routing node. The leaf node structure is extended to contain global best route pointer r_g, which points to the best route of leaf nodes containing same network address routes of local forwarding tables. Now each active node contains the routes in local forwarding table and the global best routes coming from other active nodes. Passive nodes only keep the global best routes from other active nodes. With smaller forward table size, passive nodes can concentrate on packets forwarding without excrescent expensive storage and computing resources. Route synchronization happens only when a few global best routes are deleted or active nodes receive new routes, which are better than original global best routes.

In order to minimize ARS's synchronization costs, which are caused by frequent routes flapping or link fails, ARS uses route cache to predict secondly best route. Every leaf node caches another active nodes' local best route, which is better than all routes of local route set R_{list} except local best route. New global best route can be predicted by route cache when original global best route is deleted. A route cache pointer r_c is added in each leaf node structure. r_c can be null or caches a route from another active node and satisfies the expression $r_c < r_b \bigcap (\forall\, r, r \in R_{\text{list}} \bigcap (r = r_b \bigcup r_c > r)))$.

3.1 Symbols and Definitions

We simplify the description of route information and define route r_1 as triple $< dest, metric, nodeid >$. *dest* means the network address of route r_1. *metric* means the metric value of route r_1. *nodeid* is the identifier of route r_1's original routing node. Formula $r_1 > r_2$ means that route r_1 is better than route r_2. Formula $r_1 \equiv r_2$ means that each field of triple r_1 is equal to corresponding field of triple r_2. Tab 1 gives the functions to be used in ARS algorithm.

Formula 1 gives the define of tree leaf node structure.

$$L_d = \langle\, r_b\ , R_{\text{list}} \rangle, R_{\text{list}} = \{\ r_i \mid 1 \le i \le k\ \} \cap\ r_b \in R_{\text{list}} \\ \cap\ (\ \forall\, r_i\ (\ r_i \in R_{\text{list}} \cap\ (\ r_i \equiv r_b \cup r_i < r_b\ ))) \tag{1}$$

ARS algorithm extends L_d, add global best route field r_g and route cache field r_c. New leaf node L'_d structure is defined in formula 2.

$$L'_d = \langle\, r_g\ ,\ r_b\ ,\ r_c\ , R_{\text{list}} \rangle, R_{\text{list}} = \{\ r_i \mid 1 \le i \le k\ \} \\ \begin{cases} r_b \in R_{\text{list}} \ \bigcap\ (\forall r_i (r_i \in R_{\text{list}} \bigcap (r_i \equiv r_b \ \ \bigcup r_i < r_b))) \\ (\ r_g\ >\ r_b\ \bigcap\ r_g\ \notin R_{\text{list}}) \ \ \bigcup\ r_g\ \equiv\ r_b \\ r_c\ =\ null\ \bigcup\ (\ r_c\ \notin R_{\text{list}} \ \ \bigcap\ r_c\ <\ r_b\ \bigcap \\ \qquad\qquad (\ \forall r, r \in R_{\text{list}} \ \ \bigcap (\ r \equiv r_b \ \ \bigcup r_c > r))) \end{cases} \tag{2}$$

Table 1. Functions of ARS algorithm

Function name	Description
$Dest(r_1)$	return the *dest* field of triple r_1
$ID(r_1)$	returns the *nodeid* field of triple r_1
$myid()$	return the identifier of local routing node
$Broadcast(opt, r_1)$	Broadcast route r_1 with the *opt* code on inner network $opt = \begin{cases} ADD \text{ announce new route } r_1 \\ DEL \text{ withdraw route } r_1 \end{cases}$
$Getleafnode(dest)$	match a leaf node with network address 'dest' in forwarding table and return it
$Best(R_list)$	select the best route in set R_{list} and return it

3.2 The Description of ARS

ARS includes two parts: one is the API (Application Program Interface) procedures of global forwarding table, which are provided for high routing protocols layer to add/delete routes. The other is the procedures of route synchronization messages among routing nodes. The API procedures of ARS includes two functions:

- *Proto_Addroute* is used to add new route into global forwarding table.
- *Proto_Dleroute* is used to delete original route from global forwarding table.

Fig 2 gives the pseudo-codes of these two functions. The route r_i is the route of high layer routing protocol to be added into or delete from global forwarding table. When routing protocols call function *Proto_Addroute* to add route r_i into global forwarding table, the leaf node L'_d need be located firstly. Function *GetLeafNode* traverses forwarding table structure and locate the leaf node of route r_i. It can also create a new leaf node structure automatically, if there doest not exist route r_i's leaf node. Then route r_i is added into set R_{list} and compared

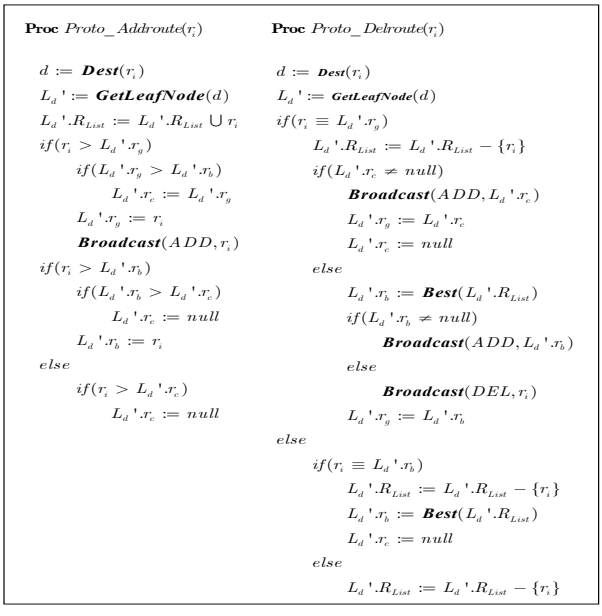


Fig. 2. API implementation algorithm of global forwarding table

with the global best route of leaf node L'_d. If r_i is better than route $L'_d.r_g$, r_i is assigned to $L'_d.r_g$ and a new route synchronization message is broadcasted to other routing nodes. At last original global best route is checked whether it can be cached in route cache pointer $L'_d.r_c$.

When deletion function *Proto_Delroute* is called by high layer routing protocol, route r_i is compared with global best route $L'_d.r_g$. If route r_i is equal to $L'_d.r_g$, new global route need be predicted immediately. The prediction rules are listed in following orderly.

1. If the route cache pointer $L'_d.r_c$ is not null, $L'_d.r_c$ is assigned to $L'_d.r_g$ and broadcasted as new global best route.
2. If the local best route $L'_d.r_b$ is not null, $L'_d.r_b$ is assigned to $L'_d.r_g$ and broadcasted as new global best
3. broadcast route r_i's deletion message and notify other nodes to withdraw it.

If route r_i is not equal to global best route $L'_d.r_g$, r_i is removed from the route set $L'_d.R_{\text{list}}$ and no synchronization message is broadcasted. Local best route $L'_d.r_b$ need be recomputed sometimes.

There are only two types of synchronization messages during route synchronization: route update message and route withdraw message. Fig 3 give the pseudo-codes of message processing functions *Node_Addroute* and *Node_Delroute*. Parameter *srcid* is the routing node identifier, which has sent the route synchronization message. Parameter r_i is the route broadcasted. When

routing node receives a update message with route r_i from inner network, r_i must be compared with global best route $L'_d.r_g$. $L'_d.r_g$ is replaced with route r_i, only if r_i is better than it. $L'_d.r_g$ is deleted or replaced by r_i impliedly if r_i comes from the same routing node as $L'_d.r_g$. Route r_i must meet one of the following conditions when it has occurred:

1. r_i is the route cached by route cache pointer $L'_d.r_c$ of routing node *srcid* and broadcasted as a new global best route candidate. If r_i is a route belonging to local node and equal to local best route $L'_d.r_b$, it is valid. Otherwise local node need broadcast message to correct the invalid candidate route r_i.
2. r_i is the local best route of routing node *srcid*. Now $L'_d.r_b$ is compared with route r_i. $L'_d.r_b$ is broadcasted only if it is better than r_i.

```
Proc Node_Addroute(srcid,r_i)
d := Dest(r_i)
L_d' := GetLeafNode(d)
if((r_i > L_d'.r_g)OR(srcid ≡ ID(L_d'.r_g)))
    if(ID(r_i) ≠ myid)
        if(r_i > L_d'.r_b)
            L_d'.r_g := r_i
        else
            Broadcast(ADD, L_d'.r_b)
            L_d'.r_g := L_d'.r_b
            if((ID(r_i) ≡ srcid)AND(r_i > L_d'.r_c)AND
                (r_i > BEST(L_d'.R_List − {L_d'.r_b})))
                L_d'.r_c := r_i
    else
        if(L_d'.r_b ≠ r_i)
            if(L_d'.r_b ≡ null)
                Broadcast(DEL, r_i)
            else
                Broadcast(ADD, L_d'.r_b)
        L_d'.r_g := L_d'.r_b
else
    if((ID(r_i) ≡ srcid)AND(r_i > L_d'.r_c)
        AND(r_i > BEST(L_d'.R_List − {L_d'.r_b})))
        L_d'.r_c := r_i
```

```
Proc Node_Dlroute(srcid,r_i)
d := Dest(r_i)
L_d' := GetLeafNode(d)
if(r_i ≡ L_d'.r_g)
    if(L_d'.r_b ≠ null)
        Broadcast(ADD, L_d'.r_b)
        L_d'.r_g := L_d'.r_b
else
    if(r_i ≡ L_d'.r_c)
        L_d'.r_c := null
```

Fig. 3. The procedures of route synchronization messages

4 Algorithm Evaluation

We construct three cases to simulate typical scenes of core router. The first case simulates that a routing protocol of core router receives prefix burst updates and inserts them into global forwarding table in a few seconds. The second case simulates that a critical link failed or a routing protocol restarted make a active node's local forwarding table flap. The third case is that continuous route

flapping of BGP routing protocol and other protocols make all active nodes' local forwarding tables flap in different probabilities. When new routes are inserted into local forwarding tables, only the routes better than global best routes are broadcasted in ARS and CBSA algorithms. So only RRA algorithm is compared with ARS in the first case. ARS, CBSA and RRA are compared in second and third cases.

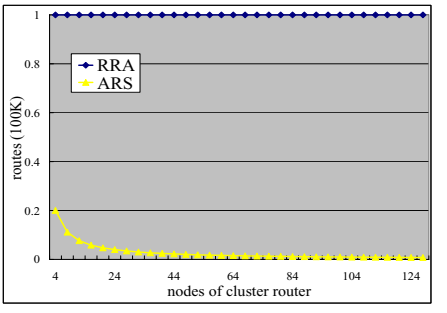


Fig. 4. Burst insertion of 100K routes

All routing nodes are interconnected with inner switch network during the simulation. Each routing node can reach another one directly. Inner switch network supports broadcast and unicast. We use the amount of synchronized routes to evaluate the performances of three algorithms. It is a fair measurement and can ignore the influences of different implementation mechanisms, such as route packing, transmission mechanism and different data structures.

We select an active node randomly and insert 100K routes into its local forwarding table in the first case. Fig 4 compares the performances of RRA and ARS. X-coordinate shows the amount of active nodes in clustered router. Y-coordinate shows the routes synchronized among active nodes. The amount of ARS algorithm's synchronization routes decreases dramatically with the increase of active nodes. ARS is much better than RRA.

Fig 5 shows the average route amount of three algorithms in the second case. A active node is selected randomly and 100K routes of its local forwarding table are flapped each time. RRA need notify all other routing nodes when local forwarding table's route is changed. Its synchronization route amount is constant 200K. ARS and CBSA need route synchronization only when global best routes are changed. Their performance curves are very close in Fig 5.

The frequent route flapping of routing protocols makes active node's local forwarding table flap. We compare three algorithms' performance when all active nodes' local forwarding tables flap in given probability. Fig 6 shows the curves of active nodes increasing from 4 to 128 and each active node flapping in probability 0.5. Fig 7 compares the synchronization route amount of 128 active nodes in different flapping probabilities. ARS is close to CBSA in flapping probability 0.5. Its synchronization cost is only about 25 percent of RRA at the extremity of 128 active nodes and very high flapping probability.

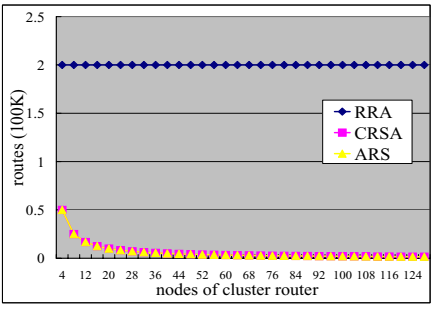


Fig. 5. Route flapping of an active node

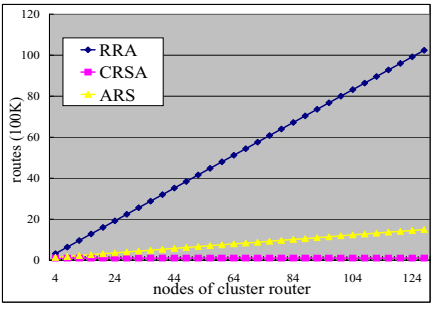


Fig. 6. All active nodes flapping in probability 0.5

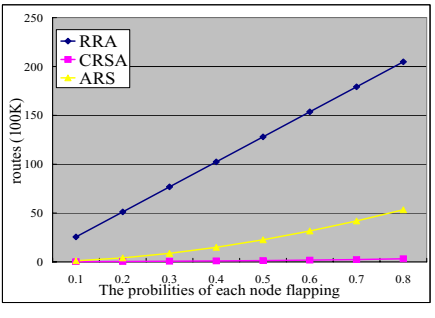


Fig. 7. 128 active nodes in different flapping probabilities

5 Conclusion

The synchronization mechanism of maintaining local forwarding tables' single image is very important to the performances of control plane and forward plane. ARS is an asymmetric route synchronization algorithm. It uses route cache to predict the new best route when original global best route is deleted. It decreases

synchronization cost among clustered route nodes greatly. ARS has following properties:

1. supports asymmetric clustered router platform inherently.
2. supports flexible distribution ways of routing protocols among active nodes.
3. enhances the scalability of clustered router.
4. lower resources requirement for each routing node.
5. lower synchronization costs.

References

1. Cisco Networks. http://www.cisco.com. 2004
2. Juniper Networks. http://www.juniper.net. 2004
3. Avici Systems Technology. http://www.avici.com. 2003.
4. Sam Halabi. Pluris Massively Parallel Routing. Pluris Inc. Whit Paper, 1999
5. P. Pradhan, T. Chiueh. A Cluster-based Scalable and Extensible Edge Router Architecture. ECSL Techincal Report (http://www.cs.sunysb.edu/~prashant/papers/design.ps.gz), 2000.
6. Gong Zhen-Hu,Sun Zhi-Gang. The Architectures of Cluster Router. Research Report, 2004
7. Mitsuru MARUYAMA, Naohisa TAKAHASHI, Members and Takeshi MIEI. CORErouter-1:An Experiental Parallel IP Router Using a Cluster of Workstations, IEICE TRANS.COMMUN, 1997, E80-B(10): 14071414
8. Xipeng Xiao and Lionel M.Ni. Parallel Routing Table Computation for Scalable IP Routers, Proceedings of the IEEE International Workshop on CANPC, Las Vegas, Nevada, USA, 1998, 144158
9. Xiaozhe Zhang, Peidong Zhu and Xicheng Lu. Fully-Distributed and Highly-Parallelized Implementation Model Of BGP4 Based on Clustered Routers, 4th International Conference on Networking, France, 2005.
10. Nina Taft. The Basics of BGP Routing and its Performance in Today's Internet RHDM (Resaux Haut Debit et Multimedia), High-Speed Networks and Multimedia Workshop, France. 2001
11. Gianluca Iannaccone, Chen-Nee Chuah, Richard Mortier, Supratik Bhattacharyya, and Christophe Diot. Analysis of link failures in an IP backbone. In: Proceedings of the second ACM SIGCOMM Workshop on Internet measurment table of contents. Marseille, France: ACM Press, 2002, 237-242
12. Craig Labovitz, G.Robert Malan, Farnam Jahanian. Internet Routing Instability. IEEE/ACM Trans. Networking, vol. 6, no. 5, pp. 515-558, 1998
13. Craig Labovitz, G. Robert Malan, Farnam Jahanian. Origins of Internet Routing Instability. In Proceedings of the IEEE INFOCOM 99, NewYork,1999.

Design and Implementation of Control-Extensible Router*

Baosheng Wang and Xicheng Lu

School of Computer, National University of Defense Technology,
410073 Changsha, China
bswang@nudt.edu.cn

Abstract. The rapid development of Internet requires that control and forwarding plane of traditional IP router should be uncoupled and implemented in standard protocol. At the same time performance scalability and easiness of new function deployment also burden traditional router. In this paper, we propose and implement a new control plane and forwarding plane architecture called Controlling Extensible Router (CER). CER separates the implementation of control plane functions from forwarding plane and hosted it at general purpose server. Forwarding functions of CER are based on network processor and control software is designed on the basis of modular objects. They permit fast creation, configuration and deployment of new services and protocols. Compared with traditional IP router architecture, CER could rapidly employ the new control protocol and enhance the performance of control plane and forwarding plane independently, especially for control plane with using common high computer server.

1 Introduction

The Internet continues along a path of seemingly inexorable growth, at a rate that has almost doubled in size each year. In order to match the Internet expansion speed, network devices must provide more powerful computing ability and packet forwarding ability, and support huge density of physical interfaces. It makes traditional control plane of router unable to meet the requirements of next generation Internet in terms of reliability, performance scalability and service scalability in future.

In current routers, implementations of the control and forwarding planes are intertwined deeply in many ways. The control processors implementing control plane functions are collocated with the line cards that implement forwarding functions and often share the same router backplane. Communication between the control processors and the forwarding line cards is not based on any standards-based

* This work is supported by National Natural Science Foundation of China (NSFC), under agreement no. 90412011 and 90104001, National Basic Research Priorities Programme of China, under agreement no.2003CB314802.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 762 – 771, 2005.

mechanism, making it impossible to interchange control processors and forwarding elements from different suppliers. This also leads to a static binding between forwarding elements and line cards. A router typically has at most two controllers (live and stand-by) running control plane software. Each line card is statically bound to these two controllers. The two controllers, the line-cards to which they are statically bound, and the switch fabric together constitute the router. Current commercial routers are designed for the flexibilities of supporting different interface types and interface bandwidth upgrade. But the upgrade of route controller is troublesome and also very expensive. Network administrators can not use powerful computation and storage resources provided by current PC to assist the poor performance of control plane except that they buy new but expensive router shelf with more powerful controller from same vendor.

Router software is also not based on any standard mechanisms at the same time. The router shelf determines network administrator's choice. The coupling among different routing protocols and interaction between control plane and forwarding plane are in private. Routing protocol modules from different suppliers cannot interchange. With the rapid emergences of many kinds of new applications, new protocols or services cannot be updated into control plane in time. Network administrator has to wait for tardy but expensive supplier's updates and cannot plan, deploy, configure and administrate new services as his/her wish.

In this paper, we propose and implement a new control plane and forwarding plane architecture called the Control Extensible Router (CER). It has a few advantages over traditional router architecture and other similar research work: (a) CER separates the implementation of control plane functions from packet forwarding functions. In this architecture, all control plane functions are implemented on general-purpose servers called the control elements (CEs) that could be multiple hops away from the line cards of forwarding elements (FEs). (b) Forwarding elements is based on Network Processor (NP). NP has great flexibilities over ASIC implementation. New services can not only be deployed in CE but also be downloaded into FEs. There are no architectural limitations for new routing protocols, forwarding protocol and network services implementation in CER. (c) CER is open and extensible. We design a standardized interface between the CEs and FEs similar to that being standardized in the IETF ForCES[1] working group. CER's control plane software is implemented on the basis of Modular Object (MO). A MO can be a kind of routing protocol, forwarding protocol or network service, such as QOS. The coupling among MOs is also in standard interfaces and new MO, which supports these interfaces, can be deployed easily. (d) We have verified the feasibility of CER by implementing CER router system.

The rest of this paper is organized as follows. In the next section, we summarize related work. Section 3 describes the architecture overview of CER. In Section 4, we give the detail descriptions of CER's implementation. To demo the powerful extensibility of CER software, an IPv6 implementation case is also discussed in Section 4. We present performance results of CER router in Section 5, and conclude in Section 6.

2 Related Work

TV Lakshman presents the SoftRouter architecture[2] that separates the implementation of control plane functions from packet forwarding functions and permits dynamic association between control and forwarding elements. He discusses the challenges of SoftRouter architecture in general and demonstrates the benefits of open controlling in reliability, scalability, security, ease of adding new functionality and lower costs. The SoftRouter is very similar to our work in router architecture. But CER has more extensibility and verified by a practical system. It shows a few drawbacks which are against the proposed benefits of the SoftRouter.

The Open Signaling approach[3] advocates the separation of control plane from the forwarding plane in ATM networks for increased network programmability. This separation enables the Tempest framework[4] to create architectures where multiple control planes could simultaneously control a single ATM switches network.

The Internet Engineering Task Force (IETF) is working on standardizing a protocol between the control element and the forwarding element in the ForCES[1] working group. ForCES focuses on the protocol oriented issues of decoupling networking control and forwarding elements (FE) to provide fast, reliable, and scalability. ForCES describes the standard interfaces between CE and FEs, which are basic building blocks of network. NPForum[5] also focuses on the definition of standardized interfaces among basic building blocks. But it concentrates on lower logical function blocks level. The case for separating some of the routing protocols multiple hops away from the routers have been made by several researchers[6,7].

CER architecture makes it easier to add and deploy new protocols into existed router. Researchers have proposed other techniques such as Active Routers[8,9] or open programmable routers[10] to increase flexibility in deploying new protocols in the Internet. CER hosts control plane on general purpose servers, which has a lot more computation and storage resources, and makes control plane more scalable and reliable.

3 Architecture Overview

3.1 CER Architecture

A typical Control Extensible Router (CER) consists of a powerful CE and many FEs. The primary function of a FE is to switch packets among its physical interfaces. The switching functions are driven by a simple local forwarding table which is computed and downloaded by a remote CE. CE is a general purpose server with powerful computation and memory resources. The biggest difference between CER and traditional router is that the sophisticate control functions, such as routing protocols, administration module and forwarding table computation, are separated from traditional router and hosted at CE. CE executes control logic on the behavior of its FEs, responds to the events from FE and configures FE forwarding logic. So there must be a binding between CE and its FEs. Any combination between CEs and FEs is valid in theory. But it can sophisticate the synchronization logic among CEs when one FE is bound to multi CEs. A FE can only be bound to one CE in CER router but a powerful CE can have multi FEs. Sometimes there is a backup CE in order to achieve high availability. It can take over those FEs only when original CE is crashed.

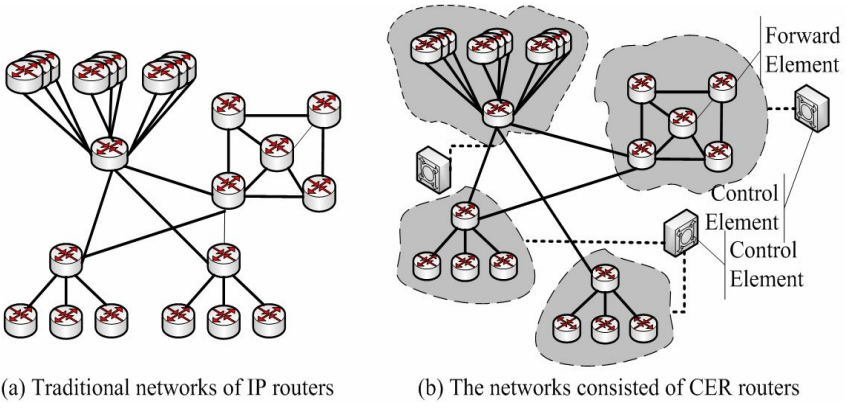


Fig. 1. An example network of CER routers

Fig. 1 shows the comparison between traditional IP network and CER network. In CER network, there are two different views: device view and routing view. The device view of CER network is very similar to traditional IP network except that a few CE servers are interconnected with FE devices by physical links. But there is great difference in routing view. The FEs in dark area are bound to a CE and they consist a logical IP router in Fig. 1 (b). CER can simplify the complex topology of Fig. 1 (a) into three interconnected logical routers. With the deployment of CER in traditional IP networks, it can cluster neighboring FEs into a single router and provide significant simplification. The reduction of network devices can simplify the complexities of routing protocols, improve protocol convergence time and ease network administration.

3.2 Modular Object

Modular Object (MO) is basic building block of CER software architecture. A MO fulfills an independent control function of control plane, such as IP forwarding, QoS or routing protocol. Typical MO is consisted of control object and forward object. Control object is a software module or process hosted at CE. Forward object is a downloadable picocode module of network processor, which concentrates on packet forwarding or scheduling. CER provides standard communication interface between control object and forwarding object. Each MO registers itself into CER and sends/receives private messages between its control object and forward object via standard interface directly.

CER has two types of MO: forwarding MO and routing MO. Forwarding MO deals with packet forwarding and forwarding table management functions. It is a typical MO and must include control object and forward object. However routing MO is not a typical one, has only control object. Routing MO is high layer routing protocol and does not operate forwarding table directly. It exchanges routing information with other CEs or traditional routers, determines best routes and installs them into forwarding MO.

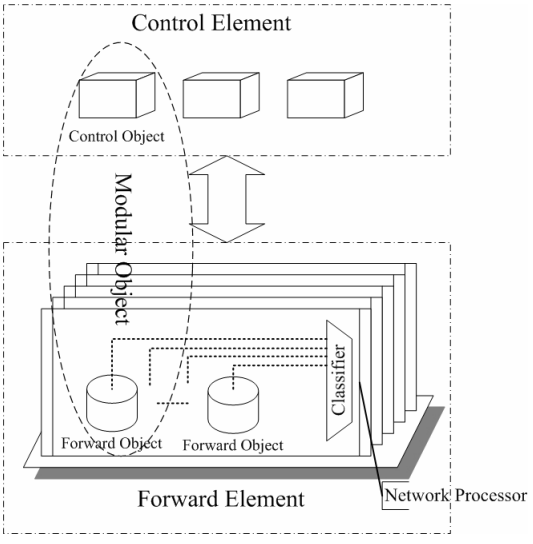


Fig. 2. The Modular Objects (MO) of CER

Fig. 2 shows the MO view of CER software architecture. CE is much as a container of MO's control objects. It provides not only basic infrastructure to create, deploy and download MO, but also standard communication interfaces between MO's control object and forward object, routing MO and forwarding MO and among routing MOs. Forward object is hosted at all NPs of FEs. It registers a classifying policy into packet classifier and processes booked packets. For the limitation of NP architecture, forward objects are configured and packed as a NP image during CER boots up and downloaded when a FE is bound to CE now. It does not influence the extensibility of CER and we believe that MO can work in real-time mode with the rapid development of NP technology in the future.

3.3 Standard Protocols and Interfaces

In order to make CER open and extensible, we have proposed a number of standard protocols and interfaces. We describe the four most important ones here.

Discovery Protocol: In order for a FE to establish a binding with a CE, it must first know about the existence of the CE and be able to reach it using some route. We design a discovery protocol finds out what CEs are available and lists available paths to them for the FEs.

FE/CE Control Protocol: Once a FE is bound to a CE, the FE and CE communicate using a standard protocol. FEs register their physical descriptions into CE and report link status information in a real-time mode. CE configures the status of FEs and downloads all kinds of control information by control protocol. Our control protocol is based on ForCES protocol. We extend it to support MO deployment and MO inner communication.

MO Interface: During the boot of CE, MO registers itself into CER by MO interface. MO interface assign a unique identifier for each MO, initializes control objects and packs forward objects into a downloadable NP image. The communication between control object and forward object is also provided by MO interface. On the downlink direction, control objects send messages to update data structures of forward object, such as forwarding table, filter lists and QoS configuration. Forward objects redirect abnormal or local packets on uplink direction.

Forwarding Operation Interface: Forwarding operation interface provides the routes operate functions on forwarding MOs for upper layer routing MOs. It is very similar to the interfaces between routing protocols and IP protocol. But forwarding operation interface is more flexible and supports different forwarding protocols. Routing MOs bind itself on a forwarding MO when they are registered into forwarding operation interface. They export exchanged routing information into bound forwarding MO and influence the forwarding behaviors of all FEs.

4 Implementation Case for Prototype of an IPv6 Router

We have implemented a prototype CER for IPv6 case based on a traditional IPv4 router platform, which is called OpenRouter6. The software view of OpenRouter6 is showed in Fig. 3.The system consists four parts: Routing Services and Stack (Control Service), Control Master (CER Master), Control Slave (CER Slave), and IPv6 Forwarding Picocode (Forward Engine).

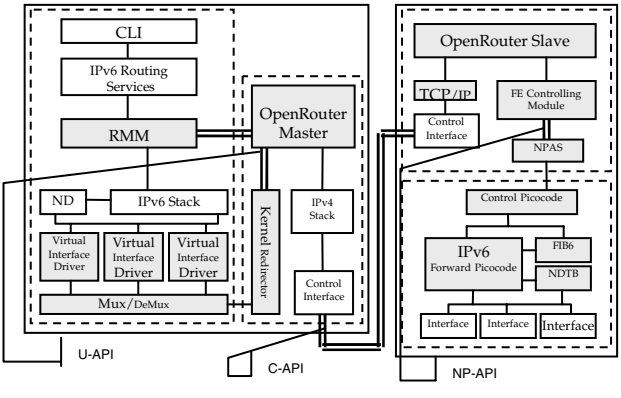


Fig. 3. Function view of CER software

The RMM(Route Management Module) daemon is a layer of abstraction between each protocol and the operating system routing table. It maintains a table of the routes and the protocol that sent the route to it. The RMM daemon will then update the operating system routing table with the new route.

We use the Linux kernel network IPv6 stack (include TCP6, ND, ICMP6, ET.) with some virtual NIC drivers and IPV6 Forwarding picocode as a forwarding MO, to which the RMM run to control as the controlled object.

The Control Master (C-master) is implemented with a user space process. It uses two TCP/IP connections, control-connection and data-connection, to interactive with Control Slave (in FE router). C-master completes the master fuction of FE/CE control protocol, provider the U-API to stack and routing service. The C-master interacts with routing service for control message via a IPC method, interact with stack for redirect packet via a kernel communicate module.

The Control Slave (C-slave) is implemented with a task of VxWork. It completes the slave function of FE/CE control protocol, provider the C-API (on-wire) to master. It also completes the FE-Specified control behavior with FE Controlling Module. NPAS module does the map between the FE-Specified control behavior and NP-based FE model. NPAS implements the NP-API similar with NP Forum CPIX.

IPv6 Forwarding object runs on NPs and completes the normal IPv6 packet forwarding based on forward table (established by routing service). Packet and control flows work as follows:

Data Flow: In the receiving case, a packet will enter the system via one of ports serviced by the NP-platform. The IPv6 forward object will process to make a decision how to dispose the packet. It may choose to forward it via another port of the system, in which case, the network stack will not be exposed to the packet. The classifier may also discover that the packet is a routing update, or control packet destined for CE. In this case, the packet is forwarded to the local control point (Slave). CER Slave converts the packet into a redirection data message of control protocol, and sends to separated control server (Master) through the CER's data-connection.

The redirection data message arrives to CER Master of separated control server, CER Master discover the redirection data message and construct the metadata to send to kernel redirector. The metadata include the information about receiving port, kernel redirector construct the 2-layer-frame to specified virtual interface for injection into the network stack. From there, CE protocol stack will process the packet as usual.

If packet is a routing protocol packet. It will be handed to RMM, which will process it and make decisions about routing policy.

In the transmission cases, Routing service should send routing protocol packet to RMM, and RMM then call OS system call to send the packet to network stack and the stack could generate control packets too.

The stack will select one of the virtual interfaces presented by the stack adapter as the destination interface. The packet will be encapsulated into a layer 2 frame passed across virtual interface, which will send the frame to the kernel redirector. The kernel redirector will construct the metadata with the frame, and send metadata to CER master.

CER master receives the metadata, and converts the metadata into a redirection data message. With the FE/CE control protocol, master sends the message to slave via data-connection.

Control Flow: There are 2 directions of control messages: ahead (CE to FE) and reverse (FE to CE). These messages include IPv6 forward table add/modify/delete, IPv6 neighbor table add/modify/delete, port status query, forwarding logic configure, exception notify, etc. In the ahead direction, the process is synchronous, CE send a control message to FE, and wait for a VALUE or ACK from FE to indicated the results. In the reverse direction, the process is asynchronous, FE send a notification to CE, CE deal with the message to apply FE. Now we use a example to explain the control flow. One is the IPv6 forwarding table update message. The routing protocol calculate a routing entry for a destination, then put the routing entry in the kernel stack, at the same time, call CER master.

5 Experience and Practice (Evaluation)

To evaluate CER architecture, we add a control card into FE and make it traditional IP router. Control card executes tightly coupled router software and communicate with line-cards by internal crossbar. We call it FE router. CE uses 2GHz Pentium CPU with 512MB DRAM. The control card uses 600MHz PowerPC750 CPU with 128MB DRAM. The distinction of two controlling platform is obvious. Extensibility is the one of the key benefits of CER architecture.

We use Spirent AX4000 to test the BGP and OSPF performance of CER system in a one-hop separated environment. The result is compared with traditional IP router and listed in following:

Table 1. BGP performance comparison

	CER	FE router
Route Learning Time for 100K routes (second)	117	673
Maximum Number of Routing Table Entries	512K	133K
pass-through delays (ms)	350	330

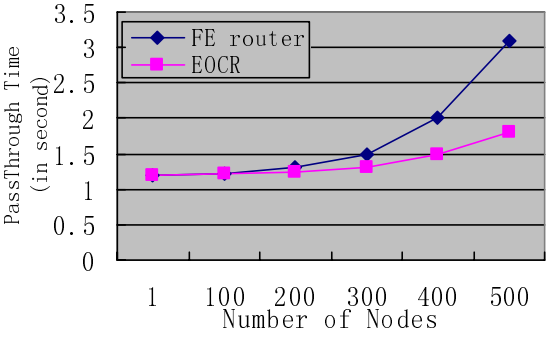


Fig. 4. OSPF performance comparison

The comparison of BGP basic capabilities is shown in Table 1. CER can achieve high protocol processing ability and huge BGP table capacity with a scalable and powerful controller. Pass-through delay of BGP is not improved but gotten worse for dominated network transmission delay and packet redirection delay. Sometimes long delay of route update may be helpful. It can combine multi BGP update packets into one and reduce the influence of dramatic route flapping.

Fig. 4 shows the pass-through time comparison of OSPF protocol. CER is very near to FE router with small node number for that SPF computation delay is lower than all kinds of OSPF timer and task schedule delay. When node number is increased over 300, SPF computation burdens FE router controller greatly and pass-through time is in exponential growth trend. CER is much better than FE router.

We have also tested the performance impact of CER in a long hops environment, the control server (CE) is located at Changesha in china, and one FE is located at Beijing and bound to the remote CE. The results are not perfect. The efficiency of control is low and CE endures long transmission delay. It suggests that the creation of a separate controlling network connecting CEs and FEs can make the control plane much more resilient and flexible.

6 Conclusion

In this paper we explained that current marked trends push for both flexible and high performance routers. Current router options are either high performance (commercial routers) or flexible (open source-based PC routers).

As a solution to this problem, we propose a router architecture that consists of the combination of NP-based forward engine, and open Linux control server. The combination of these components provides:

- A performance level that can easily be compared to the switching performances of commercial routers;
- The scalability through the use of multi-FE support in Open Router Protocol.
- The flexibility at the control path equal to that of an open source PC router;
- The extensive developer support that have been engaged on Linux-based routers;
- A clear separation between forwarding and control planes.

With CER architecture, we could rapidly employ the new control protocol than monolithic router software suit, and we could enhance the control or forward performance independently, especially for control performance with using common high computer server.

In a far future technology view, the network could separate into two physical layer: Forwarding layer and Controlling layer. The concept of router could be changed a software system as describes in [2]. Following that, the services of network could be deployed from the software center.

References

1. L. Yang et al. Forwarding and Control Element Separation (ForCES) Framework. RFC 3746, 2004.
2. T. V. Lakshman, T. Nandagopal et al. The SoftRouter Architecture. Hotnets III: Networking Seminar. UC Berkely. Novemer, 2004.
3. A. Lazar. Programming Telecommunication Networks. IEEE Netowrks, vol 11(8-18). 1997.
4. Sean Rooney, Jacobus E. van der Merwe, Simon Crosby and Ian Leslie. The Tempest, a Framework for Safe, Resource Assured, Programmable Networks. IEEE Communications, Vol 36, Num 10 pp.42-53. 1998.
5. NPFroum
6. N. Feamster, H. Balakrishnan et al. The Case for Separating Routing from Routers. In Proc. Of FDNA workshop. 2004.
7. R. Govindan, C. Alaettinoglu, K. Varadhan and D. Estrin. Route server for inter-domain routing. Computer Networks and ISDN systems. Vol 30 pp. 1157-1174. 1998.
8. DARPA Active Network Program. http://www.darpa.mil/ato/programs/activenetworks. 1996
9. D. Wetherall, U. Legedza and J. Guttag. Introducing New Internet Servies: Why and How. IEEE Network Magazine. 1998.
10. M. Handley, O. Hudson, and E. Kohler. XORP: An open platform for network research. In HotNets. 2002.

Dependable Propagating Routing Information in MANET*

Zhitang Li, Wei Guo, and Fuquan Xu

School of Computer Science & Technology, Huazhong University
of Science Technology, Wuhan, 430074, Hubei, P.R.China
{leeying, fqx}@mail.hust.edu.cn
guo_w@jhun.edu.cn

Abstract. The dynamic topologies and membership in MANET make it difficult to use conventional routing protocol for propagating trusted routing message between individual nodes. However, it is the most important thing to implement the secure communication in MANET that how to propagate routing message dependably to discover the secure shortest paths. This paper defines a trusted routing discovery model, *MARD*, which proposed to authenticate the protocol participants in order to prevent the malicious node from providing secure shortest paths, negotiate one times session keys to encrypt the communication between the source and destination nodes by using the routing discovery mechanism at the same time. In the end, the analysis shows that secure goals have been achieved.

Keywords: Secure Routing Protocol, Mobile Ad Hoc Networks, Routing Discovery, Passive Attack, Active Attack.

1 Introduction

Mobile Ad hoc Networks (MANET) is a collection of mobile nodes, which has no fixed available infrastructure and has no pre-determined organization of available wireless links. Due to the limit of wireless transmission range, nodes of MANET are required to relay packets for other nodes in order to deliver data across the network. It means that individual nodes need using routing protocol to be responsible for dynamically discovering which is the best effective route and deliver the packets to the next node. Without accurate routing information, the network links may become more congested, and the overall load on the network may increase, and it even may cause the entire network to paralyze.

However, the open and distributed wireless communication environment where mutually distrusting mobile nodes leave or enter MANET at any time is a major challenge to propagate routing message dependably between nodes that need to trust and cooperate mutually in the process of route discovery. Unfortunately, mobile nodes are susceptible to be compromised and the adversary may use the malicious nodes to inject false routing information into the network. Even though the malicious nodes have not interrupted the operation by fabricating, modifying or disrupting the

* **Foundation item:** Supported by the National network & information secure guarantee sustainable development plan (2004 Research 1-917-C-021).

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 772 – 780, 2005.

routing information, which also may eavesdrop the application data deliver subsequently. Obviously, secure operation of routing protocols is based on how to establish the trust and protected creditability of message between nodes is the most important thing for the precise routes in MANET.

This paper presents a model of mutual authentication routing discovery (*MARD*), which not only propagate dependably routing message to build the trust relationship associated with MANET nodes chain that would be used for routing message from source to destination, but also negotiates one times session keys to establish a private communication channel between source and destination nodes that will use to route data securely. In Section 2, this paper discusses related work on the subject, and in section 3, secure mutual authentication routing discovery model has been described. The section 4 analyzes the security of the routing discovery model, and finally Section 5 draws the conclusions.

2 Security Threats in Routing Discovery of MANET

Most existing MANET routing protocols, such as *DSDV*, *WRP*, *FSR*, *AODV*, *TORA*, *ABR*, *SSR*[1], trust implicitly all participator, and focus on how to depend on the neighboring nodes cooperative to spread routing information quickly once the network changes. This naive trust model allows erroneous routing information easily inserting by malicious attacker [2], [3].

2.1 The Particularly Vulnerability of Routing Information Exchange

In MANET, routing protocols distribute reachable information to various destinations and dynamically adjust the paths based on topology and other kinds of changes. However, the dynamic topology and cannot predict apriority, which leads the membership of routing to protean and insecure nature. The adversary have many chances to insert malicious node in the dynamic and uncertainty route and difficult to detect. As nodes will route the message for other nodes until the final destination is reached, such a malicious node can subvert the routing operation throughout the MANET.

The wireless link is another insecure nature that relate to the routing information exchange. Because of open medium and lack of clear line of defenses, the wireless links of MANET is susceptible to link attacks. Sniffing and analyzing the routing information in traffic, attacker can discover valuable information, such as topology information.

2.2 The Threats for Routing Security

Because of the insecure natures discussed above, there are several threats to propagate routing information dependably in this environment. This paper concentrates on how to protect the routing mechanism from malicious nodes and ignore physical attacks on the transmission medium. The protocol has to face various attacks with different intensity and intentions that can be categorized into two kinds: Passive and Active [4]. The active method, which adversary use malicious nodes to inject actively arbitrary

routing information into propagating routing message of MANET routing protocols, can be classified into modification, impersonation, and fabrication [5]. The active attack enables malicious nodes to launch a variety of attacks in MANET including creation of black-holes, grey-holes, denial of service and so on.

Attacks Using Modification. Conventional routing protocols for MANET neglect that intermediate nodes alter maliciously the control fields of messages to distribute falsified values. Thus, it is easy for malicious nodes to compromise the integrity of routing message and modify routing information, which cause network traffic to be dropped, redirected to a different destination, or take a longer route to the destination increasing communication delays. The typical modification attack includes: modified route sequence numbers or hop counts in *AODV* to implement redirection attack, such as black-hole, routing loops or increasing in route length; modifying source routes in *DSR* to implement Denial-of-Service attack.

Attacks Using Fabrication. Conventional routing protocols are difficult to identify whether routing messages they received are legitimate, so the messages fabricated by another node cannot be detected. The typical fabrication attack includes: fabricating routing error messages in both *AODV* and *DSR* assert that a neighbor can no longer be contacted and broadcast spoofed packets in *DSR* to poison route caches.

Attacks Using Impersonation. Conventional routing protocols are weakness of safeguarding the identifier of the node in message. A malicious node can launch many attacks under this environment by misrepresenting its identity as another node (spoofing) to filch unauthorized resource or combined with modification attacks. For example, a spoofing attack allows the creation of loops in routing information collected by a node as the result of partitioning in the network.

Passive Attacks. In passive attack, the malicious entity does not disturb actively or modify routing information, just only attempts to discover valuable information by eavesdropping on the network traffic. Attacker analyses the routing message and reveals the relationships between nodes. For example, which nodes are trying to establish routes to another nodes or which nodes are pivotal to proper operation of the network, and then it can launch attack to collapse the entire network. Furthermore, This attack is virtually impossible to be detected in the wireless environment and hence also extremely difficult to avoid.

2.3 Previous Work

In order to propagate dependably routing information in MANET, routing protocol must ensure that no node can prevent successfully routing discovery and maintenance. Several secure routing protocols have been proposed previously [6]. The recent development includes Papadimitratos and Haas have proposed *SRP* (Secure Routing Protocol) based on *DSR*, which assumes the existence of a security association between the source and destination to validate the integrity of a discovered route [7]. Dahill have proposed the *ARAN* (Authenticated Routing for Ad hoc Networks), which uses public key cryptography instead of the shared security association used in the

SRP [8]. Yi, et al. have proposed a generalized *SAR* (Security-Aware Ad hoc Routing) protocol for discovering routes that meet a certain security criteria, which requires that all nodes meet a certain criteria to share a common secret key [9]. However, Both *SRP* and *ARAN* are focused on detection whether the protocol participants are legal when the route is discovered, and that *ARAN* use an optional second discovery stage that provides non-repudiating route discovery. Furthermore, without the privacy of the routing information, *ARAN* and other protocols are weak against the passive attacks, such as eavesdropping and spoofing, and intermediate nodes that handle control messages can easily find the identity of the communicating nodes.

3 Mutual Authentication Routing Discovery (*MARD*) Model Describe

This paper proposes a secure routing discovery mechanism. *MARD* is designed for pledging routing message propagates and preventing the malicious node from inserting into route at any hop to discovery a secure route from the source node to destination. Besides, this model can negotiate the session encryption algorithm and establish private communication channel between the source and destination nodes to transmit the application data.

3.1 Assumptions and Initialization

To adapt to the special characteristics, this model use certificate and signature to guarantee the trust of routing information propagated in MANET. Every legit node joining MANET must obtain an identity certificate from a Certificate Authority (*CA*) that can guarantee the reliability of identity. Because certificate authenticates the unique identity of node and can be trusted and verified by all nodes, bound in routing message, so can establish association with origin node. All nodes must maintain fresh certificates with the *CA*. The identity certificates have the following form:

$$Cert_A=\{\ ID_A, IP_A, ID_{CA}, K_A, Cert_V\ \}_{K_{CA}^{-1}}$$

In the certificate, ID_A is the identifier of A, IP_A is the IP address of A, ID_{CA} is the identifier of *CA* that issues the certificate, K_A is the public key of A, and the $Cert_V$ is the version number of the certificate.

Each node also has a pair of public and private keys, and other legal nodes know the public key. In order to authenticate and verify the integrity of the message, nodes sign encrypt the message by using its private key during the routing discovery process. The security of the cryptographic algorithms is irrespective, and the definition of the model and the negotiation of the session encryption algorithm are disjoint as well so that other cryptography can be used[1].

[1] Considering the expanding on unidirection link, we use mutual symmetrical key exchange here. To improve the security of mutual authentication, the other key negotiation algorithm can be replaced for the symmetrical key exchange that is being used currently, such as: Diffie-hellman, etc.

3.2 Dependable Routing Discovery Mechanism

The routing discovery mechanism of *MARD* is accomplished by broadcasting a routing discovery message from the source node that will be replied to unicast by the destination node. The flooding-based routing ensures that a packet will traverse every link and hence reach its intended destination, as long as a non-faulty path exists. Essentially, the *MARD* is an extension to *AODV* routing protocol[10]. The routing message forwarding process is achievable and very secure, as the trust is propagated from one node to another by exchanging certificates and verifying signatures, the routing messages are authenticated at each hop from source to destination, as well as on the reverse path from the destination to the source. The process can be divided into follow phases:

Route Request. The source node *S* triggers the routing discovery process by broadcasting route request message(*RREQ*) to all neighbors in order to discover the shortest paths to destination node *R*.

$$S \rightarrow broadcast: [RREQ, RD\_ID, IP_R, IP_S, \{SN\_ID, TK_S\}_{K_R}, Cert_S]_{K_S^{-1}}$$

The source node uses the public key K_R of destination receiver that may learn from certificates to encrypt the session identifiers (SN_ID) and the symmetric key(TK_S). With the packet type identifiers (*RREQ*), routing discovery process number (RD_ID), the IP address of the destination (IP_R), and the certificate of the source($Cert_S$), all the parameters of *RREQ* are signed to encrypt by the private key of *S*. Essentially, the SN_ID is a random number, which is large enough to be recycled within a special period, and is mapped to the symmetric key (TK_S) that will encrypt the data to send from *S* after route has been discovered.

Route Request Forward. When the neighbor node *B* receives the *RREQ* from *S*, it tries to decrypt the *RREQ* with the public key of the source node and verifies whether the message has been modified. *B* is also checked SN_ID subsequently to verify whether the message has received previously. If *B* is not the expectant receiver, it sets up the reverse path back to the source by recording the neighbor from where it receives the *RREQ*. After appending its own certificate, *B* signs encrypt the contents of the message, and rebroadcast it to its other neighbors**:**

$$B \rightarrow broadcast: [[RREQ, RD\_ID, IP_R, IP_S, \{SN\_ID, TK_S\}_{K_R}, Cert_S]_{K_S^{-1}}, Cert_B]_{K_B^{-1}}$$

Let *C* a neighbor of *B* and that has received the *RREQ* from *B*, which validating the *B*'s signature, and repeats those step to verify the *RREQ*. If *C* is not the expectant receiver either, it sets up the reverse path back to the source. After removing *B*'s certificate and signature, *C* appends its own certificate, signs to encrypt the message, and forward broadcasting the message to its other neighbors as the predecessor:

$$B \rightarrow broadcast: [[RREQ, RD\_ID, IP_R, IP_S, \{SN\_ID, TK_S\}_{K_R}, Cert_S]_{K_S^{-1}}, Cert_C]_{K_C^{-1}}$$

Each intermediate node that receives the route request message (*RREQ*) repeats these steps like node *C*, until the *RREQ* reach the expectant destination.

Route Reply and Forward. When the expectant receiver *R* receives the *RREQ*, it also tries to decrypt the previous node's signature and ensures that the message has not been modified. If it receives this *RREQ* firstly, it sets up the reverse path back to

the source. Then the destination node R unicasts a route reply message($RREP$) back along the reverse path to the source, which triggers the route reply:

$$R \rightarrow C: [RREP, RD_{\_ID}, IP_S, IP_R, \{SN_{\_ID}, TK_R\}_{K_S}, Cert_R]_{K_R^{-1}}$$

The destination node uses the public key K_S of source node that may learn from certificates to encrypt the session identifiers ($SN_{\_ID}$) and the symmetric key (TK_R). With the packet type identifiers ($RREP$), routing discovery process number ($RD_{\_ID}$), the IP address of the source (IP_S), and the certificate of the destination ($Cert_R$), all the parameters of $RREP$ are signed to encrypt by the private key of R. The TK_R is the symmetric key, which will be mapped by $SN_{\_ID}$ and encrypt the data that send from S after route has been discovered.

Let C the first node on the reverse path that received the $RREP$ from the destination, it tries to decrypt the $RREP$ with the public key of the destination and verifies whether the message has been modified. Being not the expectant receiver, C sets up the path to the destination by recording the neighbor from where it receives the $RREP$. After appending its own certificate, C signs to encrypt the contents of the message, and unicasts the message to the next node on the reverse path back to the source:

$$C \rightarrow B: [[RREP, RD_{\_ID}, IP_S, IP_R, \{SN_{\_ID}, TK_R\}_{K_S}, Cert_S]_{K_R^{-1}}, Cert_C]_{K_C^{-1}}$$

When the next node B receives the message, it validates the previous node's signature, and repeats those steps to verify the $RREP$. If B is not the expectant receiver either, it sets up the path to the destination. After removes the signature and certificate of previous node, B appends its own certificate and signs to encrypt the message, unicast the message to next node on the reverse path as the predecessor:

$$B \rightarrow S: [[RREP, RD_{\_ID}, IP_S, IP_R, \{SN_{\_ID}, TK_R\}_{K_S}, Cert_S]_{K_R^{-1}}, Cert_B]_{K_B^{-1}}$$

Each intermediate node that receives the route reply message ($RREP$) repeats these steps as node B, until the $RREP$ reach the origin node.

3.3 Establish Secure Communications Channel and Transfer Data

When the source node gets the $RREP$, it also tries to decrypt the previous node's signature and ensure that the message is correct. After that, a concatenation of the trust nodes composes a path, which also be considered to be secure route data between source and destination node.

Once secure route has been established the data packets need to be encrypted between source and destination node. The source node uses the symmetric key(TK_R) to encrypt the data X, which will send to destination node. With the packet type identifiers ($RDAT$), the session identifiers ($SN_{\_ID}$), the IP address of the destination (IP_R), and the certificate of the source($Cert_S$), all the parameters of $RDAT$ are signed to encrypt by the private key of S. when receives the data message($RDAT$), the destination node will accord to the $SN_{\_ID}$, which is contained in the message to choose the symmetric key to decrypt the data X. Similarly, the destination node uses the symmetric key(TK_s) to encrypt the reply data Y, and construct the route data message($RDAT$) as the source node.

4 Security Analysis

In *MARD*, the routing information has been protected in the whole routing discovery process. To guarantee the dependability of routing information, the outgoing routing messages in which the certificate is encapsulated have been signed and encrypted before being sent to the network. When receive the routing message, nodes need to decrypt and decapsulate it to verify the dependability of the message. A number of cited frequently attacks against MANET routing protocol could be prevented by *MARD*, which has these characteristics.

The first, adversary must overcome the authentication. As the messages are signed by all hop-by-hop, each node must encapsulate the identity certificate attested by a trusted third party to claim its authenticity, which enables the receiver can verify who propagate the routing message. An authorized node just only creates the message. Malicious nodes cannot generate and inject false routing messages for other nodes, or masquerade as another node and gain unauthorized access to resources and sensitive information. Mutual authentication is the basis of a trust relationship between nodes. The other security characteristic in *MARD*, such as confidentiality, integrity and non-repudiation all rely on the accurateness of this authentication.

The second is the integrity of routing information. The integrity is the property that ensures the message has not been altered in an unauthorized manner while being transferred. In *MARD*, every node uses digital signatures to guarantee the integrity of the content, and any unauthorized manipulation of information should be detected [11]. As the contents in every transmitted message are replicated regularly and forward, the contents of these messages should be guaranteed. The routing information carried in the routing message could not be corrupted, and the path spreads over the secure nodes so as to all function of routing discovery will prevent from malicious node.

The third is the non-repudiation of routing information. The non-repudiation ensures that the origin of a message cannot deny having sent the message. In *MARD*, even though false routing information has been injected into network, nodes cannot repudiate the malicious behavior with non-repudiation. In active attack, adversary needs to manipulate the routing message abnormally to carry out malicious activity later. It means that, once it starts the malicious activity, the secure routing discovery algorithm can use intrusion detection systems to detect which nodes attempt to compromise network by propagating fake routing information and then isolate malicious nodes [12].

The fourth is the confidentiality of routing information. The confidentiality assures that only the authorized nodes are able to obtain protected information from message. In *MARD*, all the routing message of routing discovery process are encrypted before transmission, and only legit nodes that have the decryption key can decrypt the messages and participate in the routing to protect the messages that are exchanged between nodes. It means that even though adversary can sniff data transmitted on the wireless link, sensitive information also cannot be leaked. After establishing secure route, all application data exchange between source and destination nodes occur over the private channels equipped with cryptographic protections, and that the cryptographic techniques are sufficient in this context to protect the confidentiality of the communication.

5 Conclusion and Future Work

This paper studied the security issues of routing protocol in MANET and has pointed out that there are several threats to the trust of routing information that have been propagating in this environment. To fit to the special characteristics of protecting the message, a secure mutual authentication routing discovery model (*MARD*) for MANET is proposed, which can guarantee the authentication, integrity, non-repudiation and confidentiality of the routing information that propagate in whole routing discovery process along with confidentiality. Depending on validating the signature and the identity certificates of protocol participants, routing discovery in *MARD* coulḍprovide both end-to-end and hop-to-hop authentication of routing query and reply messages to ensure that no malicious node can prevent successfully routing discovery and maintenance. As privacy is a key issue in wireless ad hoc networks, the source and destination nodes of *MARD* use the session key negotiated in the routing discovery phase to encrypt the data in order to establish private communication.

As to the future work, we will implement our model using simulation tools such as NS-2 to verify the dependability in various attacks and compare the performance with other typical secure routing protocols in MANET, such as *SAODV*, *ARAN*.

References

1. Royer, E. M., Toh, C. K.: A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks. *IEEE Personal Communications Magazine*, vol.6, no.2, (1999) 46-55.
2. Zhou, L., Haas, Z. J.: Securing Ad Hoc Networks. *IEEE Network Magazine*, vol. 13, no.6, (1999).
3. Bayya, A. K., Gupte, S., Shukla, Y.K., Garikapati, A.: Security in ad hoc network. University of Kentucky. http://cs.engr.uky.edu/~singhal/term-papers/Fourth-paper.doc
4. Hu, Y. C., Perrig, A., Johnson, D. B.: Ariadne: A secure on-demand routing protocol for ad hoc networks. *Proceedings of the eighth Annual International Conference on Mobile Computing and Networking*. (2002) 12–23.
5. Pirzada, A. A., McDonald, C.: Establishing Trust In Pure Ad-hoc Networks. *Proc. of 27th Australasian Computer Science Conference (ACSC'04)*, vol. 26, no. 1, (2004) 47-54.
6. Pirzada, A. A., McDonald, C.: A Review of Secure Routing Protocols for Ad hoc Mobile Wireless Networks. *Proc. of 2nd Workshop on the Internet, Telecommunications and Signal Processing (DSPCS'03 & WITSP'03)*, (2003)118-123.
7. Papadimitratos, P., Haas, Z. J.: Secure Routing for Mobile Ad hoc Networks. SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002), San Antonio, TX, January (2002)27-31.
8. Dahill, B., Levine, B. N., Royer, E., Shields, C.: A Secure Routing Protocol for Ad Hoc Networks. *Proceedings of 2002 IEEE International Conference on Network Protocols (ICNP)*. November (2002).
9. Yi, S., Naldurg, P., Kravets, R.: Security-Aware Ad Hoc Routing Protocol for Wireless Networks. The 6th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2002), 2002.
10. Perkins, C., Belding-Royer, E., Das, S.: Ad hoc On-Demand Distance Vector (AODV) Routing. IETF RFC 3561, (2003).

11. Murphy, S., Badger, M., Wellington, B.:OSPF with Digital Signatures. RFC 2154. June. (1997)
12. Y. Zhang, W. Lee. Intrusion Detection in Wireless Ad hoc Networks. in *Proc. of the 6th International Conference on Mobile Computing and Networking*, (2000).

Data Structure Optimization of AS_PATH in BGP

Weirong Jiang

Research Institute of Information Technology, Tsinghua University,
Beijing 100084, P.R. China
jwr2000@mails.tsinghua.edu.cn

Abstract. With the fast growing size and complexity of core network, the hash based data structure of current AS_PATH implementation in BGP is facing challenges in performance, mainly caused by the static attribute of the simple hash. This paper proposed a splay tree based data structure and an optimal index generation algorithm specifically designed for AS_PATH. Exploiting the innate characteristics of AS_PATH, the proposed algorithm shows superior performance.

1 Introduction

The Border Gateway Protocol (BGP) is an inter-Autonomous System (AS) routing protocol. One of the most important attributes in BGP is AS_PATH [1]. AS_PATH serves as a powerful and versatile mechanism for policy-based routing [2]. With the rapid development of Internet and wide deployment of BGP [10], storage and comparison of AS_PATH entries become a potential performance issue to be addressed.

This paper is a forward-looking exploration on optimizing the data structure of AS_PATH. The rest of the paper is organized as follows: In Section 2, we will present the inherent problems of hash data structure of AS_PATH and propose the possible solutions briefly. In Section 3, we discuss the optimization of the AS_PATH data structure by comparative study. In Section 4, we provide results of our simulation experiments. In Section 5, we put forward our conclusion and our expectations for future work.

2 Background and Challenges

The global Internet has experienced tremendous growth over the last decade. Figure 1 shows the BGP statistics [5] from Route-Views Data with trend curves added. As shown in Figure 1 (c), the number of unique AS_PATHs is growing at nearly an exponential speed, which motivates research in optimized algorithms to provide higher performance.

In typical BGP implementations [8, 9], hash table is preferred since in early days, when number of AS_PATH is small, it is the most simple and efficient way. To deal with the collision, different AS_PATH entries with same hash value will be stored in a linked list and be distinguished through linear search by comparing the whole AS_PATH. In theory [6, 7], the time complexity to insert, lookup or delete an entry in hash table is O(1), which obviously is perfect in AS_PATH attribute update and retrieval. To reach high efficiency, nearly half the hash table should be empty, and ac-

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 781 – 788, 2005.

cordingly the hash table size should double the size of the existing unique AS_PATH entries, and thus the space complexity is O(2*n*) where *n* is the number of AS_PATH entries. For instance, in [8], the table size is 32,767, almost twice as the number of unique AS_PATH entries in the global routing table.

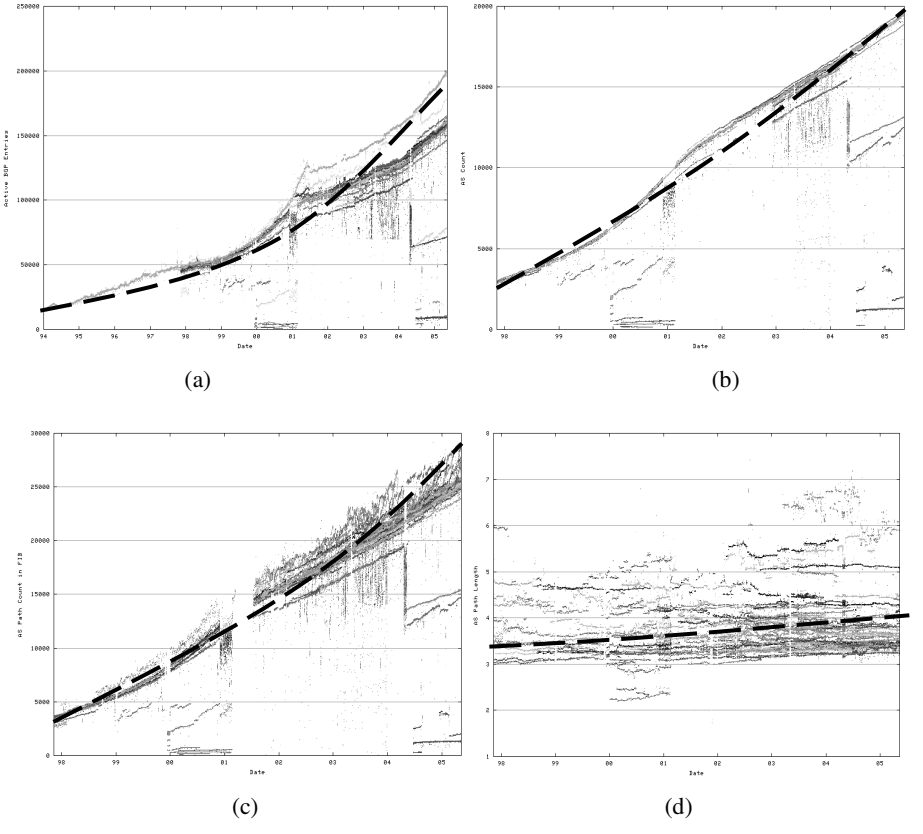


Fig. 1. BGP Statistics: (a) Active BGP entries (FIB); (b) Unique ASes; (c) Unique AS Paths used in FIB; (d) Average AS Path Length

These hash based implementations perform well nowadays in most of cases, but are expected to face severe challenges as follows.

Hash is a static data structure and the main disadvantage is the constant hash table size. The efficiency of hash will decline quickly since the hash table size will not catch up with the increasing number of AS_PATH entries. In addition, it is difficult to get it right for all situations. For example, the software including BGP routing needs to work on both a low end router with small memory and small number of routes and a high end router with large amount of memory and large number of routes. But there is no way to set a universal value for hash table size for both high and low ends. Obviously, to resolve this challenge, dynamic data structures such as binary trees could be good substitutes for hash.

The AS_PATH attribute is of various lengths and hardly can be used directly as an index. An index of constant length for each AS_PATH entry can be generated by encoding the AS_PATH attribute. Nevertheless, possible collision needs to be taken into consideration when two different AS_PATH entries are encoded into the same index. To reduce the probability of collision, folding is an easy and popular method to generate index. That is, split an AS_PATH into several equally sized sections and add all sections together. However, both splitting and adding up consume time. Since the AS_PATH is getting longer due to the rapid growth of AS number, the cost of folding is getting much more expensive. Thus there is need to find an algorithm more efficient to generate indexes.

Linking different entries with identical index is a simple solution for collision. However, increasing entries incline to cause more collisions and longer links. Then the efficiency of linked list operations (i.e. insert, lookup and delete) will also decline since entry comparison is usually expensive. One way to relieve this challenge is to construct different entries with the same index to be a secondary tree, rather than a linked list.

3 Optimizations by Exploiting the Characteristics of AS_PATH

3.1 Characteristic Observations

Table 1 shows a sample output of a BGP routing table from our test set which from real life includes more than 100,000 AS_PATH entries with 17,520 unique entries. Using this example, the following characteristics can be observed.

Table 1. A sample output of a BGP routing table

Network	Next Hop	Metric	LocPrf	Weight	AS_PATH
12.42.72.190/32	10.1.1.235	0	100	100	14207 3944 7777 i
12.43.128.0/20	10.1.1.235	0	100	100	14207 3944 2914 7018 16711 16711 16711 i
12.43.144.0/20	10.1.1.235	0	100	100	14207 3944 2914 7018 16711 i
12.65.240.0/20	10.1.1.235	0	100	100	14207 3944 2914 7018 17231 i
12.66.0.0/19	10.1.1.235	0	100	100	14207 3944 2914 7018 17231 i
12.66.32.0/20	10.1.1.235	0	100	100	14207 3944 2914 7018 17231 i
12.79.224.0/19	10.1.1.235	0	100	100	14207 3944 2914 7018 5074 i
13.13.0.0/17	10.1.1.235	0	100	100	14207 3944 2914 7018 22390 i
13.13.128.0/17	10.1.1.235	0	100	100	14207 3944 2914 4323 22390 i
13.16.0.0/16	10.1.1.235	0	100	100	14207 3944 2914 5511 5388 i
15.0.0.0/8	10.1.1.235	0	100	100	14207 3944 2914 209 71 i
15.130.192.0/20	10.1.1.235	0	100	100	14207 3944 2914 5400 1889 i
15.142.48.0/20	10.1.1.235	0	100	100	14207 3944 2914 3561 5551 1889 i
15.166.0.0/16	10.1.1.235	0	100	100	14207 3944 2914 209 71 i
15.195.176.0/20	10.1.1.235	0	100	100	14207 3944 2914 3561 1273 1889 i

Characteristic 1. Many routes share one AS_PATH entry but few entries share one origin AS. In Table 1, there are at most three different AS_PATH entries originating from the same origin AS 1889. In our test set, over 80% AS_PATH entries monopolize one origin AS. Hence in most of cases a path could be specified by its distal AS indicating the origin AS of the entry.

Characteristic 2. Scanning the AS_PATH field in Table 1 from left hand side, we find that, nearer to the origin AS, two AS_PATH entries are more likely to be different. On the other hand, the nearest AS numbers, which indicate the ASes closer to local AS, are mostly the same. This can be explained that local AS usually has very few neighbors and hence left parts of most AS_PATH entries are similar.

Characteristic 3. Considering the update process, the efficiency of the data structure/algorithm is very important when it faces a burst of route updating, which might happen when one AS's state alters and all the AS_PATHs originating from it have to be updated. It requires the entry operated most frequently to be visited most promptly. This characteristic is coincident with a type of dynamic binary trees: splay tree [7].

3.2 Constructing Splay Trees

Similar to the means done in hash, as for a splay tree, each AS_PATH entry is stored as a whole, with an index whose value is calculated by the functions discussed in next section. The entries with identical index value are linked off the same node. Figure 2 shows the data structure of the tree node and an example process to construct a splay tree from left side to right.

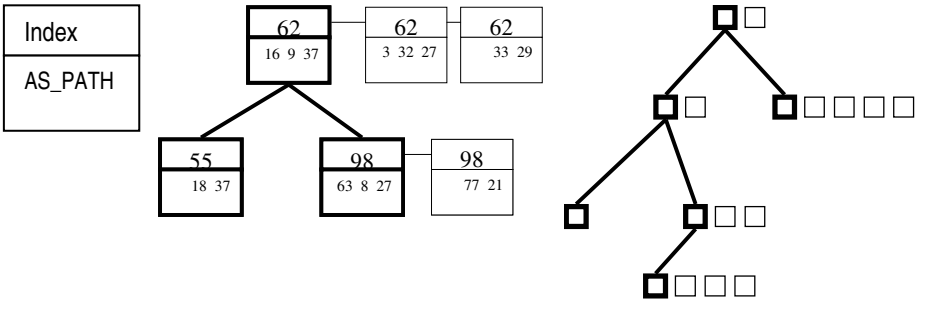


Fig. 2. Construct a Splay Tree for AS_PATH

3.3 Optimizing Index Generation

Definition of k-step Golden AS for an AS_PATH Entry. We assume the golden section ratio is $\beta \approx 0.618$ and the length of an AS_PATH entry is m. The function *Position*(AS) indicates the position of an AS in the entry and its value range is $\{1,2,\ldots, m\}$. Herein, *Position*(origin AS) = m. Then we use P_k to denote *Position*(k-step golden AS).

Definition 1. The 1-step golden AS is the one on the golden section point of the entry, that is, $P_1 = \lceil \beta m \rceil$.

Definition 2. The k-step golden AS is the one on the golden section point of the short section after last golden section, that is,

$$P_k = P_{k-1} + \lceil \beta(m - P_{k-1}) \rceil, 1 \le P_k \le m. \quad (1)$$

We impose the condition $P_k \neq P_{k-1}$, and consequently k has an upper boundary for each certain m. For our test set, $m \ge 3$, $k = 1, 2$.

Comparison on Different Index Generation Functions. As we have discussed, folding is expensive. According to characteristic 1, we employ the origin AS number as the index of an entry. Moreover, according to characteristic 2, we design other index generation functions whose time-consuming is on the same level. All the functions are presented as follows.

1. *Folding.* Split an AS_PATH entry into 16-bit sections and add all sections together to a 32-bit integer.
2. *Origin AS.* Directly get the rightmost AS number.
3. *Sum of rightmost two ASes.* Add the rightmost two AS numbers together.
4. *Sum of rightmost three ASes.* Add the rightmost three AS numbers together.
5. *Golden section.* Get the 1-step golden AS and add it to the origin AS.
6. *Golden section2.* Get the 2-step golden AS and add it to the origin AS.
7. *Golden section3.* Add the 1-step golden AS, the 2-step golden AS and the origin AS together.

We construct splay trees using our test set and regard the number of tree nodes and links, average length of all the tree nodes, average and maximum length of links and the time cost as the main judge of efficiency of index generation functions. Larger amount of tree nodes, less links, shorter length, and cheaper time cost, indicate the higher efficiency. The results are presented in Table 2.

Table 2. Efficiency of different Index Generation Functions

Index Generation	Number Of Tree Nodes	Total Average Length	Number of Links	Average Link Length	Max Link Length	Time Cost
Folding	16287	1.075705	1152	1.070313	3	O(*N*)*
Origin AS	13436	1.303960	2639	1.547556	12	O(1)
Sum 2	13492	1.298547	3133	1.285669	6	O(1)×2
Sum 3	14358	1.220226	2652	1.192308	5	O(1)×3
Golden 1	13077	1.339757	3366	1.319964	7	O(1)×2
Golden 2	13619	1.286438	3141	1.241961	6	O(1)×2
Golden 3	13921	1.258530	3014	1.194094	5	O(1)×3

* *N* indicates the number of sections after splitting.

According to the results, regardless of the time cost, folding seems most efficient, since it utilizes more information in an entry than any other function. However, the time cost of index generation influences much the efficiency of operations to insert, lookup and delete entries, especially when AS_PATH is getting longer. The other six types of index generation functions perform almost equal in efficiency. Hence using the origin AS as index is preferred for its simplicity.

3.4 Further Improvement

As we have discussed, when links get longer, the efficiency will decline badly for its linear data structure [6, 7]. This problem may come true soon owing to the astonishing increase of ASes and AS_PATH entries. If the link is replaced by a splay tree, our splay tree with links then alters to be a splay tree with a secondary tree, which might be called double-splay tree. We use the origin AS as index of the primary splay tree while we could use the 2-step golden AS or the second rightmost AS as index of secondary splay tree. Two different entries owning the same two indexes still have to be linked but the length of the link will be much shorter and hence the efficiency will be improved. Figure 3 shows an example process to construct a double-splay tree.

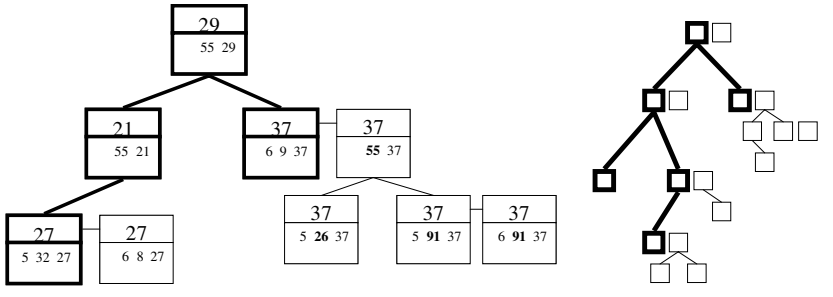


Fig. 3. Construct a Double-Splay Tree for AS_PATH

Limited by the size of test set, this improvement is not remarkable in our experiments since over 80% links are as short as just one node. We temporarily do not present the meaningless results in this paper. Nonetheless, we believe this improvement will be verified when AS_PATH entries in real life is getting much increased.

4 Simulation Experiments

4.1 Experiment Environment

For all the experiments we use a computer with a Pentium M processor running at 1.4GHz and 256 Mbytes of main memory. The host operating system is Windows XP professional with SP2. We develop and compile our program with the Microsoft Visual C++6.0 with default settings. In our program, each AS number is treated as a four-byte integer [4].

4.2 Splay Tree vs. Hash

To simulate the fact that hash is static while the number of AS_PATH entries is increasing explosively, yet limited by the condition that the number of existing AS_PATH entries is certain, we have to set the hash table size a small value (e.g. 37). We augment the size of test set from 100 to 100,000 entries, and observe the time cost to insert, lookup and delete entries. Results are shown in Figure 4(a ~ c).

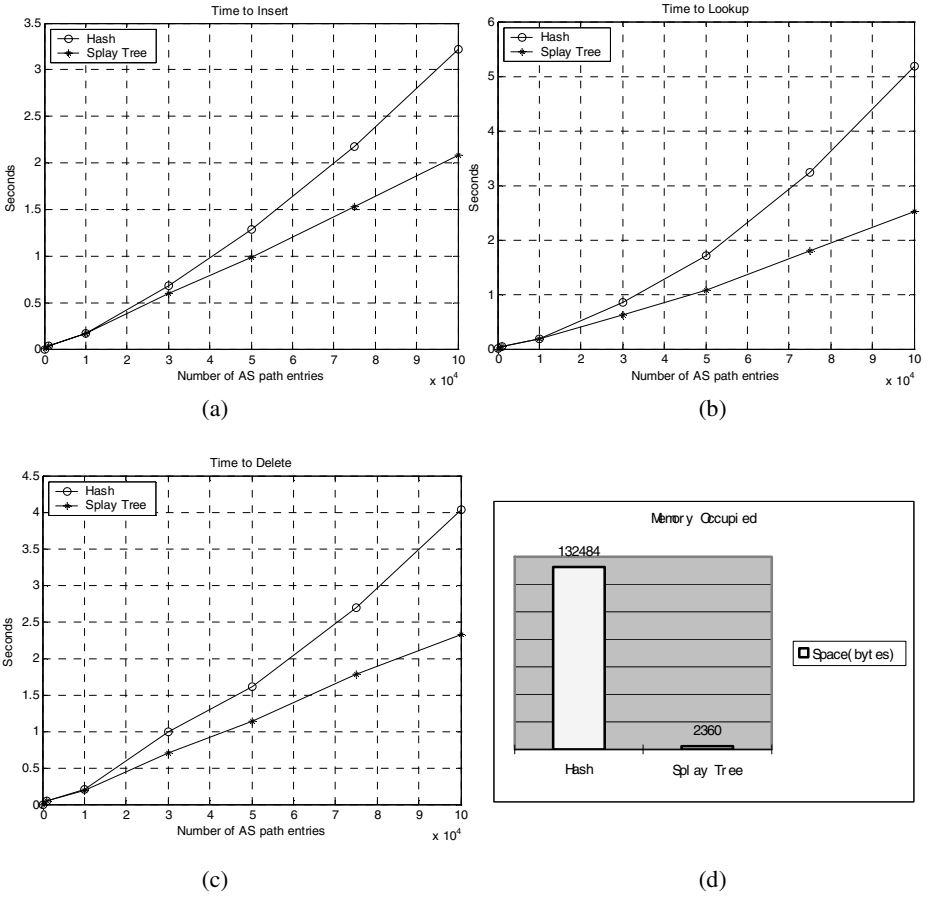


Fig. 4. Hash vs. Splay Tree using Origin AS as index

Furthermore, to verify that static hash table size is not universal for both high and low end routers, we set the hash table size an appropriate value (e.g. 32,767) and experiment with small size of entries (e.g. 1,000 route entries). Figure 4 (d) reveals the memory waste for low end routers.

These results firmly demonstrate that, hash is not suitable as the data structure of AS_PATH because of its static feature. AS_PATH should be encoded into dynamic structures such as splay trees.

5 Conclusions and Future Work

According to our above discussions and experiments, hash is no longer fit for the data structure of AS_PATH for its fatal defects under the background of the explosive development of Internet. Instead, splay trees are more suitable for their dynamic attribute. To reduce collisions, we studied several functions to generate index after exploiting inherent characteristics of AS_PATH. And we suggest using the origin AS as the index. Furthermore, a novel binary tree named double-splay tree, is proposed and waiting for future's verifications.

Based on what we have done, we try to build a test bed in future to experiment with more dynamic data structures to seek more efficient data structure for AS_PATH.

Acknowledgements

The author would like to thank Dr. Enke Chen at Cisco Networks for pointing out this research topic and providing much information and advice. He also would like to thank his supervisor Dr. Jun Li for his earnest enlightenment and comprehensive help. At last but not least, he would thank the Zebra community for the source code.

References

1. Rekhter, Y., and Li, T.: A Border Gateway Protocol 4 (BGP-4). IETF RFC 1771. (1995)
2. Traina, P.: BGP-4 Protocol Analysis. IETF RFC 1774. (1995)
3. Chen, E., and Yuan, J.: AS-wide Unique BGP Identifier for BGP-4. IETF draft-ietf-idr-bgp-identifier-04. (2004)
4. Vohra, Q., and Chen, E.: BGP support for four-octet AS number space. IETF draft-ietf-idr-as4bytes-08. (2004)
5. BGP Statistics from Route-Views Data. http://bgp.potaroo.net/rv-index.html. (2005)
6. Sahni, S.: Data structures, algorithms, and applications in C++. China Machine Press. (1999)
7. Shaffer, C.A.: Practical Introduction to Data Structure and Algorithm Analysis (C++ Edition). China Publishing House of Electronics Industry. (2002)
8. Zebra-0.94. http://www.zebra.org.
9. MRT-2.2.0. http://www.mrtd.net.
10. Meng, X., Xu, Z., Zhang, B., Huston, G., Lu, S., Zhang, L.: IPv4 Address Allocation and the BGP Routing Table Evolution. ACM SIGCOMM Computer Communications Review. 35(1): 71-80. (2005)

A Framework for Designing Adaptive AQM Schemes[1]

Wen-hua Dou, Ming Liu, He-ying Zhang, and Yan-xing Zheng

Computer College, National University of Defense Technology,
Hunan, 410073, PR China
liutomorrow@hotmail.com

Abstract. Active Queue Management(AQM) is an effective method to improve the performance of end-to-end congestion control. Several AQM schemes have been proposed to provide low delay and low loss service in best-effort networks in recent studies. This paper presents a generic framework which encompasses RED, P, PI, PIP, PD, SMVS, REM, AVQ as special cases by using Single neuron-based PID control. In addition, the framework expands the current AQM controllers by augmenting the update laws of packet-drop probability and provides an adaptive mechanism. Based on this framework and adaptive mechanism, we develop an adaptive single neuron-based PI controller. Simulation studies under a variety of network and traffic situations indicate that the proposed scheme exhibits more robust and adaptive congestion control behavior than the prior schemes.

1 Introduction

Active Queue Management (AQM) is an active research area in networking and have been recommended at intermediate nodes to improve the end-to-end congestion control and provide low delay and low loss service in best-effort networks by actively signaling congestion early [1]. Recently, several AQM schemes were proposed and analyzed using either a control theoretic model or an optimization model. Hollot et al. have used a control theoretic approach to analyze the Random Early Detection (RED) algorithm and have proposed two AQM schemes, called Proportional (P) and Proportional and Integral (PI) control[2]. The Proportional Integral and Position feedback compensation algorithm (PIP) [3] and the Proportional-Differential control algorithm (PD) [4] were developed also using the control theoretic approach. The sliding mode variable structure algorithm (SMVS) [5] is another recently proposed AQM controller which applies a sliding mode variable structure control and shares PI's goal of maintaining a stable average queue length. The TCP/AQM algorithms were interpreted as an optimization problem in [6] and the Random Exponential Marking (REM) [7] and the Adaptive Virtual Queue (AVQ) schemes were developed using this model. However, we find there are some generic implementations in their update laws of packet-drop probability.

[1] This research was supported by the National Natural Science Foundation of China Grant No.90104001 and the National Grand Fundamental Research 973 Program of China under Grant No. 2003CB314802.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 789 – 799, 2005.

In this paper, we try to present a generic framework which encompasses RED, PI, PIP, PD, SMVS, REM, AVQ as special cases by using Single neuron-based PID control. We will analyze these AQM schemes under this framework and show that these AQM schemes can be classified as Single neuron-based PID controller. The framework expands the current AQM controllers by augmenting the update laws of packet-drop probability and offers a generic mechanism which can adjust its control parameters to optimize the AQM controllers. Based on this framework, we develop an adaptive single neuron-based PI controller. ASNPI scheme adjusts its control parameters according to the changing network conditions, and thus, is more robust in maintaining system stability.

The rest of the paper is organized as follows. In section 2, we analyze the single neuron-based PI controller and summarize the generic framework. The proposed algorithms are presented and analyzed in Section 3. A comprehensive simulation study is given in Section 4, and the conclusions are drawn in Section 5.

2 Single Neuron-Based PID Control Model

2.1 The Single Neuron-Based PID Controller

As we know, neuron is a basic element of neural networks. From the structure and function points of view, an artificial neuron can be regarded as a nonlinear multi-input and multi-output processing unit, which performs a transfer function f of the following type: $y = f(\sum_{i=1}^{n} \omega_i x_i - \theta)$, where y is the output of the neuron, $x_i (i = 1, \cdots n)$ are the neuron inputs, $\omega_i (i = 1, \cdots n)$ are the neuron connecting weights, the weights are determined by a learning strategy, which is based on Delta or Hebbin learning rules. θ is the threshold of the neuron. **A** conventional continuous PID control algorithm is shown as follows: $u(k) = K_p e(k) + K_I \sum_{i=0}^{k} e(i) + K_D [e(k) - e(k-1)]$.

The incremental PID controller can be represented as:

$$u(k) = u(k-1) + K_p [e(k) - e(k-1)] + K_I e(k) + K_D [e(k) - 2e(k-1) + e(k-2)]$$

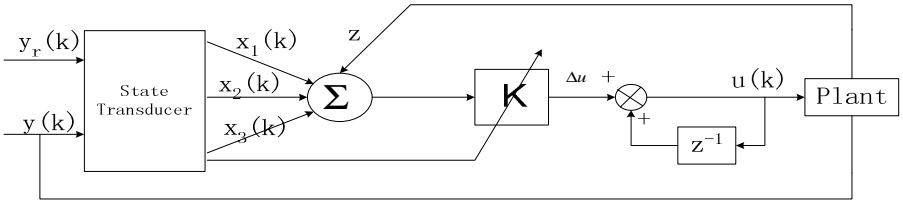


Fig. 1. Block diagram of a single neuron-based PID controller

Traditional PID control algorithms have the advantages of intuitive structure and algorithm. However, they are not very efficient for nonlinear and time-varying systems. Single neuron, which is the basic unit of neural networks, has the capability of self-adaptation. The fusion of single neuron with conventional PID would result in an 'auto-tuning' controller that benefits the merits from both sides. In [8], the authors present a single neuron-based PID controller, they demonstrate that their method is concise but efficient in tuning the PID controller parameters.

A single neuron-based PID controller can be depicted as in Figure 1. The input of the state transducer are the system output $y(k)$ and the reference input $y_r(k)$, the output of the state transducer are the state variables x_1, x_2, x_3:

$$\begin{cases} x_1(k) = y(k) - y_r(k) = e(k) \\ x_2(k) = e(k) - e(k-1) \\ x_3(k) = e(k) - 2e(k-1) + e(k-2) \end{cases} \tag{1}$$

In Figure 1, K is a gain coefficient in the single neuron. The control strategy of our single neuron-based PID controller is described as:

$$y(k) = y(k-1) + K\sum_{i=1}^{3} \omega_i(k) x_i(k) \tag{2}$$

Adaptive weights $\omega_1, \omega_2, \omega_3$ are introduced to act in a similar way like the regular PID parameters: K_P, K_I, K_D.

2.2 The Single Neuron-Based PID Control Model for AQM

Considering the averaging weight in RED, we introduce the weight ω_z into the element z^{-1} in figure 1, our model for AQM is described as:

$$y(k) = \omega_z y(k-1) + K\sum_{i=1}^{3} \omega_i(k) x_i(k) \tag{3}$$

We will analyze RED, P, PI, PIP, PD, SMVS, REM, AVQ based on (3) and show that these AQM schemes can be classified as Single neuron-based PID controller.

RED

RED configuration is specified through four parameters: the minimum and the maximum of the threshold($\min_{th}, \max_{th}$), the maximum dropping probability in the region of random discard $\max_p$, and the memory coefficient ω_q. RED can be described by the following equations:

$$avg(k) = (1-\omega_q) \times avg(k-1) + \omega_q \times q(k) \tag{4}$$

$$p(k) = \begin{cases} 0 & avg \le \min_{th} \\ \dfrac{avg(k) - \min_{th}}{\max_{th} - \min_{th}} \max_p & \min_{th} < avg < \max_{th} \\ 1 & avg \ge \max_{th} \end{cases} \tag{5}$$

considering the linear part output($\min_{th} < avg < \max_{th}$), we have:

$$p(k) = (1-\omega_q)\,p(k-1) + \omega_q \frac{\max_p}{\max_{th} - \min_{th}} (q(k) - \min_{th}) \tag{6}$$

we find it can be evolved from (3) when $y(k) = p(k)$, $\omega_z = 1 - \omega_q$, $y(k-1) = p(k-1)$, $e(k) = q(k) - \min_{th}$, $\omega_1 = 1$, $\omega_2 = \omega_3 = 0$, $K = \omega_q \frac{\max_p}{\max_{th} - \min_{th}}$.

PI

C.V.Hollot linearized the TCP/AQM control system model and proposed a PI controller expecting better responsiveness by calculating packet drop probability based on the current queue length instead of the average queue length [2]. PI controller has a transfer function of the form:

$$C(s) = K_{PI}(1/\omega_g + 1/s)$$

This can be converted into a difference equation, at time $t = kT$ where $T = 1/f_s$

$$p(kT) = p(kT - T) + a * (q(kT) - q\_ref) - b * (q(kT - T) - q\_ref) \tag{7}$$

where q_ref is the target queue length.

we find (7) can be evolved from (3) when $y(k) = p(kT)$, $\omega_z = 1$, $y(k-1) = p(kT-T)$, $e(k) = q(k) - q\_ref$, $\omega_1 = a - b$, $\omega_2 = b, \omega_3 = 0$, $K = 1$.

REM

In [7], an approach known as random exponential marking (REM) was developed and analyzed. REM measures congestion by a quantity called price. Price is computed by each link distributively using local information and is fed back to the sources through packet dropping or marking. It attempts to match user rates to network capacity while clearing buffers (or stabilize queues around a small target), regardless of the number of users. For queue l , the price $p_l(k)$ in period t is updated according to:

$$p_l(k) = \left[p_l(k-1) + \gamma(q_l(k) - (1-\alpha)q_l(k-1) - \alpha q^*)\right]^+ \tag{8}$$

where $\gamma > 0$ and $\alpha > 0$ are small constants, $[z]^+ = \max\{0, z\}$, q^* is the target queue length. When $p_l(k) > 0$, (8) can be evolved from (3) when $y(k) = p_l(k), \omega_z = 1$, $y(k-1) = p_l(k-1)$, $e(k) = q(k) - q^*$, $\omega_1 = \alpha$, $\omega_2 = 1 - \alpha, \omega_3 = 0$, $K = \gamma$.

AVQ

The AVQ algorithm maintains a virtual queue whose capacity is less than the actual capacity of the link. When a packet arrives in the real queue, the virtual queue is also updated to reflect a new arrival. Packets in the real queue are dropped when the virtual buffer overflows. The virtual capacity at each link is then modified so that flow entering each link achieves a desired utilization. The virtual link speed is determined by:

$$C_v(k) = C_v(k-1) + \alpha * \gamma * C * (t - s) - \alpha b(k) \tag{9}$$

where α is a smoothing parameters, γ is the desired utilization, C is the actual link speed, s is the arrival time of previous packet, t is current time, b is the number of bytes in current packet. $b^*(k) = \gamma * C * (t - s)$ shows the desired arrival bytes during $t - s$. From (9), we yield:

$$C_v(k) = C_v(k-1) - \alpha(b(k) - b^*(k))) \tag{10}$$

(10) can be evolved from (3) when $y(k) = C_v(k)$, $\omega_z = 1$, $e(k) = b(k) - b^*(k)$, $\omega_1 = -\alpha$, $\omega_2 = 0, \omega_3 = 0$, $K = 1$.

SMVS

SMVS controller for AQM was put forward based on Sliding Mode Variable Structure Control. SMVS can be described by the following equations:

$$p(k) = \begin{cases} \alpha X_1(k) & X_1(k)z(k) > 0 \\ -\alpha X_1(k) & X_1(k)z(k) \le 0 \end{cases} \tag{11}$$

$X_1(k) = q(k) - q_0$, $X_2(k) = f * (X_1(k) - X_1(k-1))$, $z(k) = 2X_1(k) + X_2(k)$, q_0 is the target queue length, f is the sampling frequency. (11) can be evolved from (3) when $y(k) = p(k)$, $\omega_z = 0$, $e(k) = q(k) - q_0$, $\omega_1(k) = \begin{cases} +\alpha & X_1(k)z(k) > 0 \\ -\alpha & X_1(k)z(k) \le 0 \end{cases}$, $\omega_2 = 0, \omega_3 = 0$, $K = 1$.

PD

PD is another AQM scheme developed using control theoretic approach.. it can be described by the following equation:

$$p(k) = p(k-1) + k_p \frac{e(k)}{B} + k_d \frac{e(k) - e(k-1)}{B} \tag{12}$$

where $e(k) = avg(k) - Q_T$, $avg(k) = (1-\beta)avg(k-1) + \beta q(k)$, here avg is average of the queue length; Q_T is the target queue length; β is the filter gain, $0 < \beta < 1$ which appears as an exponentially weighted average of the queue length; k_p is the proportional gain, k_d is the derivative gain. B is the router buffer size. (12) can be evolved from (3) when $y(k) = p(k)$, $\omega_z = 1$, $e(k) = avg(k) - Q_T$, $\omega_1 = k_p$, $\omega_2 = k_d$, $\omega_3 = 0$, $K = 1/B$.

PIP

PIP is the fusion of PI and Position feedback compensation. By choosing appropriate feedback compensation parameters, the properties of the corrected system can be determined mainly by series and feedback compensation elements. Thus, PIP can eliminate errors due to inaccuracies in the linear system model. The transfer function of the drop probability is

$$p(s) = \frac{1 + \tau s}{Ts} \delta q(s) - K_h q(s)$$

The update law of packet-drop probability can be described as:

$$p(k) = p(k-1) + \frac{1}{T}(q(k) - q_0) + (\frac{\tau}{T} + K_h)[q(k) - q(k-1)], \qquad k \geq 1 \tag{13}$$

where q_0 is the target queue length, τ is cascade feedback coefficient and K_h is position feedback coefficient.

(13) can be evolved from (3) when $y(k) = p(k)$, $\omega_z = 1$, $e(k) = q(k) - q_0$, $\omega_1 = 1/T$, $\omega_2 = \tau / T + K_h$, $\omega_3 = 0$, $K = 1$.

3 The Proposed Algorithm

Consolidating the advantages of single neuron and PID controller, the single neuron-based PID controller has the ability of coping with nonlinear and time-varying plants. This is our main purpose in introducing the framework for AQM. In this section, we propose an adaptive single neutron-based PID controller using square error of queue length as performance criteria.

One of the goals of an AQM scheme is to regulate queue length to a desired reference value with changing levels of congestion. So we use square error of queue length as our performance criteria. $J_1 = [q(k+1) - q\_ref]^2 / 2 = z^2(k+1)/2$, here q_ref is the target queue length.

The adjust value of connecting weights $\omega_i (i = 1,2,3)$ in (3) should make J_1 decrease, so $\omega_i (i = 1,2,3)$ adjust themselves along the direction of $-\partial J_1 / \partial \omega_i$:

$$\Delta\omega_i(k) = \omega_i(k+1) - \omega_i(k) = -\eta_i \frac{\partial J_1}{\partial \omega_i(k)} = \eta_i z(k+1) \frac{\partial q(k+1)}{\partial p(k)} \frac{\partial p(k)}{\partial \omega_i(k)} \tag{14}$$

(14) have explicit physical meaning-to decrease J_1. As the leaning rule, (14) together with (3), describe an adaptive scheme. We know PI fits into the model (3), so we use PI as the foundation to rebuild an adaptive single neutron-based PI controller for AQM(ASNPI). We evaluate our design by NS simulator and use common network topology with a single bottleneck link between r1 and r2 as depicted in Figure 2.

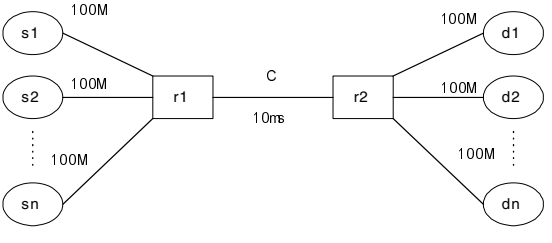


Fig. 2. Network topology

C is 3750pkt/s and the average packet size is 500B. Connections are established between s_i and d_i. The propagation delay ranges uniformly between 40ms and 220ms.

The buffer size in r1 is 800 packets and our target queue length is 100 packets. r1 runs AQM and supports ECN, while the other router runs Drop Tail. 200 FTP flows start during 0~1s, the queue lengths are depicted in Figure 3.

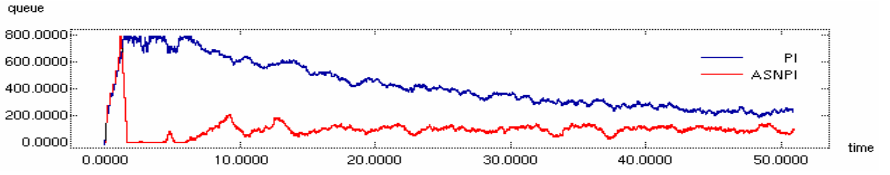


Fig. 3. The queue lengths of r1

As shown in figure 3, when r1 runs ASNPI, the system is faster acting but has larger overshoot. Because only the queue length appeared in the performance criteria, the scheme produces large Δp, it gives risk to large overshoot which is not allowed in our application. So we use $Pe^2(k)+Q\Delta u^2(k)$ as our performance criteria:

$$J_2 = 1/2\{P*[q(k)-q\_ref]^2 + Q*\Delta p^2(k)\}$$

here P is the weight of output error, Q is the weight of control variable. The adjust value of connecting weights $\omega_i(i=1,2,3)$ in (3) should make J_2 decrease, so $\omega_i(i=1,2,3)$ adjust themselves along the direction of $-\partial J_2/\partial\omega_i$:

$$\Delta\omega_i(k) = -\eta_i\frac{\partial J_2}{\partial\omega_i(k)} = \eta_i K\{Pe(k)x_i(k)\operatorname{sgn}(\frac{\partial q(k)}{\partial p(k)}) - QK[\sum_{i=1}^{3}\omega_i(k)x_i(k)]x_i(k)\}$$

$$\operatorname{sgn}(x)=\begin{cases}+1 & x\geq 0\\ -1 & x<0\end{cases} \tag{16}$$

Using PI as foundation to reconstruct our adaptive single neutron-based PI controller for AQM, from (7), we have

$$p(k) = p(k-1)+\omega_1(k)*(q(k)-q\_ref)+\omega_2(k)*(q(k)-q(k-1)) \tag{17}$$

($\omega_1(0)=a-b$, $\omega_2(0)=b$), according to (16), we yield(18):

$$\begin{cases}\omega_1(k+1)=\omega_1(k)+\eta_I(q(k)-q\_ref)*\{Pe(k)\operatorname{sgn}(\frac{\partial q(k)}{\partial p(k)})-Q[\sum_{i=1}^{2}\omega_i(k)x_i(k)]\}\\ \omega_2(k+1)=\omega_2(k)+\eta_P(q(k)-q(k-1))*\{Pe(k)\operatorname{sgn}(\frac{\partial q(k)}{\partial p(k)})-Q[\sum_{i=1}^{2}\omega_i(k)x_i(k)]\}\end{cases}$$

Now we summarize the adaptive single neutron-based PI controller for AQM(ASNPI):

step1: calculate p(0) using (17) ;
step2: read the new sample data q(k);
step3: calculate $\omega_1(k)$ and $\omega_2(k)$ using (18) ;
step4: calculate p(k) using (17), output the new value ;
step5: return to step2.

4 Simulation Results

In this section we study the performance of ASNPI in various traffic conditions and compare it with PI and ARED. The simulation topology and default configurations are shown in Fig.2. The buffer size in r1 is 800 packets and our target queue length is 100 packets. r1 runs AQM and supports ECN, while the other router runs Drop Tail. We use ns default parameters set in PI and ARED scheme, and set $\eta_I = (a-b)/(qlmit)^2, \eta_P = b/(qlmit)^2, P = 0.1, Q = 0.9$ in ASNPI scheme, where qlmit is the buffer size in r1, here is 800. To imitate real network situations, we adopt three ordinary traffic types, i.e., infinite FTP flows and burst HTTP flows based on TCP-Reno, exponential ON/OFF flows based on UDP. Among them, FTP flows always have data to send during simulation runtime. In contrast to long-lived FTP flows, HTTP flows are short-lived with an average page size of 10240B and an average request interval of 3s. The burst and idle times of the ON/OFF service model are 0.5s and 1s respectively, and the sending rate during "on" duration is 200Kbps.

A Experiment 1

In this experiment, we analyze the performance of the AQM schemes under varying traffic load. We compare the responsiveness and queue size of ASNPI, PI and ARED in the presence of long-lived FTP flows only. The number of FTP flows is 200 at the beginning, 100 FTP flows leave the link 100 seconds later, they join the link again when t=200s. The total simulation lasted for 300s.

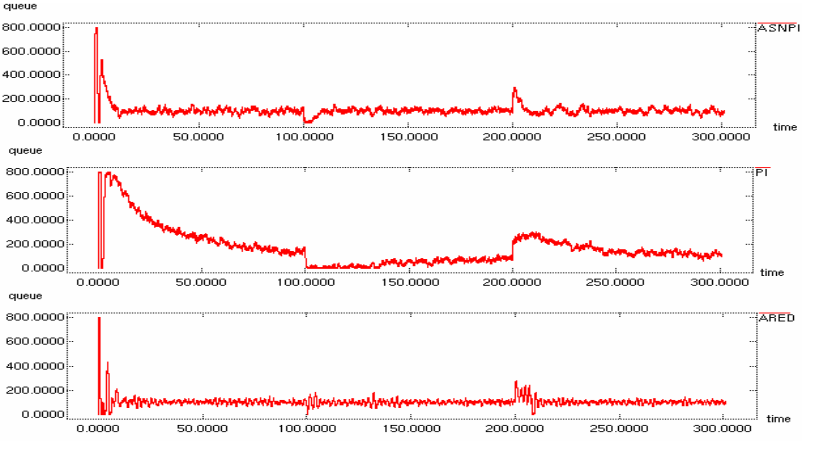


Fig. 4a. Experiment1.Evolution of the queue lengths with FTP flows

When the link bandwidth is 15 Mb/s, the queue lengths for the three algorithms are depicted in Figure 4a. ASNPI and ARED can regulate queue length to the desired reference value quickly, the queue lengths of PI climbs to the highest point when the number of FTP flows increases from zero to 200, then falls towards the target value when the load level stabilizes, it last small when the number of FTP flows decreases

from 200 to 100. Once the number of FTP flows increases suddenly, the queue length increases and converges slowly.

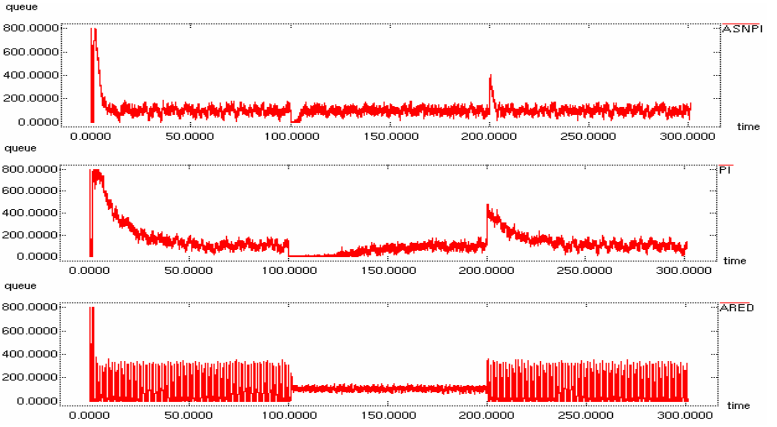
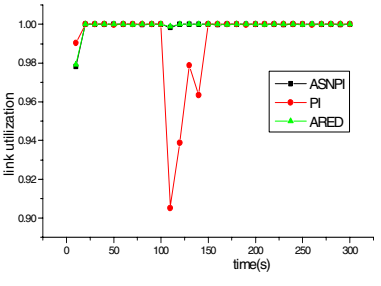


Fig. 4b. Experiment1.Evolution of the queue lengths with FTP flows

When the link bandwidth is 100 Mb/s, the queue lengths for the three algorithms are depicted in Figure 4b. As shown in figure 4b, ASNPI and ARED can regulate queue length to the desired reference value quickly, but ARED keeps the queue length at the desired value with large oscillations when N=200. The queue length of PI converges slowly once again.

The link utilizations are illustrated in Figure 5a and Figure 5b.



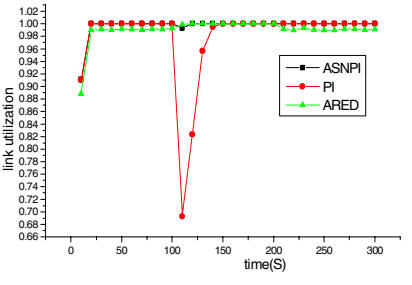


Fig. 5a. link utilizations when C=15Mbps **Fig. 5b.** link utilizations when C=100Mbps

When the queue lengths become very small, the link is under utilization. ASNPI and ARED have higher utilization than PI in experiment 1.The results of experiment 1 show that ASNPI and ARED have better transient response property, and ASNPI is more robust than ARED in the experiment environment set.

B Experiment 2

In this experiment, we analyze the performance of the AQM schemes when unresponsive flows exist. Here, the link bandwidth is 15 Mb/s, we use two mixtures: FTP and ON/OFF flows. The number of FTP flows is 100 at the beginning, 50 ON/OFF flows arrive at the link 50 seconds later. The queue lengths, plotted in Figure 6, show that ASNPI reaches the steady state in a short time, whereas PI takes longer time to stabilize. ARED keeps the queue length at the desired value with large oscillations. The results of experiment 2 show that ASNPI is more robust when unresponsive flows exist.

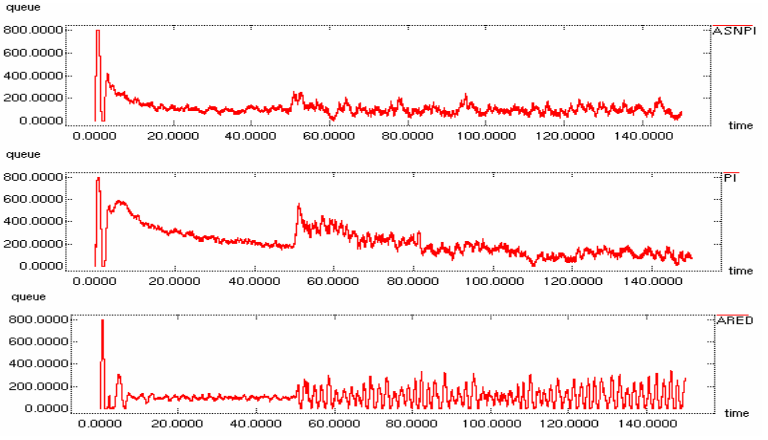


Fig. 6. Experiment2.Evolution of the queue length with FTP and ON/OFF flows

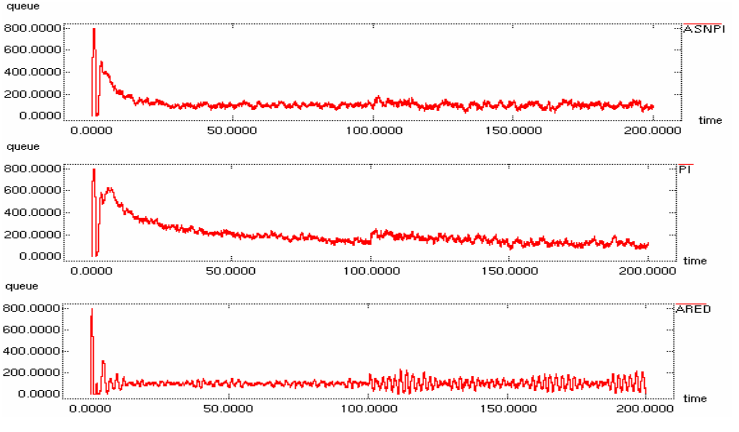


Fig. 7. Experiment3.Evolution of the queue length with FTP and HTTP flows

C Experiment 3

In this experiment, we analyze the performance of AQM when short-lived TCP flows exist. Here, the link bandwidth is 15 Mb/s, we use a mixture of FTP and HTTP flows. The number of FTP flows is 100 at the beginning, 300 HTTP flows arrive at the link 100 seconds later. The queue dynamics are plotted in Figure 7. As shown in figure 7, smaller oscillations for ASNPI and PI are observed. The results show that in experiment 3, ASNPI and PI are more robust than ARED when short-lived TCP flows exist.

5 Conclusions

In this paper, we present a generic framework which encompasses RED, P, PI, PIP, PD, SMVS, REM, AVQ as special cases by using Single neuron-based PID control, an adaptive mechanism is presented also. Based on the framework and adaptive mechanism, we develop an adaptive single neuron-based PI controller for AQM. The performance of ASNPI is evaluated by simulations and compared with PI and ARED. The results under a variety of network and traffic situations indicated that the proposed scheme exhibits more robust and adaptive congestion control behavior.

References

1. B. Braden, D. Clark, J. Crowcroft, B. etc, Recomedations on Queue Management and Congestion Avoidance in the Internet,RFC2309, April 1998.
2. C. Hollot, V. Misra, D. Towsley, and W. Gong. On designing Improved Controllers for AQM Routers Supporting TCP Flows. Infocom, 2001.
3. Zhang Heying, Liu Baohong, Dou Wenhua. Design of a robust active queue management algorithm based on feedback compensation. SIGCOMM2003, Germany, August 2003
4. J.S.Sun G.Chen S.Chan PD-controller: A new active queue management scheme Globecom 2003 - Next Generation Networks and Internet, San Francisco, USA, Dec.
5. Ren Fengyuan, Lin Chuang. A Robust Active Queue Management Algorithm Based on Sliding Mode Variable Structure Control. INFOCOM 2002 , New York,USA . 2002.
6. S.H.Low, A duality model of TCP and queue management algorithm, ITC Specialist Seminar on IP Traffic Management, Modeling and Management, Monterey, CA, 2000.
7. Athuraliya S, Low S H, Li V H, Yin Qing-He. REM: Active queue management. IEEE Network, 2001, 15(3):48-53
8. S. Yanagawa, and I. Miki, PID Auto-tuning controller using a single neuron for DC servomotor. IEEE International Symposium on Industrial Electronics, pp. 277-280, May 1992.

Designing Adaptive PI Algorithm Based on Single Neuron

Li Qing[1], Qingxin Zhu[2], and Mingwen Wang[3]

School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, P. R. China
[1] qingli_new@163.com
[2] qxzhu@uestc.edu.cn
[3] sohuwmw@sohu.com

Abstract. PI is a newly proposed Active Queue Management algorithm that has many important applications. But in case of heavy congestion its response is sluggish, and because of its static parameter's setting PI is sensitive to network status, such as RTT, capacity of neck-link, and number of TCP flows. To overcome these shortcomings of PI algorithm, we propose a novel AQM scheme, called Neuron based PI or NPI. NPI takes PI controller as an ADALINE with two inputs, and the proportional and integral factors of the controller are adjusted online by LMS algorithm. Simulation results show that under NPI the queue length converges to the desired value quickly and the oscillation is small.

1 Introduction

Internet uses end-to-end congestion control scheme, such as TCP, to prevent network congestion. Routers maintain a FIFO queue and drop packets only when the buffer is overflow, termed Drop-Tail. It is becoming increasingly clear that TCP coupling with Drop-Tail is not sufficient to provide acceptable performance [1]. As a more efficient feedback strategy than Drop-Tail, Active Queue Management (AQM) is proposed to enhance the endpoint congestion control. AQM enhances routers strength to detect and notify end-systems of impending congestion earlier by dropping or marking packets before the buffer is overflow. Hence AQM can improve network performance such as delay, link utilization, packet loss rate, and system fairness.

Random Early Detection (RED) [1] is the algorithm recommended by IETF to realize AQM [2]. However, RED is extremely sensitive to parameters setting and cannot prevent buffer's overflow under the situation of heavy congestion. Misra et. al. used control theory to analyze the performance of RED [3,4]. It is revealed that the direct coupling between queue length and the packet loss probability causes the above problems. To achieve the performance objectives including efficient queue utilization, regulated delay and robustness, AQM should be able to stabilize the queue length on a target value. Thus the PI controller is proposed as a better realization for AQM [5]. PI controller can reduce the steady state error by introducing an integral factor. But the parameters of PI controller are deployed statically and can't be adjusted online to adapt the changes of network status. It is shown that PI is very sluggish especially under heavy congestion situation, resulted in buffer's overflow or emptiness, correspondingly heavy packets losing or low link utilization [6].

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 800 – 807, 2005.

To overcome the shortcoming of PI, the configuration of parameters should be able to adapt the changes of network environment. Many adaptive AQM schemes are proposed recently. Among them we mention API [6] R-PI [7], S-PI [8] and STAQM [9]. STAQM is based on the estimation of network bandwidth and number of TCP flows. API, R-PI and S-PI adjust the gain of PI controller in order to quicken the response to the burst flows. There is tradeoff between response and stability in AQM. It is necessary to find a better way to enhance the adaptability for AQM.

In this paper the neural network control theory is introduced to design adaptive PI. A new AQM scheme, namely NPI or neuron based PI, is proposed to speed up the response of PI without sacrificing system's stability. NPI treats PI controller as an ADALINE (ADAptive LInear NEuron) [10] with two inputs, and the proportional factor and integral factor of the controller are adjusted online by LMS (Least Mean Square) algorithm. The structure of a single neuron is very simple and easy to realize. NPI solves the problem that PI can't control congestion efficiently, and at the same time it keeps its merit of low computing complexity. Under the control of NPI, the packets drop probability is not changed smoothly. Thus the response of NPI is faster than PI. In this paper we discuss some guidelines to design this adaptive PI controller.

The rest of the paper is organized as follows. In Section 2 we introduce ADALINE and its learning algorithm LMS. In Section 3 we present the control structure of an adaptive PI controller based on single neuron and analyze the convergence of NPI algorithm. We also discuss some design guidelines of NPI in this section. In Section 4 we give the simulation results of NPI with ns-2 [11] and compare its performance with PI controller. Some conclusions are given in Section 5.

2 The Adaptive Linear Neuron

A neuron is the basic unit of a neural network, which has the capability of self-study and self-adapt. A single neuron is easy to compute for its simple structure. ADALINE [10] uses both linear function as its propagation function and LMS algorithm as its learning rule. Fig. 1 shows an ADALINE with N inputs $x_1, x_2, \cdots, x_N$. The output is given by

$$u = \sum_{i=1}^{N} x_i w_i + b \,, \tag{1}$$

where, $w_i (i = 1, 2, \cdots, N)$ and b are weights.

The self-study capability of neuron is achieved by adjusting its weights. ADALINE uses LMS algorithm to regulate the weights. LMS algorithm is a kind of approximate steepest descent method. Let $F(W)$ be the objective function, where W is the weight vector. Suppose ∇F is the gradient of $F(W)$, the learning process with LMS is denoted by

$$W(n+1) = W(n) - \alpha \nabla F \,, \tag{2}$$

where α is the learning step. It is easy to see from (1) and (2) that ADALINE with LMS learning algorithm is self-adapted and fit for adaptive control. In the next section, we use the idea of ADALINE to improve PI controller and propose a new AQM, called NPI.

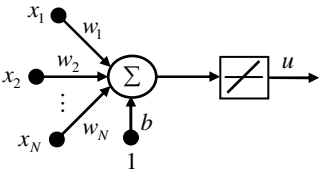


Fig. 1. ADALINE

3 The NPI Algorithm

3.1 The Feedback System with PI Controller

Before the buffer is overflow, AQM drops or marks packets randomly with probability p. When end hosts detect the network congestion status from the lost or marking sign of the packets, they can adjust their TCP window accordingly. Thus AQM in routers and TCP in end hosts together form a feedback system [4].

In the TCP/AQM dynamic model, the control object consists of the TCP mechanism of the end system and the queue lengths of the routers, and the AQM control law is regarded as the compensator. Considering the performance objectives such as efficient queue utilization, regulated queuing delay and robustness, AQM should be able to stabilize the queue length q to a target value q_{ref}. The AQM controller controls the packets arrival rate λ by setting a packet drop probability p as the control signal.

PI controller is proposed to achieve efficient stability [5]. Let f_s be the sample frequency. At each sampling instant $t = nT = n/f_s$, the PI algorithm updates the dropping probability p as follows:

$$p(n) = p(n-1) + a[q(n) - q_{ref}] - b[q(n-1) - q_{ref}], \tag{3}$$

where a and b are constants, q_{ref} is the target queue length. When the network parameters including flow number, RTT, and link capacity are known, a and b can be chosen according to the design guidelines of PI to make the TCP/PI system stable. It is known that PI is sluggish under heavy congestion condition [12]. According to control theory, it's a sacrifice of bigger phase margin to achieve higher stability.

3.2 Designing NPI Algorithm

The discrete PI controller can be described as:

$$p(n) = p(n-1) + K_p \delta q(n) + K_I(\delta q(n) - \delta q(n-1)), \tag{4}$$

where K_p and K_I are the coefficients of proportional factor and integral factor respectively, and $\delta q(n) = q(n) - q_{ref}$. Comparing (3) and (4), we have

$$K_p = a - b, K_I = b. \tag{5}$$

Let

$$\Delta p(n) = p(n) - p(n-1),\ x_1 = \delta q(n),\ x_2 = \delta q(n) - \delta q(n-1).$$

We can regard a PI controller as an ADALINE with two inputs x_1 and x_2. The output of the neuron is $\Delta p(n)$. Then K_p and K_I are weight numbers of ADALINE and they are adjusted by LMS learning algorithm.

As we described above, AQM should stabilize the queue length q on a target value q_{ref}. Therefore, we construct an objective function for the PI controller based on single neuron as follows:

$$J(n) = \frac{1}{2}[q(n) - q_{ref}]^2. \tag{6}$$

Suppose $\nabla J(n)$ is the gradient of $J(n)$, we have

$$K_p(n+1) = K_p(n) - \eta_p \delta q(n+1) x_1 \frac{\partial q(n+1)}{\partial p(n)}, \tag{7}$$

$$K_I(n+1) = K_I(n) - \eta_I \delta q(n+1) x_2 \frac{\partial q(n+1)}{\partial p(n)}, \tag{8}$$

where η_p and η_I are learning steps, $\partial q(n+1)/\partial p(n)$ is not known beforehand. It can be determined by pattern recognition method. For simplicity we may substitute $\partial q(n+1)/\partial p(n)$ with $\mathrm{sgn}[\partial q(n+1)/\partial p(n)]$. The introduced error can be compensated by regulating learning step.

To make the TCP/AQM system stable, we should adjust K_p and K_I in (7) and (8) to lead the objective function converging. The following proposition guarantees the convergence of $J(n)$.

Proposition 1. Suppose that the capacity of queue is B, and the target queue length is q_{ref}. In the AQM controller described by (4), (7) and (8), if η_p and η_I satisfy the following:

$$0 < \eta_p \le \frac{2}{\max\{\max_{n=1,2,\cdots}[q_{ref}^2, (q(n) - q_{ref})^2]\}}, \tag{9}$$

$$0 < \eta_I \le \frac{2}{\max_{n=1,2,\cdots}\{[q(n) - q(n-1)]^2\}}. \tag{10}$$

Then the objective function $J(n)$ is convergent.

Proof. Let

$$\Delta J(n+1) = J(n+1) - J(n).$$

Then

$$\Delta J(n+1)=\frac{1}{2}\eta_P[\delta q(n+1)\frac{\partial \delta q(n+1)}{\partial K_P(n)}]^2[-2+\eta_P(\frac{\partial \delta q(n+1)}{\partial K_P(n)})^2].$$

The objective function is convergent when $\Delta J(n+1)<0$, and this is equivalent to

$$\eta_P[-2+\eta_P(\frac{\partial \delta q(n+1)}{\partial K_P(n)})^2]<0.$$

Substituting $\partial q(n+1)/\partial p(n)$ with $\mathrm{sgn}[\partial q(n+1)/\partial p(n)]$ in (7), we have

$$0<\eta_p<2/(\frac{\partial \delta q(n+1)}{\partial K_p(n)})^2=2/[\delta q(n)]^2. \tag{11}$$

Similarly, we have

$$0<\eta_I<2/(\frac{\partial \delta q(n+1)}{\partial K_I(n)})^2=2/[q(n)-q(n-1)]^2. \tag{12}$$

(11) and (12) hold whenever (9) and (10) are true. This completes the proof.

Remark 1. From (9) and (10) we have $0<\eta_p,\eta_I\le 2/B^2$. Hence in practice η_p and η_I can be replaced by $2/B^2$. In this paper we choose $\eta_p=2/[\max(q_{ref},B-q_{ref})]^2$ and $\eta_I=\min[{q_{ref}}^2,(B-q_{ref})^2]$, to get better control performance and guarantee the convergence of the algorithm.

Remark 2. The initial values of K_p and K_I are crucial for the stability of TCP/AQM system. Motivated by the stability of PI controller, we set the initial value of K_p and K_I by equation (5), where a and b are chosen according to the design guidelines of PI in [5]. We use the same sample frequency f_s as that of PI. Thus from proposition 1 and the stability of PI we know that the feedback control system with new AQM is stable.

Remark 3. To increase the flexibility of NPI, we may add a coefficient k to the neuron, then (3) becomes

$$p(n)=p(n-1)+k[K_p\delta q(n)+K_I(\delta q(n)-\delta q(n-1)]. \tag{13}$$

We use k to adjust the amplitude of K_p and K_I, thus control the drop probability $p(n)$. The increasing of k lead to faster response of NPI, but the throughput will decrease sharply because of high drop rate as k becomes too large. The recommended value is $1\le k\le 4$. Thus the NPI algorithm is represented by (13), (7), and (8).

4 Simulation Results

In this section, we give some simulations with ns-2 [11] simulator to evaluate NPI controller and compare it with PI, ARED. The network topology used here is the same as in [5]. There are three kinds of traffic sources, i.e. TCP, UDP, and HTTP. In the following we start with different combinations of such sources. Round trip time is set to be 200ms. The capacity of AQM queue is set to be 350 packets, and target queue length is 150packets. Table 1 lists some notions and the default parameter settings in the corresponding AQM scheme.

Table 1. Parameters of AQM Controller

Controller	Parameters
PI[5]	$a = 0.00001822$, $b = 0.00001816$, $f_s = 160$
ARED[13]	$\min_{th} = 100$, $\max_{th} = 200$, $w_g = 1 - \exp(-1/C)$
NPI	$K_p = a - b$, $K_I = b$, $\eta_p = 0.000050$, $\eta_I = 0.000089$, $k = 4$

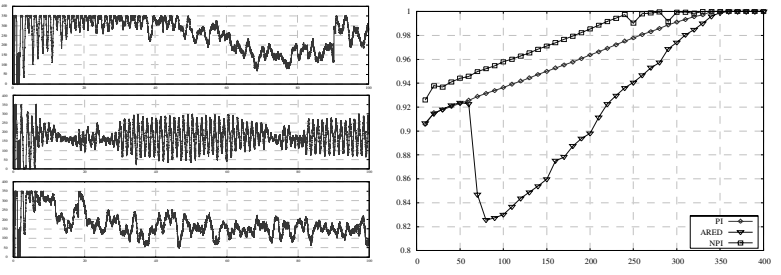


Fig. 2. Performance of PI, ARED and NPI under heavy network load. Left compares queue length vs. time, and right compares utilization vs. TCP number

4.1 Heavy Traffic Load

When the network traffic load is heavy, AQM should stabilize the queue length at a target value to avoid congestion. In this simulation, we start 500 TCP/Reno connections randomly at the beginning. The simulation lasts 50s. Fig. 2 (Left) shows the results for PI, NPI and ARED. The queue under the control of PI is overflow for almost 50s at the beginning, and then oscillation appears. ARED oscillates between 100 packets and 250 packets. Heavy oscillation will cause jitter and queuing delay. NPI shows better performance in response and stability. It takes only 10s to set the queue down to the equilibrium. Fig. 2 (Right) presents the link utilization of each AQM algorithms under different traffic load levels. The link utilization with NPI is higher than with other AQM algorithms.

4.2 Long-Lived TCP Flows Mixed with HTTP and UDP Connections

In this experiment, we consider the ability of NPI under the disturbances caused by HTTP and UDP connections. There are 350 long-lived TCP connections for all the

simulation time. The bursty HTTP traffic involves 200 sessions, and the number of pages per session is 150. We start 30 UDP flows at 20s. Each of the UDP sources follows an exponential ON/OFF traffic model. Both the idle and the burst times have mean of 0.5 second, and packet size is set at 500 bytes. The sending rate during on-time is 64kb/s. Fig. 3 shows the results of queue length vs. time for PI, NPI, and ARED. NPI can cope with the disturbances caused by HTTP and UDP connections well.

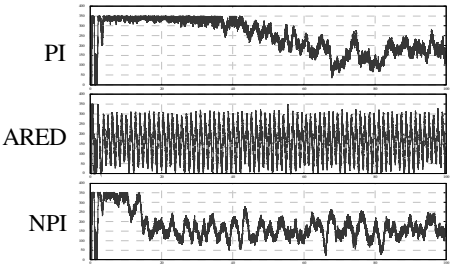


Fig. 3. Compare queue length vs. time of PI, ARED and NPI under mixed network load

4.3 Variation in TCP Traffic Load

To investigate the response time of NPI, we change the number of long-lived TCP flows. There also exist disturbances caused by 50 HTTP connections. The number of TCP flows is 150, 50, 250, and 100 at time 0, 50, 100, and 150 respectively. The result in Fig. 4 (Left) shows that the queue length of PI and ARED oscillates heavily, while NPI reaches the equilibrium point quickly. We show the drop probability of PI and NPI in Fig. 4 (Right). It is clearly to see that PI updates its packet drop probability slowly, while NPI adapts packet drop probability to the variations of traffic load.

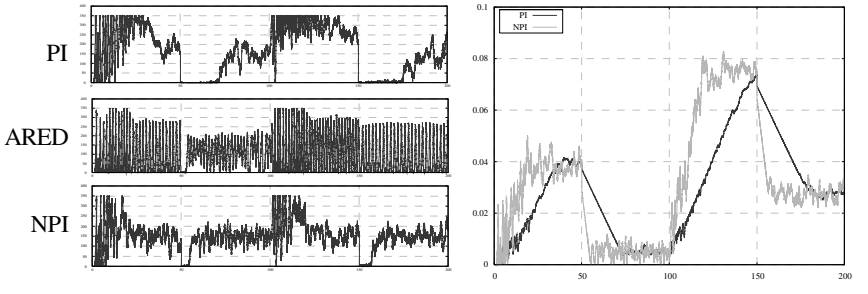


Fig. 4. Performance of PI, ARED and NPI under heavy network load. Left compares queue length vs. time, and right compares drop probability vs. time

5 Conclusions

The research of adaptive AQM is a hot topic of Internet. To overcome the sluggish problem of PI, a new adaptive AQM, called NPI, is proposed. The NPI algorithm is based on single neuron of ADALINE. LMS learning rule is used to tune the coeffi-

cients of proportional factor and integral factor. NPI can quickly regulate queue length to the desired value, respond much faster than PI and keep the oscillation small. Simulation results with ns-2 validate the better performance of NPI.

References

1. Floyd, S., Jacobson, V.: Random early detection gateways for congestion avoidance. IEEE/ACM Trans on Networking (1993) 1(4): 397-413
2. Braden, B., Clark, D., Crowcroft, J., et al.: Recommendations on queue management and congestion avoidance in Internet. RFC 2309, IETF (1998)
3. Misra, V., Gong, W. B., Towsley, D.: Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. In Proceeding of ACM/SIGCOMM (2000) 151-160
4. Hollot, C. V., Misra, V., Towsley, D., Gong, W. B.: A control theoretic analysis of RED. In: Proceedings of the INFOCOMM, Anchorage, Alaska (2001) 1510-1519
5. Hollot, C. V., Misra, V., Towsley, D., Gong, W. B.: On designing improved controllers for AQM routers supporting TCP flows. In Proceeding of INFOCOM (2001) 3: 1726 – 1734
6. Wang, C., Li, B., Sohraby, K.: API: Adaptive Propertional-Intergal Agorithm for Active Queue Management under Dynamic Environments. High Performance Switching and Routing (2004) 51-55
7. Chang, X. L., Muppala, J. K., Yu, J.: A robust nonlinear PI controller for improving AQM performance. IEEE International Conference on Communications (2004) 4:2272-2276
8. Wu, W., Ren, Y., Shan, X.: A Self-configuring PI Controller for Active Queue Management. In Asia-Pacific Conference on Communications (APCC), Session T53, Tokyo Japan (2001)
9. Zhang, H., Hollot, C., Towsley, D., Misra, V.: A self-tuning structure for adaptation in TCP/AQM networks. IEEE GLOBECOM (2003) 7: 3641 - 3646
10. Widrow, B., Hoff, M. E.: Adaptive Switching Circuits. IRE WESCON convention record: part 4. computers: Man-machine Systems, Los Angeles (1960) 96-104
11. UCN/LBL/VINT: Network simulator-ns2. http://www.isi.edu/nsnam/ns/
12. Ren, F., Lin, C., Ying, X., Shan, X., Wang, F.: A Robust Active Queue Management Algorithm Based on Sliding Mode Variable Structure Control. IEEE INFCOM (2002) 13-20
13. Floyd, S., Gummadi, R., Shenker, S.: Adaptive RED: An Algorithm for Increasing the Robustness of RED. http://www.aciri.org/floyd/papers/adaptiveRed.ps.

An Optimal Component Distribution Algorithm Based on MINLP*

Kebo Wang, Zhiying Wang, Yan Jia, and Weihong Han

National University of Defense Technology, China
kebowang@gmail.com

Abstract. While deploying distributed components, a key decision to be made is the location of each component in the target distributed environment. Unappropriate distribution may lead to bad performance. Existing distribution algorithms use criterion such as minimal communication bandwidth or response time to optimize object distribution. But for distributed applications processing massive parallel requests, which require both low response time and high throughput. In such situation single-criterion algorithms may output distribution with a very low throughput sometimes. We propose an algorithm called OCDA based MINLP(Mixed Integer Non-Linear Programming), which meets the requirement that with a restricted average response time, maximize throughput capacities of applications. Finally we discuss the advantages and limitations of OCDA.

1 Introduction

Last few years have seen a tremendous growth in the area of web-based application. As these applications are user oriented, their main target is to keep their huge amount of users satisfied by meeting certain QoS requirement like response time. These Internet oriented applications have to process massive parallel client requests. They must have good scalability, which means a low response time while processing large amount of requests simultaneously. For such application the more requests they can process the better it is. Low latency and high throughput are critical performance factors for these applications.

Performance of distributed application depends on many factors, one of them is the location of each software component in a set of networking servers. For a certain distributed application, different distribution scheme usually leads to different performance. This is important especially for distributed component application which is based on EJB or CCM technology.

There are already some solutions to work out the problem. MIP(Mixed Integer Programming) based algorithms are provides to output components distribution scheme optimized with single criterion like minimal communication

* This work has been partially supported by 863 Hi-Tech Research and Development Program of China(No.2004AA112020, 2003AA115210, 2003AA111020, 2003AA115410).

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 808–816, 2005.

bandwidth scheme or minimal response time scheme. While single-criterion distribution scheme like this often cannot meet the requirements in some situation. A minimal response time distribution scheme, for example, may result in a minimal response time, whereas the response time is unacceptably long for users. We need a distribution scheme with both good response time and high throughput, especially for Internet-oriented applications.

Minimal response time and high throughput contradict each other in distributing components. A minimal response time distribution scheme often means all components be deployed on the least number of fastest servers with least network delay. While a high throughput distribution scheme requires component be deployed on maximize number of servers os that application can utilize all processing powers, but is has a bigger response time due to network delay.

In this paper, we provide a performance model for distributed component based application, and a optimizing algorithm based on the performance model with restricted response time and highest throughput been taken into account.

2 Related Works

Some other researches[12][13] focus on applying performance model into component technology domain. Those models are developed for reasoning for non-functional properties of application or in reverse. They fall into three categories [11]. QN(Queueing networks)[10] is commonly used analytical model to predict performance. Some researches have been done to extending QN for more precise model, like LQN(Layered Queueing Networks)[7] and QPN(Queueing Petri Networks)[8]. But it is much more different to solve.

[1][2] thinks that object execution time on a node can be ignored compared to network delay and then develops a a binary integer programming(BIP) model that takes into account an application's communication patterns and the target network characteristics. The goal is to minimize the overall remote communication bandwidth for the application, while achieving a meaningful distribution of components in the target network setting. [5][9] take into account both distributed objects execution time and network transporting time, present a algorithm of minimal response time distribution. Karin Högstedt provide an approach to minimize the network bandwidth using a graph cutting algorithm[6][4].

3 Distributed Application Performance Model

A performance prediction must be provided in order to quantitatively calculate the response time of the distributed component-based application. Simulation based approach is not suite for this situation, we choose QN as the performance model basis. LQN and QPN are too complex for optimizing problem.

In a distributed system, A client call is usually accomplished by several components interacting each other. Response time of a client call depends on many factors that range from the complexities of implementations to the processing power of nodes and the interaction patterns of components.

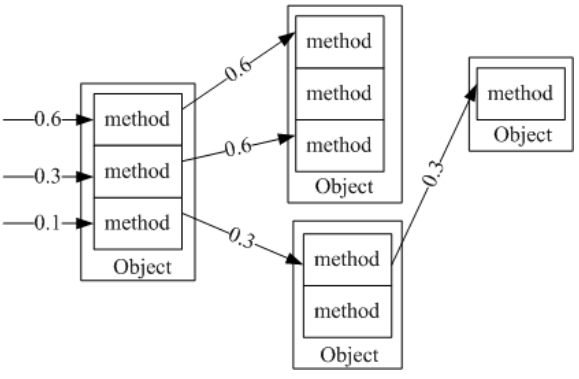


Fig. 1. Components Interact Each Other Following a Certain Pattern

Assumed that the application consist of N nodes and M components represented by $C_j(1 \leq j \leq M)$. These components are deployed on the internetworking nodes. Every node has been simplified by a CPU queue and IO queue. CPU processing power of node i is described as S_i^{CPU}. S_i^{CPU} is the ratio of processing power of node i and the fastest node, so we have $0 < S_i^{CPU} \leq 1$. Again, we can describe IO processing power as S_i^{IO}, $0 < S_i^{IO} \leq 1$.

There are access pattern and interacting pattern for most distributed applications. For a certain distributed system, we can evaluate accessing frequency of every method of component by these patterns. For component C_j, its fth method($1 \leq f \leq F_j$, F_j is the number of methods of component C_j) accessing frequency ratio is λ_{fj}. And the accessing frequency of the most accessed component is t. The average accessing frequency of fth method of component C_j is $\lambda \times t$. The bigger t is, the more distributed application's average throughput.

The average CPU demand of fth method of component C_j described as d_{fj}^{CPU}, so its average CPU service time is d_{fj}^{CPU}/S_i^{CPU}.

We describe network latency is a constant l for simplicity.

In a $M/M/1$ queueing network, the average utilization of a server can be computed by: $\overline{\mu} = \overline{f} \times \overline{d}$, where $\overline{f}$ is the average arrival rate and $\overline{d}$ is the average service time.

If component C_j is deployed on node i, then the CPU utilization of node i can be calculated by $\sum_j \frac{\lambda_{fj} \times t \times d_{fj}^{CPU}}{S_i^{CPU}}$, can be simplified as: $\sum_j \lambda_{fj} \times d_{fj}^{CPU} \times \frac{t}{S_i^{CPU}}$.

$\sum_j \lambda_{fj} \times d_{fj}^{CPU}$ describe the load characteristic of applications implementations. It is related to component's CPU demanding, inter-networking and accessing pattern. We express it by L_j.

The average response time of a distributed component can be given through Little's Law: $R = \sum_{r=1}^{K} R_i$ and $R_i = \frac{d}{1-U_i}$.

According to Little's Law, the average response time for a method of a component is sum of CPU service time and IO service time. So we can write:

$\frac{d_{fj}^{CPU}}{1-U_i^{CPU}} + \frac{d_{fj}^{IO}}{1-U_i^{IO}}$, where U_i^{CPU} is the average CPU utilization of node i, U_i^{IO} is the average utilization of IO device of node i.

4 A MINLP Problem

For any distributed systems, the following constraints must be satisfied:

- CPU utilization of any node can not exceed 1
- IO utilization of any node can not exceed 1
- An distributed component can not be deployed more than one node

We defined $M \times N$ decision variables where M is the number of application components and N is the number of nodes in the network.

$$x_{ij} = \begin{cases} 1 & \text{if component } C_j \text{ is assigned to node } i \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The completeness constraint specifies that each component is assigned to one and only one node, and that all components in the application are assigned,

$$\sum_i x_{ij} = 1 \tag{2}$$

The run time CPU and IO utilization for each node can not exceed 1. So we can write:

$$\sum_j \sum_f \lambda_{fj} \times d_{fj}^{CPU} \times x_{ij} \times \frac{t}{S_i^{CPU}} < 1 \tag{3}$$

$$\sum_j \sum_f \lambda_{fj} \times d_{fj}^{IO} \times x_{ij} \times \frac{t}{S_i^{IO}} < 1 \tag{4}$$

For the methods f and their owner components j which was traversed by a client call($j \in J$ and $f \in F$, where J is a set of components traversed by the call and F is a set of methods traversed by the call), we can calculate the response time of the call by : $\sum_j^{j \in J, f \in F} \sum_i \left(\frac{d_{fj}^{CPU}}{1-U_i^{CPU}} + \frac{d_{fj}^{IO}}{1-U_i^{IO}} \right)$.

Then we compute network delay separately. For co-located components the network delay is zero. Otherwise for component C_{j_1} deployed on node i_1 and component C_{j_2} deployed on node $i_2 (j_1 \neq j_2)$, we can write the following about the network delay: $l \times \sum_{i_1}^{N} x_{i_1 j_1} \times \sum_{i_2 \neq i_1}^{N} x_{i_2 j_2}$. That is: $\sum_{i_1}^{N} \sum_{i_2 \neq i_1}^{N} x_{i_1 j_1} \times x_{i_2 j_2} \times l$.

While modelling performance by queueing networks, we usually assume that the whole application system to be a *markoviansystem*, which means that the distribution of the inter-arrival times and service times are exponential distributions(Poisson). So if we say the average response time is less than a constant T by a probability of α, then we can get that the average response time C satisfy: $C = \frac{1}{-\ln(1-\alpha)} \times T$.

And we have:

$$\sum_{j}^{j\in J, f\in F} \sum_{i} \left(\frac{d_{fj}^{CPU}}{1 - U_i^{CPU}} + \frac{d_{fj}^{IO}}{1 - U_i^{IO}} \right) \times x_{ij} + \sum_{j_1, j_2}^{j_1\in J, j_2\in J} \sum_{i_1}^{N} \sum_{i_2\neq i_1}^{N} x_{i_1 j_1} \times x_{i_2 j_2} \times l \leq C \tag{5}$$

The highest throughput means that we must compute largest t, so we have the following:

$$t > 0 \tag{6}$$

and the following objective function:

$$Max\ t \tag{7}$$

We can get a MINLP defined by (1) (2) (3) (5) (6) (7) which regard t and x_{ij} as variables. If we get rid of constraint (5) and let $a = \frac{1}{t}$, then the MINLP can be simplified to a BIP(mixed Binary Integer Programming) problem.

5 OCDA Algorithm

From the solution of the MINLP we can know the distribution of components. There is no universal way to solve to MINLP problem. BB(Branch and Bound) is standard solution to MIP. We present a resolution to the MINLP which is a variation of BB.

In a standard BB , the problem is branched into two sub-problem. In our variation we use (5) to cut branches as well as upper bound and lower bound. The following describe the OCDA algorithm:

Input: performance metadata of servers and distributed components, network delay;
Output: The Optimal component distribution;

1. Compute to obtain the MINLP problem according to the inputs. We define the MINLP as p(MINLP). let $a = \frac{1}{t}$ and get rid of constraint (5), we get a MILP defined as p(MILP) and a relaxation LP of the MILP defined as p(LP). then:
   – if p(LP) is unfeasible then exit. p(MINLP) is unfeasible, too.
   – if p(LP) is feasible and the solution satisfy constraint (5) and it is integer, then output the solution and exit.
   – if p(LP) is feasible but the solution do not satisfy constraints (5) or it is not integer, then let the reciprocal of the value of objective function as the upper bound $\overline{z}$
2. Find the solution for deploying all components to the fastest computer, represent by x^*, and let the objective function value as the lower bound $\underline{z}$
3. let $\Delta = \{(MINLP)\}$
4. Cut the branches whose objective function value is less than lower bound $\underline{z}$, if $\Delta = \phi$, then output the x^* as the solution and exit, otherwise go to 5.

5. Choose a branch from Δ and let it as (CP), $\Delta = \Delta \backslash \{(CP)\}$.
6. For (CP), let $a = \frac{1}{t}$ and get rid of constraint (5) and represent the relaxation problem as (CP'), (CP') is a linear programming problem. solve (CP').
7. if (CP') is unfeasible then go to 4, otherwise go to 8.
8. if (CP') has a objective function value less than $\frac{1}{\underline{z}}$, then go to 4, otherwise go to 9.
9. if (CP') has the solution which can satisfy constraints of (CP), then go to 11, otherwise go to 10.
10. if (CP') has the solution which is integer but do not satisfy constraint (5), then cut this branch and go to 4, otherwise from (CP'), select a non-integer variable x'_j, let it equal 0 and 1 respectively to break (CP) into two subproblem (CP_1) and (CP_2), let them join Δ then go to 5.
11. if the objective function value of (CP') is less than $\frac{1}{\underline{z}}$, then let $x^* = x'$ and $\underline{z} = \frac{1}{x'_0}$, otherwise go to 4.

End of the algorithm. We'll proof the correctness of the algorithm.

Proposition 1. *The Algorithm is correct.*

Proof (of proposition). The output of the algorithm is x^* and z. If $p(MINLP)$ is feasible and there exists a optimal solution x^{**} and its objective function value z^* for $p(MINLP)$, so $z^* \geq z$ holds.

In OCDA algorithm we can easily know that x^{**} and z^* has its corresponding subproblem $(CP^* \subset \{(MINLP)\}$ because OCDA enumerate all possible subproblems. From step 4 and 11 of OCDA, we can get $z^* \leq z$.

From $z^* \geq z$ and $z^* \leq z$ we get $z^* = z$

End of proof.

OCDA is derived from branch and bound, which implicitly enumerate all possible subproblems. In the worst situation complexity of OCDA is $2^{M \times N}$. But considering (2), that is, one component can be deployed on one server only. So the complexity reduce from $2^{M \times N}$ to N^M. Branch and bound converge fast usually. Our variation due to the existence of (5) constraint is a little slower than standard BB.

The speed of OCDA is determined the speed the lower bound $\underline{z}$ approaching to z. In step 10 of OCDA, the sequence of how to choose a variable to partition the original problem into two subproblem determined how far the lower bound $\underline{z}$ go forward to z.

6 Experiment Result

We implements our distributing algorithm on StarCCM[14] platform. StarCCM is an open source CORBA Component compliant application server in C++. Taking that calculating IO requirement is homogeneous as CPU requirement into account, we experiment on CPU requirement only in various components

distribution scheme. Our experiment environment is mode of 5 CCM components and 3 server nodes and a client. The computers are interconnected by a 100M Ethernet switch. Each CCM component has different CPU requirement interacting each other with only one component made of 3 business method interacting with client. The processing power of each server node is list in table 1.

Table 1. Processing power of nodes

Server Node	Server1	Server2	Server3
Server Information	PIV 2.4G 512M	PIV 1.6G 512M	Celeron 733M 128M
Processing Power	1(Base Node)	0.603	0.340

The processing power is computed by the average ratio of a CPU density CCM component running on each node in turn.

After testing 1000 requests, load factor L_j of each CCM component is listed in table 2.

Table 2. Load factor of component

	Component1	Component2	Component3	Component4	Component5
L_j	7.657	3.132	5.331	3.060	13.301

A multi-thread client program run on the client node to simulate multiple clients. For a certain client, a request is issued at 2 second interval to one of the three methods by predefined probability with 1000 requests in total. Number of simulated client vary from 1 to 500 for each distribution scheme.

Each simulated client issues 100 requests as warm up in test so that each service enter stabilization state to get a more accurate result. At the same time the following should be monitored:

1. The average CPU utilization of client node;
2. The average CPU utilization of each server node;
3. The average percentage of page fault *per* second of each server node;
4. The average throughput of the network.

In test the CPU utilization of client node should be no more that 30%. The network throughput should be less that 10% of 100M. The percentage of page fault *per* second of the server node should be less that 5% or 6%.

The distribution scheme include a optimal distribution calculated by OCDA and other 2 distribution as table 3.

In plan2 we distribute all component on the fastest node, while in plan1 we distribute the component on random server.

Table 3. Distribution scheme

	Server1	Server2	Server3
OCDA	2,5	1,4	3
Plan1	5	1,3,4	2
Plan2	1,2,3,4,5	*N/A*	*N/A*

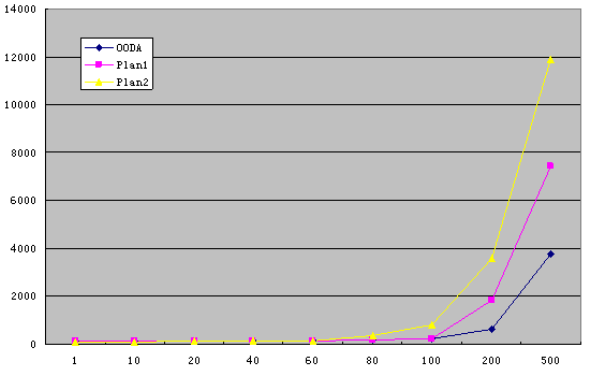


Fig. 2. Experiment Result

Response time of distributed application vary when we scale the number of their clients. A scalable application have an acceptable response time while dealing massive client requests. The experiment result is showed as figure 2.

In figure 2, horizontal axis show the number of clients(nonlinear), vertical axis represent response time. For 10 clients situation, response time of plan1 is 109ms, while OCDA and plan1 are 119ms and 121ms respectively. In plan2, all components are deployed on one computer, so its response time is better because of no network delay. In the situation of 100 clients, for plan2, CPU processing power becomes bottleneck, its response time increase fast comparing to 10 clients situation. Response time of plan1 increase slower than plan2. For OCDA, its response time increase slowest in the three.

7 Conclusion

Our solution is suitable for most distributed application which can modelled by Queueing Networks. The algorithm will calculate an optimal distribution of components. For complicated interaction pattern among components like distributed transaction and component replica is our future research goal. A heuristic search should also be considered so as to get a faster algorithm than OCDA.

References

1. Bastarrica, C. Shvartsman, A.A., Demurjian, S.A. A Binary Integer Programming Model for Optimal Object Distribution. In Proc. of 2nd International Conference On Pronciples of Distributed Systems, OPODIS'98. pp. 91-105, Amiens, France, December 1998.
2. C. Bastarrica, S. Demurjian and A.A. Shvartsman, Software Architectural Specification for Optimal Object Distribution, in Proc. of XVIII International Conference of the Chilean Society of Computer Science, IEEE Press, 1998.
3. C. McArdle, T.Curran. Optimal Object Placement, Load Distribution and Load Control for Distributed Telecommunication Service Applications. In Proc. of 17th International Teletraffic Congress, Salvador, Brazil, December 2001.
4. Karin Högstedt, Doug Kimelman, V.T. Rajan, Tova Roth, Mark Wegman. Graph Cutting Algorithms for Distributed Applications Partitioning. ACM SIGMETRICS Performance Evaluation Review. v.28 n.4, p.27-29, March 2001.
5. B. Liu, S. Jha, C. Chou, P. Ray. Resource Allocation for Networked Application Using Binary Integer Programming. In Proc. of the 7th International Symposium on DSP and 2nd WITSP 2003.
6. Karin Högstedt, Doug Kimelman, Nan Wang. Fast Optimality-Preserving Graph Reduction for Dynamics-Based Partitioning of Distributed Object Applications.
7. J.A. Rolia, K.C. Sevcik. The Method of Layers. IEEE Transaction on Software Engineering, vol.21, no.8 pp.689-700, August 1995.
8. Falko Bause. Queueing Petri Nets –A Formalism for the Combined Qualitative and Quantitative Analysis of Systems. 5th International Workshop on Petri Nets and Performance Models, Toulouse, France, p.14-23. 1993.
9. B. Liu, S. Jha, C. Chou, P. Ray. Optimized Allocation of Distributed Applications Across Local Area Networks. 28th Conference on Local Computer Networks, p. 291-292, 2003.
10. P. Kahkipuro. Performance Modeling Framework for CORBA Based Distributed Systems. Ph.D. Dissertation, Technical Report A-2000-3, Department of Computer Science, University of Helsinki, Finland, 2000.
11. Viktoria Firus, Steffen Becker. Towards Performance Evaluation of Component Based Software Architectures. In Proc. of Formal Foundation of Embedded Software and Component-Based Software Architectures, 2004
12. Shiping Chen, Ian Gorton,Anna Liu, and Yan Liu,Performance Prediction of COTS Component-based Enterprise Applications, CBSE5, Orlando, Florida, USA, May 2002.
13. Antonia Bertolino, Raffaela Mirandola: Modeling and Analysis of Non-functional Properties in Component-based Systems. Electronic Notes in Theoretical Computer Science, 82(6), 2003.
14. StarCCM, An CORBA Component Platform. http://starccm.sourceforge.net.

An Efficient Anomaly Detection Algorithm for Vector-Based Intrusion Detection Systems

Hong-Wei Sun[1], Kwok-Yan Lam[1], Siu-Leung Chung[2], Ming Gu[1], and Jia-Guang Sun[1]

[1] School of Software, Tsinghua University, Beijing 100084, PR China
{sunhw, lamky, guming, sunjg}@tsinghua.edu.cn
[2] School of Business & Administration, The Open University of Hong Kong
slchung@ouhk.edu.hk

Abstract. This paper proposes a new algorithm that improves the efficiency of the anomaly detection stage of a vector-based intrusion detection scheme. In general, intrusion detection schemes are based on the hypothesis that normal system/user behaviors are consistent and can be characterized by some behavior profiles such that deviations from the profiles are considered abnormal. In complicated computing environments, users may exhibit complicated usage patterns that the user profiles have to be established using sophisticated classification methods such as vector quantization (VQ) technique. However, anomaly detection based on the data set in a high dimension space is inefficient. In this paper we focus on the design of an algorithm that uses principal component analysis (PCA) to improve the anomaly detection efficiency. The main contribution of this research is to demonstrate how the efficiency of the anomaly detection can be raised while the effectiveness of the detection in terms of low false alarm rate and high detection rate can be maintained.

Keywords: Intrusion Detection; Multivariate Data Analysis; Vector Quantization; Principal Component Analysis.

1 Introduction

In the information age, our society is increasingly relying on information infrastructures to support critical operations such as telecommunications, banking, transportation, defence, etc. Intrusions into information systems have become a significant threat with potentially severe consequences. In order to protect information systems, it is highly desirable to detect intrusive activities so that remedial actions can be taken before severe damages are done to the systems.

Intrusion detection [1] aims to detect a wide range of security violations – from attempted break-ins by outsiders to system penetrations and abuses by insiders. Many intrusion detection techniques have been proposed [2,3,4,5,6,7,8]. Existing intrusion detection techniques fall into two major categories: misuse

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 817–825, 2005.

detection and anomaly detection [2,3,4]. Misuse detection techniques, also referred to as rule-based approach in some literatures, identify and store misuse patterns of known intrusions, and signal intrusions when there is a match between on-going activities and the stored patterns. However, misuse detection techniques fail to detect intrusions whose misuse patterns are not known [5,6]. Anomaly detection schemes are based on the hypothesis that normal systems/users are engaged in specific functions/tasks, hence tend to exhibit some consistent behavior patterns; and deviations from such patterns can be detected and considered suspicious. Hence, anomaly detection techniques can detect unknown attacks if system/user behavior profiles are established effectively [7,8].

Anomaly detection systems typically involve three stages, namely, monitoring of system/user activities, analysis of audit data and profile creation, and anomaly detection. The effectiveness of anomaly detection is based upon the hypothesis that individual users logging into the system are engaged in specific functions or tasks, hence tend to exhibit some consistent behavior patterns. These behavior patterns are represented in the form of user profiles. For individual users, it has been demonstrated that the resulting user profiles are relatively simple and can be built using statistical approaches such as the Q-statistic of the NIDES system [9]. One possible complication of the resulting user profiles is that they may be composed of multiple groups of correlated activities. [10] proposed the use of multivariate data analysis to identify groups of correlated user activities for representing user behavior profiles. However, the detection performance of [10] is affected by its ability to quantify and measure the deviations between user sessions and the behavior profiles.

A number of recent efforts in anomaly detection adopted vector-based clustering algorithms[11,12,13]. [11] used fuzzy clustering to analyze data streams that captures process, system and network states and detects anomalous behaviors. [12] used Y-means which can partition the training data set into an appropriate number of clusters automatically. [13] used vector quantization (VQ) technique to represent user profiles for intrusion detection in complicated computing environment. All these systems use vectors to represent user activities in the computer systems and establish user profiles. The user profiles established are then used for the detection of abnormal activities. However, they all put the emphasis in the data analysis and profile creation stage while in the detection stage, simple full search algorithms are used in most cases. For vector-based intrusion detection systems, it will be computationally intense if a full search algorithm is used in the detection stage.

For a practical deployment of an intrusion detection system, anomaly detection is usually required in real time, hence, the efficiency of the detection stage is of vital importance. In this paper, we propose an algorithm, which uses the principal component analysis (PCA) technique to enhance the efficiency of the detection stage of a vector-based intrusion detection system. Basically, PCA is a dimensionality reduction method, by which the original high dimensional data set will be projected into a lower dimensional space. The process will

inevitably leads to fidelity loss if analysis is done based on the projected data set. For intrusion detection systems, fidelity loss may influence the effectiveness of the detection. In this respect, we need to find a way to improve the efficiency of the anomaly detection without sacrificing the effectiveness. The proposed anomaly detection algorithm will demonstrate how the balance between the efficiency and effectiveness can be maintained. This algorithm has been applied to an intrusion detection system which contains user profiles built using the VQ technique [13].

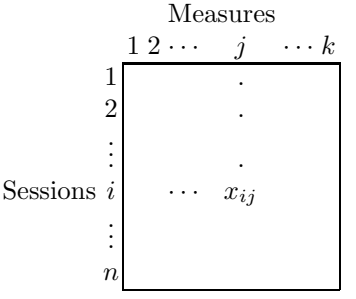
The rest of the paper is organized as follows. In Section 2, we will have a brief recap of the VQ-based technique for the building of user profiles. In Section 3, we describe the fundamental concepts of principal component analysis (PCA) and the anomaly detection algorithm based on PCA. Experimental results demonstrating the efficiency and effectiveness of this algorithm are presented in Section 4. The paper is concluded in Section 5.

2 Intrusion Detection Based on Vector Quantization

As mentioned, in order for anomaly detection to be successful, an essential step is to capture the behavior patterns of the users and represent them in the form of user profiles. In cases where the user patterns exhibit multiple groups of correlated user activities, simple statistical approaches such as the Q-statistic in the NIDES [9] or the χ^2 statistic proposed by [7,8] are not sufficient to represent the complex usage patterns.

[13] formulated the user profiles generation problem based on the vector quantization (VQ) technique. VQ is an efficient technique in data compression which is used extensively in image processing [14]. The VQ technique aims to extract the the most important characteristics of a large data set (the global image data in image processing) and represent them with a small set of data called the codebook. In the codebook, there are a number of codewords and each codeword is in fact a representative vector of a certain partition of the global data set. The codebook generation process in VQ is an optimization problem in a high dimensional space with the codebook generated aims to be an optimal approximation of the large data set. The codebook generation process does not require the structure of the original global data set to be known a priori and the finding of the optimal codebook is formulated as an optimization problem and the optimal solution is generated automatically through a series of iterations.

With these characteristics, the codebook generation technique in VQ is highly suitable for user profile generation in intrusion detection. In a computer system, a user typically uses more than one command in a single session. Therefore data collected from audit log are naturally taken as a multivariate data set. The user profiles for each user are built upon the set of multivariate measures that represent the counts of events over a number of sessions. Let the measures on k different events from n sessions, which are extracted from the audit files that are known to be from a particular user, be represented in the form of matrix M shown as follows:



where x_{ij} denotes the value of measure $j(1 \leq j \leq k)$ in session $i(1 \leq i \leq n)$ of the particular user being monitored. The matrix M also denotes a collection of n vectors $(x_i : i = 1, \ldots, n)$ of dimension k.

According to the intrusion detection hypothesis, user activities during each session are governed by behavior patterns which are influenced by the user's job duty and habit. Thus, sessions belonging to the same user are closely correlated. During the profile generation stage, all these n sessions are assumed to be normal user sessions of the genuine user. If we treat these n sessions as n vectors in the high dimensional space, these vectors should be able to be partitioned into groups of closely correlated points in the high dimensional space. The user profiles should then be formulated into a problem of finding the most representative vector of each of these partitions. As we have mentioned in the earlier paragraphs, the codebook generation in VQ should be a suitable technique. To proceed with the profile generation (or in the VQ context the codebook generation), the n vectors that represent the n user sessions will be treated as a set of training vectors. This set of training vectors will then be used to generated a codebook $C = (c_1, \ldots, c_N)$ where $c_i, i = 1 \ldots, N$ are the codewords. The number of codewords N is a predetermined value. Which number N should be chosen to give the best result depends on the nature of the global data set which is possibly another future research topic. There are a number of existing algorithms in VQ for the generation of the codebook C. In this research, we adopt the LGB algorithm proposed by Linde et al and for the details of the LBG algorithm for the codebook generation, please refer to [13] or [14]. The codebook C generated is then stored as the user profiles of the particular user for the future use of anomaly detection.

For anomaly detection of a new user session, suppose the event counts of the user session extracted from the audit log is represented by the vector $y = (y_1, \ldots, y_k)$. In order to determine whether this new user session is normal or not, we have to measure how "close" is y with the user profiles, i.e. the codebook C. This "closeness" is measured by the shortest squared Euclidean distance d of y and C which is given as

$$d = \min_l \sum_{i=1}^{k} (y_i - c_{li})^2.$$

To determine whether the session is normal or not, d is compared with a pre-determined threshold τ. If $d \leq \tau$, the session is labelled as normal, otherwise it is labelled as intrusive.

In this anomaly detection process, we can see that we have to compute the squared Euclidean distance between y and all the codewords c_l of C in order to find the minimal d which is inefficient in terms of amount of computations in real time. In this research, we propose an efficient detection algorithm making use of Principal Component Analysis (PCA) which is presented in the next session.

3 Efficient Detection Algorithm Based on Principal Component Analysis

In order to understand our detection algorithm, we firstly introduce the basic idea of principal component analysis (PCA). The idea of PCA is to project vectors in a high dimensional Euclidean space into a subspace where the variance among the original vectors can be maximally retained. The projected subspace of dimension 1 is called the principal axis of the vectors.

Given a set of N vectors $C = \{c_i : i = 1, \ldots, N\}$ and each vector $c_i = (c_{i1}, \ldots, c_{ik})$ is in the k-dimensional Euclidean space. The principal axis of the set of vectors is given by the unit vector

$$V = (v_1, \ldots, v_k),$$

such that the sum of projection of all the N vectors onto V, i.e. $\sum_{i=1}^{N} c_i^T V$ is the maximum among all possible V in R^k.

The algorithm to find the principal axis of the N vectors $\{c_i : i = 1, \ldots, N\}$ is stated as follows:

Step 1: Construct a matrix $A = (c_{ij})_{N*k}$.
Step 2: Construct the normalized matrix $\hat{A} = (\hat{c}_{ij})_{N*k}$, where

$$\hat{c}_{ij} = \frac{c_{ij}}{\sqrt{\sum_{j=1}^{k} c_{ij}^2}}.$$

Step 3: Compute the covariance matrix $\hat{A}^T \hat{A}$.
Step 4: Compute the largest eigenvalue $\lambda_{\max}$ and the corresponding eigenvector $V_{\max}$ of the covariance matrix $\hat{A}^T \hat{A}$.

$V_{\max}$ is the principal axis of the N vectors.

In the previous section, we have introduced the VQ method that generates a codebook $C = (c_1, \ldots, c_N)$ where $c_i, i = 1 \ldots, N$ are the codewords to represent the user profiles. In the anomaly detection stage, instead of using the full search algorithm, which computes the squared Euclidean distance between current session vector and all the codewords, we propose to use PCA as a tool to accelerate the process. The algorithm is as follows:

Step 1: Compute the principal axis V of all the codewords c_i in the codebook C.
Step 2: Compute the projections $c_i^T V$ of the N codewords on V.
Step 3: Sort the N codewords based on the magnitude of their projections on V, i.e. to establish the order such that

$$c_i \leq c_j \Leftrightarrow c_i^T V \leq c_j^T V.$$

This constitute a mapping $S : C \rightarrow \{1, \ldots, N\}$, with $S(c) = i$ means the projected value $c^T V$ ranks i in the sorted list of the total N projections.
Step 4: To test the anomaly of a new user session represented by vector y, compute the projection $p_y = y^T V$.
Step 5: Search for the codeword c_k in the sorted codebook whose projection is closest to p_y.
Step 6: Compute the squared Euclidean distance between y and the codewords whose indices are in the interval $[k - r/2, k + r/2 - 1]$, where r is a pre-determined search parameter. If $k - r/2 < 1$, then the interval is $[1, r]$. If $k + r/2 - 1 > N$, then the interval is $[N - r + 1, N]$.
Step 7: Let d be the minimal Euclidean distance between y and the codewords within the interval. Compare d with a pre-determined threshold τ. If $d \leq \tau$, the session is labelled as normal, otherwise it is labelled as intrusive.
Step 8: Repeat steps 4 to 7 for new user sessions.

In the algorithm, the choice of the search parameter r will affect the efficiency and effectiveness of the algorithm. If too large the value of r is chosen, too many codewords will be included in the computation in which the gain in efficiency may not be significant. However, if the value of r is too small, too few codewords will be included which will increase the false alarm rate. In order to make a balance between the efficiency gain and the false alarm rate, Step 7 of the algorithm is modified as follows:

Step 7: Let d be the minimal Euclidean distance between y and the codewords within the interval. Compare d with a pre-determined threshold τ. If $d \leq \tau$, the session is labelled as normal. Otherwise using the full search algorithm to compute the distances between y and all the codewords and identify the minimal d_m. If $d_m \leq \tau$, the session is labelled as normal, otherwise it is labelled as intrusive.

4 Experimental Results

In this experiment, the audit data from a Solaris host machine were analyzed. The Solaris operating system has a security extension, called the Basic Security Module (BSM). The BSM extension supports the monitoring of activities on a host by recording security-relevant events [15].

This section presents results of experiments that test the performance of the proposed detection algorithm for vector-based intrusion detection systems. In this study, for easy bench-marking, our experiments used system audit logs

from the well-known MIT DARPA data. The experiments used seven weeks of the 1998 audit data set from the MIT Lincoln Laboratory. Since there are about 243 different types of BSM audit events in the audit data, we considered 243 event types in our experiments. We used those data that are labelled as "normal" as the training data set to generate the codebook. During testing, we removed the label of all audit data and use the proposed technique to generate a label for each audit data. The labels generated in our test are then compared with the given labels to evaluate the performance of our proposed technique.

There are in total 35 days of data in the seven weeks of the audit data. We used the **telnet** sessions from all these data for training and testing. The anomaly detection algorithm for vector-based intrusion detection systems was implemented in Visual C++ and executed in a Pentium IV computer. We use the VQ-based intrusion detection algorithm [13] to generate a codebook, with size N chosen as 64. According to the intrusion hypothesis, user profiles are established based on the user activities over a period of time. In this respect, we use the first 4 weeks of the audit data which contains 2181 normal **telnet** sessions to generate the user profiles (codebook). The data which contains 1730 normal **telnet** sessions from the last 3 weeks of the audit data mixed with 40 intrusive **telnet** sessions from all the 7 weeks of audit data were used as the testing data set. To compare the performance of the proposed algorithm with the full search algorithm, the running of the test data set was repeated 5 times each with a different value of search parameter r equal 4, 8, 16, 32, and 64 respectively. When the search parameter equals to the codebook size of 64, the proposed algorithm becomes the full search algorithm. The threshold value τ is chosen at the 99% level threshold value as in [13]. The resulting false alarm rate, detection rate, and execution time of the running of the test data set are presented in Table 1.

Table 1. Performance of anomaly detection algorithm based on PCA

Search parameter (r)	False alarm rate	Detection rate	Execution time (ms)
4	4.34%	100%	207
8	1.79%	100%	332
16	1.10%	100%	617
32	0.81%	100%	1156
64	0.75%	100%	2188

According to results presented in Table 1, when the value of the search parameter is 4, the speed of execution is about 10 times faster than the full search algorithm. However, the false alarm rate has increased almost by a factor of 6. This is not acceptable in terms of the effectiveness of the intrusion detection. The test was repeated with the same set of test data and search parameters but with the modified Step 7 of the detection algorithm. The results are presented in Table 2.

Table 2. Performance of the modified anomaly detection algorithm based on PCA

Search parameter (r)	False alarm rate	Detection rate	Execution time (ms)
4	0.75%	100%	335
8	0.75%	100%	413
16	0.75%	100%	674
32	0.75%	100%	1191
64	0.75%	100%	2188

As we can see from the results in Table 2, when the value of the search parameter equals 4, the speed of the detection has been increased by a factor of 6 when compared with the full search algorithm while the false alarm rate has remained unchanged. This has indicated that the algorithm with modified Step 7 has a very promising performance in which the efficiency has raised significantly without sacrificing the detection effectiveness.

5 Conclusion

An efficient anomaly detection algorithm for vector-based intrusion detection systems was presented. In general, intrusion detection schemes are based on the hypothesis that normal system/user behaviors are consistent and can be characterized by some behavior profiles such that deviations from the profiles are considered abnormal. There were researches that proposed the use of simple statistics such as the Q-statistic and χ^2 statistic to represent user profiles. However, for usage patterns that exhibit complex structures, simple statistics are not sufficient. Recently, there are researches that propose the use of multivariate data analysis or vector-based analysis to extract and represent complicated user profiles. [13] has proposed a VQ-based technique for the extraction of complicated usage patterns and represented user profiles in the form of a codebook. In VQ, the codebook generation is in essence the optimal partition of a global set of multivariate data (in vector form) and represents each each partition with a single vector, i.e. a codeword. Hence, for a large volume of audit data, we can make use of the codebook generation technique to build representative user profiles in the form of codebook.

While most of these vector-based intrusion detection researches have focused on the efficient and effective ways of user profiling, another important aspect of intrusion, the anomaly detection stage was seldom mentioned. Most of the intrusion detection systems use the full search algorithm in this stage, which will be computationally intense. In terms of real time anomaly detection, it is not efficient enough to perform a full search especially the data set is vector-based and high dimensional in nature. In this paper, we propose an efficient algorithm that based on the PCA technique to perform anomaly detection using user profiles that were built using the VQ technique. The main contribution of this research is that we have demonstrated how the efficiency of the detection can be raised significantly using the PCA technique while the fidelity loss due

to the reduction in dimensionality of the original data set can be avoided. This way, the effectiveness of the anomaly detection can be maintained.

Acknowledgement. This research was funded by the National 973 Plan (Project Number: 2004CB719400) and the National 863 Plan (Project Numbers: 2003AA413030 and 2003AA414030), P. R. China.

References

1. D. E. Denning, "An Intrusion Detection Model", *IEEE Transactions on Software Engineering*, 1987, 13(2): 222-232.
2. R.A. Kemmerer, G. Vigna, "Intrusion Detection: A Brief History and Overiew", *Security & Privacy*, 2002: 27-30.
3. E. Biermann, E. Cloete, L. M. Venter, "A Comparison of Intrusion Detection System", *Computers & Security*, 2001, 20:676-683.
4. T. Verwoerd, R. Hunt, "Intrusion Detection Techniques and Approaches", *Computer Communications*, 2002, 25:1356-1365.
5. U. Lindqvist, P.A. Porras, "Detecting Computer and Network Misuse through the Production-based Expert System Toolset", *Proc. IEEE Symp. Security Privacy*, 1999: 146-161.
6. W. Lee, S.J. Stolfo, K.W. Mok, "A Data Mining Framework for Building Intrusion Detection Models", *Proc. IEEE Symp. Security Privacy*, 1999: 120-132.
7. N. Ye, X. Li, Q. Chen, M.M. Xu, "Probabilistic Techniques for Intrusion Detection Based on Computer Audit Data", *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 2001, 31(4): 266-274.
8. N. Ye, S.M. Emran, Q. Chen, S. Vilbert, "Multivariate Statistical Analysis of Audit Trails for Host-based Intrusion Detection", *IEEE Transactions on Computers*, 2002, 51(7): 810-820.
9. H.S. Javitz, A. Valdes, "The NIDES Statistical Component Description and Justification", *Technical Report A010*, Menlo Park, Calif: SRI Int'l, 1994.
10. K.Y. Lam, L. Hui, S.L. Chung, "Multivariate Data Analysis Software for Enhancing System Security", *Journal of Systems Software*, 1995, 31: 267-275.
11. H. Shah, J. Undercoffer, and A. Joshi, "Fuzzy Clustering for Intrusion Detection", *Proc. of the* 12^{th} *IEEE International Conference on Fuzzy Systems*, 2003(2): 1274-1278.
12. Y. Guan, A.A. Ghorbani, and N. Belacel, "Y-Means: A Clustering Method for Intrusion Detection", *Proc. of 2003 Canadian Conference on Electrical and Computer Engineering*, 2003(2): 1083-1086.
13. H. W. Sun, K. Y. Lam, S. L. Chung, M. Gu, J. G. Sun, "Anomaly Detection in Grid Computing Based on Vector Quantization", *Proc. of the* 3^{rd} *International Conference on Grid and Cooperative Computing*, 2004: 883-886.
14. Y. Linde, A. Buzo, R.M. Gray, "An Algorithm for Vector Quantizer Design", *IEEE Transactions on Communications*, 1980, 28(1): 84-95.
15. D. Endler, "Intrusion Detection using Solaris' Basic Security Module" at http://www.securityfocus.com/focus/ids/articles/idsbsm.html

Applying Mining Fuzzy Association Rules to Intrusion Detection Based on Sequences of System Calls

Guiling Zhang

Department of Electronic Information Engineering, Tianjin University
glzhang808@sohu.com

Abstract. Intrusion detection is an important technique for computer and information system. S. Forrest and coworkers present us that short sequences of system calls are good signature descriptions for anomalous intrusion detection [10]. This paper extends their works by applying mining fuzzy association rules to intrusion detection. After giving a primary classification of system calls based on threat level and its classification identifier numbers, we generate series short sequences of *sendmail* trace data and transform them into fuzzy expression. Then we extract the Most Dangerous Sequences Database (MDSD) from the fuzzy expression data, according to the specific threshold. For the MDSD database, we apply mining fuzzy association rules to detect each sequence is "normal" or "abnormal". The prototype experimental results demonstrate that the proposed method gives enough ability for intrusion detection.

1 Introduction

Intrusion detection techniques can be categorized into misuse detection and anomaly detection. Misuse detection uses patterns of well-known attacks to identify intrusion. The main shortcomings of such systems are that known intrusion patterns have to be hand-coded into the system and they are unable to detect any future (unknown) intrusions. But the advantage of anomaly detection is that it can detect new attacks and vulnerabilities in the system. The main difficulties of these systems are how to select the system features.

Works of analyzing sequences of system calls for intrusion detection are described in [10,11,12,13]. System calls are usually requests to the operating system to do a system-specific or privileged operation. Forrest's group uses the concept of window size (The length of the sequence) to compare exact matches between a normal profile and the new trace of a process. If a counter of anomalies exceeds a user threshold, an intrusion may present.

The concept of prediction of sequence calls can be implemented using neural networks [2]. The main advantages of such an approach are the lack of dependence on any statistical assumption, noise tolerance, and abstraction.

R. Agrawal and R. Srikant first report fast algorithms for mining association rules [4]. In [6], mining quantitative association rules are proposed. The algorithm finds the association rules by partitioning the attribute domain and combining adjacent partitions, then transforms the problem into binary one. This method can cause the

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 826 – 835, 2005.

sharp boundary problem. Correspondingly, C. M. Kuok, A. Fu and M. H. Wong combine fuzzy theory with mining association rules to build mining fuzzy association rules in paper [5]. This method can solve sharp boundaries problem in database. In paper [8], the author applies fuzzy data mining techniques to security system and builds a fuzzy data mining based network intrusion detection model. In [9], the authors use sets of fuzzy association rules that are mined from network audit data as models of "normal behavior". In these applications of detecting anomalous behaviors, they generate fuzzy association rules from new audit data and compute the similarity with sets mined from "normal" data. If the similarity values are below a specified threshold, then an alarm is generated. Other reports [1,15,16,17] also demonstrated that *machine learn techniques* play a very important role in intrusion detection system.

In this paper, we transform the sequences of *sendmail* trace data into fuzzy expression. Then we extract all sequences that their fuzzy labels are greater than the specific threshold to generate the Most Dangerous Sequences Database (MDSD). For the MDSD database, we apply mining fuzzy association rules to detect each sequence is "normal" or "abnormal". The rests of this paper is organized as follows: In section 2 we briefly report algorithms of mining fuzzy association rule; User behavior descriptions based on sequences of system calls and their fuzzy expressions are described in section 3; Section 4 shows some prototype experimental results; Section 5 presents conclusion and outlines future works.

2 Algorithms of Mining Fuzzy Association Rules

2.1 Fast Algorithms for Mining Association Rules

R. Agrawal and R. Srikant proposed two new algorithms, Apriori and AprioriTid, for discovering all significant association rules between items in a large database of transactions. The association rule is defined as [4]: Let $I=\{i_1, i_2,\ldots, i_m\}$ be a set of binary attributes called items and T be a set of transactions. Each transaction t in T is a set of items ($t \subseteq I$). Associated with each transaction t has a unique identifier, called t-ID. So an association rule is an implication of the form $X \Rightarrow Y$, c, s, where $X \subset I$, $Y \subset I$ and $X \cap Y=\varphi$. The rule $X \Rightarrow Y$, c, s holds in the transaction set T with *confidence* c and *support* s. The *confidence* c is percentage of transactions in T that contain X also contain Y. The *support* s in the transaction set T is percentage of transactions in T contain $X \cup Y$.

Given a set of transactions T, the problem of mining association rules is to generate all association rules that have *support* s and *confidence* c greater than the user-specified minimum value (*minsup* and *minconf*).

Apart from algorithms to find binary association rules in large databases, algorithms to find quantitative association rules will also need to be developed. Because a database may contain quantitative attributes, e.g. integer, categorical, numerical attributes. R. srikant and R. Agrawal define the problem of mining association rules over quantitative and categorical attributes in large relation tables and present techniques for discovering such rules [6]. They refer to this mining problem as the Quantitative Association Rules problem. But this algorithm can generate sharp boundaries problem.

2.2 Mining Fuzzy Association Rules

C. M. Kuok and coworkers combine fuzzy theory with mining association rule algorithm to discover fuzzy association rules [5]. The definition of fuzzy association rule is as follows: Let $T=\{t_1, t_2,\ldots, t_n\}$ be the database and t_i represents the ith transaction in T and $I=\{i_1, i_2,\ldots, i_m\}$ to represent all attributes appeared in T and i_j represents the jth attribute, where I is named *itemset*. For each attribute i_j will associate with several fuzzy sets. For example $F^{i_j}=\{f_{i_j}^1, f_{i_j}^2,\ldots, f_{i_j}^k\}$ represent set of fuzzy sets associated with i_j and $f_{i_j}^k$ represents the kth fuzzy set in F^{i_j}. The proposed fuzzy association rule by C. M. Kuok is as the following form:

If X is A then Y is B.

In the above rule, X and Y are *itemsets*. $X=\{x_1, x_2, \ldots, x_p\}$ and $Y=\{y_1, y_2,\ldots, y_q\}$ are subsets of I and $X \cap Y=\varphi$. $A=\{f_{x_1}, f_{x_2}, \ldots, f_{x_p}\}$ and $B=\{f_{y_1}, f_{y_2}, \ldots, f_{y_q}\}$ contain the fuzzy sets associated with the corresponding attributes in X and Y. We use *significance* and *certainty* factors to determine the satisfiability of rules [5].

Definition of a fuzzy rule is interesting: Given a fuzzy rule "If X is A then Y is B", if the rule has enough significance and higher certainty factor than the user specified threshold, then it is interesting.

Definition of *large k-itemsets*: If *itemsets* have higher significance than the user specified threshold, then the *itemsets* is large *k-itemsets.*

In order to generate fuzzy association rule, we should first to find out all large *k-itemsets*.

Significance factor: The *significance* factor is calculated as follows: first summing all votes of each record with respect to the specified *itemset*, then dividing it by the total number of records. A *significance* factor reflects not only number of records supporting the *itemset*, but also their degree of support. We use the following formula to calculate the significance factor of <X, A>, i.e. S(X,A) [5]:

$$Significance = \frac{\text{Sum of votes satisfying <X,A>}}{\text{Number of records in T}}$$

$$S(X,A) = \frac{\sum_{t_i \in T} \prod_{x_j \in X} \{\alpha_{\alpha_j}(t_i[x_j])\}}{total(T)} \quad \text{where}$$

$$\alpha_{\alpha_j}(t_i[x_j]) = \begin{cases} m_{\alpha_j \in A}(t_i[x_j]) & \text{if } m_{\alpha_j} \geq \omega \\ 0 & otherwise \end{cases}$$

In the above equation, <X,A> represents the *itemset-fuzzy* set pairs, where X is the set of attributes x_j and A is the set of fuzzy sets α_j. A record satisfies <X, A> means that the vote of the record is greater than zero. The vote of a record is calculated by the membership grade of each x_j in that record. The membership grade should not be less than the user specified threshold ω such that low membership values will not be considered. We use $t_i[x_j]$ to obtain the value of x_j in the ith records, then transform the value into membership grade by $m_{\alpha_{j \in A}}(t_i[x_j])$ which is the membership function of x_j. After obtaining all membership grades of each x_j in a record, we use

$\prod_{x_j \in X} \{m_{\alpha_{j \in A}}(t_i[x_j])\}$ to calculate the vote of t_i. After summing up the votes of all records, we divide the value by the total number of records.

Certainty factor. We use the discovered large *itemsets* to generate all possible rules. When we obtain a large *itemset* <Z, C>, we want to generate fuzzy association rules of the form, 'If X is A then Y is B', where $X \subset Z$, Y=Z-X, $A \subset C$ and B=C-A. The *certainty* factor can be calculated as follows [5]:

$$\text{Certainty} = \frac{Significance-of-<Z,C>}{Significance-of-<X,A>}$$

$$C_{<<X,A>,<Y,B>>} = \frac{\sum_{t_i \in T} \prod_{x_k \in Z} \{\alpha_{a_k}(t_i[z_k])\}}{\sum_{t_i \in T} \prod_{x_j \in X} \{\alpha_{a_j}(t_i[x_j])\}}$$

where

$$\alpha_{a_k}(t_i[z_k]) = \begin{cases} m_{a_k \in C}(t_i[z_k]) \; if \; m_{c_k} \geq \omega \\ 0 \qquad\qquad otherwise \end{cases}$$

The *certainty* reflects fraction of votes support <X, A> also support <Z, C>. If the union of antecedent and consequent has enough *significance* and the rule has sufficient *certainty*, this rule will be considered as interesting.

2.3 The Relation of Mining Fuzzy Association Rules and Intrusion Detection

W. Lee and others successfully apply two data mining algorithms (the association rules algorithm and the frequent episodes algorithm) to intrusion detection [1]. They demonstrate that these algorithms can construct concise and accurate classifiers to detect anomalies. But these algorithms do not use itself fuzzy features of intrusion detection.

In paper [9], the authors successfully use fuzzy data mining techniques to extract patterns that represent normal behavior for intrusion detection. They employ sets of fuzzy association rules to mine network audit data as models of "normal" behavior. Then they generate fuzzy association rules from new audit data and compute the similarity with sets mined from "normal" data. If the similarity values are below a threshold value, an alarm is produced. They also show that fuzzy logic is appropriate for the intrusion detection problem because security itself involves fuzziness.

This paper extends their works by applying fuzzy association rules to anomaly detection based sequences of system calls.

3 Data Sets of Sequences of System Calls

3.1 System Calls Primary Classification According to Its Threat Level

In paper [14] the authors classify the UNIX (LINUX) system calls into six categories according to their functionality. The six categories are file system group, process group, network group, module group, signal group and other group. Each call category is further classified into four groups according to their threat level, i.e. 1(Harmless), 2(Used for subverting the invoking process), 3(Used for a denial of service attack), 4(Allows full control of the system). In this paper, larger threat level numbers are assigned to more dangerous system calls.

In this classification, the most high threat level 4 has 42 calls, the next threat level 3 has 76 calls, the threat level 2 has 42 calls and the threat level 1 has 80 calls. We assign identifying number 401, 402,…, 442 to each calls in threat level 4, 301,302,…, 376 to each calls in threat level 3, 201, 202,…242 to each calls in threat level 2, 101, 102, 180 to each calls in level 1, respectively. These identifier numbers will be fuzzification as follows.

3.2 Using Membership Function to Transform System Call Identifiers into Fuzzy Representation

Each sequence of system calls has k position features. We should transform crisp value (the identifier number of every system call) into fuzzy subjection function value by selecting appropriate membership function. There are many different types of membership functions for fuzzification the crisp input, such as larger fuzzy distribution (S type), smaller fuzzy distribution (Z type) and medium fuzzy distribution (πtype) . In this paper, the larger fuzzy distribution function (S type) is adopted as follows:

$$s(x:a,b)=\begin{cases}0 & x\le a\\ 2\left(\frac{x-a}{b-a}\right)^2 & a<x\le\frac{a+b}{2}\\ 1-2\left(\frac{x-a}{b-a}\right)^2 & \frac{a+b}{2}<x\le b\\ 1 & x>b\end{cases}$$

For example, there is the following trace of system calls to define “normal” or “abnormal” behavior [10]:

open, read, mmap, mmap, open, getrlimit, mmap, close

Its corresponding identifiers are:

412, 215, 233, 233, 412, 147, 233, 301

To suppose that we select a=200 and b=400 then using the above S membership function to transform the trace to fuzzy identifier as follows:

1, 0.0125, 0.05445, 0.05445, 1, 0, 0.05445, 0.48995

After traces of system calls from programs are gained, we slide the window (The window length is k) across the traces and record each call that follows it at position 1, position 2, and so forth, up to position k. Then we transform them into fuzzy representation. For example: Suppose we choose k=3 and the following database is produced for above sequences of system calls:

Table 1. The column μ is the fuzzy expression for sequences of system calls

Position 1			Position 2			Position 3			Label
Calls	ID	p1	Calls	ID	p2	Calls	ID	p3	μ
open	412	1	read	215	0.0125	mmap	233	0.0545	0.1344
read	215	0.0125	mmap	233	0.0545	mmap	233	0.0545	0.3684
mmap	233	0.0545	mmap	233	0.0545	open	412	1	0.7458
mmap	233	0.0545	open	412	1	getrlimit	147	0	0.1380
open	412	1	getrlimit	147	0	mmap	233	0.0545	0.7411
getrlimit	147	0	mmap	233	0.0545	close	301	0.4899	0.2727

The value of label for each record is calculated as follows:

If a record is a "normal" sequence then let $l=|(p1 + p2 + p3)/3 - 0.5|$; otherwise, if a record is an "abnormal" sequence then let $l=|(p1 + p2 + p3)/3 + 1|$. In this paper, we use the follow membership function to fuzz the label column:

$$Label = \mu(l) = 1 - e^{-l} \qquad l > 0$$

After the above fuzzy database has been established, we will use fuzzy sequences similar degree π to compare a new sequence with all sequences in the database. If the similar degree π is greater than specified threshold (e.g. 0.7, 0.8 or 0.9) then these two sequences are regarded as similarity sequences. So the new sequence is labeled with *product* of π and the corresponding old sequence label. The similar degree (π) of two fuzzy sequences can be calculated by Hamming distance or other methods.

If we select π=0.8 then we extract all sequences that their labels$\geq 0.5\pi$ (=0.4) to construct a new database, we call it the Most Dangerous Sequences Database (MDSD). After the MDSD is established, we apply mining fuzzy association rules to the database. If the fuzzy association rules from the MDSD database have enough *significance* and *certainty* then the corresponding sequence is labeled as "abnormal" sequence, otherwise, it is labeled as "normal" sequences.

3.3 Preparing Training Datasets

S. Forrest provided us with set of traces of the *sendmail* program used in her experiments [10]. The *sendmail* traces detailed in [10], each file of the traces data has two columns of integers: the first is the process ids and the second is the system call "numbers". These numbers are indexed into a lookup table of system call names. For example, the number "5" represents system call open. But we instead these "numbers" by "threat level identifying numbers" we defined in section 3.1, e.g. using "412" instead of "5" for system call "open". But in this paper, the lookup table is fuzzy expressions of system calls like table 1. The training set of traces includes both "normal" and "abnormal" sequences. We use the same training data set as described in the paper [18].

The "normal" traces is produced by the *sendmail* daemon; The "abnormal" traces include: 3 traces of the *sscp* attacks, 2 traces of the *syslog-remote* attacks, 2 traces of the *syslog-local* attacks, 2 traces of the *decode* attacks, 1 trace of the *5x* attack and 1 trace of the *sm565a* attack. These are the traces of (various kinds of) abnormal runs of the *sendmail* program.

The training data is composed of 80% normal sequences and 20% abnormal sequences.

As the same method in paper [1,15], in order to prepare the training data sets, we use a sliding window to scan the normal traces and to create a list of unique sequences of system calls (The size of the sliding window may be 2l+1, e.g. 7, 9, 11, 13, etc). This list is called the "normal" list. Next, we scan each of the intrusion traces. For each sequence of these system calls, we first look it up in the normal list. If an exact match can be found then the sequences is labeled as "normal". Otherwise it is labeled as "abnormal".

After the training datasets have been well prepared, we should first transform them into fuzzy representations as in table 1. Then we apply the fuzzy association rules algorithm to generate detecting rules (see section 3).

4 Experimental Results

Our experimental datasets are downloaded from http://cs.unm.edu. But we randomly combine different percent "normal" and "abnormal" data into experiment A, B, C and D as W. Lee's method in paper [1,15].

Table 2. Comparing Detection of anomalies

	%abn	%abn in Paper [1]				%abn in This Paper			
Traces	Paper[10]	A	B	C	D	A	B	C	D
sscp-1	5.2	41.9	32.2	40.0	33.1	38.9	29.8	37.0	32.1
Sscp-2	5.2	40.4	30.4	37.6	33.3	38.1	28.4	34.6	31.8
sscp-3	5.2	40.4	30.4	37.6	33.3	38.4	27.9	35.1	31.6
syslog-r-1	5.1	30.8	21.2	30.3	21.9	27.5	18.6	27.8	20.1
syslog-r-2	1.7	27.1	15.6	26.8	16.5	24.4	13.1	24.3	14.9
syslog-l-1	4.0	16.7	11.1	17.0	13.0	13.2	8.1	14.2	11.7
syslog-l-2	5.3	19.9	15.9	19.8	15.9	16.7	12.9	15.8	12.9
decode-1	0.3	4.7	2.1	3.1	2.1	3.1	1.3	2.4	1.9
decode-2	0.3	4.4	2.0	2.5	2.2	2.9	1.4	1.9	1.7
sm565a	0.6	11.7	8.0	1.1	1.0	8.7	6.1	0.9	0.8
sm5x	2.7	17.7	6.5	5.0	3.0	14.7	4.8	3.8	2.3
sendmail	0	1.0	0.1	0.2	0.3	0.7	0.2	0.5	0.6

Experiment A: 46% normal and 54% abnormal, sequence length is 11.
Experiment B: 46% normal and 54% abnormal, sequence length is 7.
Experiment C: 46% abnormal and 54% normal, sequence length is 11.
Experiment D: 46% abnormal and 54% normal, sequence length is 7.

When the test dataset is well prepared, we use a sliding window of length $2l+1$ (l=3, 5) to scan the dataset and producing sequences of system calls. The sliding (shift) step is selected l. Then, we use fuzzification algorithm described in section 3 to transform these sequences into fuzzy expression and apply mining fuzzy association rules algorithm to determine the sequences is "normal" or "abnormal".

Because we slide window with step l, the determined "abnormal" is an "abnormal region", not only an "abnormal sequence" [1,15]. As described in the paper [1,15], we look for "abnormal regions" that contain more abnormal sequences than the normal ones. And calculate the percentage of "abnormal regions". If the percentage of abnormal regions is above user specified threshold then the trace is an intrusion.

We compare our experimental results with the results in paper [1] and in paper [10] (see table 2).

From these comparisons we can obtain that the proposed algorithm in this paper is suitable for intrusion detection. These experimental results show that the algorithm proposed in this paper has as similar stronger as presented method in paper [1].

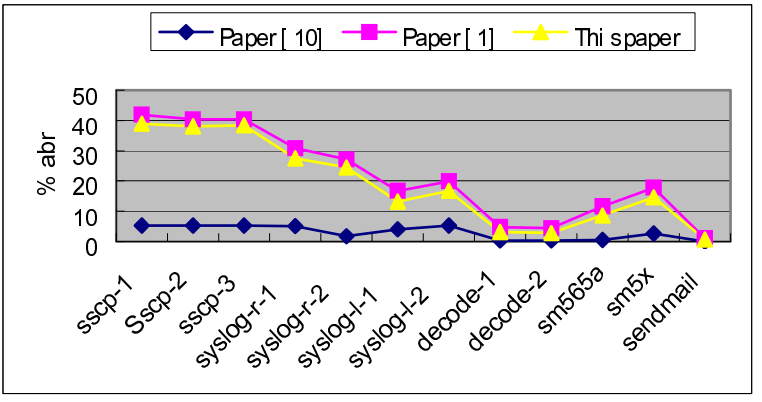


Fig. 1. The comparison curves of abn % for experiment A

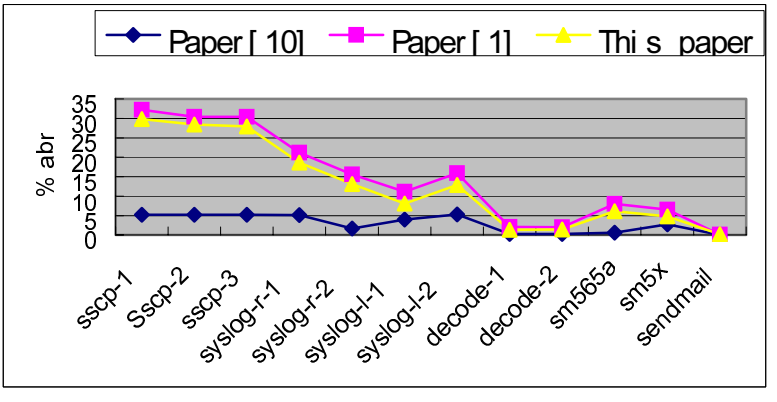


Fig. 2. The comparison curves of abn % for experiment B

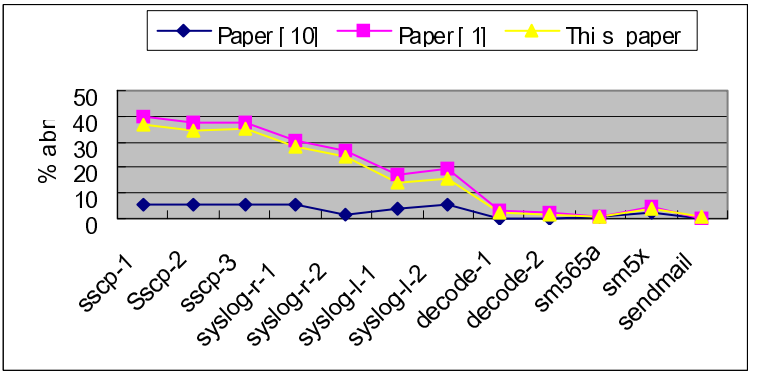


Fig. 3. The comparison curves of abn % for experiment C

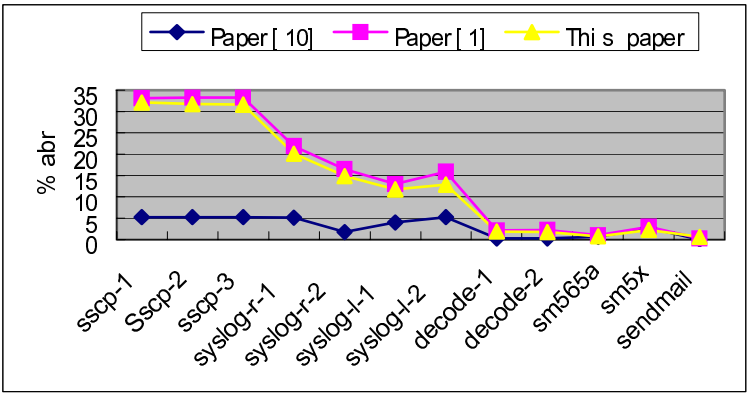


Fig. 4. The comparison curves of abn % for experiment D

The main disadvantage of the algorithm in the paper is that it is more complexity than two other algorithms in paper [1, 10].

5 Conclusions and Future Work

Intrusion detection is an important technique for computer and information system. Many AI techniques have been widely used in intrusion detection system. System calls provide rich sources of description information about the behavior of a program. Many different algorithms have been used to analyze these types of data. In this paper, we proposed a new fuzzy method to apply fuzzy association rule to intrusion detection based on sequences of system calls. By using fuzzy theory, sharp boundaries problem in intrusion detection have been well solved. The prototype experiments on *sendmail* system call data demonstrated the effectiveness of the proposed algorithm in detecting anomalies. The accuracy of the detection models largely depends on sufficient training data.

But we still have many problems to be studied in the future.

- To investigate more accuracy threat level for every system call. To study more system calls statistic features for many intrusion behaviors.
- To compare several fuzzification membership function for our system and improve the performance of the system.
- To study the methods and benefits of combing multiple AI intrusion detection methods.

References

[1] W. Lee and S. Stolfo: Data Mining Approaches for Intrusion Detection. Proc. The Seventh USENIX Security Symposium, January 1998.

[2] Z. Liu, G. Florez , and S.M. Bridges: A Comparison Of Input Representations In Neural Networks: A Case Study in Intrusion Detection. International Joint Conference on Neural Networks (IJCNN), Honolulu, Hawaii,2002

[3] Z. Liu, S.M. Bridges, R.B. Vaughn: Classification of Anomalous Traces of Privileged and Parallel Programs by Neural Networks. Proceeding of the 12th IEEE International Conference on Fuzzy Systems, 2003.
[4] R. Agrawal and R. Srikant: Fast Algorithms for Mining Association Rules. In 20th International Conference on Very Large Databases, Santiago, Chile, Sept. 1994
[5] Kuok C., Fu A., Wong M.: Mining Fuzzy Association Rules in Databases. SIGMOD Record 17(1):41-46
[6] R. Srinkant, R. Agrawal. Mining Quangtitative Association Rules in Large Relation Tables. SIGMOD, 1996
[7] Dickerson J E., Juslin J, Loulousoula O, Dickerson J A.: Fuzzy Intrusion Detection. IFSA World Congress and 20th North American Fuzzy information Processing Society (NAFIPS) International Conference, 2001
[8] Hai J., Jianhua S., Hao C., Zongfen H.: A Fuzzy Data Mining Based Intrusion Detection Model. Proceedings of the 10th IEEE international workshop on future trends of distributed Computing System (FTDCS'04) 2004, IEEE.
[9] G. Florez, S. M. Bridge and R. B. Vaughn: An Improved Algorithm for Fuzzy Data Mining for Intrusion Detection. 2002, IEEE.
[10] S. Forrest, S.A. Hofmeyr, A. Somayaji, and T.A. Longstaff: A Sense of Self for UNIX Processes. Proceedings of the 1996 IEEE Symposium on Security and Privacy, Los Alamitos, CA, 1996. IEEE Computer Society Press.
[11] JBD. Cabrera, L. Lewis, and RK. Mehra: Detection and Classification of Intrusions Using System Calls. SIGMOD RECORD 30 (4): 25-34 DEC 2001.
[12] S.A. Hofmeyr, S. Forrest, and A. Somayaji: Intrusion Detection Using Sequences of System Calls. Journal of Computer Security, 1998.
[13] C. Warrender, S. Forrest, and B. Pearlmutter: Detecting Intrusions Using System Calls: alternative data models. IEEE Computer Society, 1999.
[14] X. Ming, C. Chun, Y. Jing: Anomaly Detection Based on Sytem Call Classification. Journal of Software, China, 2004.
[15] Wenke Lee, Sal Stolfo, and Phil Chan: Learning Patterns from UNIX Process Execution Traces from Intrusion Detection. AAAI Workshop: AI Approaches to Fraud Detection and Risk Management, July 1997.
[16] T. Verwoerd, R. Hunt: Intrusion Detection Techniques and Approaches. Computer Communications, 25 (15), SEP 15 2002.
[17] C.C. Michael: Finding the Vocabulary of Program Behavior Data for Anomaly Detection. Proceeding of DARPA Information Survivability Conference and Exposition, VOL 1, 2003.

A Novel and Secure Non-designated Proxy Signature Scheme for Mobile Agents

Jianhong Zhang[1], Jiancheng Zou[1], and Yumin Wang[2]

[1] College of sciences, North China University of Technology, Shijingshan district, 100041, Beijing, China
{jhzhang,zjc}@ncut.edu.cn
[2] State Key Lab. on ISN, Xidian University, 710071, Xi'an, Shaanxi, China
ymwang@xidian.edu.cn

Abstract. Mobile agents are able to migrate across different execution environments through network. However, proxy signature is a signature that an original signer delegates his signing capability to a proxy signer. If a mobile agent may act as a proxy signer to produce signatures on behalf of the agent owner autonomously on remote hosts, this will increase efficiency of the agent owner. In this work we propose a novel and secure non-designated proxy signature scheme with revocation. This work is different from other related proxy signature scheme in that in addition to providing non-repudiation, the method provides the proxy signer's privacy which be realized by Alias Issuing authority. if need be, the proxy signer's privacy can be revoked. and a prominent character is that Alias Issuing authority is untrusty, even though he colluded with the original signer, they also cannot forge a proxy signature. These features are attractive for proxy signature in agent-based paradigm in which proxy signers are mobile agents that are executed in remote untrustworthy host.

1 Introduction

Mobile agents[1,2] are autonomous software entities which can migrate across different execution environments through network,and one of the most prominent technologies believed to be playing an important role on future electronic commerce systems. The characteristics of mobile agents: mobility and autonomy, make them ideal for mobile computing. Mobile software agents have emerged as a promising paradigm for a number of applications ranging from ubiquitous computing to mobile/electronic commerce (m-/e-commerce). One of the tasks a mobile agent is expected to perform is to sign a digital signature on behalf of its owner. In other words, a mobile agent may act as a proxy signer to produce signatures on behalf of the original signer (i.e. the owner of the agent) autonomously on remote hosts. Such signatures are sometimes referred to as proxy signatures in cryptography. As the remote hosts are not trustworthy, or may be malicious, the challenges faced by signature delegation in mobile agent systems are great. Among these challenges are, firstly, the signing key carried by an agent (also

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 836–844, 2005.

called proxy key) may be abused by the agent. For example, the agent may use the key to sign messages on which the original signer has not authorized the agent to sign. Alternatively, the proxy key may be spied on by other agents or remote hosts on which the agent is executed. Consequently, the agents or hosts may forge this agent's proxy signature. Furthermore, the agent or the owner of the agent may falsely deny having signed a specific signature, i.e. repudiation of signing or repudiation of signature origin. Recently,to overcome the problems above, B.Lee,H.Kim and K.kim[5] proposed a secure mobile agent using strong non-designated proxy signature. Unfortunately, the scheme has repudiation yet and cannot realize proxy signer' privacy.

To satisfy non-repudiation of proxy signature and assure the proxy signer's privacy, in this paper we propose a novel and secure non-designated proxy signature scheme with revocation. This work is different from other related proxy signature schemes in that in addition to providing strong non-repudiation, the method provides the proxy signer's privacy . if need be, the proxy signer's privacy can be revoked. These features are attractive for proxy signature in agent-based paradigm in which proxy signers are mobile agents that are executed in remote untrustworthy host.

The rest of this paper is organized as follows. In section 2, we give basic definition and security requirements . In section 3, we propose a novel and secure non-repudiation proxy signature with revocation; security our proposed scheme is analyzed in section 4. Finally, we draw a conclusion of the paper in section 5.

2 Non-designated Proxy Signature with Revocation

2.1 Definition and Security Requirements

In 1996, Mambo *et al*[3] proposed the notion of proxy signature. A proxy signature scheme allows a signer to delegate the signing capability to a designated person,call a proxy signer. After the concept of proxy signature was first introduced by Mambo *et al.*, many researchers have done a lot of work in this field, several kinds of proxy signature schemes[11,13] have been put forth.

In real application, we sometimes hope that the original signer doesn't specify his proxy signer in proxy key issuing phase. Anyone who owns the warrant and some secret parameters issued by the original signer can construct his proxy signing key. if need be, the identity of proxy signer can be revoked. We define the Non-designated Proxy signature with Revocation (NPSR) as follows.

Definition 1. *(Non-designated Proxy Signature with Revocation) Let O be an original signer who has authentic key pair (sk_O, pk_O), T be an Alias Issuing Authority who has authentic key (sk_T, pk_T) and P be a proxy signer who has authentic key pair (sk_P, pk_P). Let m_w be $A's$ warrant information for the delegation which does not specify a proxy signer. Let $\delta_O = Sign(sk_O, m_w)$ be $A's$ signature on warrant m_w using his private key sk_O. then NPSR is constructed as the following six algorithms $(AI, DA, PKG, PS, PV, PR)$*

– AI is an Alias issuing algorithm that takes the proxy signer identity ID_P and Alias issuing authority 's private key sk_T, and outputs an Alias (h_P, k_P, δ_T).

$$(h_P, k_P, \delta_T) \longleftarrow AI(ID_P, sk_T)$$

– DA is a delegation Algorithm that takes the original signer's private key sk_O and warrant m_w, and outputs the signature δ_O of warrant m_w.

$$\delta_O \longleftarrow DA(sk_O, m_w)$$

– PKG is proxy key issuing algorithm that takes original signer's signature δ_O on the warrant m_w and Alias Issuing authority's signature δ_T on the proxy signer's Alias, and outputs a proxy signing key sk_P.

$$sk_P \longleftarrow PKG(\delta_T, \delta_M)$$

– PS is a proxy signing algorithm that takes proxy private key sk_P and the message m and outputs a proxy signature δ_P. It is executed by the proxy signer.

$$\delta_P \longleftarrow PS(sk_P, m)$$

– PV is a proxy verification algorithm that takes (δ_P, m, m_w) and outputs either accept or reject. It is executed by any verifier.

$$PV(\delta_P, m, m_w) \stackrel{?}{=} accept\ or\ reject$$

– PR is a proxy signature revocation algorithm. if need be , Alias Issuing authority can revoke the identity of the proxy signer by (δ_P, m, m_w). It is executed by the Alias authority

$$ID_P \longleftarrow PR(\delta_P, h_P)$$

An NPSR scheme should satisfy the following basic security requirements:

Verifiability: from a proxy signature a verifier can be convinced of the original signer's agreement on the signed message.

Strong unforgeability: A proxy signer can create a valid proxy signature for the original signer. But the original signer and Alias Issuing authority cannot create a valid proxy signature with the name of proxy signer' alias h_P.

Non-designateability: his warrant issued by the original signer does not specify who the proxy signer is. It is also transferable among proxy signers.

Strong non-repudiation: Once a proxy signer creates a valid proxy signature on behalf of an original signer. the proxy signer cannot repudiate his signature creation against anyone.

Proxy privacy: No one (except Alias Issuing authority)can determine the identity of the proxy signer only from the proxy signature.

Revocation: It should be confident that proxy key pair cannot be used for other purpose. In the case of misuse, Alias Issuing authority can revoke the actual identity of the proxy signer explicitly by proxy signature, however, others cannot do it.

Resistant-collusion: even though Alias Issuing authority colludes with the original signer, they are also able to produce a proxy signature with name of the proxy signer h_P .

2.2 Knowledge Signature

As a basic building blocks, we use signatures converted from honest-verifier zero-knowledge proofs of knowledge , which are called as signature of knowledge. We abbreviate them as SPK. The SPK are denoted as

$$SPK(\alpha, \beta, \cdots) : R(\alpha, \beta, \cdots)(m)$$

which means the signature for message m by a signer with secret knowledge $\alpha, \beta, \cdots$ satisfying the relation $R(\alpha, \beta, \cdots)$. Refer to [7,8] for the detail content.

3 The Proposed Proxy Signature Scheme

In the proposed scheme, there are four entities:the Alias Issuing Authority T, a Original signer O , a Proxy signer P and a Verifier V ,where T is responsible for issuing an alias for every proxy signer,O and P denote an original signer and a proxy signer respectively, V denotes a verifier. the parameters of the system Setup are as follows:

p, q: large prime numbers, where $q|(p-1)$

g : an element of order q in Z_p

$h(.)$: an one-way hash function

m : the signed message

m_w: the warrant issued by the original signer O

$(x_M, y_M = g^{x_M})$: the private/public key pair of the original signer O .

$(x_T, y_T = g^{x_T})$: the private/public key pair of the Alias Issuing Authority .

$\delta = sign(m, x)$: a signature algorithm based on the Discrete-Logarithm by using a secret key x to sign a message m and δ is the digital signature of the message m .

$verify(m, \delta, y)$: the corresponding signature verification algorithm by using a public key y to verify the validation of the signature δ of message m.

Issuing Alias phase: T issues an alias h_p, a public parameter r_T and a secret key s_T to P and records the triple (h_P, r_α, ID_P) into the his database, where ID_P is the identity of P, after receiving the s_T, P will check the validation of the secret key s_T by the following steps. if it holds, P will use it to produce the proxy signature in the proxy signature generation phase. if need be, the Alias Issuing authority can reveal the ID_P to revoke the proxy signer's privacy according to the h_P. the detail procedure is as follows:

- step 1: the proxy signer P first randomly chooses a number $\alpha \in_R Z_q$ to compute $r_\alpha = g^\alpha modp$ and sends his identity ID_P and r_α to Issuing Alias Authority T.
- step 2: T computes $h_P = h(r_\alpha, ID_P)$, and randomly chooses a number $k_T \in_R Z_P^*$ to compute $r_T = g^{k_T} modp$, $r = r_T \cdot r_\alpha modp$ and $s'_T = x_T h(h_P, r) + k_T modq$.
- step 3: T records the data (h_P, k_P, r, ID_P) in his local database and sends (s'_T, h_P, r_T) to the proxy signer P.

– step 4: After obtaining (s'_T, h_P, r_T), the proxy signer P computes $r = r_T \cdot r_\alpha mod p$ and $s_T = s'_T + \alpha \ mod \ q$, then verifies whether the following equation holds.

$$g^{s_T} = y_T^{h(h_P,r)} \cdot r \ mod \ p$$

Delegation phase: in this phase, the original signer O delegates his signing capability to P, he produces a secret s_M and a public parameter r_M by the following steps and sends it to the proxy signer P along with the warrant m_w. After obtaining the s_M , P will check the validation of it. If it holds, P will combine it with s_T to form the proxy signing key. the detail delegation procedure is as follows:

– step 1: First, the original signer randomly chooses a number $k_M \in_R Z_q^*$ and computes $r_M = g^{k_M} modp$ and $s_M = x_M h(m_w, r_M) + k_M modq$
– step 2: send (m_w, r_M, s_M) to the proxy signer P.
– step 3: the proxy signer verifies whether the following equation is hold or not.
$g^{s_M} \stackrel{?}{=} y_M^{h(m_w,r_M)} r_M modp$
– step 4: compute the proxy signing key $x = s_M \cdot h(r, r_M) + s_T \ mod \ q$

Signing and verifying phase: To produce a proxy signature, the proxy signer P first generates the proxy signing key x as $x = s_M \cdot h(r, r_M) + s_T modq$, then he produces the proxy signature $\delta = sign(x, m)$ on the message m by a secure signature algorithm $sign(\cdot)$ such as Schnorr signature. The resultant proxy signature is $(\delta, r_M, m_w, h_p, r)$. if a verifier V wants to verify the correction of the proxy signature, he executes the following steps:

Step 1: first he computes the public key of the proxy signer

$$y = y_M^{h(m_w,r_M)\cdot h(r,r_M)} \cdot r_M^{h(r,r_M)} \cdot y_T^{h(h_p,r)} \cdot r modp$$

Step 2: the proxy signature is verified by by the verifying algorithm $verify(\cdot)$.

$$verify(m, \delta, y) \stackrel{?}{=} 1$$

Revoking privacy phase: If need be, the verifier V sends signature $(\delta, r_M, m_w, h_p, r)$ to Alias Issuing authority T. T will reveal ID_P to revoke the proxy signer $P's$ privacy.

– Step 1:V sends $(\delta, r_M, m_w, h_p, r)$ to T.
– Step 2: check whether the proxy signature $(\delta, r_M, m_w, h_p, r)$ is valid.
– Step 3: After T sent back r_α, ID_P, the verifier V checks $h_P \stackrel{?}{=} h(r_\alpha, ID_P)$.

4 Security Analysis of Our Proposed Scheme

In this section, we will analysis our proposed scheme's security, and show the security of this scheme is base on Schnorr signature. we know that Schnorr signature scheme is secure in the random oracle Model[6]. Thus, if an adversary A can forge our proxy signature, then there is an adversary A' who can forge a Schnorr signature.

Theorem 1. *Alias Issuing Authority in the proposed Non-designated proxy signature scheme with revocation reaches Girault's trusted level 3.*

Proof. According to the signature scheme above, we know if Alias Issuing Authority colludes the original signer to impersonate a proxy signer,whose identity information is ID_P, in the name of h_P. They can perform as follows:

- step 1: T randomly chooses a number $k_T \in_R Z_P^*$, then computes $r_T = g^{k_T} \ mod\ p.$ and $s_T = x_T h(h_P, r_T) + k_T \ mod\ q$
- step 2: T records the data (h_P, ID_P) in his local database and sends (s_T, h_P, r_T) to the original signer O.
- step 3: the original signer randomly chooses a number $k_M \in_R Z_q^*$ and computes $r_M = g^{k_M} modp$ and $s_M = x_M h(m_w, r_M) + k_M modq$.
- step 4: they combine s_M with s_T to produce the proxy private key $x = s_M h(r_M, r_T) + s_P \ mod\ q$.

it is obvious that the proxy public key $y = g^x \ modp$ has the following form

$$g^x = y_M^{h(m_w, r_M) \cdot h(r_T, r_M)} \cdot r_M^{h(r_T, r_M)} \cdot y_T^{h(h_p, r_T)} \cdot r_T \ mod\ p$$

Thus they can forge the proxy signature of the proxy signer ID_P with using the private key x.(Note that: the identity of proxy signer P is hide into the proxy signature; in fact, x consists of the original signer's signature and Alias Issuing authority's signature.)

However, this proxy signer can provide a proof to convince the three Party that the signature is forged by Alias Issuing authority and the original signer. Supposed that the proxy private key of the proxy signer is x_p, then he provide the knowledge signature $SPK(g, y')$ of $y'(= y_M^{h(m_w, r_M) \cdot h(r_T, r_M)} \cdot r_M^{h(r_T, r_M)} \cdot y_T^{h(h_p, r_T)} \cdot r_T \ mod\ p)$ to the base g. it means that the proxy signer with identity ID_P possesses original signer's signature and Alias Issuing authority's signature. Because only original signer and Alias Issuing authority can produce their signature. Thus, it means that the proxy signer with identity ID_P has already been delegated by the original signer.

Therefore, our scheme reaches Girault's trusted lever 3 [10], i.e. the Alias Issuing authority does not know the proxy private key of the proxy signer with the original signer, and it can be proven that they generate false witness if they does so.

Theorem 2. *If an existential forgery of our proposed signature scheme, under an adaptively chosen message attack, has non-negligible probability of success, then the Schnorr signature scheme can be forged in polynomial time.*

Proof. According to our proposed scheme above, a signature verification consists of two steps: in first step, the proxy public key was produced; in second step, the proxy signature is verified by the produced proxy public key. Because we adopt a secure signing algorithm, the proxy signature is secure, provided the proxy public key is correct and corresponds to the proxy private key. Thus, an

adversary can attack the scheme only through forging the proxy private key x' which satifies

$$g^{x'} = y = y_M^{h(m_w,r_M)\cdot h(r,r_M)} \cdot r_M^{h(r,r_M)} \cdot y_T^{h(h_p,r)} \cdot rmodp$$

In other words, the corresponding public key of x' has the above form. From the above equation, we know that the original signer's public key y_M and Alias Issuing authority's public key y_T are included in y. In fact, x' is a Schnorr signature variation.

Thus, in the scheme, the original signer O and the Alias Issuing authority are two powerful attacker. In the following ,we consider two attack cases, the first case is that there exist an algorithm make the original signer O to produce the proxy private key x' without $T's$ help, and the second case is that T produce a proxy private key x' without $O's$ agreement.

1. **Forgery by O:** Supposed that there is a NPSR breaker which takes (m, m_w, k) as input and outputs a valid proxy private key x' and it satisfies $g^{x'} = y_M^{h(m_w,r_M)\cdot h(r,r_M)} \cdot r_M^{h(r,r_M)} \cdot y_T^{h(h_p,r)} \cdot r \ modp$. Because of the group property of discrete logarithm problem. we can obtain

$$g^{x'} = g^{x_M \cdot h(m_w,r_M)\cdot h(r,r_M)+kh(r,r_M)} \cdot y_T^{h(h_p,r)} \cdot r \ mod \ p$$

where $r_M = g^k \ mod \ p$

Then, $x' - x_M \cdot h(m_w, r_M) \cdot h(r, r_M) + kh(r, r_M)$ is the Schnorr signature on message h_P of Alias Issuing T, it means that the original signer O can forge T's Schnorr signature without knowing his private key x_T .

2. **Forgery by T:** Assumed that there is a NPSR breaker which takes (m, h_P, m_w, k') as input and outputs a valid proxy private key x' and it satisfies $g^{x'} = y_M^{h(m_w,r_M)\cdot h(r,r_M)} \cdot r_M^{h(r,r_M)} \cdot y_T^{h(h_p,r)} \cdot r \ mod \ p$. Because of the group property of discrete logarithm problem. we can obtain

$$g^{x'} = g^{x_T h(h_P,r)+k'} g_M^{h(m_w,r_M)h(r,r_M)} r_M^{h(r,r_M)} modp$$

where $r = g^{k'} \ mod \ p$

Then, $\frac{x'-(x_T h(h_P,r)+k')}{h(r,r_M)}$ is a Schnorr signature on warrant m_w of the original signer O, it means that the Alias Issuing authority T can forge O's Schnorr signature without knowing his private key x_M .

Because Schnorr signature is secure in random oracle model [6],it means that our proposed scheme is as secure as the Schnorr signature scheme.

From the above theorem 1 and theorem 2, we know that our proposed scheme satisfies Verifiability, resistant-collusion, Strong unforgeability,proxy privacy and revocation.

5 Conclusion

Mobile agent is very good application instance of proxy signature. The based-agent proxy signature is very suitable to application of mobile devices and limited

computing devices in mobile environment.In this work, we propose a novel and secure non-designated proxy signature scheme with revocation. This work is different from other related proxy signature scheme in that in addition to providing non-repudiation, the method provides the proxy signer's privacy. if need be, the proxy signer's privacy can be revoked. and a prominent character is that Alias Issuing authority is untrusted, even though he colluded with the original signer, they also cannot forge a proxy signature in the name of h_P. These features are attractive for proxy signature in agent-based paradigm in which proxy signers are mobile agents that are executed in remote untrustworthy host.

Acknowledgements

The authors would like thank the anonymous referees for their value comments. This work was supported in part by National 973 program of China (2002CB312104), NSF of Beijing (4002011), Beijing New Star Program, Beijing Edu. Com. Program, and was also a project supported by doctor research starting.

References

1. W.Farmer, J.Gutmann and V.Swarup,"Secutiy for Mobile Agents: Authentication and State Appraisal", Prof.of the European Symposium On Research in Computer Security Lecture Notes in Computer Science, Vol.1146. Springer-Verlag, Berlin Heidelberg New York (1996) 118-130.
2. P.Kotzanikolaous,G.Katsirelos and V.Chrissikopoulos,"Mobile Agents for secure electronic Transactions" Recent Advances in Signal Processing and Communications,World Scientific and Engineering Society Press, (1999) 363-368.
3. Mambo, M., Usuda, K., and Okamoto, E. Proxy Signature: Delegation of the Power to Sign Messages. IEICE Transactions Fundamentals, E79-A(9): (1996) 1338-1353.
4. Eric Jui-Lin Lu, Min-Shiang Hwang, Cheng-Jian Huang, "A new proxy signature scheme with revocation", Applied Mathematics and Computation 161 ,Elsevier,(2005) 799-806.
5. Byoungcheon Lee, Heesun Kim, and Kwangjo Kim,"Secure Mobile Agent using Strong Non-designated Proxy signature",ACISP2001 Lecture Notes in Computer Science, Vol.1446. Springer-Verlag, Berlin Heidelberg New York (2001) 145-157.
6. Pointcheval.D,and Stern.J, Security Proofs for Signature Schemes[C],In Eurocrypt'96 , LNCS 1070, Springer-verlag, (1996) 387-398.
7. G. Ateniese and J. Camenisch and M. Joye and G. Tsudik, A practical and provably secure Coalition-Resistant group signature scheme, Advances in Cryptology-Proceedings of CRYPTO 2000, LNCS 1880, (2000) 255-270.
8. J. Camenisch, Group signature schemes and payment systems based on the discrete logarithm problem, PhD thesis, vol. 2 of ETH-Series in Information Security an Cryptography, Hartung-Gorre Verlag, Konstanz, 1998, ISBN 3-89649-286-1.
9. J.H.Zhang,Q.H. Wu, J.L. Wang,Y.M. Wang, An improved nominative proxy signature scheme for mobile communication,18th International conference on Advanced Information Networking and Applications, (2004),23-26.

10. M. Girault, Self-certified public keys, Advances in Cryptology-Eurocrypt 1991, LNCS 547, Springer-Verlag, (1991) 490-497.
11. T.-S. Chen, T.-P. Liu and Y.-F. Chung, A proxy-protected proxy signature scheme based on elliptic curve cryptosystem, TENCON 02. Proceedings. 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering, (2002) 184-187.
12. M. Jakobsson, K. Sako, and R. Impagliazzo, " Designated verifier proofs and their applications," Advances in Cryptology-Eurocrypt 1996, LNCS 1070, Springer- Verlag, (1996)143-154.
13. B. Lee, H. Kim and K. Kim. Strong proxy signature and its applications. In Proceedings of SCIS, (2001) 243-247.
14. FUJISAKI, E. and OKAMOTO, T., A Practical and Provably Secure Scheme for Publicly Verifiable Secret Sharing and Its Applications, Advances in Cryptology-EUROCRYPT '98, Lecture Notes in Computer Science, Vol. 1403, Springer-Verlag, (1998) pp. 32-46.
15. J.H Zhang, J.C Zou and Y.M Wang,Two Modified Nominative Proxy Signature Schemes for Mobile Communication, 2005 IEEE Networking, Sensing and Control Conference Proceedings, Tucson, Arizona, U.S.A, (2005) 1054-1058

Identity Based Conference Key Distribution Scheme from Parings*

Shiqun Li[1,2], Kefei Chen[1], Xiangxue Li[1], and Rongxing Lu[1]

[1] Department of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai 200030, China
[2] National Laboratory for Modern Communications
{sqli,chen-kf,rxlu,xxli}@cs.sjtu.edu.cn

Abstract. A conference key distribution system can provide a session key of a conference with more than three participants via network. In this paper,we propose an ID-based conference key distribution scheme from pairings. The proposed scheme provides user anonymity. At the same time, we also show that the proposed scheme is secure against replaying attacks and conspiratorial impersonation attacks.

1 Introduction

A Conference Key Distribution System (CKDS) is used for sharing a secret key among the attending participants of the conference via open network. Thereafter the participants use the secret key to establish secure communication. However, anyone nonattending the conference should reveal no useful information about the secret key and further the communication message.

In recent years, many CDKSs have been proposed [1,5,28,16,13,15,27]. Some of them [1,5,28,16] are key-distribution protocols, whereas others are key-agreement protocols. In the key-distribution protocol, the conference organizer initializes a conference key and distributes it to all the other participants. While in the key-agreement protocol, it does not need a conference organizer, all the participants compute a common conference key.

In 1984, Shamir[22] asked for identity-based cryptosystem to simplify key management.The idea of ID-based cryptosystem is to use identity information as the user's public key. Under Shamir's concept, many ID-based schemes have been proposed[25,26,7,10]. And several ID-based public key distribution systems were also developed[19,6,8,12,29]. However, most of them are not fully satisfactory.

Until 2001, Boneh and Franlin proposed a fully functional ID-based encryption scheme using bilinear pairings[2]. From then on, many ID-based key agreement protocol[24,30] and ID-based signatures schemes[4,10,20] were proposed from parings.

Based on bilinear pairings, we propose an ID-based CKDS in this paper. The proposed scheme is very efficient and satisfies the following properties:

* This work is partially supported under NFSC 60273049, 60303026. The first author's work is also supported under Foundation of NLMC 51436050304JW0317.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 845–852, 2005.

- Anonymity of the participants. Not only the nonattending participants but also the attending ones can't know who are involved in the conference except for the conference organizer.
- Security against impersonation attacks. Any adversary can not impersonate the conference organizer by intercepting the broadcast message.
- Security against conspiracy attacks. Conspiratorial participants in the conference can not derive the secret information of the conference organizer.

The rest of the paper is organized as follows: Section 2 describes bilinear parings and Bilinear Diffie-Hellman Assumption. Our ID-based CKDS is presented in Section 3. The security and performance of the scheme are discussed in Section 4 and Section 5, respectively. Section 6 gives conclusions.

2 Preliminaries

Before we begin our scheme, let's briefly review the Bilinear parings and Bilinear Diffie-Hellman Assumption.

2.1 Bilinear Parings

Let G be a cyclic additive group of prime order q, and G_1 be a cyclic multiplicative group of the same order q. Suppose the discrete logarithm problem is hard. A bilinear pairing $e : G \times G \longrightarrow G_1$ is a map that satisfies the following conditions:

- Computable: There is an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in G$.
- Bilinear: $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in G$ and all $a, b \in Z_q^*$.
- Non-degenerate: There exists $P, Q \in G$ such that $e(P, Q) \neq 1$.

For details about bilinear parings, please refer to[2,3,11].

2.2 Bilinear Diffie-Hellman (BDH) Assumption

We now review the BDH problem[14,2]. Let $e : G \times G \longrightarrow G_1$ be a bilinear map and let P be a generator of G, the BDH problem is described as follows: Given $< P, aP, bP, cP >$ for $a, b, c \in Z_q^*$, compute $W = e(P, P)^{abc}$. An algorithm A is said to solve the BDH problem with an advantage of ϵ if

$$Pr[A(P, aP, bP, cP) = e(P, P)^{abc}] > \epsilon \tag{1}$$

BDH Assumption: *We* assume that the BDH problem is hard, which means there is no polynomial time algorithm to solve BDH problem with non-negligible probability.

3 The Proposed Scheme

In this section, we will introduce our ID-based conference key distribution scheme. Basically, the scheme is made of four phases which are the following: *Setup phase, Key distribution phase and Key recovery phase.*

- Setup: The PKG picks a security parameter k and generates the systems public parameters and the master-key. Then the PKG generates a private key corresponding to the user's identity. The private key is given to the user in a secure channel. This procedure is done only once for every identity and uses the same Setup data for different identities.
- Key distribution: The conference organizer generates a conference key and computes a message according to it. Then the organizer broadcasts this message to all the users in the group.
- Key recovery: All users compute the conference key when they receive the message containing the conference key information. They can authenticate the message.But only the proper participants can correctly get the conference key and thereafter take part in the following conference.

3.1 Setup Phase

Suppose there are n users, the set of all the users is denoted by $A = \{U_0, U_1, ..., U_{n-1}\}$.At the Setup phase, the PKG computes as the following:

- Step 1: Given a security parameter k, the PKG randomly chooses groups G and G_1 of prime order q, a generator P of G, a bilinear map $e : G \times G \to G_1$ and hash functions $H : \{0,1\}^* \to Z_q^*$, $H_1 : \{0,1\}^* \to G$, $H_2 : G_1 \times \{0,1\}^* \to Z_q^*$, $H_3 : G_1 \times \{0,1\}^* \to Z_q^*$.
- Step 2: It chooses a master-key $s \in Z_q^*$ randomly and computes $P_{pub} = sP$.
- Step 3: For each participant $U_i (i = 0, 1, ..., n-1)$, the PKG computes $Q_{ID_i} = H_1(ID_i)$ and the corresponding private key $d_{ID_i} = sQ_{ID_i}$.
- Step 4: The PKG publishes system's public parameters $\{G, G_1, k, e, P, P_{pub}, H, H_1, H_2, H_3\}$ and delivers the private Key d_{ID_i} to $U_i \in A$ over a secret channel.

3.2 Key Distribution Phase

Without loss of generality, let the conference organizer be U_0, and user of the set $B = \{U_1, ...U_m\}$ be the conference participants. U_0 calculates the message CT as the following:

- Step 1: Computes $X_i = e(d_{ID_0}, Q_{ID_i})$ for $i = (1, ..., m)$.
- Step 2: Get a timestamp T from the system.
- Step 3: Chooses $SK \in Z_p^*$ randomly.
- Step 4: Calculates $H_2(X_i, T)$ and $h_i = H_3(X_i, T)$ for $i = (1, ..., m)$.

– Step 5: Formulates the following equations:

$$\begin{cases} H_2(X_1,T) & = C_m h_1^m + ... + C_2 h_1^2 + C_1 h_1^1 + SK \pmod p \\ H_2(X_2,T) & = C_m h_2^m + ... + C_2 h_2^2 + C_1 h_2^1 + SK \pmod p \\ . & \\ . & \\ . & \\ H_2(X_m,T) & = C_m h_m^m + ... + C_2 h_m^2 + C_1 h_m^1 + SK \pmod p \end{cases} \tag{2}$$

and solves $C_m, ..., C_2, C_1$ from Eq.2.
– Step 6: U_0 computes $C_0 = H(SK \parallel ID_0 \parallel T)$, where $\parallel$ means concatenation.
– Step 7: U_0 broadcasts $CT = (ID_0, C_m, ..., C_2, C_1, C_0, T)$ to all users.

3.3 Key Recovery Phase

On receiving $CT = (ID_0, C_m, ...C_2, C_1, C_0, T)$, user $U_i (i = 1, ..., n-1)$ performs the following steps to extract the session key SK:

– Step 1: Checks the validity of the T. If it is out of date, U_i terminates the Key Extraction phase.
– Step 2: Computes $X_i' = e(d_{ID_i}, Q_{ID_0})$.
– Step 3: Calculates $H_2(X_i', T)$ and $h_i = H_3(X_i', T)$. And then U_i calculates $SK' = H_2(X_i', T) - (C_m h_i^m + ... + C_2 h_i^2 + C_1 h_i) \pmod p$.
– Step 4: Checks the validity of the CT. U_i computes $C_0' = H(SK \parallel ID_0 \parallel T)$. and test whether $C_0' = C_0$, halt if this is not the case.

4 Security Analysis

Basicly, a secure conference key distribution scheme should satisfy security requirements: correctness, unforgery, confidentiality, and can resist known attacks, such as impersonation attacks, replaying attacks and conspiratorial impersonation attacks.

4.1 Correctness

Theorem 1: *All the attending participants of the conference get the common conference key SK if they follow the protocol.*

Proof: On receiving the message CT, user $U_i \in B$ can get

$$\begin{aligned} X_i' &= e(d_{ID_i}, Q_{ID_0}) \\ &= e(sQ_{ID_i}, Q_{ID_0}) \\ &= e(Q_{ID_i}, Q_{ID_0})^s \\ &= e(d_{ID_0}, Q_{ID_i}) \\ &= X_i \end{aligned}$$

$$H_2(X_i', T) = H_2(X_i, T)$$

and

$$\begin{aligned} h_i' &= H_3(X_i', T) \\ &= H_3(X_i, T) \\ &= h_i \end{aligned}$$

and further U_i can get

$$\begin{aligned} SK' &= H_2(X_i', T) - (C_m h_i^m + ... + C_2 h_i^{m-1} + C_1 h_i^m) \pmod p \\ &= SK \end{aligned}$$

Theorem 2: *Anyone non-attending the conference can't reveal SK.*

Proof: Because Eq.2 was constructed by $H_2(X_i, T)$ and $h_i = H_3(X_i, T)$, those non-attending the conference can not obtain those parameters. Hence it is impossible for any adversary to reveal SK.

Theorem 1 and Theorem 2 show that the proposed scheme works correctly.

4.2 Security Against Impersonators, Conspiratorial Impersonation Attack

Theorem 3: *Any adversary can not generate conference key message or replay the message to impersonate the organizer.*

Proof: Firstly, by the expiration of T, the adversary can not take replaying attack. If the adversary can forge a valid T, he has to get all the $H_2(X_i, T)$. On the OWH[17,18,23] assumption, the adversary can forge all the $H_2(X_i, T)$ only if he knows d_{ID_0}, which is protected by the BDH assumption.

Lemma 1: *The identities of the attending participants are anonymous to each other, and even anonymous to the unattended one except for the conference organizer.*

Proof: This is very obvious from Theorem 3.

Theorem 4: *Conspiratorial participants cannot successfully replay the intercepted message for originating a conference by the name of the organizer.*

Proof: This can be proved by the same method as in theorem 3.

5 Performance Analysis

As for the proposed scheme, computations in the setup phase can be precomputed. So we mainly consider the computation complexity of the key distribution phase and key recovery phase. To show the computation complexity clearly, the following notations are used:

- T_M: the computation time for modular multiplication over $GF(q)$.
- T_H: the computation time for hash function.
- T_I: the computation time for inverse modulo q.
- T_D: the computation time for division.
- T_P: the computation time for pairing.

At the key distribution pase, the computation complexity of the conference organizer is as following:

$$\begin{aligned} T_{KDP} = m \times T_P(for\ computing\ X_i) + 2m \times T_H(for\ H_2\ and\ H_3\ hashing) \\ +m \times (m-1) \times T_M(for\ formulating\ Eq.2) \\ +(m+2) \times (m^2 \times T_M) + T_I(for\ solving\ C_i) + T_H(for\ H\ hashing) \end{aligned}$$

At the key recovery pase, the computation complexity of the user U_i is as following:

$$\begin{aligned} T_{KRP} = T_P(for\ computing\ X_i') + 2 \times T_H(for\ H_2\ and\ H_3\ hashing) \\ +(m-1) \times T_M(for\ reconstructing\ SK) \\ +T_H(for\ verifying\ C_0) \end{aligned}$$

Note that in this scheme $e(d_{ID_j}, Q_{ID_i})$ can be pre-computed once and for all so that key distribution phase and key recovery phase do not require any pairing computations. Alternatively, $e(d_{ID_j}, Q_{ID_i})$ can be included in the system parameters. So further efficiency can be achieved, not withstanding more storage complexity of the participants.

6 Conclusion

In this paper, we have proposed an identity-based efficient conference key distribution scheme from pairings. The scheme provided user anonymity. At the same time, we also show that the proposed scheme is secure against replaying attacks and conspiratorial impersonation attacks. In the proposed scheme, the security also depends on U_0, the organizer. So further work should be to improve the scheme to provide security without depending on the security of the organizer.

References

1. S. Berkovits. How to Broadcast a Secret, Proc. Advances in Cryptology-Eurocrypt'91,pp.535-541, 1991.
2. D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing, Advances in Cryptology-Crypto 2001, LNCS 2139, pp.213-229, Springer-Verlag, 2001.
3. D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing, In C. Boyd, editor, Advances in Cryptology-Asiacrypt 2001, LNCS 2248, pp.514-532, Springer-Verlag, 2001.
4. J.C. Cha and J.H. Cheon. An identity-based signature from gap Diffie-Hellman groups, Cryptology ePrint Archive, Report 2002/018, available at http://eprint.iacr.org/2002/018/.

5. C.C. Chang and T.H. Lin. How to Converse Securely in a conference, Proc. IEEE 30th Ann. Int'l Camahan Conf., pp.42-45, 1996.
6. T. Chikazawa and A. Yamagishi. Improved identity- based key sharing system for multiaddress communcation, Electr. Lett., vol. 28, no.11, pp.1015-1017, 1992.
7. C. Cocks. An identity based encryption scheme based on quadratic residues, In Cryptography and Coding, LNCS 2260, pp.360-363, 2001.
8. L. Harn and S. Yang. ID-based cryptographic schemes for user identification, digital signature, and key distribution, IEEE J. on sele. areas in comm., vol.11, No.5 pp.757-760, 1993.
9. F.Hess. Efficient identity based signature schemes based on pairings. Proceedings of 9th workshop on selected areas in cryptography-SAC2002. Lecture Notes in Computer Science.
10. F. Hess. Exponent group signature schemes and efficient identity based signature schemes based on pairings, Cryptology ePrint Archive, Report 2002/012, available at http://eprint.iacr.org/2002/012/.
11. J. Horwitz and B. Lynn. Toward hierarchical identity-based encryption, Proc. Eurocrypt 2002, LNCS 2332, pp.466-481, Springer-Verlag, 2002.
12. T. Hwang and J. L. Chen. Identity-based conference key broadcast systems, IEE Proc. Comput. Digit. Tech., vol.141, No.1 pp.57-60, 1994.
13. I. Ingemarsson, D.T. Tang, and C.K. Wong. A Conference Key Distribution System, IEEE Trans. Information Theory, vol.28, no. 5, pp. 714-720, 1982.
14. A. Joux. A one round protocol for tripartite Diffie-Hellman, Proc. Fourth Algorithmic Number Theory Symposium, Lecture Notes in Computer Science, Vol. 1838, Springer-Verlag, pp. 385-394, 2000.
15. B. Klein, M. Otten, and T. Beth. Conference Key Distribution Protocols in Distributed Systems, Proc. Codes and Ciphers-Cryptography and Coding IV, pp. 225-242, 1995.
16. J.Mao, B.Yang. Anonymous and dynamic conference-key distribution system, The 14m IEEE 2003 lntemational Symposium on PersonalJndoor and Mobile Radio Communication Proceedings. pp. 2784- 2788, 2003.
17. R. Merkle. One way hash functions and DES, Advances in Cryptology-CRYPTO '89, Lecture Notes in Computer Science Vol. 435, pp.428-446, 1989.
18. M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications, In 21st Annual ACM Symposium on Theory of Computing, 1989.
19. E. Okamoto and K. Tanaka. Key distribution system based on identification information, IEEE Journal on Selected areas in comm., vol.7, No.4, 481-485, 1989.
20. K.G. Paterson. ID-based signatures from pairings on elliptic curves, Cryptology ePrint Archive, Report 2002/004, available at http://eprint.iacr.org/2002/004/.
21. A. Shamir. How to Share a Secret, Communications of the ACM, vol. 22, no. 11, 612-613, 1979
22. A. Shamir. Identity-based cryptosystems and signature schemes, Advances in Cryptology-Crypto 84, LNCS 196, pp.47-53, Springer-Verlag, 1984.
23. V. Shoup. A composition theorem for universal one-way hash functions,In Advances in Cryptology-Eurocrypt'2000, 2000.
24. N.P. Smart. An identity based authenticated key agreement protocol based on the Weil pairing, Electron. Lett., Vol.38, No.13, pp.630-632, 2002.
25. H. Tanaka. A realization scheme for the identity-based cryptosystem, proc. Crypto 87, LNCS 293 pp.341-349, Springer-Verlag, 1987.
26. S. Tsuji and T.Itoh. An ID-based cryptosystem based on the discrete logarithm problem, IEEE Journal of Selected Areas in Communications, Vol.7, No.4, pp.467-473, 1989.

27. W.G. Tzeng. A Secure Fault-Tolerant Conference-Key Agreement Protocol, IEEE Trans. on Computers, Vol.51 No.4, pp.373-379, 2002.
28. TC. Wu. Conference key distribution system with user anonymity based on algebraic approach, IEE Proceedings Computers and Digital Techniques, Vol. 14, No. 2, pp. 145-148, 1997.
29. S. Xu and H. Tilborg. A new identity-based conference key distribution scheme, in Proc. IEEE Int. Symp. on Information Theory, pp. 269, 2000.
30. F.Zhang and K.Kim. Efficient ID-based blind signature and proxy signature from bilinear pairings. ACISP 2003, LNCS 2727, pages 312-323, 2003.

Some Remarks on Universal Re-encryption and A Novel Practical Anonymous Tunnel

Tianbo Lu, Binxing Fang, Yuzhong Sun, and Li Guo

Software Division, Institute of Computing Technology,
Chinese Academy of Sciences 100080, Beijing, P.R. China
Graduate School of Chinese Academy of Sciences, 100039, Beijing, P.R. China
lutianbo@software.ict.ac.cn, {bxfang,yuzhongsun,guoli}@ict.ac.cn

Abstract. In 2004 Golle, Jakobsson, Juels and Syverson presented a new encryption scheme called the universal re-encryption [GJJS04] for mixnets [Cha81] which was extended by Gomulkiewicz et al. [GKK04]. We discover that this scheme and its extension both are insecure against a chosen ciphertext attack proposed by Pfitzmann in 1994 [Pfi94]. Another drawback of them is low efficiency for anonymous communications due to their long ciphertexts, i.e., four times the size of plaintext. Accordingly, we devise a novel universal and efficient anonymous tunnel, rWonGoo, for circuit-based low-latency communications in large scale peer-to-peer environments to dramatically decrease possibility to suffer from the attack [Pfi94]. The basic idea behind rWonGoo is to provide anonymity with re-encryption and random forwarding, obtaining practicality, correctness and efficiency in encryption in the way differing from the layered encryption systems [Cha81] that can be difficult to achieve correctness of tunnels.

1 Introduction

In 1981, Chaum proposed the concept of a mixnet in his seminal paper [Cha81]. A mix is a store-and-forward device that attempts to hide the correspondence between its incoming and outgoing messages . A mixnet is a collection of mixes that relay messages between each other from their original senders to their final receivers. Many modern anonymous systems are based on Chaum's layered encryption idea. That's to say, a message is encrypted into a layered data structure in a manner starting from the last stop of the path. As the data passes through each node along the way, one layer of encryption is removed according to the recipe contained in the data. We can roughly classify them into two categories, i.e., message-based high-latency email [DDM03, MCPS03] and circuit-based low-latency communications [DMS04, RP02, RR98]. A difference between them is that within the former class, every packet has to enwrap the IP addresses of nodes on the path; while in the latter class a user firstly establishes a circuit to the receiver, then all packets pass along the circuit, without enwrapping IP addresses in packets.

However, Chaum's original mixnet design included a system of signed receipts to assure senders that their messages have been properly processed by

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 853–862, 2005.

the network, which is hard in layered encryption [JJ01]. A whole body of research [PIK93, KS95] has concentrated on creating such verifiable systems that could offer a proof of their correct functioning alongside the mixing. Such systems are commonly called re-encryption mixnet and have been closely associated with voting. But all the conventional re-encryption techniques are impractical for anonymous communications because of the complex key distribution and management. Fortunately in 2004, Golle et al. introduced a new practical cryptographic scheme called universal re-encryption (URE) that leads to new types of functionality in mixnet architectures since it has no requirements of key generation, key distribution, and private key management [GJJS04]. Fairbrother has improved the scheme for sending large files [Fai04] and Gomulkiewicz et al. have extended the scheme (called as a EURE). However, Fairbrother's improvement is still very costly overall and unpractical for we can't enwrap too many ciphertext blocks in a fixed-size packet. And we find that the two schemes, URE and EURE, are insecure against a chosen ciphertext attack (called Pfitzmann attack for short) proposed firstly by Pfitzmann [Pfi94]; further the two schemes increase the ciphertext to four times the plaintext size, while a conventional re-encryption scheme is double.

We make two contributions in this paper. Firstly, we discover that the URE scheme and its extension EURE are vulnerable to Pfitzmann attack [Pfi94]. Secondly, we devise a novel practical and efficient anonymous tunnel - rWonGoo with ElGamal re-encryption and random forwarding whose ciphertext is only double the size of the corresponding plaintext, instead of four times as the URE or EURE. And our tunnel is universal due to re-encrypt without knowing the public key used for the original encryption. We also discuss the security of rWonGoo.

The rest of the paper is structured as follows. Section 2 presents an analysis of the URE scheme and its extension EURE. Section 3 details rWonGoo's tunneling approach. Section 4 analyzes the security of rWonGoo, and we finally conclude on Section 5.

2 Analysis of Universal Cryptosystems

2.1 ElGamal Encryption

ElGamal is a probabilistic public-key cryptsystem, defined by the following parameters: a group G of prime order p, a generator g of G, a private key x and the corresponding public key $y = g^x$. Plaintexts are in G and ciphertexts in $G \times G$. The encryption of a message $m \in G$ is $(a, b) = (g^k \mod p, my^k \mod p)$, where $k \in_u G$ is chosen anew per encryption ("$\in_u S$" denotes selection uniformly at random from the set S). Decryption of the ciphertext (a, b) returns $m = b/a^x \mod p$. Such a pair (a, b) can be re-encrypted by choosing $k' \in_u G$ and forming $(a', b') = (ag^{k'} \mod p, by^{k'} \mod p)$. Such a re-encrypted pair is homomorphic to the input ciphertext, i.e., the two ciphertexts correspond to the same plaintext. Therefore decryption of the ciphertext (a', b') also returns $m = b'/a'^x \mod p$.

2.2 Golle et al's Systems

Golle et al. described two practical mixnets, called *universal mixnet* and *hybrid universal mixnet* based on their URE scheme, respectively [GJJS04]. Fairbrother improved the latter using the Pohlig-Hellman secret key algorithm [Fai04], which has turned out to be insecure [GKK04]. We will prove that universal mixnet is also insecure in the following.

We first outline (with slightly modifications) the universal mixnet. Suppose that Alice (V_0) wants to send message m to Bob (V_λ) along the path $V_0V_1\cdots V_i\cdots V_{\lambda-1}V_\lambda$, the universal mixnet works as follows:

- Bob chooses a private key x at random, and publishes the corresponding public key $y = g^x$;
- To encrypt message m for Bob, Alice generates numbers $0 < k_0, k_0^{'} < p$ uniformly at random, then the ciphertext of m is computed as a quadruple:

$$C_0 = (\alpha_0, \beta_0; \alpha_0^{'}, \beta_0^{'}) = (my^{k_0}, g^{k_0}; y^{k_0^{'}}, g^{k_0^{'}}) = (my^{r_0}, g^{r_0}; y^{r_0^{'}}, g^{r_0^{'}}) \quad (1)$$

  where $r_0 = k_0$, $r_0^{'} = k_0^{'}$. Alice sends the ciphertext C_0 to node V_1;
- Node V_1 generates values $0 < k_1, k_1^{'} < p$, and re-encrypts message C_0 received from node V_0 by the following formula:

$$\begin{aligned} C_1 &= (\alpha_1, \beta_1; \alpha_1^{'}, \beta_1^{'}) = (\alpha_0(\alpha_0^{'})^{k_1}, \beta_0(\beta_0^{'})^{k_1}; (\alpha_0^{'})^{k_1^{'}}, (\beta_0^{'})^{k_1^{'}}) \\ &= (my^{r_0+r_0^{'}k_1}, g^{r_0+r_0^{'}k_1}; y^{r_0^{'}k_1^{'}}, g^{r_0^{'}k_1^{'}}) = (my^{r_1}, g^{r_1}; y^{r_1^{'}}, g^{r_1^{'}}) \end{aligned} \quad (2)$$

  where $r_1 = r_0 + r_0^{'}k_1$, $r_1^{'} = r_0^{'}k_1^{'}$. Then node V_1 sends C_1 to V_2. Each node $V_i (1 < i < \lambda)$ repeats this process, i.e., re-encrypts message C_{i-1} received from node V_{i-1} and sends it to V_{i+1}.
- When Bob receives the message

$$C_{\lambda-1} = (\alpha_{\lambda-1}, \beta_{\lambda-1}; \alpha_{\lambda-1}^{'}, \beta_{\lambda-1}^{'}) = (my^{r_{\lambda-1}}, g^{r_{\lambda-1}}; y^{r_{\lambda-1}^{'}}, g^{r_{\lambda-1}^{'}}) \quad (3)$$

  from node $V_{\lambda-1}$, he computes

$$m_{\lambda-1} = \alpha_{\lambda-1}/(\beta_{\lambda-1})^x \quad \text{and} \quad m_{\lambda-1}^{'} = \alpha_{\lambda-1}^{'}/(\beta_{\lambda-1}^{'})^x \quad (4)$$

  using his private key x and accepts message $m_{\lambda-1} = m$, if and only if $m_{\lambda-1}^{'} = 1$. Note that each node V_i $(0 < i < \lambda)$ will permute the ciphertexts from different predecessors (A node may locate on different paths). If a node receipts only a ciphertext, it will produce dummy ciphertexts and permute them all together.

Problems. The security of the system depends on the secret of the private key x, i.e., it guarantees only *external anonymity* described in [GJJS04]. Anyone who has the ability to eavesdrop the whole network can trace Alice by computing the plaintext m given x from one node in the path step by step. But the worst drawback is that the universal mixnet is subject to Pfitzmann attack [Pfi94].

In this case, the attacker doesn't need to trace step by step, but only need to compromise the second node V_1. We will illustrate this as follows.

Theorem 1. The universal mixnet is vulnerable to Pfitzmann attack.

Proof. Usually, the sender Alice knows the receiver Bob, but Bob doesn't know Alice in anonymous communications. Because we assume that the attacker knows the secret key x, for convenience, we assume the receiver Bob was compromised. If the the node V_1 is also compromised, then the attacker can relate Alice and Bob by construction a related ciphertext. Suppose that the receiver V_λ and node V_1 are compromised but the attacker doesn't know they are on the same path, others the anonymity is broken. The sender V_0 and node V_i for some $(i \in 1, \cdots, \lambda - 1)$ are honest (and at least one node, others there is nothing to hide). The outside eavesdropper can't relate the sender and the receiver because the honest node has shuffled the traffic. However, when node V_1 received message C_0 (1) from the sender. It knows that C_0 contains the secret message m of the sender. In order to relate the sender Alice and the receiver Bob, V_1 generates value t and prepares the second, related ciphertext $C_1^{'}$ based on C_1 (2) as follows:

$$C_1^{'} = (\alpha_1{}^t, \beta_1{}^t; \alpha_1^{'\,t}, \beta_1^{'\,t}) = (m^t y^{tr_1}, g^{tr_1}; y^{tr_1^{'}}, g^{tr_1^{'}}) = (m^t y^{R_1}, g^{R_1}; y^{R_1^{'}}, g^{R_1^{'}})$$

where $R_1 = tr_1$, $R_1^{'} = tr_1^{'}$. Then node V_1 sends $C_1^{'}$ to node V_2 together with message C_1. Node V_2 can't relate $C_1^{'}$ and C_1 because we assume that discrete logarithms are hard in group G. Then V_2 re-encrypts C_1 and $C_1^{'}$ into C_2 and $C_2^{'}$ respectively, and sends them to the next node V_3:

$$C_2 = (my^{r_1 + r_1^{'} k_2}, g^{r_1 + r_1^{'} k_2}; y^{r_1^{'} k_2^{'}}, g^{r_1^{'} k_2^{'}}) = (my^{r_2}, g^{r_2}; y^{r_2^{'}}, g^{r_2^{'}})$$

$$C_2^{'} = (m^t y^{R_1 + R_1^{'} K_2}, g^{R_1 + R_1^{'} K_2}; y^{R_1^{'} K_2^{'}}, g^{R_1^{'} K_2^{'}}) = (m^t y^{R_2}, g^{R_2}; y^{R_2^{'}}, g^{R_2^{'}})$$

where $R_2 = R_1 + R_1^{'} K_2$, $R_2^{'} = R_1^{'} K_2^{'}$. Each node V_i $(3 \leq i \leq \lambda - 1)$ also repeats this process. When the receiver Bob receives the message $C_{\lambda-1}$ and $C_{\lambda-1}^{'}$, he can get m according to formula (4) and m^t as the following:

$$M_{\lambda-1} = m^t y^{R_{\lambda-1}} / (g^{R_{\lambda-1}})^x \quad \text{and} \quad M_{\lambda-1}^{'} = y^{R_{\lambda-1}^{'}} / (g^{R_{\lambda-1}^{'}})^x.$$

Bob accepts message $M_{\lambda-1} = m^t$, if and only if $M_{\lambda-1}^{'} = 1$. Then he can easily relate m and m^t based on parameter t, as a result the attacker knows that Bob communicates with Alice. □

The attack can be extended to any malicious node on a path. That's to say, a malicious node can determine whether it's on a path to the receiver by preparing a related ciphertext. Furthermore, the hybrid universal mixnet and the EURE scheme [GKK04] are also vulnerable to this attack. Due to space limitations, we don't discuss this here. As Pfitzmann described [Pfi94], we do not see any successful countermeasures against this attack. However, we usually trust the recipient is honest in practice, hence the probability of this attack is trivial.

3 rWonGoo

3.1 Motivation

rWonGoo is an anonymous tunnel for circuit-based low-latency communications that achieves its strong anonymity and high efficiency with ElGamal re-encryption and random forwarding [RR98]. Here a tunnel is a circuit composed of a set of hops. Generally speaking, the longer the tunnel, the stronger the anonymity, but the cost such as processing time, latency and bandwidth is also higher. Hence, how to achieve a tradeoff between anonymity and efficiency is a challenge. Our random forwarding approach reduces the cost when we lengthen the anonymity tunnel for higher anonymity. rWonGoo is practical since it has no complex key distribution and management as an extension and application of the scheme proposed for voting in [PIK93]. It is also universal due to re-encryption without knowing the key used for the original encryption. Furthermore, our tunnel mainly used in large decentralized peer-to-peer networks is more efficient than URE and EURE schemes because its ciphertext is only double the size of the corresponding plaintext, instead of four times as the latter two schemes. Compared with layered encryption mixnets, rWonGoo can effectively defend against replay attack and achieve correctness.

The ElGamal cryptosystem has a very interesting property. Let $E_k(m)$/ $D_k(m)$ denotes encryption/decryption of message m with key K. Suppose that the private key is SK, and the corresponding public key is PK. In RSA algorithm, we have the following properties: $D_{PK}(E_{SK}(m)) = D_{SK}(E_{PK}(m)) = m$. This means that we can swap the roles of public and private keys. However, this doesn't hold for ElGamal encryption. We only can encrypt m with public key PK, and decrypt it with private key SK, i.e., $D_{SK}(E_{PK}(m)) = m$. We can't phrase it backwards. Hence, when building the tunnel via re-encryption scheme, we must consider the way from Alice to Bob and the reverse respectively, not like the way in layered encryption systems.

Because in a rWonGoo tunnel, nodes re-encrypt the messages from Alice to Bob and the reverse using different keys, we call the tunnel transmitting messages from Alice to Bob as a *forward tunnel*, and the reverse as a *backward tunnel*. However, they are in fact the same tunnel. Further, in order to defend the simple passive attack described in [Pfi94], we let G described in section 2.1 is a q-order subgroup of Z_p^*, where p, q are two large prime numbers and $q|p-1$. The generator g of order q is public and usually we can choose $p = 2q + 1$.

3.2 Establishing the Tunnel

There are two kinds of nodes, i.e., *fixed nodes* and *probability nodes* in a tunnel. The former, representing by P_i $(0 \leq i \leq \lambda)$ will re-encrypt the messages received from its predecessor, while the latter, representing by Q_j^i $(0 \leq i < \lambda, j = 1, 2, \cdots)$, only forwards the message. We also use link by link encryption to defend against some powerful attacks such as timing attack, counting attack and so on. Figure 1 outlines the whole process of establishing a rWonGoo tunnel.

1. *The sender Alice* (P_0) *chooses a node* P_1 *from its neighbors and establishes a tunnel to* P_1 *based on random choice described in following extension process. Then* P_1 *sends its ElGamal public key to Alice.*
2. *Alice extends the tunnel to fixed node* P_i $(1 \leq i \leq \lambda)$ *step by step and refreshes the key-pairs of the tunnel.*
3. *When the tunnel is extended to the receiver Bob* (P_λ) *and the key-pairs are refreshed, the establishment of the tunnel from Alice to Bob is finished.*

Fig. 1. The establishment process of a rWonGoo tunnel

Extension Process. We detail how to extend the tunnel. Suppose that the tunnel from Alice to node P_i $(1 \leq i \leq \lambda)$ has been established. In order to extend this tunnel to the next node, Alice sends a message m_1 to P_i along the tunnel (when we say sending a message along the tunnel, it means that the message will be re-encrypted by the nodes on the tunnel as described in section 3.3. For convenience, we only say sending a message unless otherwise stated) and tells it to choose the next node. When P_i receives m_1, it chooses some possible next nodes from its neighbors to form a set U_{i+1}, then sends it to Alice. Each item in U_{i+1} includes the IP address, PORT number and ElGamal public key of a node. Alice selects the next node P_{i+1} from the set U_{i+1} and sends a message m_2 to P_i and tells it to extend the tunnel to node P_{i+1}. The reason that we don't let P_i choose P_{i+1} directly is to defend against malicious attack [RP02]. When P_i receives m_2, it then makes a random choice to either extend the tunnel to P_{i+1} directly or to another node. P_i flips a biased coin to determine whether or not to extend the tunnel to P_{i+1} directly; the coin indicates to extend directly with probability p_f. If the flipping result is 'heads', then P_i extends the tunnel to P_{i+1} directly; else randomly selects another node Q_1^i from its neighbors and extends the tunnel to it. Then P_i sends a message m_3 contained the IP address, PORT number and ElGamal public key of node P_{i+1} to Q_1^i. When Q_1^i receives m_3, it does the similar operation as P_i. If Q_1^i doesn't extend the tunnel to P_{i+1} directly, it extends the tunnel to a randomly selected node Q_2^i from its neighbors and sends m_3 to Q_2^i. When the tunnel is extended to P_{i+1}, it sends a message m_4 to P_i and tells it that the tunnel extension has done. Note that this message m_4 from P_{i+1} to P_i hasn't been re-encrypted due to P_{i+1} doesn't yet have the re-encryption key. When receives m_4, P_i sends it to Alice who knows that this tunnel has been extended to node P_{i+1} after receiving it.

Then Alice starts to refresh the key-pair that is composed of a *forward key* (used in forward path) and a *backward key* (used in backwark path) for each fixed node on the tunnel. Firstly Alice encrypts the message $((y_1 \cdots y_{i+1})^r, g^r; y_0^{r'}, g^{r'})$ with the public key y_1 of P_1 and sends the ciphertext to P_1. P_1 gets the new key-pair K_1 when decrypting this message with its own private key x_1 as follows:

$$K_1 = ((y_1 \cdots y_{i+1})^r/(g^r)^{x_1}, g^r; y_0^{r'}, g^{r'}) = ((y_2 \cdots y_{i+1})^r, g^r; y_0^{r'}, g^{r'}).$$

Then P_1 produces the message

$$((y_2 \cdots y_{i+1})^r, g^r; y_0^{r'}\,(g^{r'})^{x_1}, g^{r'}) = ((y_2 \cdots y_{i+1})^r, g^r; (y_0 y_1)^{r'}, g^{r'})$$

and sends it to P_2 after encryption it with the public key y_2 of P_2. Each node P_j $(2 \leq j \leq i)$ repeats this process. When P_{i+1} receives the message $({y_{i+1}}^r, g^r; (y_0 \cdots y_i)^{r'}, g^{r'})$ from P_i, it decrypts this message with its own private key x_{i+1} and get the new key-pair

$$K_{i+1} = ({y_{i+1}}^r/(g^r)^{x_{i+1}}, g^r; (y_0 \cdots y_i)^{r'}, g^{r'}) = (1, g^r; (y_0 \cdots y_i)^{r'}, g^{r'}).$$

In this way, all the key-pairs along the tunnel are refreshed. Note that node P_{i+1} has only the backward key, no forward key, representing with 1 in K_{i+1}. Probability nodes only forward the message received, and have no key-pairs to refresh. Our key-pairs along the tunnel are protected with the random number r and r' compared with the scheme in [PIK93].

3.3 Data Transition

When the tunnel has been established, Alice can communicate with Bob on this tunnel. Let the tunnel be $P_0 - P_1 \cdots P_i - Q_1^i - Q_2^i - P_{i+1} \cdots P_{\lambda-1} - P_\lambda$. Assume that Alice wants to send a message m to Bob and the responding message m', and the key-pair of fixed node i is $K_i = ((y_{i+1} \cdots y_\lambda)^r, g^r; (y_0 \cdots y_{i-1})^{r'}, g^{r'})$, $(0 < i < \lambda)$, especially $K_0 = ((y_1 \cdots y_\lambda)^r, g^r; 1, g^{r'})$ and $K_\lambda = (1, g^r; (y_0 \cdots y_{\lambda-1})^{r'}, g^{r'})$. The data transition is as follows.

From Alice to Bob:

$P_0 \rightarrow P_1 : (G_0, M_0) = ((g^r)^{k_0} \mod p, m(y_1 \cdots y_\lambda)^{rk_0} \mod p)$

$= (g^{R_0} \mod p, m(y_1 \cdots y_{\lambda-1})^{R_0} \mod p)$, $(k_0 \in_u G, R_0 = rk_0)$

$P_i \rightarrow P_{i+1} : (G_i, M_i) = (G_{i-1} g^{rk_i} \mod p, M_{i-1}(y_{i+1} \cdots y_\lambda)^{rk_i} / G_{i-1}^{x_i} \bmod p)$

$= (g^{R_{i-1}+rk_i} \mod p, m(y_{i+1} \cdots y_\lambda)^{R_{i-1}+rk_i} \mod p)$

$= (g^{R_i} \mod p, m(y_{i+1} \cdots y_\lambda)^{R_i} \mod p)$, $(k_i \in_u G, R_i = R_{i-1} + rk_i, 1 \leq i < \lambda)$

The receiver P_λ computes $m = M_{\lambda-1}/G_{\lambda-1}^{x_\lambda} \mod p$.

From Bob to Alice:

$P_\lambda \rightarrow P_{\lambda-1} : (G'_\lambda, M'_\lambda) = ((g^{r'})^{k'_\lambda} \mod p, m'(y_{\lambda-1} \cdots y_0)^{r'k'_\lambda} \mod p)$

$= (g^{R'_\lambda} \mod p, m'(y_{\lambda-1} \cdots y_0)^{R'_\lambda} \mod p)$, $(k'_\lambda \in_u G, R'_\lambda = r'k'_\lambda)$

$P_i \rightarrow P_{i-1} : (G'_i, M'_i) = (G'_{i+1} g^{r'k'_i} \mod p, M'_{i+1}(y_{i-1} \cdots y_0)^{r'k'_i} / (G'_{i+1})^{x_i} \mod p)$

$= (g^{R'_{i+1}+r'k'_i} \mod p, m'(y_{i-1} \cdots y_0)^{R'_{i+1}+r'k'_i} \mod p)$

$= (g^{R'_i} \mod p, m'(y_{i-1} \cdots y_0)^{R'_i} \mod p)$, $(k'_i \in_u G, R'_i = R'_{i+1} + r'k'_i, 1 \leq i < \lambda)$

The sender P_0 computes $m' = M'_1 / {G'_1}^{x_0} \mod p$.

Each node (including fixed nodes and probability nodes) determines whether or not to re-encrypt the message based on whether it has the corresponding re-encryption key or not. Note that the messages (G_i, M_i) may pass through some probability nodes between node P_i and P_{i+1}. These probability nodes only do link by link encryption on the message, not re-encrypt it. So does on message (G'_i, M'_i). We can draw the following conclusion from the data transmission process.

Theorem 2. The size of ciphertext in our rWonGoo tunnel is double the size of corresponding plaintext.

4 Tunnel Security Analysis

Our anonymous tunnel is immune against replay attack and mainly used in large decentralized peer-to-peer networks in which Pfitzmann attack is hard. We assume a well-funded adversary who can observe some fraction of network traffic; who can operate nodes of its own; and who can compromise some fraction of the nodes on the network. However, the adversary is not able to invert encryptions and read encrypted messages.

4.1 Variable Anonymity

An outside eavesdropper can't relate the incoming and outgoing messages about a node due to link by link encryption, that's to say, he cannot distinguish between probability nodes and fixed nodes. However, inside compromised nodes can do this. Assume that all nodes between two malicious nodes controlled by an adversary are probability nodes, then the two nodes can recognize a message m between them because the probability nodes doesn't re-encrypt the message m. This means that the adversary can reduce the length of the tunnel by excluding all the probability nodes between them. However, if there is at least one fixed node between them, then the adversary can't do that because the encoding of the message m has changed when it passed through the fixed node. Hence, on condition that the tunnel length is fixed, in order to achieve stronger anonymity, we should add fixed nodes and reduce probability nodes, but the cost also increases. Sometimes, we don't need strong anonymity, then we can add probability nodes and reduce fixed nodes, achieving high efficiency. In a word, rWonGoo is a trade-off between anonymity and efficiency, providing a variable anonymity influenced by the number of fixed nodes and forwarding probability p_f.

4.2 Correctness

In rWonGoo, a node will take an input list of ciphertexts from different previous nodes and output a permuted and re-encrypted version of the input list. We define correctness as the outputs from a node should correspond to a permutation of the inputs. Sako and Killian [KS95] required each node leak side information so that anybody can verify correctness of the result. Later, Michels et al. pointed out that the side information could violate anonymity [MH96]. Hence, in our tunnel, we don't require each node reveals side information to achieve correctness. However, for correctness, we can draw on essentially any of the proof techniques presented in the literature on mixnets, as nearly all apply to ElGamal ciphertexts. For example, we can use the proof techniques in [FS01, Nef01]. Especially, the paper [FS01] presents a completely different approach than that of [KS95] to efficiently prove the correctness of a shuffle.

4.3 Pfitzmann Attack

As Pfitzmann pointed out that the scheme [PIK93] is insecure against Pfitzmann attack described in section 2.2, our tunnel based on it is also vulnerable to this

attack. However, we argue that such an attack is not realistic to rWonGoo that is mainly used in large decentralized peer-to-peer environment with thousands of nodes distributed in the Internet. Firstly, the sender usually trusts the receipt is honest. Secondly, our tunnel establishment guarantees that nodes in a rWonGoo tunnel are chosen uniformly at random from the whole network due to we let each node select the next hop. This makes the probability of Pfitzmann attack is very close to zero. Assume that the size of the network is N, and there are C corrupted nodes that are controlled by the adversary. Then the probability of Pfitzmann attack is $(\frac{C}{N})^2$. Lastly, in peer-to-peer networks, all nodes both originate and forward traffic. Thus a malicious node along the tunnel cannot know for sure whether its immediate predecessor is the initiator or not.

4.4 Replay Attack

The decryption of a message done by each node is deterministic in layered encryption mixnets. Hence if an adversary records the input and output batches of a node and then replays a message, that message's decryption will remain the same. Thus an attacker can completely break the security of the mixnet. It is a challenge work to defend against replay attack in layered encryption systems, and so far there is no effective approaches to prevent it though much work has been done [DDM03, DMS04]. However, rWonGoo is immune to this attack due to it adopts ElGamal cryptosystem that is probabilistic, not deterministic. We know that the parameters used for re-encryption are chosen anew per encryption, so when the same message passes a node again, the result of the re-encryption sent by the node would be different. Even if an attacker can observe the incoming and outgoing messages, he can't recognize which message is sent twice.

5 Conclusions and Future Work

We reviewed the universal re-encryption scheme and its extension, and discovered its vulnerability to Pfitzmann attack that can hard be prevented. Then we proposed a novel anonymous tunnel, rWonGoo, for low-latency communications. We presented how to establish the tunnel and transmit data. Our universal re-encryption tunnel can ensure the correctness of the outputs from a node, which is hard in layered encryption systems. rWonGoo is a tradeoff between anonymity and efficiency, providing variable anonymity. In addition, thanks to ElGamal re-encryption algorithm rWonGoo is immune to replay attack. It is not realistic to carry out Pfitzmann attack in large decentralized peer-to-peer systems though so far we haven't any effective countermeasures to it. Our tunnel is practical and more efficient than URE and EURE schemes. We believe that rWonGoo is a first step towards building re-encryption mixnets with correctness and variable anonymity for anonymous communications.

In the future we will do more research on analyzing the performance of rWonGoo. It is necessary to further develop our tunnel rWonGoo to increase the spectrum of attacks we can defend against except for Pfitzmann attack and replay

attack. We also plan to incorporate rWonGoo into layered encryption systems for message-based high-latency email to effectively defend against replay attack. At last, it is important to find a feasible way to facilitate deployment of our tunnel in reality.

References

[Cha81] D. Chaum. Untraceable electronic mail, return addresses and digital pseudonyms. Communications of the ACM., 24(2):84-88, February 1981.

[DDM03] G. Danezis, R. Dingledine, and N. Mathewson. Mixminion: Design of a Type III Anonymous Remailer Protocol. In Proceedings of the 2003 IEEE Symposium on Security and Privacy, May 2003.

[DMS04] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The Second-Generation Onion Router. In Proceedings of the 13th USENIX Security Symposium, 2004.

[Fai04] P. Fairbrother. An Improved Construction for Universal Re-encryption. In Proceedings of Privacy Enhancing Technologies, 2004. Springer-Verlag.

[FS01] J. Furukawa and K. Sako. An Efficient Scheme for Proving a Shuffle. In Proceedings of Crypto '01, pages 368-387, 2001. Springer-Verlag.

[GJJS04] P. Golle, M. Jakobsson, A. Juels, and P. Syverson. Universal re-encryption for mixnets. In Proceedings of CT-RSA 2004, February 2004. Springer-Verlag.

[GKK04] M. Gomulkiewicz, M. Klonowski, and M. Kutylowski. Onions Based on Universal Re-Encryption - Anonymous Communication Immune Against Repetitive Attack. In Proceedings of Workshop on Information Security Applications (WISA'2004), 2004. Springer-Verlag.

[JJ01] M. Jakobsson and A. Juels. An optimally robust hybrid mix network. In Principles of Distributed Computing (PODC 2001), pages 284-292, August 2001.

[KS95] J. Kilian and K. Sako. Receipt-free MIX-type voting scheme - a practical solution to the implementation of a voting booth. In Proceedings of Advances in Cryptology - Eurocrypt'95, May 1995. Springer-Verlag.

[MCPS03] U. Möller, L. Cottrell, P. Palfrader, and L. Sassaman. Mixmaster Protocol - Version 2. Draft, July 2003.

[MH96] M. Michels and P. Horster. Some remarks on a receipt-free and universally verifiable mix-type voting scheme. In Proceedings of Advances in Cryptology - Asiacrypt'96, November 1996. Springer-Verlag.

[Nef01] A. Neff, A verifiable secret shuffle and its application to e-voting. In Proceedings of ACM CCS '01, pages 116-125, 2001.

[Pfi94] B. Pfitzmann. Breaking efficient anonymous channel. In Proceedings of Advances in Cryptology - Eurocrypt'94, May 1994. Springer-Verlag.

[PIK93] C. Park, K. Itoh, and K. Kurosawa. Efficient anonymous channel and all/nothing election scheme. In Proceedings of Eurocrypt'93, LNCS 765, pages 248-259, 1994. Springer-Verlag.

[RP02] M. Rennhard and B. Plattner. Introducing MorphMix: Peer-to-Peer based Anonymous Internet Usage with Collusion Detection. In the Proceedings of the Workshop on Privacy in the Electronic Society (WPES'02), November 2002.

[RR98] M. K. Reiter and A. D. Rubin. Crowds: Anonymity for Web Transactions. ACM Transactions on Information and System Security, 1(1): 66-92, November 1998.

An Integrated Information Retrieval Support System for Multiple Distributed Heterogeneous Cross-Lingual Information Sources

Lin Qiao, Weitong Huang, Qi Wen, and Xiaolong Fu

Computer and Information Management Center, Tsinghua University,
Beijing 100084, PR China
{qiaolin, hwt, vendy, fuxiaolong}@cic.tsinghua.edu.cn

Abstract. This paper presents a new integrated information retrieval support system (IIRSS) which can help Web search engines retrieve cross-lingual information from heterogeneous resources stored in multi-databases in Intranet. The IIRSS, with a three-layer architecture, can cooperate with other application servers running in Intranet. By using intelligent agents to collect information and to create indexes on-the-fly, using an access control strategy to confine a user to browsing those accessible documents for him/her through a single portal, and using a new cross-lingual translation tool to help the search engine retrieve documents, the new system provides controllable information access with different authorizations, personalized services, and real-time information retrieval.

1 Introduction

A digital campus is an information-rich and seamlessly connected environment that brings the world to a student's fingertips and lets the student move freely about the globe. This digital campus functions through the integration of a multiplicity of technologies in a unified network environment – a seamless, secure, collaborative environment for learning, achievement and administration that is available to everyone.

In building a digital campus, information retrieval system (IRS) is the key task of the information resource management system (IRMS) which provides the university community with policies, procedures and support for secure access to information resources to complement its teaching, learning, research, and outreach as well as to support administrative operations. IRS must solve three questions, that is, how to support administering authorization, authentication and security access controls to information technology resources to the university community, how to improve the effectiveness and efficiency of services provided to the university community, and how to make authorization rules and access information accessible for review to appropriate levels of management within the university community for decision making and strategic planning.

Most of information retrieval services in a digital campus use embedded Web-based full-text information retrieval systems, e.g. Google (http://www.google.com)

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 863 – 872, 2005.

and Baidu (http://www.baidu.com). The former is good at searching documents written in English, and the latter good at searching documents written in Chinese. These information retrieval systems have their advantages of high-level efficiency and supporting simple cross-lingual retrieval. However, they can not support retrieving databases, not guarantee effectiveness of retrieved information, and not provide security access control.

In this paper we present a new integrated information retrieval support system (IIRSS) for multiple distributed heterogeneous cross-lingual information sources. The IIRSS with a three-layer architecture, which is integrated in Tsinghua University Campus Network (TUCN), cooperates with other application servers running in the campus network. By using intelligent agents to collect information and to create indexes on-the-fly, using an access control strategy to confine a user to browsing those accessible documents for him/her through a single portal, and using a new cross-lingual translation tool to help the search engine retrieve documents, the IIRSS can

- support information collection from multiple distributed heterogeneous cross-lingual information sources,
- support access control of users and documents,
- filter pieces of inaccessible information from query results according to user's access control information,
- construct the index while a piece of information is put out on-the-fly, and
- reconstruct the index of a document by using document information stored in the document databases.

This paper is organized as follows. Section 2 introduces the three-layer framework of the IIRSS. In Section 3 a multiagent-based information collection technique is discussed, in Section 4 an access control strategy which supports safe access to documents is deployed, and in Section 5 a cross-lingual information retrieval tool for the IIRSS is presented. Section 6 gives experimental results, while Section 7 draws conclusions.

2 The IIRSS Framework

The TUCN uses a single information portal as the solution of integrating large-scale information resources and applications. Users access application systems through the portal which controls and manages information resources and applications distributed in different servers physically placed in different departments and laboratories, as shown in Fig. 1.

As a part of TUCN, the IIRSS has to cooperate with other application servers running in the campus network. Fig. 2 shows the framework of the IIRSS, which is divided into three layers, i.e. Interface Layer, Management Layer, and Search Engine Layer.

In order to interact with distributed heterogeneous application systems, an Information Collector Module and a Web Searcher Module are included in the Interface Layer. The former provides interfaces for application servers and collects information of documents, while the latter provides interfaces for the portal, through which a user can submit his/her query and browse results.

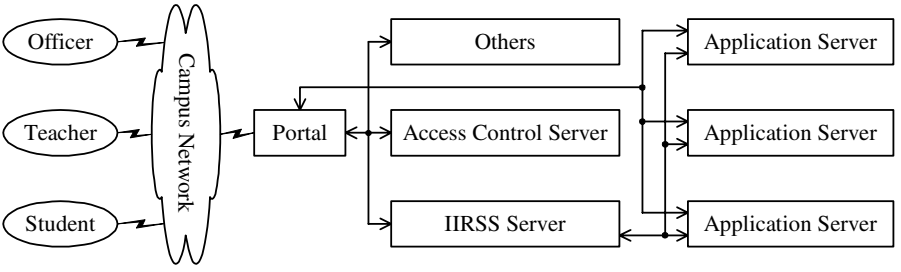


Fig. 1. Infrastructure of Tsinghua University Campus Network

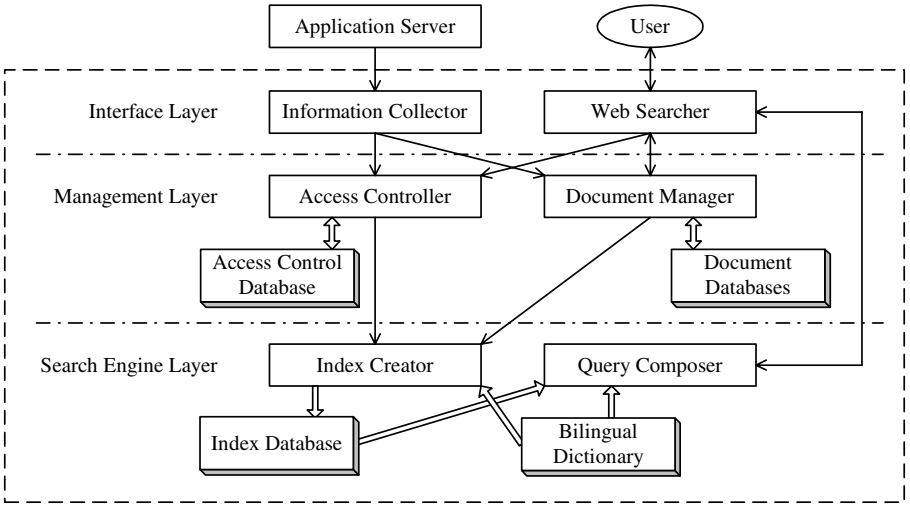


Fig. 2. The three-layer framework of the IIRSS

In the Management Layer, an Access Controller Module is used for managing and maintaining access control information of documents stored in a specific access control database, and a Document Manager Module for constructing and/or reconstructing indexes of documents.

The Search Engine Layer consists of four parts, i.e. an Index Creator Module, an Query Composer Module, an Index database, and a Bilingual Dictionary. The Index Creator Module is responsible for creating an index for a piece of information, and stores it in the Index database, while the Query Composer Module searches the Index database by keywords that a user inputs. If need be, both of the Index Creator Module and the Query Composer Module look up the Bilingual Dictionary and use a cross-lingual translation tool that this paper introduces to translate some keywords into other languages.

3 Multiagent-Based Information Collection

In recent years, agents have been developed for information management applications. The goal of intelligent search agents is to allow end-users to search effectively, be it either a single database of bibliographic records or a network of distributed, heterogeneous, hypertext documents [1]. Intelligent agents, which can locate, retrieve, and integrate the answers to a query into one result, alleviate the work of both a developer and a user. They communicate with other agents by means of message-passing or blackboards, making requests and performing requested tasks [1]. In a distributed environment agents managing applications and sources need to coordinate and cooperate with each other in order to achieve a goal. This is typically accomplished by using specification sharing, contract nets, or federated system [2].

The IIRSS uses JATLite (Java Agent Template, Lite) [3], a package of programs written in the Java language that allow users to quickly create new software agents that communicate robustly, to construct the Information Collector Module. JATLite is a readily available tool and provides a basic infrastructure for multiple agents.

In general, the Information Collector Module consists of some monitor agents and collector agents, as illustrated in Fig. 3. A monitor agent, running in an application server, obtains modification information of documents stored in the application server, and then wrap it in a kind of standard formatted information that can be used by our retrieval support system directly. A collector agents invokes corresponding functions defined in the Access Controller Module and the Document Manager Module to process modification information of documents.

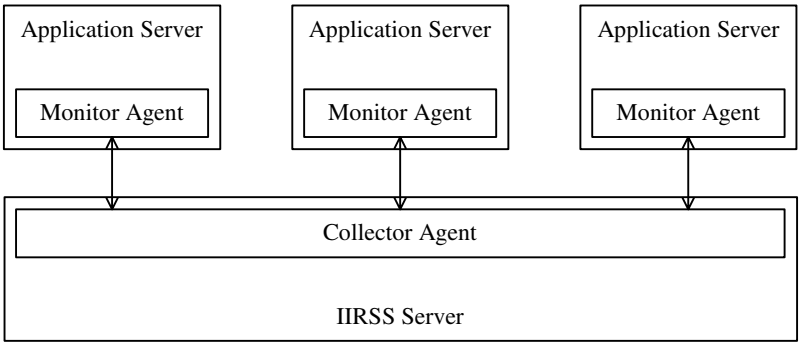


Fig. 3. Multiagent-based information collection

In order to get an index of a document in the intranet offhand, the Information Collector Module obtains document information passively: an application impels the retrieval support system to modify its indexes via a monitor agent. In other words, an application triggers a collector agent by invoking the corresponding function of the monitor agent running in the application server when a new document is posted; the collector agent checks document information to ensure its availability and validation,

and then calls corresponding functions implemented in the Document Manager Module and the Access Controller Module to append the document and its access control information respectively; the Document Manager Module invokes a function implemented in the Index Creator Module to update indexes.

The flow of information collection is as follows:

- Step 1. An application calls the *CreateFileRec* method of the monitor agent running in the application server when a document is posted. The *CreateFileRec* method creates a *fileRec* object instance of the *FileRec* class to store the document information, and an *authorRec* object instance of the *AuthorRec* class to store access control information of the document.
- Step 2. According to the content of document information, the monitor agent sets parameters, *fileId*, *subject*, *content*, and *location*, etc., of the *fileRec* object and then pass it to a collector agent.
- Step 3. According to access control information of the document, the monitor agent sets parameters, *fileId*, *userId*, *userRole*, *group*, and *department*, etc., of the *authorRec* object.
- Step 4. The collector agent invokes the *add* method of a *fileManagement* object instance of the *FileManagement* class to insert the *fileRec* object into a document database, to insert the *authorRec* object into an access control database. And then an index of the document is generated.

In the IIRSS, a monitor agent, which is platform-independent, has a configurable interface in order to deal with documents with various formats (such as Dynamic HTML page, Adobe PDF format, and Microsoft Word format, etc.) and access control information of them flexibly. Through the interface an user can customize document format and/or access mode. In addition, the type definition of a monitor agent is scalable, which makes it possible to support new document formats.

4 Access Control

As discussed above, a user accesses applications accessible for him/her through a single portal. In similar, he/she accesses the IIRSS through the portal, as shown in Fig. 4. In retrieving information the IIRSS executes following procedures:

- Step 1. The portal gets a user's name and password.
- Step 2. The portal gets the user's access control information stored in the Access Control database, creates a *userRec* object of the *UserRec* class, and sets its parameters, *userId*, *userRole*, *group*, and *department*, etc.
- Step 3. The portal gets the user role control information built in every application system.
- Step 4. The portal sends the IIRSS keywords, which typically are inputted in a search Web page, and his/her access control information.
- Step 5. The IIRSS invokes the method *searchFile* of a *searchManagement* object and transforms the user's query request into formatted ones that a retrieval system can read. In this step, translating keywords into another language might be needed. Under our experimental environment a Baidu search

engine serves as the retrieval system, which responses the formatted requests and then pushes query results, stored in a *fileRecList* object, to the portal.

- Step 6. The portal lists links to those documents accessible by the user by comparing the user's access control information with user role control information built in the application system.
- Step 7. When the user clicks a link, the portal makes the corresponding application system push the document to user according to the application information attached to the document information.

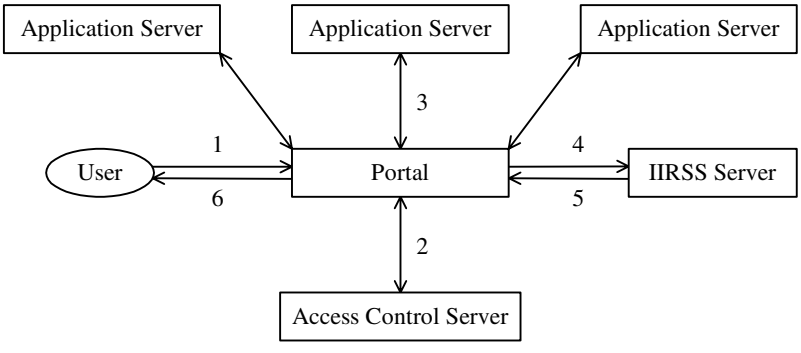


Fig. 4. Information retrieval with access control by a single portal

Usually, different users have different access control information. Thus, the Access Controller Module of the IIRSS supports several authorization modes: (a) authorizing a single user, (b) authorizing a user role, (c) authorizing a user group that can be customized, and (d) authorizing a department. Every piece of information in the access control database includes authorized user(s), authorized user role(s), authorized user group(s), and authorized department(s), etc. This makes it possible to authorize users cumulatively.

5 Cross-Lingual Information Retrieval

With the popularity of the Internet and Intranet, more and more languages are becoming to be used for Web documents. As far as the TUCN is concerned, for example, most of documents are written in Chinese and/or English. It is very difficult for the monolingual users to retrieve documents written in non-native languages, and besides, there might be cases, depending on the user's demand, where information written in non-native languages is rich. Needs for retrieving such information must be large. Cross-lingual information retrieval (CLIR) derives itself from these needs.

In general, recent CLIR studies concentrate on three categories: document translation, query translation, and inter-lingual representation. The document translation approach achieves better retrieval effectiveness than those based on query translation by using existing machine translation systems [4], but it is not suitable for

multi-lingual, large-scale, and frequently-updated collection of the Web because it is impossible to translate all of documents in the Intranet beforehand and it is difficult to extend this approach to new languages [5]. The third approach transfers both documents and queries into an inter-lingual representation, such as bilingual thesaurus classes or a language-independent vector space. It requires a training phase using a bilingual corpus as a training data [5].

The method this paper presents belongs to the second category. In our method, a source language query is first translated into target language (usually Chinese) using a bilingual dictionary. The method has an advantage that the translated queries can simply be fed into existing search engines (such as Baidu which is apt at processing Web queries written in Chinese).

The key problem in query translation approaches is that queries submitted from ordinary users of Web search engines tend to be very short (approximately two words on average [6]), and usually consist of just an enumeration of keywords. However, existing cross-lingual Web search engines, such as Google and Baidu, can not process Chinese-English and English-Chinese translation accurately, and this leads to get many irrelevant results.

Existing cross-lingual Web search engines have two disadvantages. The first is that they are not good at processing abbreviations of Chinese proper nouns which often appear in documents in the TUCN. For example, *Computer and Information Management Center* means 计算机与信息管理中心 in Chinese, commonly abbreviated to 计算中心 or seldom to 计算机中心. When a Web query is submitted, most of search engines cannot treat 计算中心 as a whole word, and translate it into two words, *computation* and *center*, by searching a Chinese-English dictionary. In order to satisfy such needs on these retrieval systems, the user has to manually translate the query by using a dictionary.

Proper nouns, especially abbreviations of Chinese proper nouns, are usually essential terms for retrieval in the TUCN, and the failure to translate them might heavy decrease in retrieval effectiveness. In order to solve this problem the IIRSS uses a customable English-Chinese dictionary instead of traditional English-Chinese dictionary. Each proper noun has an abbreviation list where all of standard and nonstandard abbreviations are stored. Before a Web query is submitted, the Query Composer Module searches the abbreviation list of the keyword and submits its abbreviations together.

The second disadvantage of the existing cross-lingual Web search engines is that they have to deal with Web queries in many different kinds of topics or fields. It is well known that for different users a keyword has different meanings. For example, the word *mouse* might be 鼠标 (a computer input device) or 鼠 (such as pleural mouse, a medical term). In the TUCN a user who submits a Web query might be a student at Department of Computer Science or a teacher at School of Medicine; hence the query translation must be capable of reflecting how the user really thinks.

To attack the second problem the IIRSS employs user information (such as *userId*, *userRole*, *group*, and *department*, etc.) as a feature term selector of keywords. In detail, words in our English-Chinese dictionary is arranged in discipline categories, where a discipline category a_q in query language corresponds to a category a_t in target language. When constructing indexes of a document, the Index Creator Module first

extracts terms from the document which belongs to a discipline category a_q using necessary discipline information that has been attached to the document, and then calculates the weights of the extract terms. The higher-weight terms are put into the feature term set f_q of the discipline category a_q. Last, the feature term set are translated into target language. These discipline category pairs, a_q and a_t, are used to retrieval.

Weights of feature terms are calculated by term frequency/inverse category frequency (TF/ICF), a variation of term frequency/inverse document frequency (TF/IDF), which is reported in [5].

When translating a query keyword, the Query Composer Module first looks up the feature term in the bilingual dictionary and extracts all translation candidates, and then picks up the highest-weighted translation candidate from the discipline category a_q, which coincides with the user's interests, by using user information. If no translation candidate for a feature term exists in the feature term set of the discipline category a_q, the term itself is as a feature term in the target language. This is very useful because some English terms (mostly abbreviations, such as *CIC*, *IIRSS*, etc.) are commonly used in Chinese documents.

6 Experiment Results

The IIRSS has been implemented and integrated in the local office network of Computer and Information Management Center, Tsinghua University. There are totally 673 documents distributed in three servers of the local office network when following experiment is performed. Among them, 631 documents are written in Chinese, 40 documents written in Chinese and English, and 2 documents written in English; 407 documents are with dynamic HTML format, 42 documents with Adobe PDF format, 135 documents with Microsoft Word format, 37 documents with Microsoft Excel format, 52 documents with Microsoft PowerPoint format.

Among all of documents, there are 101 articles including keyword 计算机与信息管理中心 or 计算中心 and 2 articles including keyword *Computer and Information Management Center* or *CIC*; there are 12 documents accessible by everyone, 82 documents accessible by Director of Computer and Information Management Center, 39 documents accessible by everyone belonging to office group, 54 documents accessible by everyone belonging to infrastructure group.

Retrieval results using keyword 计算中心 is shown in Table 1, where an *officer* belongs to the office group, a *developer* belongs to the infrastructure group, and

Table 1. Retrieval results using the keyword 计算中心

Role of User	Before the new document added	After the new document added
Anonymous	12	12
Administrator	103	104
Director	82	83
Officer	39	40
Developer	54	54
ClusterA	63	62

Table 2. Retrieval results using the keyword *Tsinghua Univ*

Role of User	Before the new document added	After the new document added
Anonymous	30	30
Administrator	276	277
Director	239	239
Officer	183	184
Developer	209	210
ClusterA	236	237

Table 3. Retrieval results using the keywords *Tsingha University* and *CIC*

Role of User	Before the new document added	After the new document added
Anonymous	12	13
Administrator	81	82
Director	79	80
Officer	20	21
Developer	38	39
ClusterA	41	42

clusterA includes all members in the office group or in the infrastructure group. The second column gives first-time retrieval results, and the last column gives second-time retrieval results after a new document including keyword 计算机与信息管理中心 is added, which can only be accessed by Director and members of the office group. The experiment also shows that even though 计算机与信息管理中心 is not the same as keyword 计算中心, IIRSS also find those documents including 计算机与信息管理中心.

Among all of documents, there are 244 articles including keyword 清华大学 or 清华 and 32 articles including keyword *Tsinghua University* or *Tsinghua Univ*; there are 30 documents accessible by everyone, 239 documents accessible by Director, 183 documents accessible by everyone belonging to the office group, 209 documents accessible by everyone belonging to the infrastructure group.

Retrieval results using keyword *Tsinghua Univ* is shown in Table 2. The second column gives first-time retrieval results, and the last column gives second-time retrieval results after a new document including keyword 清华大学 is added, which can only be accessed by members in ClusterA.

Among all of documents, there are 81 articles including keyword 清华大学/清华/*Tsinghua University*/*Tsinghua Univ* and keyword 计算机与信息管理中心/计算中心/*Computer and Information Management Center*/*CIC*. There are 12 documents accessible by everyone, 79 documents accessible by Director, 20 documents accessible by everyone belonging to the office group, 38 documents accessible by everyone belonging to the infrastructure group.

Retrieval results using keywords *Tsinghua Univ* and *CIC* is shown in Table 3. The second column gives first-time retrieval results, and the last column gives second-time retrieval results after a new document including keyword 清华大学 is added, which can be accessed by anyone.

7 Conclusion

The paper proposes a new integrated information retrieval support system, IIRSS, which can help Web search engines retrieve cross-lingual information from heterogeneous resources stored in multi-databases in a campus network. The IIRSS, with a three-layer architecture, can cooperate with other application servers running in the campus network. Even though the IIRSS is designed for the TUCN, it should be suitable for other Intranet environments, such as a government office network or an enterprise Intranet.

By using intelligent agents to collect information and to create indexes on-the-fly, using an access control strategy to confine a user to browsing those accessible documents for him/her through a single portal, and using a new cross-lingual translation tool to help the search engine retrieve documents, the IIRSS provides controllable information access with different authorizations, personalized services, and real-time information retrieval.

The experimental results this paper presented are compendious. More experimental results and detailed performance analysis can be obtained once the IIRSS is integrated into the TUCN. This is our ongoing work.

Acknowledgement

This work was partially supported by National Nature Science Foundation, grant number 69773028, of *P. R. China*.

References

1. Haverkamp, D. S., Gauch, S.: Intelligent Information Agents: Review and Challenges for Distributed Information Sources. Journal of the American Society for Information Science and Technology 49 (1998) 304-311
2. Takkinen, J.: Intelligent Agents for Information Retrieval and Integration. http://www.ida.liu.se/~juhta/publications.shtml (1998)
3. JATLite. http://java.stanford.edu (1998)
4. Sakai, T.: MT-Based Japanese-English Cross-Language IR Experiments Using the TREC Test Collections. Proceedings of the Fifth International Workshop on Information Retrieval with Asia Languages, Hong Kong (2000) 181-188
5. Kimura, F., Maeda A., Yoshikawa M., Uemura, S.: Cross-Language Information Retrieval Based on Category Matching Between Language Versions of a Web Directory. In: Proceedings of the 6th International Workshop on Information Retrieval with Asia Languages in Conjunction with the 41th Annual Meeting of the Association for Computational Linguistics, Sapporo, Japan (2003), 153-159
6. Jansen, B. J., Spink, A., Saracevic, T.: Real Life, Real User and Real Needs: A Study and Analysis of User Queries on the Web. Information Processing & Management 36 (2000) 207-227

DHAI: Dynamic Hierarchical Agent-Based Infrastructure for Supporting Large-Scale Distributed Information Processing

Jinlong Wang, Congfu Xu, Huifeng Shen, Zhaohui Wu, and Yunhe Pan

Institute of Artificial Intelligence, Zhejiang University,
Hangzhou 310027, China
zjupaper@yahoo.com, xucongfu@cs.zju.edu.cn,
yaekee@hotmail.com, wzh@cs.zju.edu.cn, panyh@sun.zju.edu.cn

Abstract. The emergence of Internet, Intranet, local area networks, and ad hoc wireless networks introduces a plethora of new problems in information processing. In order to overcome these problems, we combine the Grid and P2P technologies, and propose a dynamic hierarchical agent-based infrastructure. By dint of agents, we can not only overcome a lot of limitations of the mechanisms which Grid and P2P support for data management and interchange in large-scale dynamic distributed systems, but also provide a suitable paradigm for implementing systems that need to configure adaptively, and especially keep users' information local to preserve individual privacy.

1 Introduction

With the rapid collection of data in wide variety of fields, the demands in data analysis, from business decisions to scientific researches, are ever growing. Today's challenges are less related to data storage and information retrieval, but in the analysis of data on a global scale in a heterogeneous distributed information system. These systems are characterized with: (1) Existence of very large number of distributed and heterogeneous participants; (2) Dynamic nature of participation; (3) No global synchronization.

In order to satisfy the above requirements, we combine the Grid [1] and Peer-to-Peer (P2P) [2] technologies, and propose a dynamic, hierarchical and agent-based [3,4] infrastructure for large-scale distributed information processing. The hierarchical structure of our architecture takes advantages of both P2P and Grid. The introduction of hierarchy in the architecture increases the scale. Moreover, the hierarchical structure is more stable when nodes join and leave the network frequently. By dint of agent technique, a new way of analyzing, designing, and implementing large-scale distributed complex data analysis systems is provided, and a number of limitations in the mechanisms which P2P and Grid support for data management and interchange can be overcome. In

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 873–882, 2005.

the infrastructure, top-level nodes, equipped with super computers, connected with high-speed network comprise Grid, and bottom level network composed of desktop and other pervasive computers connected intermittently to the network are organized with P2P. At the same time, we introduce agent technology to nodes in the system. Through the infrastructure, we can implement efficient large-scale distributed information processing which does not require all-to-all communication, and supply the support of security and trustworthiness.

The rest of this paper is organized as follows. Next section describes the infrastructure. Section 3 presents information processing in the infrastructure. Section 4 gives an application of large-scale distributed privacy preserving data mining based on the infrastructure. Finally, we conclude with some remarks and the ongoing and future work.

2 Infrastructure Description

Herein we describe the dynamic hierarchical distributed agent-based Infrastructure. The infrastructure is based on P2P-Grid whose nodes are agents. The main goal of the infrastructure is to supply a platform for large-scale distributed information processing.

2.1 Infrastructure Goals

Scalability
Scalability is the ability of a system to operate without a noticeable drop in performance despite variability in its overall operational size. The infrastructure is intended to be capable of scaling to millions of nodes.

Decentralized Control
Decentralized control gives the possibility of balancing the loadings, making systems scalable for a large number of communication nodes.

Global Asynchrony
Any nodes can take action independently and asynchronously.

Dynamic Nature
Dynamic nature is the key characteristic of the large-scale distributed systems. Through managing the processing of node joining and leaving in a more intelligent method, we can solve the unstable topological problem.

Transparency
When a node leaves or joins system, network topology in the bottom-level is completely transparent to the top-level.

Local Communication
Each node can only be familiar with a small set of other nodes – its immediate neighbors – and communicate with them, this is valuable for real-world large-scale distributed systems.

2.2 Architecture Components

The overall framework is shown as Fig.1 below.

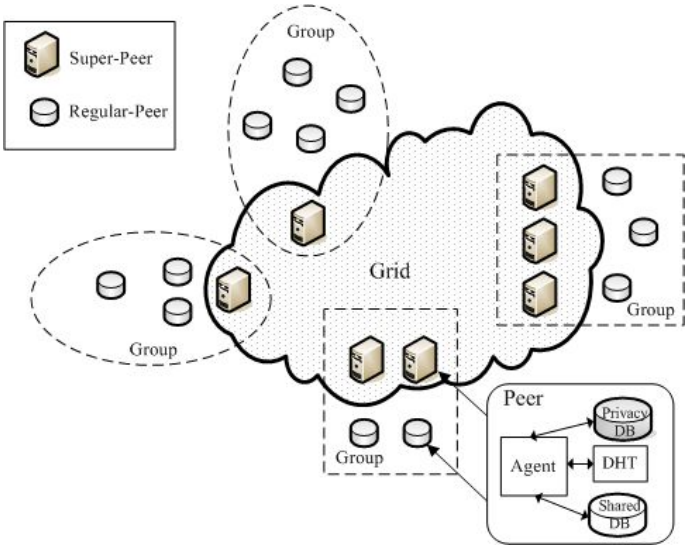


Fig. 1. Overall Framework of Architecture

Definition 1. ***Peer.*** *Each peer is symbolized as p. All of peers* p_i *form a set of* $\mathbb{P} = \{p_i | i \in \mathbb{N}\}, \mathbb{N} = \{1, 2, \ldots, n\}$*, here, n is the number of the peers. The peers are interconnected through a network of links and switching equipments, such as routers, bridges, etc.*

Definition 2. ***Group.*** *All of peers are organized into groups. Let* $\mathscr{I}$ *be the number of groups,* $G_i = \{p_{ij} | j \in \mathbb{N}\}$ *(*$i \in \{1, \ldots, \mathscr{I}\}$*) are the peers in group i, which satisfies* $G_i \bigcap G_j = \emptyset$ $(i \neq j)$ *and* $\bigcup_{i \in \mathscr{I}} G_i = \mathbb{P}$*. The groups may or may not be such that the peers in the same group are topologically close to each other, depending on the application needs.*

Definition 3. ***Super-peer.*** *Each group is required to have one or more super-peers. Let* $SP_i = \{sp_{ij} | j \in \mathbb{N}\}$ *(*$i \in \{1, \ldots, \mathscr{I}\}$*) be the set of super-peers in group i,* $SP_i \subseteq G_i$*.*

Definition 4. ***Top-Level Overlay Network (Grid).*** *In every group, there are one or more super-peers. The super-peers from all the groups are organized into a top-level overlay network (GRID) described by a directed graph* (SP, E)*, where, SP is the set of all the super-peers and E is a given set of virtual edges between the super-peers in SP. It is important to note that this overlay network defines directed edges among groups and not among specific peers in the groups.*

Definition 5. ***Regular peer.*** *Let $R_i = G_i - SP_i$ ($i \in \{1, \ldots, \mathscr{I}\}$) be the set of "regular peers" in group G_i. In intra-group, these homogeneous regular peers are organized into a hierarchy, which can be used to address the problem of scalability.*

Definition 6. ***Agent.*** *Agents are software which helps/assists the peers (super-peer and regular-peer) for the modelling and implementation of complex systems. Each agent is one portion of a peer. Agents my initiate tasks on behalf of peers. Each agent is symbolized as α. All of agents form a set of $A = \{\alpha_i | i \in M\}, M = \{1, 2, \ldots, m\}$, here, m is the number of the agents. In general, each agent corresponds to a peer ($m = n$).*

Definition 7. ***Privacy Database (PDB).*** *PDB is owned by each peer. PDB may not be open to other peers.*

Definition 8. ***Shared Database (SDB).*** *SDB is owned by each peer, but SDB my be shared by other peers.*

Definition 9. ***Dynamic Hash Table (DHT).*** *DHT is the table of route, indicating those peers the messages will be sent to. There are two types of peers: up-peer and down-peer.*

Definition 10. ***Virtual Organize (VO).*** *VO is a fabric structure that is composed of peers and is established by a series of protocols.*

Definition 11. ***Protocol.*** *The Protocol between peers is not discussed in the paper, since it can be implemented in many ways. What's more, we can also use one protocol in controlling but another distinct protocol in transmission under different situations.*

2.3 Overall of Infrastructure

The distributed infrastructure is a virtual and dynamic hierarchical architecture (as Fig.1), where peers are organized into disjoint Virtual Organizes (VOs). The VOs are generally hierarchically arranged according to the related domains. Each VO maintains its own overlay network. A top-level overlay network is defined among the VOs. The top-level overlay network is organized with super-peers. Normally, the so-called super-peers are a subset of peers with sufficient bandwidth and more processing power. These super-peers constitute a Grid environment. Grid technologies have been investigated by many researchers [1,5,6] and will not be discussed in detail in this paper.

Because the classification of group is according to the node distances and functions, there will be much more frequent communication among the same group than that between groups. And, in a VO, we make use of asymmetrical bidirectional communication patterns to improve communication efficiency. Upward communication transfers all types of information, including system messages (peers joining or leaving) and data messages (data updating) to update up-peers, the downward communication transfers only system messages to adjust

to the topology. In this way, the architecture can use the bandwidth more effectively, gain the advantage over Grid or P2P respectively, and thus suit a great variety of network environments better. By virtue of agent technology, we can make the architecture more intelligent, and selectively hide sensitive data and information to suit the demands of privacy preserving. Through this architecture, a great diversity of the cryptographic methods can be adopted. Within the different groups, we can utilize various cryptographic methods. Even in the same group, we can sustain diversity of the security methods based on the demands.

2.4 Intra-group

Agent and P2P are complementary concepts, in that cooperation and communication among peers can be driven by those agents that reside in each peer. A peer depends on its agents to "Deal with Problem Requests", and to give "Solutions". Also, the peer expects these agents to be "Failure Tolerant", and to have good performance in terms of "Good Bandwidth Optimization". Within every peer, each agent controls the communication of message, receives message from other peers, and sends information to other peers for system control or further processing. In a decentralized P2P environment, each peer does not know beforehand which partners to communicate with. Agents provides a function with searching and registration capabilities, which allow the peer to get to know other peers. This ability is based on *DHT*. The logical hierarchy of peers forms a group domain by the *DHT*. Especially, agent supplies peers with services for information processing, such as data analysis, knowledge discovery, data mining, data transforming, data hiding, *etc.*

Through agents, the peers receive the message in the network, and then process the message. The peer has state database, which is typically used to record information and state. Through processing between received data and local state database, the peer will act to decide which information should be sent to, and send these information to other peers adjacent to it by virtue of *DHT*.

When a peer joins the system, it registers with one of existing peers. A peer can only have one connection to another peer higher in the hierarchy to register with, but be registered with many lower level peers. A peer in the hierarchy has the identities of both its upper peers and lower peers in order to communicate with them. A peer can also leave the system at any time, which means it is not available to the system any more. In this situation, its lower peers must be informed to dynamically adjust the network topology to keep the hierarchical relations. The hierarchical model is used to address the problem of scalability.

3 Information Processing in the Infrastructure

In this section, we will describe information processing based on the infrastructure. This has broad value at present.

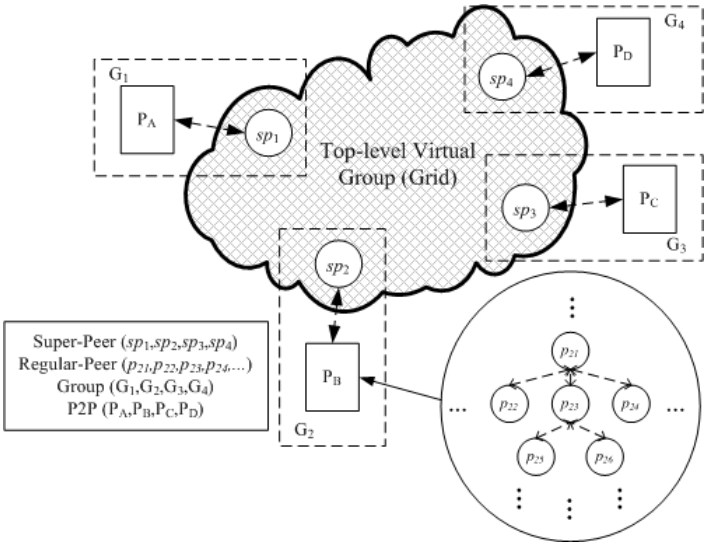


Fig. 2. The hierarchical infrastructure for distributed information processing

Just as Fig.2 (In the interest of clarity, we make an example where each group comprises a super-peer.), within each group, an super-peer manages a range of peers. Regular peers collaboratively process information in P2P based on agent.

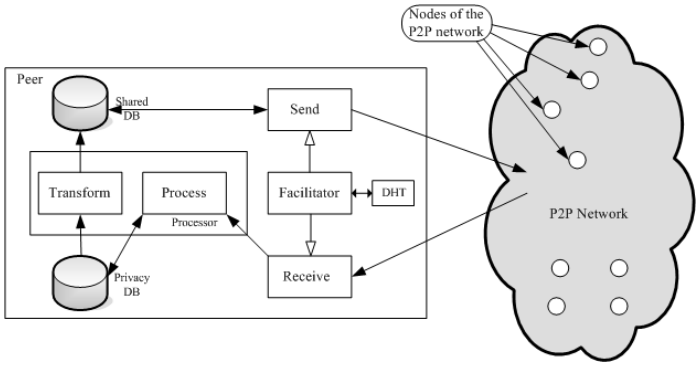


Fig. 3. Peer in information processing through P2P

In a group, for each peer, agent receives the information from other peers through P2P networks, and then it processes the information together with its privacy database (as Fig.3). After that, the agent updates its own privacy database and transforms the message to be sent to the shared database according to security strategies. These strategies satisfy the requirements of privacy

preserving and trustworthiness. Finally, through its own *DHT*, the agent sends relevant fresh information in its shared database to other peer adjacent to it for further processing. Layer upon layer, the super-peer comprises the up-to-date information from the same group. In the top-level overlay network, we can use relevant Grid technology to process them.

The communication is evidently reduced to a large extent by never transmitting all data values but incremental and updated information. This approach does not require CPU and I/O costs significantly higher than a similar approach and the communication may be lower.

4 An Application to Large-Scale Distributed Privacy Preserving Data Mining

One key aspect of exploiting the huge amount of autonomous and heterogeneous data sources in the Internet is not only how to retrieve, collect and integrate relevant information but also to discover previously unknown, implicit and valuable knowledge. Nowadays, data mining is one of the most important topics in large-scale distributed domains such as military, commerce and health-care *etc.*, where the precondition is that no privacy of any site should be leaked out to other sites. This requires preserving users' privacy while processing data mining in large-scale distributed systems.

Privacy-preserving data mining (PPDM) was first introduced by Agrawal and Srikant in 2000 [7]. The main objective in PPDM is to develop algorithms for modifying the original data in some way, so that the private data and knowledge remain private after the mining process. In order to broaden application of this technology in large-scale distributed systems, it must cater to users who do not own sophisticated (hardware and software) platforms and permanent network connections.

Based on the hierarchical infrastructure, we can mine large-scale distributed privacy-preserving data. The partitioned distribution of data can be not only horizontal but also vertical. The industrial chain data is a typically vertically partitioned distributed data. Through industrial chain PPDM, we can preserve individual sensitive data and knowledge of each corporation and get plenty of valuable information to improve the whole benefit and profit.

In the following, we give an example of the industrial chain of autobicycle manufacture which is composed of lots of assemble plants, engine manufactures, tire workshops, accessory manufactories (Fig.4).

This is also a representative industrial structure of Zhejiang province, China. By adopting PPDM based on this architecture, it is supposed to discover the relationship between the congregate of the industry and region economic growth.

The engine manufactures, tire workshop *etc*, respectively forms a VO, making use of super-peers of them, we can cooperate between them. Through the dynamic, hierarchical, agent-based infrastructure, we can mine data in this distributed system in a more efficient way. We assume a large-scale distributed PPDM system consisting of a set of networked, homogeneous data sites. Each

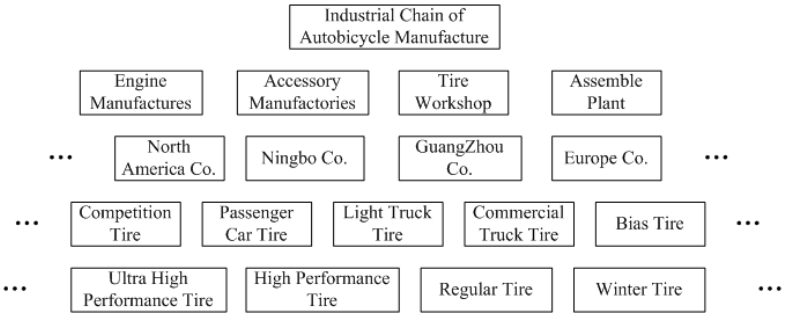


Fig. 4. Industrial Chain of Autobicycle Manufacture

of them is associated to one agent. The interactions are based on the message exchanged between agents.

Within each corporation in the industrial chain, the virtual hierarchical architecture is formed based on P2P fabric such as filiales, workshops *etc.* Then, we can process PPDM layer and layer in the corporation. After that, on the top of each company, the knowledge mined reflects the valuable information of the whole corporation.

Within the hierarchical distributed infrastructure, the mined messages which need be transmitted are encrypted so that the privacy of the individual and the sources are protected. Especially, since it is hierarchical, we can use different kinds of privacy preserving methods and cryptographical primitives [8] in different layers.

Through the cooperation PPDM among the virtual groups, we can not only protect the private information of every filiales, but also offer decision-making for the production and distribution across company. For example, it will predict the demands of the market, offer the plans of production, manage the suppliers and sellers, discover the most important customers and suppliers and control the quality of products. And we can also obtain some valuable knowledge to accelerate the update of industrial structures, technology progresses and innovations, without violating individual privacy. It is extraordinarily valuable in analysis of the development of regional economy. All the enterprise, no matter what their sizes are, can get more knowledge through joining the systems and sharing the information. Besides it, it will be helpful in economics decision making for local government and the development of regional economy. In a word, it will save a lot of expenses and enhance the competitive power of the enterprises.

5 Conclusion and Future Work

In this paper, we present a dynamic hierarchical agent-based infrastructure. As stated above, the infrastructure supplies a large-scale distributed platform

which is suitable for enhancing information processing in loosely connected and constantly evolving environments, where the network will always be in changing, with numerous participants constantly arriving and departing.

With this infrastructure, a large number of homogeneous nodes are organized into a hierarchy, which can satisfy the demands in reality, improve overall system scalability, and also generate fewer messages in the wide area through local communication. In addition, in our infrastructure, upward communication transfers all types of information, including system messages and data messages, and downward communication transfers only system messages to adjust to the topology. This particular communication pattern can save more bandwidth. By adopting asynchrony communication, our infrastructure avoids centralized control and improves the ability in suiting changes. This makes it possible in current commercial network environment. In particular, making use of agent technology, we can overcome a number of limitations in the mechanisms which P2P and Grid support for data management and interchange, and supply the support of security and trustworthiness. This property is especially important in industry, business and scientific applications.

At the present time, we are currently working to deploy PPDM in industrial chain of autobicycle manufacture and hope to report more our findings in the near future. We also aim to apply our network infrastructure for supporting the cooperation information processing between banks and insurances to discover frauds of user profiles but without compromising customer privacy. It is especially paramount in current information society.

Acknowledgements

This paper was supported by the Natural Science Foundation of China (No. 60402010) and the Advanced Research Project sponsored by China Defense Ministry (No. 413150804, 41101010207), and was partially supported by the Aerospace Research Foundation sponsored by China Aerospace Science and Industry Corporation (No. 2003-HT-ZJDX-13).

References

1. I. Foster, C. Kesselman, S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International J. Supercomputer Applications, 15(3), 2001.
2. M. P. Singh. Peer-to-Peer Computing for information systems. In Proc. of AP2PC 2002. July 2002. pp. 15-20.
3. R. Nicholas, K. S. Jennings. A roadmap of agent research and development. Autonomous Agents and Multi-Agent Systems, 1(1),1998. pp. 7-38.
4. Shoham Y. What we talk about when talk about software agents. IEEE Intelligent Systems. 14(2), March-April, 1999. pp. 28-31.

5. M. Cannataro, D. Talia. KNOWLEDGE GRID: An Architecture for Distributed Knowledge Discovery. Communications of the ACM, January 2003 46(1) 89-93.
6. M. Cannataro, D. Talia, P. Trunfio. Distributed data mining on the grid. Future Generation Computer Systems, 2002, Vol. 18(8), 1101-1112.
7. R. Agrawal, R. Srikant. Privacy-preserving data mining. In Proc. of ACM SIGMOD. May 14-19 2000. pp. 439-450.
8. V S. Verykios, E. Bertino, I N. Fovino, L P. Provenza, Y. Saygin, Y. Theodoridis. State-of-the-art in Privacy Preserving Data Mining. SIGMOD Record,33(1),2004. pp. 50.-57.

Server-Assisted Bandwidth Negotiation Mechanism for Parallel Segment Retrieval of Web Objects

Chi-Hung Chi[1,*], Hongguang Wang[1], and William Ku[2]

[1] School of Software, Tsinghua University, Beijing, China 1000080
[2] School of Computing, National University of Singapore, Singapore 117543

Abstract. In this paper, we propose a server-assisted bandwidth negotiation mechanism to address this object downloading problem. Through the negotiation, client and server can agree on the amount of the *best available* bandwidth used for the parallel downloading of a given web object. Both the HTTP extension for the bandwidth negotiation mechanism and its performance estimation are given in this paper. This mechanism not only provides a simple way to maximize bandwidth usage and to speedup web object downloading within the resource availability constraints, but it also allows easy implementation for priority-based quality service for web access.

1 Introduction

The growth of the Internet is accompanied by the provision of richer content of higher quality. This usually translates into ever-growing web object sizes. For example, good quality images and video clips can easily have size in the range of a few to hundred of Mbytes. Software distribution over the network is also in the similar situation, The CD ISO images containing certain GNU/Linux distributions are about 650MB each.

Large web objects present a few problems in their retrieval. Firstly, a larger web object size usually translates into longer retrieval latency. This makes large objects highly inaccessible to all but those connected to the Internet on broadband speeds. Even with a broadband connection, the retrieval of some web objects is still prohibitive mainly because either they are simply too large in size for real-time interactive response or because the content provider (web server) has restricted the bandwidth allocated for the transfer of these objects. Proxy caching is also less likely to be effective because larger objects have lower chance than smaller objects to be cached since size is an important metric in most replacement policies [15].

Nevertheless, users want to achieve the minimum latency for every retrieval, regardless of the object size. This has led to the use of parallel connections (in order to download large objects) which contribute significantly to network congestion and server workload as well as deny other users their fair share of bandwidth allocation. Thus, this form of aggressive client behavior has caused some servers not to support parallel connections partially or at all as much as they would also want to provide maximum throughput to their clients and hence effectively increasing retrieval latency of objects from these servers.

[*] This research is supported by the funding #2004CB719400 of China.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 883 – 890, 2005.

The crux lies in the lack of a bandwidth negotiation mechanism in HTTP which would otherwise allow both clients and servers to negotiate and agree upon a fair amount of bandwidth to be allocated for object retrievals and thereby facilitate the retrieval to be at the "best available bandwidth" transfer rate. This bandwidth negotiation mechanism should also consider the prevailing network congestion level such that subsequent object retrievals do not result in network congestion. As such, having a bandwidth negotiation mechanism would help to reduce the retrieval latency of web objects, and in particular that of large objects. The focus of this paper is to propose such client-server collaborative bandwidth negotiation mechanism for parallel segment retrieval of web objects. This protocol, called the **Server-Assisted Parallel Segment Download** (**SAPAD**), is built on top of an extension to the current HTTP protocol. With SAPAD, network bandwidth can be used more effectively according to the network/server dynamics and the client's priorities, and this results in better response time to requests for large web objects.

2 Related Work

To address the "world-wide-wait" problem, the first approach people use is to deploy proxy caching in the network. With its ability to reuse web data, proxy caching is gaining its popularity in its deployment [11, 15, 25]. It is found, however, that the performance of basic proxy caching is approaching its limit. This is due to the amount of data sharing found in typical web environment. Even for large-scale Internet service providers, the observed cache hit ratio only ranges from about 40% to 60% [3, 14]. The hit ratio drops significantly as the number of clients served by a proxy cache decreases.

Recently, researchers turn their focus to the acceleration of the first time access of web pages. Currently, the two main areas of investigation are server-client connectivity and encoding scheme. For server-client connectivity, it covers path routing optimization, pre-establishment of server-client connection [5], persistent connection [13, 24], and bundle-transfer (which include GETALL / GETLIST [19], MGET [9], and BUNDLE-GET [26]), For encoding schemes, it investigates data compression [6, 18], transcoding [7, 8, 12], and data compaction [4, 26].

Another direction for accelerating web object access is to multiplex HTTP transfers. Here, a client establishes multiple connections (in lieu of a persistent connection) to a server to retrieve objects simultaneously. WebMUX is an experimental multiplexing protocol that allows multiple application-layer sockets to transfer data fragments using a single transport-layer TCP connection [GeN98]. It does not require changes to existing HTTP and web software. WebMUX can allow for the parallel retrieval of web objects via multiple application-level sessions on a single TCP connection, thereby eliminating the need to establish parallel connections to do this. In order to prevent starvation of bandwidth to any session, WebMUX has a priority scheme in which credits are allocated to the various sessions to ensure fair distribution of bandwidth.

The TCP Control Block Interdependence proposal [23] suggested a form of information sharing across TCP connections to the same remote host. Information such as RTT estimation, the MSS, the size of the congestion window size and other

general network traffic conditions can be shared to avoid the TCP congestion controls without compromising the existing network workload. There are two proposed methods to achieve this, namely ensemble and temporal sharing. Ensemble sharing refers to the sharing of state information among two or more active connections. In this way, new connections would learn about the existing network conditions without the need to go through the TCP slow-start stage. In temporal sharing, a new connection will make use of information from a previous connection (parameters from a TCP Control Block) that is already terminated. This will be more useful for a new connection that is connecting to the same remote host as that of the previous (defunct) connection.

Integrated congestion management [1, 2] is another framework based on ensemble sharing as discussed above and it manages network congestion independently of the transport-layer protocols and applications. It provides an API for applications to learn about the network traffic conditions as well as deploy it as a Congestion Manager which would determine transmission rate based on the congestion level. At the TCP layer, the Congestion Manager would decouple congestion control from TCP.

Paraloading, as described in [17, 21, 22], refers to the parallel segmented download (PSD) of an object using parallel connections to multiple mirrors (one connection per mirror). This is different from the convention of retrieving the object solely from one site. The user would first determine the mirrors from which the required object could be downloaded. Then, the user would select the mirrors to which a persistent connection would be established each.

There is an expanded version of PSD for parallel downloading from the same server. The mechanism is basically the simultaneous establishment of multiple connections to a server and retrieving specific segments of an object by using the HTTP Range method. The client would later assemble the segments to render the required object.

For our discussion on PSD, we assume the following:

- The establishment of a connection takes a fixed amount of client, server and network resources.
- The server exercises some form of bandwidth throttling in that every connection has a bandwidth cap. The client is in turn assured of a stable (fixed) transfer rate, up to the bandwidth cap.
- The server would be able to handle many multiple simultaneous connections although this number is finite.
- The server supports the HTTP `Range` method.

The advantages of PSD are:

- The user is likely to experience a shorter retrieval latency for the object, particularly if the object is large. This is after factoring in the associated costs in establishing multiple connections and assembling the segments.
- The retrieval of large objects is made more possible and feasible, because of the shorter retrieval latency. Users who are easily discouraged by the lengthy retrieval latency might now consider the retrieval of these large objects.
- Large objects may improve in their download popularity and as such have a higher chance to be cached. This will in turn lead to even shorter retrieval latency.

The disadvantages of PSD are:

- Having more connections (and hence bandwidth) than other clients would be being unfair to them. It could also mean keeping out some other clients when the server is busy.
- The server's workload is determined by the number of connections that it is handing. PSD increases the server's workload. If the server workload gets too high, the quality of the service that it provides to clients may suffer.

Note that if there is idle bandwidth available, these disadvantages will be important.

3 Server-Assisted Bandwidth Negotiation Mechanism

In this section, we are going to propose our server-assisted parallel segment download mechanism as the mean to accelerate the downloading of large web objects. The key idea behind this mechanism is to perform client-server bandwidth negotiation through an extension of the HTTP protocol.

3.1 Bandwidth Negotiation

Because of the disadvantages of PSD, some servers have completely blocked PSD by not supporting parallel connections and the HTTP `Range` method at all while some others have restricted PSD by providing the clients a maximum fixed number of parallel connections even when they could comfortably handle more workload.

The crux of the arguments against PSD is that there is no mechanism by which the server and the client can negotiate for allocation of bandwidth. The client would like to establish as many connections as possible so long as the benefits outweigh the costs while the server aims to provide as much bandwidth as possible to the client so long as its workload is not affected considerably and that its other clients are not affected. Presently, HTTP does not provide a mechanism for the client and server to negotiate for a compromise. To define a client-server bandwidth negotiation mechanism, the following factors need to be considered:

- *Number of Connections:*
  Based on our assumptions, the number of connections will take up a fixed amount of client, server and network resources. Therefore, the more connections that a client seeks to establish with a server, the more strain it would put on itself, the server and the network. The client can only open a finite number of connections not just because it has finite system resources but also that it is not of any benefit if the allocated bandwidth (via the number of connections) is more than what it can accept. For example, a user on a dialup connection can handle about only four or five parallel connections before suffering a performance drop. The server would also specify the number of connections that it can grant to a client. This parameter would be dependent on a series of factors such as server workload, client's priority and future client demands. Note that, however, current

browsers including Netscape and Microsoft IE allow simultaneous four objects fetching despite of the persistent connection support.

- *Object Size and Segment Size:*
  PSD is essentially effective only for objects of a particular size and above, in short, large objects. There is a size threshold for which the benefits of applying PSD would outweigh its costs. This size threshold is dependent on the number of connections and hence segment size. This is because every parallel connection would have to undergo the TCP slow-start and the segment size would determine the effective throughput rate. The size threshold should also be substantially greater than the typical chunk size for web data transfer. Only when the collective throughput of the parallel connections is higher than that for a transfer on only one connection, would PSD be considered more efficient. As will be discussed in Section 5, a minimum segment size threshold would be about three (3) times the Maximum Segment Size[1]. This determines the minimum size threshold over which PSD would be useful.

- *Server and Network Workload:*
  Depending on the workload, a server may be able to provide more or less bandwidth (in terms of the number of connections). The server must be able to anticipate client traffic and have an understanding of the network traffic, so as to be able to grant the correct amount of bandwidth to each client. The server should not grant a high number of parallel connections when the server workload or/and the network traffic is high or when the server expects a spike in the number of client requests in the near future.

3.2 Server-Assisted PSD (SAPSD)

Only with some form of negotiation mechanism would the costs of PSD be checked and the benefits of PSD realised. This can also be considered as a form of bandwidth negotiation mechanism. We would like to present such a mechanism whereby the client and the server can negotiate and agree on a set of parameters for the purpose of PSD. In this mechanism, the server plays an active role in the negotiation (as compared to that in HTTP content negotiation). As such, we term this mechanism as Server-Assisted PSD (or SAPSD in short).

The goals of this SAPSD proposal are to:

- Increase client throughput using otherwise idle bandwidth, thereby reducing retrieval latency.
- Maximize the cost-effectiveness of the workload on both server and network.
- Allow the server to play a more active role in bandwidth allocation.
- Allow clients to specify some rules for resource retrieval.
- Free clients from the hassle of data management.
- Allow for simple implementations.
- Minimize the amount of per-request and per-response overheads.

[1] Default is 576 bytes as specified in [20].

3.2.1 Basic Mechanism

A client *C* wants to make a request for a resource *R* from a server *S*. *C* sends a TCP connection request to *S*, completes the TCP 3-way handshake and sends its request for *R*. *S* processes this request and determines that *R* is a large object and that *R* can be effectively transmitted using multiple HTTP connections. *S* calculates a good number *n* of multiple connections that it needs to establish with *C,* with regards to server load, server/client preferences and network load.

3.2.1.1 Default SAPSD Responses. *S* proceeds to send a `HTTP 206 Partial Content` response to *C* with the additional information that *S* would establish another *n* - 1 number of connections with *C*. Meanwhile, *S* would send *n* - 1 TCP connection requests to *C*, completes the TCP 3-way handshake and then sends the respective `HTTP 206 Partial Content` responses. Upon the completion of the download of a portion of *R*, *C* or *S* can close the connection involved depending on whether or not persistent connection (`P-HTTP`) is enabled and whether *C* has any other requests for resources on *S*. When *C* has successfully downloaded all the respective portions of *R*, *C* will assemble these portions to render *R*.

In the event that *R* is actually not a large object, *S* would have sent a normal `HTTP 200 OK` response back to *C* accompanied with *R*. However, if *R* is too large to be effectively downloaded even with multiple connections and that *C* has earlier indicated its preference to abort the download in such a scenario, *S* would have sent a `HTTP 2XX` response together with the information that *S* has not sent *R* in accordance with the client's preference and in the best interests of both parties.

3.2.1.2 Other Considerations. In order for *S* to be able to establish connections with *C*, *C* should provide in its request for *R*, certain information such as a list of port numbers that *S* can use to send its TCP connection requests to. *C* also has to indicate the number of multiple connections that it is willing and able to support as well as the relative resource size above which SAPSD is to be performed and another resource size above which the download process is to be terminated. For clients behind a firewall, they would have to establish the additional connections (if applicable) themselves since the server would be blocked by the firewall. This is however outside the scope of our discussion here.

3.2.2 Specifications

We propose some modifications to the present HTTP specifications to implement SAPSD. This involves changes at both client and server sides. The use of the terms client and server could apply to intermediate entities such as proxy, in context. We would first look at the proposed client specifics followed by the server specifics.

3.2.2.1 Client Specifics. SAPSD requires a client to specify the number of connections that he is willing and able to support as well as to provide input on the expected resource size before using SAPSD to download the resource. Following current implementation practice, the client is also expected to provide a list of port numbers for the server to establish connections with. Given that SAPSD is used only when the requested resource is large, we can represent this in a `conditional-if` header in the client's request.

3.2.2.2 Server Specifics. After receiving and processing the client's request, the server can determine whether or not to use SAPSD on the requested object and then respond accordingly.

Case 1: Resource is smaller than `large-floor` (The field `large-floor` refers to the threshold over which the client would want to perform SAPSD.)

If the resource is smaller than the client-specified value of `large-floor`, the server will reply with a `HTTP 200 OK` response accompanied with the requested resource. It is optional for the server to include information such as the number of connections that the server could allocate to the client based on the prevailing conditions as well as the value of server-specified `large-floor`.

Case 2: Resource is larger than `large-ceiling` (The field `large-ceiling` refers to the threshold over which the client would want to abort the download as the client has perceived that resources of sizes above this threshold are too big to be downloaded within a time interval acceptable to the client.)

If the resource is larger than the client-specified value of `large-ceiling`, the server could reply with a `HTTP 412 Precondition Failed` response. However, this might lead to confusion when other conditional request headers are used in the client's request. We could use a `HTTP 4XX Requested Entity Too Large` response similar to that of the `HTTP 413 Request Entity Too Large` response or more likely, reuse the `HTTP 413 Request Entity Too Large` response as there should be no ambiguities as the `HTTP 413 Request Entity Too Large response` is presently defined as a reaction to the `PUT` method.

4 Conclusions

In this paper, we examined the traffic characteristics of some NLANR IRCACHE dataset to provide the definition of a large web object in today's context and factors that are undermining the retrieval of large objects. The lack of a bandwidth negotiation mechanism in HTTP is stated as an important factor to as why large objects are not being retrieved efficiently. This led to our proposal of such a mechanism in the form of Server-Assisted Parallel Segmented Download (SAPSD). We then outline the formal specifications of the SAPSD and provide a throughput estimate of this mechanism. Building as an extension of the HTTP protocol, SAPSD not only provides a simple way to maximize bandwidth usage and to speedup web object downloading within the resource availability constraints, but it also allows easy implementation for priority-based quality service for web access.

References

[1] Bakakrishnan, H., Rahul, H.S., Seshan, S., "An Integrated Congestion Management Architecture for Internet Hosts," *Proceedings of ACM SIGCOMM Conference*, Cambridge, MA, September 1999.

[2] Balakrishnan, H., Seshan, S., "The Congestion Manager," *RFC3124*, June 2001.

[3] Breslau, L., Cao, P., Fan, L., Phillips, G., Shenker, S., "Web Caching and Zipf-Like Distributions: Evidence and Implications", *Proceedings of IEEE INFOCOMM'99,* April 1999.
[4] Chen, M.C., Woo, T.Y.C., "Cache-based Compaction: A New Technique for Optimizing Web Transfer," *Proceedings of IEEE INFOCOMM*, 1999, page 117-125.
[5] Cohen, E., Kaplan, H., "Prefetching the Means for Document Transfer: A New Approach for Reducing Web Latency," *Proceedings of IEEE INFOCOMM,* 2000.
[6] Expand Networks, http://www.expand.com
[7] Fox, A., Brewer, E.A., "Reducing WWW Latency and Bandwidth Requirements via Real-Time Distillation," *Proceedings of the 5th International World Wide Web Conference (WWW-5),* May 1996.
[8] Fox, A., Gribble, S.D., Chawathe, Y., Brewer, E.A., "Adapting to Network and Client Variation Using Active Proxies: Lessons and Perspectives," *IEEE Personal Communications,* August 1998.
[9] Franks, J., "Proposal for an HTTP MGET Method,"http://ftp.ics.uci.edu/pub/ietf/http/hypermail/1994q4/0260.html
[10] Gettys, J., Nielsen, H.F., "The WebMUX Protocol," *Expired Internet Draft*, August 1998. http://www.w3.org/Protocols/MUX/WD-mux-980722.html
[11] Glassman, S., "A Caching Relay for the World-Wide Web", *Proceedings of the 1st International World-Wide Web Conference,* May 1994, pp. 69-76.
[12] Han, R., Bhagwat, P., LaMaire, R., Mummert, T., Perret, V., Rubas, J., "Dynamic Adaptation in an Image Transcoding Proxy for Mobile Web Browsing, " *IEEE Personal Communications,* December 1998, pp. 8-17.
[13] Hypertext Transfer Protocol – HTTP/1.1, http://www.w3.org/Protocols/rfc2616/rfc2616.html
[14] IRCACHE daily report, http://www.ircache.net/Statistics/Summaries/Root/
[15] Lorenzetti, P., Rizzo, L., "Replacement Policies for a Proxy Cache," *IEEE/ACM Transactions on Networking,* Volume 8, Issue 2, April 2000.
[16] Luotonen, A., Altis, K., "World Wide Web Proxies," *Computer Networks and ISDN Systems. Also in Proceedings of the 1st International Conference on WWW,* April 1994.
[17] Miu, A., Shih, E., "Performance Analysis of a Dynamic Parallel Downloading Scheme from Mirror Sites Throughout the Internet," *Term Paper, LCS MIT*, December 1999.
[18] Packeteer, http://www.packeteer.com
[19] Pasmanabhan, V.N., Mogul, J.C., "Improving HTTP Latency," *Computer Networks and ISDN Systems*, 28(1/2):25-35, December 1995.
[20] Postel, J., "The TCP maximum Segment Size and Related Topic," *RFC879*, November 1983.
[21] Rodriguez, P., Biersack, E.W., "Dynamic Parallel-Access to Replicated Content in the Internet," IEEE/ACM Transactions on Networking, August 2002.
[22] Rodriguez, P., Kirpal, A., Biersack, E.W., "Parallel-Access for Mirror Sites in the Internet," *Proceedings of IEEE INFOCOM 2000 Conference*, March 2000.
[23] Touch, J., "TCP Control Block Interdependence," *RFC2140*, April 1997.
[24] Wang, Z., Cao, P., "Persistent Connections Behavior of Popular Browsers," http://www.cs.wisc.edu/~cao/papers/persistent-connection.html
[25] Wessels, D., *Web Caching*, O'reilly & Associates Publishing, June 2001.
[26] Wills, C.E., Mikhailov, M., Shang, H., "N for the Price of 1: Bundling Web Objects for More Efficient Content Delivery," *Proceedings of the 10th International World Wide Web Conference*, Hong Kong, May 2001.

Multiple Schema Based XML Indexing

Lu Yan[1] and Zhang Liang[2]

[1] College of Information Science and Engineering,
Shandong University of Science and Technology, Qingdao 266510, China
Lu_yan_75@hotmail.com
[2] Department of Computing and Information Technology,
Fudan University, Shanghai 200433, China
Zhangl@fudan.edu.cn

Abstract. With the prevalence of XML, XML query processing becomes a hot topic. Most techniques available now are generic to XML documents that are completely schema-less or adhere to only one DTD. In this paper, we propose MXI, a XML indexing method that supports efficient path query of XML documents in both single- and multi-DTD settings. The main feature of MXI is to take advantage of information embedded in DTD for speeding up the process of XML path query. A path expression with one predicate restriction needs only zero or two structural join operations per XML document. For a path expression that is not complying with any paths in XML documents to be searched, MXI can give a judgment of no answer in much shorter time than those of indexing methods available now. We detail our techniques and key algorithms and demonstrate the superiority of MXI in path query efficiency over Lore, XISS and SphinX via experimental results.

1 Introduction

In order to efficiently retrieving the results to XML path queries, corresponding XML indexing techniques have been developed [1-8]. Some of them are generic to XML documents that are completely schema-less although as we know that information embedded in DTD can make great help in speeding up the implementation of XML path query. Some treat XML documents adhere to only single DTD. With the popular of XML, it is sure that future applications have to process XML documents that adhere not only to single DTD but also to multiple DTDs.

In this paper, we proposes MXI, a XML indexing method that support efficient query of XML documents in both single- and multi-DTD settings. The main feature of MXI is to take advantage of information embedded in DTD for speeding up the process of XML path query. MXI adopts a new coding scheme, which enables each element in XML documents carrying corresponding DTD structural information. Indices are constructed on both XML documents and DTDs. Main contributions of MXI are:

1) For path query with N elements/attributes and a predicate restriction, MXI reduces time-consuming structural join operations from at least (N-1) times per XML document in XISS to 0 or 2 times in MXI;

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 891 – 900, 2005.

2) For a path query that is not complying with any paths in XML documents, MXI can give a judgment of no answer in much shorter time than that of indexing methods in existence.
3) Experiments over a set of XML documents indicate that MXI can process path query faster than Lore, XISS and SphinX do.

2 Related Works

Many indexing structures for semi-structured data and XML documents have been developed in recent years. Dataguides[1] in Lore is a concise and accurate summary of all paths in the database that start from the root. Every label path (the string formed by the concatenation of the labels of the edges) in the database is described exactly once in the Dataguides. Dataguides reduces the portion of the database to be scanned for path queries and is useful for navigating the hierarchy of XML documents. XISS[2] is an efficient XML indexing and querying system designed to support regular path expressions. Multiple indices including element index, attribute index, structure index, as well as name and value indices have been used in this approach. XISS needs no traverse of the hierarchy of XML tree. Main idea behind XISS is to decompose the query into several simple path expressions. With the help of structural join algorithms, each simple path expression produces an intermediate result that can be used in the subsequent stage of processing. For XISS, there are two points that can be improved: First, XISS gets all element/attribute nodes appeared in query path in the XML index to participate in process of path join although some of them are irrelevant to the query. For example, for query */Buyer/Title*, all *Title* nodes including nodes in path */Book/Title*, etc will also be get out; Second, if there are N elements/attributes in query path, XISS will run structural join algorithm (N-1) times at least per XML document and as we know structural join is a time consuming process especially when the size of XML documents is large. Structural join algorithms are improved according to paper[3][4], but no work was done to reduce times for structural join operations needed per XML documents. Index Fabric[5] is an indexing structure that provides efficiency and flexibility using Patricia trees. Index Fabric considered raw paths, which are conceptually similar to dataguides in Lore, and refined paths, which support queries. ToXin[6][7] is another indexing system that supports navigation of the XML graph both backward and forward to answer regular path query. It consists of two different types of structures: value index and path index that summarizes all paths in the database.

All indexing techniques listed above focus on XML documents that are completely *schema-less*. SphinX[8] is the first XML indexing system that utilizes information of DTD to speed up the processing of path queries. In SphinX, both document graph and schema graph are created. Yet after locating nodes that satisfy the predicate restriction given in path query, SphinX does not make use of information of DTD any longer but adopts time-consuming bottom-up or top-down XML documents traversing to find answer nodes to the query. Further work can be done to avoid it to improve the efficiency of path query.

Motivated by indexing methods listed above, MXI improves XML indexing further by: 1) Locating precisely nodes that participate the structural join algorithm;

2) Reducing the number of times for structural join operations needed per XML document; 3) Eliminating an ineffective query (there are no documents matching the query structure) without operating on XML documents.

3 Key Techniques in MXI

XML documents and DTDs are modeled as unordered tree structures where nodes represent elements and attributes, and parent-child node pairs represent nesting between XML data components. A new coding scheme is adopted in MXI. Indices are constructed both on DTDs and XML documents respectively. Evaluation of path query is discussed and two main structural join algorithms are offered.

3.1 Coding Scheme

Coding DTD: There is a virtual root node *RootNode* for the set of DTDs to be managed. Every DTD tree is added to *RootNode* as a subtree. *RootNode* is labeled with a quadrumvirate (0, *bound*, 0, -1) where *bound* represents the preorder number of *RootNode*'s most right leaf node.When a new DTD tree $tree_{new}$ is added to *RootNode* as a subtree, each node n in $tree_{new}$ is labeled with a quadrumvirate (*preorder(n)+ bound, postorder(n)+ bound, level(n), element/attribute*), where *preorder(n)* and *postorder(n)* are the preorder number and the postorder number of node n in $tree_{new}$, respectively, *level(n)* is the level number of node n(root node of $tree_{new}$ is at level 1), *element/attribute* flag indicates if node n is an element node or an attribute node (in MXI, "1" represents element node and "0" represents attribute node). Then, *bound* is updated with *bound* +number of nodes of $tree_{new}$.When a DTD tree $tree_{del}$ is deleted from the DTD tree, the coding of the other DTDs in the DTD tree will not be changed.

Lemma 1[9]. For two nodes n_1 and n_2 in the DTD tree, n_1 is an ancestor node of n_2 if and only if preorder(n_1)<preorder(n_2) and postorder(n_1)>postorder(n_2).

Proposition 1. For two nodes n_1 and n_2 in the DTD tree, if preorder(n_1)<preorder(n_2) and postorder(n_1)>postorder(n_2) and level(n_2)-level(n_1)=1 then n_1 is parent node of n_2.

Coding XML: A XML document is mapped to a tree and node n in a XML tree is labeled with a quadrumvirate (*doc_id(n), dtd_preorder(n), preorder(n), postorder(n)*), where *doc_id(n)* is the ID of the XML document that node n is in; *dtd_preorder(n)* is the preorder number of corresponding node n' in the DTD tree[1]; *preorder(n)* and *postorder(n)* are the preorder number and postorder number of node n in XML tree, respectively.

XML coding scheme enables each XML element/attribute node carrying corresponding DTD structural information through item of *dtd_preorder(n)*. This guarantees precise locating of nodes that participate the structural join operation. For example, all *Title* nodes in path */Book/Title*, etc absolutely will not be get out to take part in the join for path */Buyer/Title*.

[1] Node n and node n' have the same name and at the same position of the same path except the *RootNode*.

3.2 Index Organization

DTD index is implemented as an inverted list. Each element/attribute name record points to list of element/attribute nodes having the same name and different preorder number in the DTD tree.

XML index is implemented as a B^+-tree using preorder number of nodes in the DTD tree as keys. Each entry in a leaf node points to a set of records of XML element/attribute nodes corresponding to the same element/attribute node in the DTD tree, grouped by XML document they belong to.

Furthermore, each XML document is assigned a unique document ID. For the sake of future update, a table registering DTD and its corresponding XML documents IDs and a table registering DTD and its position in the DTD tree are also kept in MXI.

3.3 Evaluation of Path Query

For MXI, a path query with one predicate restriction is implemented in two steps: path matching with the DTD tree and operations on XML documents. The latter is composed of 6 segments. Two main structural join algorithms are used.

Path matching with the DTD tree. First we should determine if there is a matching between query path and the DTD tree. If there is a matching structure of $path_{match}$ in the DTD tree, we call the path with a predicate restriction in $path_{match}$ the condition-matching path and path with node that user want to get in $path_{match}$ the object-matching path. Suppose that $node_{con}$, $node_{obj}$ and $node_{bran}$ are leaf node of condition-matching path, object node of object-matching path and branch node of condition-matching path and object-matching path, we can be sure that $node_{bran}$ is a ancestor node of $node_{con}$. Preorder numbers of $node_{con}$, $node_{obj}$ and $node_{bran}$, which are $preorder_{con}$, $preorder_{obj}$ and $preorder_{bran}$, respectively, are returned and step **2** will be continued; If there is no matching path with the DTD tree, there is no matching path with any XML documents that conforming to the DTDs in the DTD tree by all means and warning of "No Path Matching and no answer! " is returned to users.

Although the native XML documents may be highly complex and large in size, their associated DTDs are very compactly expressed. For example, the DTD for the Shakespeare's Play is only 2KB in size. This results in : 1) Path matching between the DTD tree and query path can be implemented in a very short time (MXI adopts matching method used in XISS which decomposing complex path expression to some simple path expressions. Lemma 1 and Proposition 1 is applied to judge if each of them is a valid simple path in the DTD tree); 2) An ineffective path query can be quickly figured out without processing on XML documents. For if there is no matching between query path and the DTD tree, there is surely no matching between query path and any XML documents that are to be searched. There are no other indexing methods in existence having this capability.

Operations on XML Documents. If there is only one path in the path expression (For example, */a/b/c[d="value_1"]*) and we get a $preorder_{obj}$ for the preorder number of leaf node of object-matching path in the DTD tree, answers to user's path query are all the records that key of $preorder_{obj}$ in B^+-tree pointing to in XML index.If there are two paths in the path expression (For example, path expression */a/b[c="value_1"]/d*

has two paths of */a/b[c="value_1"]/* and */a/b/d* in it) and we have got $preorder_{con}$, $preorder_{obj}$ and $preorder_{bran}$, operations on XML documents are as follows:

1. Find all XML element nodes $\{CA_1,\ldots,CA_n\}$ in XML index. $CA_i(1\le i\le n)$ is an element/attribute node set. All nodes in CA_i are of the same name, correspond to the same element/attribute node whose preorder is $preorder_{con}$ in the DTD tree, belong to the same XML document and are sorted by item of *preorder*. $\{CA_1,\ldots,CA_n\}$ is grouped by documents ID.
2. Chose $\{C_1,\ldots,C_m\}(m\le n)$ from $\{CA_1,\ldots,CA_n\}$ where value of every element/attribute node in $C_i(1\le i\le m)$ satisfies the predicate restriction given in path query. $\{C_1,\ldots,C_m\}$ is grouped by document ID. If $\{C_1,\ldots,C_m\}$ is null then there is no answer to the path query and operation is over, else go to next step.
3. According to the XML document ID got above, find all XML element nodes $\{B_1,\ldots,B_m\}$ in XML index. $B_i(1\le i\le m)$ is an element node set. All nodes in B_i are of the same name, correspond to the same element node whose preorder number is $preorder_{bran}$ in the DTD tree, belong to the same XML document and are sorted by *preorder*. $\{B_1,\ldots,B_m\}$ is grouped by documents ID.
4. According to the XML document ID got above, find all XML element/attribute nodes $\{O_1,\ldots,O_m\}$ in XML index. $O_i(1\le i\le m)$ is an element/attribute node set. All nodes in O_i are of the same name, correspond to the same element/attribute node whose preorder is $preorder_{obj}$ in the DTD tree, belong to the same XML document and are sorted by *preorder*. $\{O_1,\ldots,O_m\}$ is grouped by documents ID.
5. For each pair of $\{C_i\}$ and $\{B_i\}$ having same XML document ID, run path join algorithm ***CB-Join*** to get $\{B\_elect_i\}$. $\{B\_elect_i\}$ is a subset of $\{B_i\}$ and nodes in $\{B\_elect_i\}$ are sorted by *preorder*. Each node in $\{B\_elect_i\}$ is ancestor node of a certain node in $\{C_i\}$.
6. For each pair of $\{B\_elect_i\}$ and $\{O_i\}$ having the same XML document ID, run path join algorithm ***BO-Join*** to get $\{O\_elect_i\}$. $\{O\_elect_i\}$ is a subset of $\{O_i\}$ and nodes in $\{O\_elect_i\}$ are sorted by *preorder*. Each node in $\{O\_elect_i\}$ is descendant node of a certain node in $\{B\_elect_i\}$.

   All nodes in $\{O\_elect_i\}$ are answer nodes to the path query.

Algorithm CB-Join

Input: {C}, a set of element/attribute nodes corresponding to the same element/attribute node of C' in the DTD tree and sorted by *preorder*, and
{B}, a set of element nodes corresponding to the same element node of B' in the DTD tree and sorted by *preorder*. // B' is an ancestor node of C' in the DTD tree;
// Nodes in {C} and {B} belong to the same XML document;

Output: {B_elect}, a subset of {B} and each node in {B_elect} is ancestor node of a certain node in {C}.

1. {B_elect}={}; i=1; j=1;
2. While i≤|C| do // |C| denotes number of nodes in {C}
   While j≤|B| and preorder(B_j)<preorder(C_i) do // |B| denotes number of nodes in {B}
   If postorder(B_j)>postorder(C_i) Then //in this case, B_j is ancestor node of C_i
   {B_elect}= {B_elect}□{B_j};
   EndIf

```
        j=j+1;
      EndWhile
      i=i+1;
    EndWhile
3. Output {B_elect}.
```

Theorem 1. Time complexity of algorithm *CB-Join* is O (|B|+|C|).

Algorithm BO-Join

Input: {B}, a set of element nodes corresponding to the same element node of B' in the DTD tree and sorted by *preorder*, and
{O}, a set of element/attribute nodes corresponding to the same element/attribute node of O' in the DTD tree and sorted by *preorder*.
// Nodes in {B} and {O} belong to the same XML document;
// B' is an ancestor node of O' in the DTD tree;

Output: {O_elect}, subset of {O} and each node in {O_elect} is descendant node of a certain node in {B};

```
1. {O_elect}={}; i=1; j=1;
2. While i≤ |B| do    // |B| denotes number of nodes in {B}
     While j≤|O| and postorder(B_i)>postorder(O_j) do
       // |O| denotes number of nodes in {O}
       If preorder(B_i)<preorder(O_j) Then    //in this cases, B_i is ancestor node of O_j
          {O_elect}={O_elect}∪{O_j};
       EndIf
       j=j+1;
     EndWhile
     i=i+1;
   EndWhile
3. Output {O_elect}.
```

Theorem 2. Time complexity of algorithm BO-Join is O (|B|+|O|).

Commonly, there are more than one predicate restrictions in a path query. For a path with *m* predicate restrictions, we decompose the path to *m* segments. For example, path */a/[b="value_1"]/c/d/e[f>"value_2"]/g/h[i<"value_3"]/k* will be decomposed to */a/[b="value_1"]/c/d/e*, */e[f>"value_2"]/g/h* and */h[i<"value_3"]/k*. Then each of them will be evaluated and the answer nodes will be implied by the evaluation of next one.

4 Experiments and Remarks

Prototype of MXI was implemented with VC++. Experiments for MXI, Lore[2] and XISS were conducted on an Intel 800MHz Pentium III workstation with 256 Mbytes of main memory running Windows 2000(for MXI) or RH 7.3(for Lore and XISS). Since we cannot get source code of SphinX, we have sent data sets and testing queries to developers of SphinX. They helped us to test them and returned the results of index size, index creation time and response times for queries on single-DTD data set done on SphinX. It is said that experiments were done on an Intel 800 MHz Pentium III workstation with 512 Mbytes of main memory.

[2] http://www-db.stanford.ed/lore/release/data.html

4.1 Data Set and XML Queries

We have chosen two main data sets in common use for experiments: Shakespeare's Play[3] whose documents size is 7.9Mb and size of DTD is 0.002Mb; DBLP[4] whose documents size is 122.0Mb and size of DTD is 0.008Mb.

Table 1. Sample of XML Queries

Data Set	Queries Example	Query#
Shakespeare's Play	/PLAY/TITLE	Q1
	/PLAY/PERSONAE	Q2
	[PGROUP/PERSONA="AMIENS"]/TITLE //PERSONAE	Q3
	[PGROUP/PERSONA="AMIENS"]/TITLE //PERSONAE	Q4
	[PGROUP/PERSONA="AMIENS"]/PLAY	
DBLP	/dblp/masterstheses/title	Q5
	/dblp/masterstheses [author="Kurt P. Brown"]/title	Q6
	//masterstheses [author="Kurt P. Brown"]/title	Q7
	/dblp/masterstheses [year="2002"]/book	Q8
Multi-DTD based data sets including Shakespeare's Play, etc	PROLOGUE/SUBTITLE	Q9
	ACT//SPEEACH	Q10
	SPEECH//ACT	Q11

Four kinds of XML path queries are chosen in the experiment: simple queries without predicates restriction (Q1 and Q5 in table 1), simple queries with predicates restriction (Q2 and Q6 in table 1), general queries (Q3 and Q7 in table 1) and ineffective queries (Q4 and Q8 in table 1). In a simple query, the user fully specifies the structure of the data that is to be searched. In a general search, the application specifies the structure of the data only partially. An ineffective path query has no path matching with any of the XML documents to be searched. Since XISS processes only structural joins, so Q9, Q10 and Q11 are designed to evaluate the performance of it.

4.2 Index Construction Performance

We offer two index construction metrics: index size and index creation time to evaluate the space occupied by the index structure and the time taken to build the index structure. Here we should point out that index of MXI, Lore and XISS are accomplished in native experimental environment while index of SphinX accomplished by its developers. Indices of Lore constructed here include Dataguides, Bindex and Lindex. Detail index construction performance can be referred at table 2 and table 3.

[3] http://www.ibiblio.org/bosak/xml/eg/
[4] http://www.informatik.uni-trier.de/~ley/db

Generally size of DTD is small, so compared to index size and index creation time of XML documents in MXI, index size of DTD and index creation time are small and make little effect on the whole index size and index creation time in MXI.

From table 2 and table 3 we can see that index size and creation time of MXI are similar to that of XISS, bigger than that of SphinX and smaller than that of Lore.

Table 2. Index Size

Data Set	MXI	Lore	XISS	SphinX
Shakespeare Play	27.3M	49.5M	24.1M	16.1M
DBLP	481.7M	+		275.3M

+note: Cannot be conducted due to the memory limitation

Table 3. Index Creation Time

Data Set	MXI	Lore	XISS	SphinX
Shakespeare Play	58s	399s	43s	15s
DBLP	762s	+		320s

+note: Cannot be conducted due to the memory limitation

4.3 Query Performance

Fig.1 summarizes the performance of MXI, Lore, SphinX and XISS.

The edition of Lore system we downloaded does not cope with multi-DTD based data set. Developers of SphinX said that SphinX can cope with this situation but they didn't return response time of path query on multi-DTD based data sets. From Fig.1 (a)(b) we can see that MXI outperforms Lore and SphinX on single-DTD based XML data set, especially when query path have no matching with the DTD tree, in other word, have no path matching with any XML documents. This is due to the existence of DTD index in MXI and MXI has a pretreatment of structure matching between query path and the DTD tree before querying on XML index while Lore and SphinX have not.

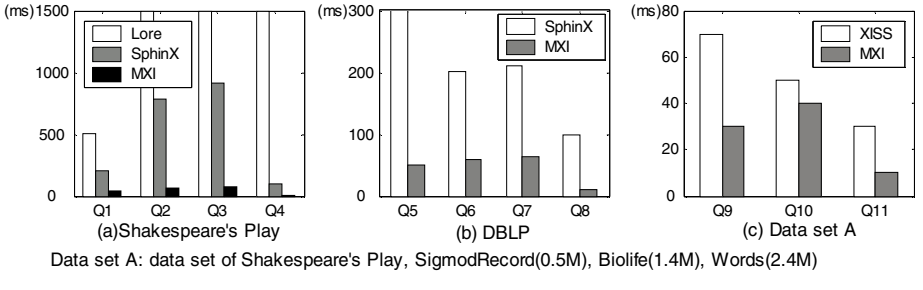


Fig. 1. Performances of MXI, Lore, SphinX and XISS

Prototype of XISS is not an integrated XML query system like MXI, Lore or SphinX, but designed for process of structural joins. So for XISS we cannot give a precise query response time for a whole path query. In order to get a comparison between the performance of MXI and XISS, we write a module in MXI specially to achieve element-element join function. Experiment is done on data sets A[5] (Fig.1(c)). We can see that MXI also outperforms XISS. That is because XISS needs to get all element/attribute nodes of the query in its XML element index and need a join process over two sets of XML element nodes while MXI does its join process on DTD index which is much smaller than XML index and needs to get only one XML element/attribute nodes to get the join results.

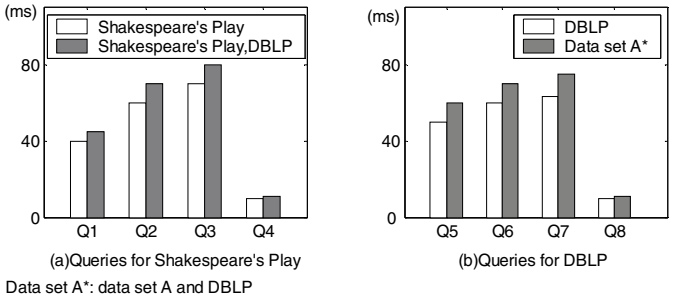


Fig. 2. Performance of same query on different XML documents in MXI

Fig.2 demonstrates the performance of same query on different XML documents that adhere to single DTD and multiple DTDs, respectively. From Fig.2 we can see that response time of the same path query on single-DTD based XML data set is of little difference to that on multiple-DTD based XML data set. This is also in virtue of the DTD index and the coding method in MXI. They provide mechanisms for determining the DTD that is relevant to the query. After $preorder_{con}$, $preorder_{obj}$ and $preorder_{bran}$ of matching path are get, XML nodes that are to take part in the structural join operations can be quickly located (Time complexity is $O(\log n)$, where n is the number of nodes in the DTD tree.

5 Conclusion and Future Work

This paper proposed a new XML indexing method---MXI for efficiently processing path query of XML data in both single- and multi-DTD settings. MXI takes the advantage of information embedded in DTD to speed up XML path query. It adopts a new numbering scheme to index both DTDs and XML documents. With MXI, a path expression with N elements/attributes and a predicate restriction needs only 0 or 2 structural join operations to evaluate per XML document. For an ineffective path

[5] Sigmod: http://www.acm.org/sigmod/record/xml/
Biolife: ftp://202.116.13.5/common/bcb6/INSTALL/Common/Borland Shared/Data/
Words: ftp://162.105.91.5/Technical Books/G R E/词汇大傻/taisha-xml/

query that is not complying with any paths in XML documents, MXI can give a judgment of no answer in much shorter time than that of indexing methods in existence. Experimental results demonstrate that MXI can process path query faster than Lore, SphinX and XISS do.

Future work will be concentrated on the expanding of MXI in settings of cyclic DTD and expanding path query to tree query. Work will also be done to replace files with RDB to store DTD and XML data.

References

1. R.Goldman and J.Widom. DataGuides: Enabling Query Formulation and Optimization in Semistructured Databases. In Proceedings of the International Conference on VLDB(1997) 436-445
2. Q.Li, B.Moon. Indexing and Querying XML Data for Regular Path Expressions, Proceedings of 27th International Conference on VLDB(2001) 361-370
3. Shu-Yao Chien, Z.vagena, D.Zhang, Efficient Structural Joins on Indexed XML Documents, Proceedings of the 28th VLDB Conference, Hong Kong(2002)263-274
4. S.Al-Khalifa, H.V.Jagadish, N.Koudas, J.M.Patel, D.Srivastava and Y.Wu. Structural Joins: A Primitive for Efficient XML Query Pattern Matching. Proceedings of ICDE(2002)
5. B.Cooper, N.Sample, M.Franklin, G.Hjaltason and M.Shadmon. A Fast Index for Semistructured Data. Proc. Of 27th Intl. Conf. On VLDB(2001)341-350
6. D.Barbosa, A.Barta, A.O.Mendelzon, G.A.Mihaila, F.Rizzolo, and P.Rodriguez-Guianolli. Tox-The Toronto XML Engine. In Proceedings of the Workshop on Information Integration on the Web(2001) 66-73
7. F.Rizzolo, A.O.Mendelzon. Indexing XML Data with ToXin. In proceedings of Fourth International Workshop on the Web and Databases(2001)
8. L.K.Poola, J.R.Haritsa. SphinX: Schema-conscious XML Indexing, Technical report. TR-2001-04, DSL/SERC(2001)
9. Paul F. Dietz. Maintaining Order in a Linked List. In Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing, California(1982)122-127

A Linked-List Data Structure for Advance Reservation Admission Control*

Qing Xiong[1], Chanle Wu[2], Jianbing Xing[1,3], Libing Wu[1], and Huyin Zhang[1]

[1] School of Computer, Wuhan University, Wuhan 430079, China
[2] National Engineering Research Center of Multimedia Software, Wuhan 430072, China
[3] The State Key Laboratory of Software Engineering, Wuhan 430072, China
netlab@whu.edu.cn

Abstract. With the development of multimedia and grid technologies, more and more distributed applications require guaranteed quality of service and maintain minimum network resource during their running sessions. Thus advance reservation is necessary because it provides a solution for the need of reserving network resources for future use. Many kinds of data structures were proposed to perform fast and efficient admission control. Most of them are based on the time-slotted method, which needs to make an appropriate tradeoff between the efficiency and the granularity of the time slots. In this paper, a linked-list data structure is proposed to perform the admission control for advance reservation. Compared with the existing bandwidth tree and time-slotted array, the proposed linked list shows better performance.

1 Introduction

Initially, the Internet was designed to support best-effort service for applications, i.e., the resources are not reserved and are provided only when they are actually available. As the development of multimedia and grid technology, especially the spread of network applications and high performance distributed applications, urgent demands are raised for high availability of network and better QoS, which leads to resource reservation. Two kinds of bandwidth reservations must be distinguished: immediate reservations and advance reservations. In contrast to the former, where the network reservations are established immediately after the request is admitted, advance reservations allow to specify and request a given QoS for a transmission a long time before the actual transmission has to be made. Advance reservation has several practical advantages. It increases the probability for call acceptance. It does not require over-provisioning of network resource. It moves control and responsibility away from the network and towards the user. It allows the network to better plan its resources taking advantage the knowledge of future calls [1, 2, 3].

Recently, bandwidth brokers are used as a managing system in differentiated services (DiffServ) networks. A single broker is only responsible for managing a certain part of a network (domain). It processes reservation requests, grants or denies access to the network. This procedure is called admission control. Reservation

* Supported by the fund of State Key Lab of Software Engineering and National High Technology Development 863 Program of China under Grant (No. 2003AA001032).

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 901 – 910, 2005.

requests have to be checked whether there are sufficient resources available for the duration of the reservation. In order to perform the admission control efficiently, a suitable data structure is necessary for the bandwidth broker. It stores relevant attributes (e.g. starting time, ending time, book-ahead time, bandwidth and duration time/interval) for each reservation request. Such a data structure must be concise, precise, easy-to-implement and efficient. The candidate data structure must satisfy the following requirements: (1) as fast as possible, i.e. to minimize the response time of a single reservation request; (2) as low-cost as possible, i.e. to minimize the memory consumed during the whole admission control phase; (3) as adaptive as possible, i.e. it should be able to adapt to the dynamically changing book-ahead time and the variable duration of the reservations.

In this paper some related works and several existing data structures were introduced first. Among them two typical data structures (a specially designed bandwidth tree [9] and a time-slotted array [8]) were examined in a number of different scenarios with respect to the issues previously described, i.e. admission time, memory consumption and adaptability to different book-ahead time and duration time of the reservation. Then a new data structure based on linked list was proposed and employed to perform admission control for advance reservation in bandwidth broker. All the three data structures were implemented by C language and some simulations are presented to evaluate their performance (mainly time and space complexity). The simulations show that the linked list is superior to the slotted array and bandwidth tree concerning both the memory requirement and the admission time. Finally, some conclusions were made and the future work was introduced.

2 Related Works

At present there are many admission control mechanisms as well as relevant data structures and corresponding algorithms. All the data structures can be divided into two types: the one based on slotted time and the one based on continuous time. The former divides time into slots, and each slot represents a fixed time interval. A time slot is the minimal unit for resource allocation. It may represents 1 second, 5 minutes, 2 hours or even longer. Under this condition, the value of all the time parameters specified in the reservation request (e.g. starting time, ending time) must be divided exactly by the value of the defined time slot. On the contrary, the method based on continuous time allows the reservations start or end at any time.

In [6] a segment tree based on slotted time was proposed to perform admission control. Each node contains a time frame (duration) and the amount of reserved bandwidth during that time frame. Each time frame is recursively divided into smaller, equally sized time frames, thus each leaf is equal to one time slot. Each node stores the aggregate bandwidth of all reservations spanning over the whole time frame represented by the nodes and the maximum sum of node values in any of the branches below current node. In [7] a binary search tree was provided in which each reservation is represented by two nodes, one for the starting time and the other for the ending time. The node at the starting time contains a positive bandwidth delta while the node at the ending time contains a corresponding negative bandwidth delta. The evaluation results in [7] show that the segment tree is better than the binary search

tree. In [8] a slotted array was presented. Each element of the array represents a time slot and stores the accumulated bandwidth allocated for it. Comparing with the segment tree, the slotted array shows much better performance.

Furthermore, there are some data structures based on continuous time. In [9] a bandwidth tree was proposed. By using the bandwidth tree, the starting time and the ending time of a reservation request can be defined as any time without dividing the whole time interval into equally sized slots. In [4] the researchers point out that dividing time into slots may result in decreasing the utilization of bandwidth. In order to solve this problem, a malleable and a flexible bandwidth reservation mechanism were proposed to improve the utilization of bandwidth in [4] and [5] respectively.

3 Data Structures

3.1 Admission Control

An admission control agent should maintain a suitable data structure storing the resource reservation information in the managed domain. When a request arrives, the agent checks the parameters and determines whether there are sufficient resources to be reserved and allocated. Each bandwidth reservation request can be described with a series of parameters, e.g. starting time, ending time, bandwidth, duration, etc. In this paper we defines reservation request as $R = (bw, t_s, t_e)$, where bw denotes the reserved bandwidth, t_s and t_e denote the starting and ending time respectively. An example for reserved bandwidth is given in Fig.1. In the following sections, we will explain the 3 mentioned data structures with the example shown in Fig. 1.

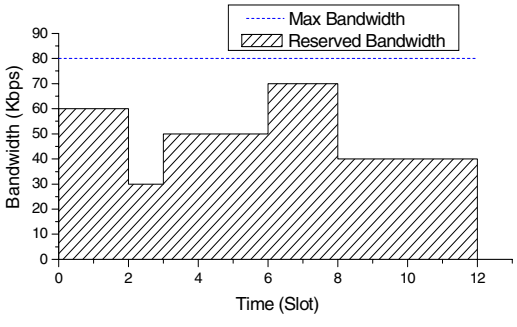


Fig. 1. An example for the reserved bandwidth. The time ranges from 0 to 12. The maximum available bandwidth is 80 Kbps. The line-hatched area denotes the reserved bandwidth

3.2 Slotted Array

The array is a basic and easy-to-implement data structure. In [8] it is used to perform admission control for advance reservation of bandwidth. Each element of the array represents a time slot and stores the accumulated bandwidth allocated for the

respective slot. For each arriving reservation request $R(bw, t_s, t_e)$, the bandwidth broker checks each element of the array between the starting time t_s and ending time t_e with the purpose of judging whether the following requirement is satisfied:

$$a[i] + bw \leq BW_{max} \tag{1}$$

where a[i] denotes the *i*-th element of the slotted array. If the condition is satisfied, the request can be admitted and the value of each element should add *bw*. Otherwise it indicates resource is not enough thus the request should be rejected.

Performing admission control with slotted array is very simple. But it also has some disadvantages. (1) For the reservation request that covers a very long period, it needs a great number of equal elements to store the reservation information. So it wastes much memory. (2) Each element of the array represents a predefined time interval, but not variable, so the granularity is not determined by the practical requirements but entirely by the time slots. (3) Since the size of the array is fixed, as the time goes by, all the elements of the array will be occupied. Although we can partly solve this problem by implementing it as ring buffers and using a pointer to mark the current time slot, the duration of the reservation will be limited inevitably.

3.3 Bandwidth Tree

In [9] a bandwidth tree was proposed to perform admission control for advance reservation. It is a tree in which each interior node has 2 or 3 child nodes and all leaf nodes have the same depth. Each node represents a non-empty time interval. Each leaf covers an interval in which the bandwidth is constant. The interval covered by a node is the union of the intervals covered by its child nodes. Each non-leaf node in the bandwidth keeps the following information.

- *l*, *r*: the left end and the right end of the interval covered by the node.
- *pChild0*, *pChild1*, *pChild2*: the pointers to the three children of the node. Because the number of the children may be 2 or 3, *pChild2* may be assigned null.
- *amb*: accumulate minimum of the available bandwidth in the node.

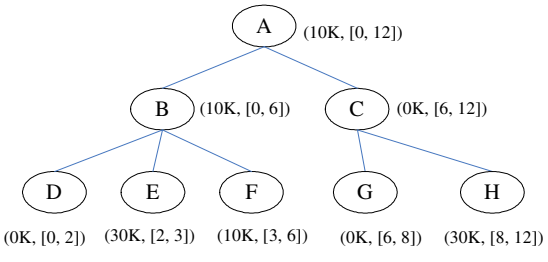


Fig. 2. The instance of a bandwidth tree corresponding to the reservation example shown in Fig.1. Here *Ts* is 0 and *Te* is 12

The bandwidth mentioned here is quite different from that in the slotted array. The bandwidth in bandwidth tree is unreserved, while in the slotted array it means the bandwidth that has been reserved. Fig.2 illustrates the bandwidth tree corresponding to Fig.1.

Initially there is only a single root node covering the whole time interval from T_s to T_e. The *amb* of the root node is BW_{max} because there is no bandwidth been reserved at this time. When a reservation request $R(bw, t_s, t_e)$ arrived and was accepted by the bandwidth broker, the whole time interval was divided into 2 or 3 (depending on whether t_s or t_e is equal to the l or r of the root node) smaller intervals. 2 or 3 new nodes were created to store the bandwidth information of the small time interval and these nodes were linked as the children of the original node. During the admission control process, the bandwidth tree is repeatedly merged, balanced and normalized. In [9] the relevant algorithm was fully described.

The bandwidth tree can perform admission control for the reservation that starts and ends at any time, so it can provide demanded time accuracy. But the algorithm is too complex to be applied to the bandwidth reservation. Moreover, each time a reservation request is admitted, the bandwidth tree should be merged, balanced and normalized. Because these operations are executed in a recursive manner, they consume much time and decrease the speed of admission greatly. In addition to that, in the same level (depth), the l (r) value of a node is equal to the r (l) value of its left (right) adjacent node. So it wastes much memory to keep the redundant information.

3.4 A Data Structure Based on Linked List

After analyzing the advantages and disadvantages of the slotted array and bandwidth tree, we proposed a new data structure based on linked list to perform admission control for advance reservation. The idea for the data structure was partly illuminated by the bandwidth tree. A node of the linked list is defined as *node*(bw, t_s, *pNext*), where t_s denotes the starting time of the time interval, *pNext* denotes the pointer to the next node, bw denotes the reserved bandwidth during the time interval of current node. Thus the time interval covered by a node (denoted as n_1) is from $n_1.t_s$ to $n_2.t_s$, where n_2 is the next node to n_1. The *pNext* of the last node in a linked list is *null*. It means that the time interval covered by the last node n_e is from $n_e.t_s$ to infinity. Usually the bandwidth of the last node is zero, because during the time interval covered by n_e (from $n_e.t_s$ to infinity) no bandwidth is reserved. Fig.3 shows the instance of a linked list recording the reservation information shown in Fig.1.

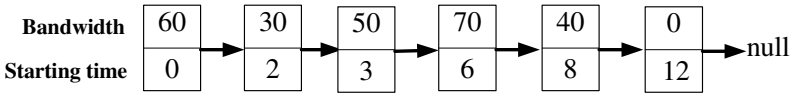


Fig. 3. The instance of a linked list corresponding to the reservation example shown in Fig.1. Each node contains the starting time, reserved bandwidth and a pointer to the next node

The process of performing admission control with linked list is described as follows. Initially there is only a single node *head*(0, 0, *null*) in the linked list. When a reservation request $R1(bw, t_s, t_e)$ arrived and the request is admitted (when $R1.bw \leq BW_{max}$), 1 or 2 (depending on whether $R1.t_s$ is equal to 0) new nodes will be added into the linked list. Their positions in the linked list are determined by their starting time. The checking and adding operations should be performed upon each arriving request. To avoid searching the proper position from the head node of the linked list

time and time again, we utilize a "local" head node to mark the start node that has been searched from for the prior request. In order to implement it, we should collect all the reservation requests during a given period, sort them by their starting time and then perform the admission control process for them together. Utilizing this mechanism will decrease the time cost for searching the proper starting node greatly.

Because the memory for the linked list is dynamically allocated, we can release the outdated nodes to recycle the memory. The so-called outdated nodes mean the time intervals covered by which are much earlier than the current time. There are two releasing strategies can be adopted.

- Set a threshold for the memory. If the amount of actually consumed memory exceeds the specified threshold, the releasing procedure will be called to recycle the memory allocated to the outdated nodes.
- Set a fixed releasing period for the memory. The bandwidth broker will perform the releasing action once during a cycle.

Adopting the first method needs to keep track of the utilization information of the memory. In this paper we chose the second mechanism to release the outdated nodes.

4 Evaluation

4.1 Simulation Environment

A simulation is carried out to contrast the performance of the linked list with the bandwidth tree and the slotted array. The network in the simulation has a single path between node u and v. In other words, path needs not to be found for reservation request in the admission control algorithm. The path provides a bandwidth capacity (denoted by BW_{max}) of 100 Mbps for advance reservations. The linked list, the bandwidth tree and the slotted array are used as the data structures of bandwidth in the simulation.

We designed a reservation request generator to produce a set of advance reservation requests on the path between node *u* and *v*. Each request includes starting time, duration and bandwidth requirement. The simulation period has a length of 20000 slots. One slot represents one second. Within this period, requests are generated with Poisson distribution with a mean varying from 0 to 1.0 requests per slot (the step is 0.001), thus the total number of the requests increases from 0 to 20000. The duration time is exponentially distributed with a mean value of 500 slots and the bandwidth requirement is uniformly distribute between 100 Kbps and 1 Mbps. The starting time is uniformly distributed between 50 to 500 slots after the request arrival time.

4.2 Performance Metrics

The performance metrics used for examinations are average admission time and total consumed memory. The average admission time is defined as

$$T_{average} = T_{total} / N_{reservation} \tag{2}$$

where T_{total} denotes the total time cost to perform the whole admission control task and $N_{reservation}$ denotes the total number of reservation requests.

4.3 Results and Analysis

Fig.4 shows the average admission time when performing admission control over a single link by using the two data structures: slotted array and linked list. The performance of the bandwidth tree is not plotted in Fig.4 because the time cost of the bandwidth tree is over thousands times more than that of the slotted array and the linked list, it is not suitable to show.

In Fig.5 we show that resources are reserved up to and above the point where we have rejections. The curve indicates at which number of reservations we start getting rejections. Once the rejection occurs, the number of rejected reservations increases linearly. In Fig.4 the average admission time decreases both for the slotted array and the linked list when there are a large number of rejections. This is because the bandwidth broker needs not to reserve bandwidth and modify the status of the data structures since it discovers there is no enough resource to be reserved. We can also see from Fig.4 that the performance of the linked list is much better than that of the slotted array when the number of reservation requests is not very large. As the number of the requests increasing, there are more and more requests being rejected. Both data structures cost less average admission time, while the average admission time of the slotted array decreases more sharply than that of the linked list until its performance exceeds that of the linked list. But the superiority is not obvious.

Fig.6 shows the memory consumption of the slotted array and the linked list. We set the total number of the reservation requests as 20000. We omit the memory consumption of the bandwidth tree again, because it is much larger than the other two data structures. As time goes on, the memory consumption of the slotted array is constant (80000 bytes), while the linked list without memory releasing consumes

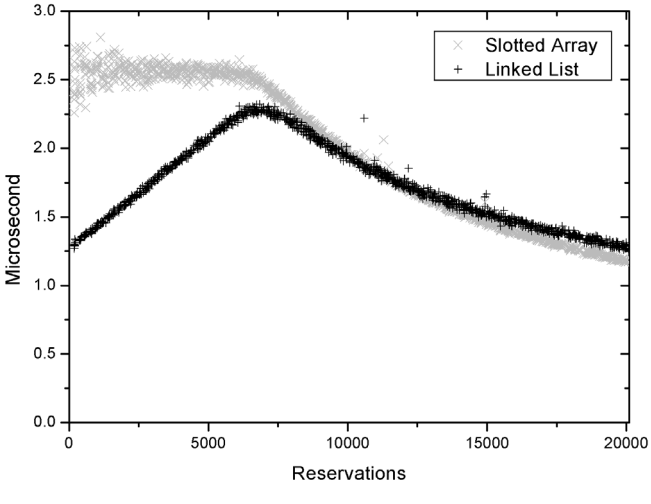


Fig. 4. The average admission time of the slotted array and the linked list. When there are not many reservation requests, the linked list performs much better than the slotted array. When the number of the reservation requests is very large, the slotted array is a little better than the linked list, but not obvious. We perform memory releasing every 2000 time slots for the linked list

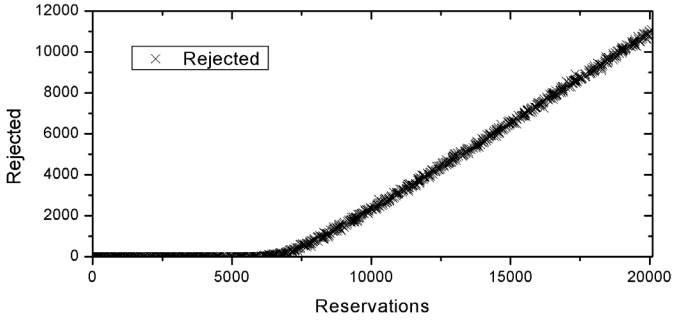


Fig. 5. The number of the rejected reservation. It increases linearly together with the total reservation request

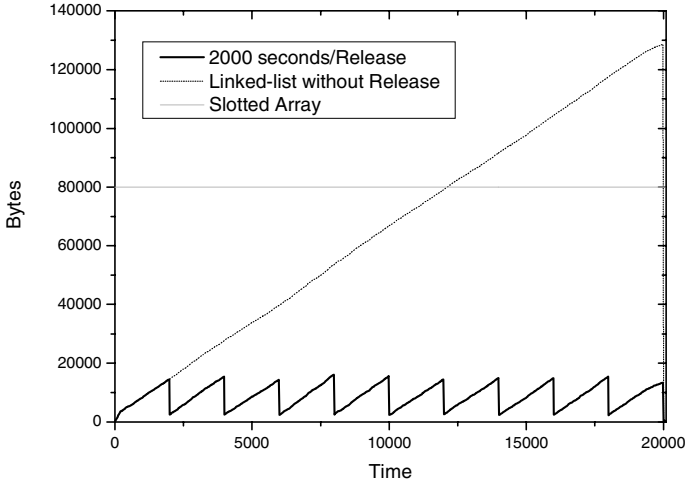


Fig. 6. Memory consumption of slotted array, periodically recycled linked list and non-recycled linked list

linearly increasing memory (illustrated in Fig. 6 as dotted line). But when we perform releasing operation for the linked list periodically (e.g. the cycle is 2000 slots), the curve of the linked list shows in a serrated pattern. Seen from Fig.6, the linked list consumes much less memory than the slotted array if we perform releasing memory periodically. Less releasing cycle leads to less memory consumption. In theory it will consume more time to perform releasing, so we should make a tradeoff between the memory consumption and the admission speed. But practically, our simulation shows that the releasing operation affects very little to the total admission time. Especially when the releasing cycle is larger than 200 slots, its effect can even be ignored. In fact, the result shown in Fig.4 was obtained by releasing the outdated nodes of a linked list with a cycle of 2000 slots.

5 Conclusion and Future Works

A suitable data structure is very important to admission control for advance reservation. In this paper we examined several existing data structures and then proposed a new data structure based on linked list. We made some simulations to compare the linked list with a slotted array and a bandwidth tree presented in [8] and [9] respectively. Our simulations show that the linked list and slotted array perform much better than the bandwidth tree in the examined scenarios. Concerning the time consumption, the linked list and the slotted array show almost the same performance when the amount of reservation requests is very large. As the number of reservations decreasing, the former shows better performance than the latter. While concerning the memory consuming, the former consumes much less memory than the latter, because dynamically allocating and releasing memory is the natural advantage of the linked list.

Theoretically, the data structure based on linked list is superior to slotted array with the following advantages:

- For the reservation covers a very large interval, the linked list needs only a single node while the slotted array must use many continuous nodes to store the reservation information.
- The reservations stored in linked list can start at any time, while in slotted array its starting time must depend on the granularity of the slots. Defining a appropriate interval for the slotted array is a challenge, because a coarsely granular decreases the accuracy while a finely granular increase the time and memory consumption.
- The linked list can perform recycling memory by dynamically allocating and releasing memory for the nodes.
- The size of slotted array is unchangeable once it is defined. If the total reservation interval is very large, the slotted array may be insufficient, while the linked list can overcome this shortcoming by freely inserting or appending nodes to it.

We have discussed advance reservation on network resource. But this mechanism can really be applied to advance reservation on other kinds of resources, such as computing, storage, etc. Our linked list is available not only for slotted-time but also continuous time scenarios. Future works will be done on better admission control algorithm to admit advance reservation more quickly.

References

1. Wilko Reinhardt: Advance Resource Reservation and its Impact on Reservation Protocols. In: Proc. of Broadband Islands'95, Dublin, Ireland, 9 (1995)
2. Nikolaos Nikou: Advance Reservation on Network. Available online at: http://www.netlab.hut.fi/opetus/s38130/k00/Papers/Topic11-Reservation.doc
3. Steven Berson, Robert Lindell, Robert Braden: An Architecture for Advance Reservations in the Internet, Technical report, USC Information Sciences Institute, July 1998
4. Lars-Olof Burchard, Hans-Ulrich Heiss: Performance Issues of Bandwidth Reservation for Grid Computing, In: Proceedings of the 15th Symposium on Computer Archetecture and High Performance Computing (SBAC-PAD'03)

5. Jianbing Xing, Chanle Wu, Muliu Tao, Libing Wu, Huyin Zhang: Flexible Advance Reservation for Grid Computing, GCC 2004, LNCS 3251, pp. 241-248, 2004
6. A. Nilsson, J. Chen, and S. Carlsson. An efficient data structure for advance bandwidth reservation on the Internet. Technical resport, CSEE, Lulea University of Technology, 1998.
7. Olov Schelen, Andreas Nilsson, Joakim Norrgard and Stephen Pink: Performance of QoS Agents for Provisioning Network Resources, Quality of Service, 1999. IWQoS '99. 1999 Seventh International Workshop on 31 May-4 June 1999 Pages: 17 – 26
8. Lars-O. Burchard, Hans-U. Heiss: Performance Evaluation Of Data Structures For Admission Control In Bandwidth Brokers, Technical Report TR-KBS-01-02, Communications and Operating Systems Group, Technical University of Berlin, May 2002.
9. Tao Wang and Jianer Chen: Bandwidth Tree—A Data Structure for Routing in Networks with Advanced Reservations. Performance, Computing, and Communications Conference, 2002. 21st IEEE International, 3-5 April 2002 Pages: 37 - 44

An Adaptive Gateway Discovery Algorithm for the Integrated Network of Internet and MANET

Tsung-Chuan Huang[1] and Sheng-Yi Wu[2]

Department of Electrical Engineering,
National Sun Yat-Sen University,
Kaohsiung 804, Taiwan
[1] tch@mail.nsysu.edu.tw
[2] m923010048@student.nsysu.edu.tw

Abstract. Since gateways are generally used to integrate the Internet with MANET (Mobile Ad Hoc Networks), gateway discovery is important for MANET mobile nodes to obtain the route to a gateway. The hybrid gateway discovery approach can be configured by adjusting a single parameter, the advertisement TTL. This paper aims to develop an adaptive gateway discovery algorithm to adjust the advertisement TTL by estimating the control overhead. Thus, the control overhead generation can be adjusted in the integrated network. The simulation results indicate that this adaptive gateway discovery algorithm can adjust the appropriate advertisement TTL in different numbers of mobile nodes which desire to access the Internet.

1 Introduction

The use of Internet has grown explosively in recent years. In the past, users relied on stations with fixed, wired interface to connect to the Internet. Nowadays, users can connect Internet via portable device by wireless interfaces such as mobile phones, laptops and personal digital assistants (PDAs).

Wireless networks can be classified into two categories, *infrastructure wireless networks* and *ad hoc networks*. In infrastructure wireless networks, MNs (mobile node) communicate directly with an AP (access point) to the Internet (i.e., the fixed network). However, an infrastructure wireless network suffers a dead-zone problem. By contrast, an ad hoc network can be flexibly deployed in the environment without AP because it is composed of MNs. Since the transmission range of MNs is limited, two nodes located beyond each other's transmission range can only communicate by routing through intermediate nodes. Besides, traditional ad hoc networks unable to access the Internet significantly limit their applicability. Therefore, to solve the dead-zone problem of infrastructure wireless network and extend the application of ad hoc networks, the integrated network of Internet and MANET (Mobile Ad Hoc Network), as shown in Fig. 1, has been proposed.

Since gateways are generally used to integrate the Internet with MANET, the gateway discovery approach is important for MNs in MANET to obtain the route to gateways. Two major implemented approaches to providing Internet connectivity to a

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 911 – 920, 2005.

MANET are *proactive* and *reactive*. Proactive gateway discovery is initiated by gateways. All gateways periodically broadcast advertisement messages, which are flooded throughout the network. If receiving the advertisements, the MNs would update the route entry and record the information about the route to the gateway. However, if the MNs do not have a route to the gateway, then they create a route entry for the gateway in their routing tables. All proactive approaches are limited in terms of the costly operation in which the advertisement message is flooded through the whole MANET periodically regardless of whether they need the information of gateways.

Reactive gateway discovery is initiated by MNs. When a MN wants to access the Internet, it broadcasts a solicitation message to obtain gateway information. The benefit of this approach is that solicitation messages are transmitted only when a MN requires the information of gateway, preventing the problem of proactive periodic flooding of the whole MANET. However, reactive gateway discovery is limited owing to its long average gateway discovery time.

To minimize the disadvantage of the proactive and reactive gateway discovery, the *hybrid* approach is proposed, in which proactive gateway discovery is used for MNs in a certain range (hops) around a gateway, and the other MNs located outside this range utilize reactive gateway discovery.

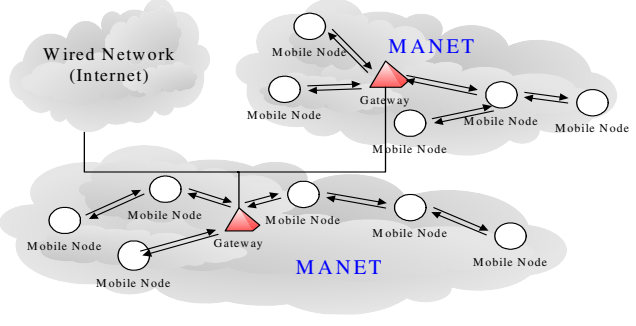


Fig. 1. The integrated network of Internet and MANET

Compared with the wired network, the wireless transmission speed is quite slow and is burdened with control overhead resulting from gateway discovery and maintaining route and Mobile IP[14][17]. Therefore, effectively limiting the control overhead generation in the network in order to increase the data transmission bandwidth becomes a very significant issue. This paper designs an adaptive gateway discovery algorithm such that the gateway adjusts, based on the estimated control overhead in the network, the hybrid gateway discovery approach to enhance the network communication performance.

The remainder of this paper is organized as follows. Section 1 provides an overview of Mobile IP, MANET and MANET routing protocol. Section 2 then describes related work. Next, Section 3 describes the proposed adaptive gateway

discovery algorithm. Section 4 summarizes the simulation results. Conclusions are finally drawn in Section 5.

1.1 Mobile IP

In a Mobile IP environment, an IP datagram sent to a MN's home address by a CN (corresponding node) is delivered to the MN's home network, even if the MN is away from its home network in another subnet (i.e., the foreign network). Then, the HA (home agent) in the home network encapsulates the datagram and tunnels it to the MN's current foreign network. The FA (foreign agent) in the foreign network then decapsulates the encapsulated datagram and forwards it to the MN. Furthermore, a MN must register its current CoA (Care-of-address) received from the FA with its HA. The HA monitors the binding cache entry of each MN in the home network.

1.2 MANET (Mobile Ad Hoc Network) and Ad Hoc Network Routing Protocol

In a MANET, MNs employ wireless interface to communicate and roam at will. Each MN serves both as a host and a router which can forward packets. Hence, the MNs can communicate beyond their transmission range using multi-hop communication. The MANET routing protocol can typically be classified into three categories, *table-driven*, *on-demand* and *hybrid* routing protocols. The table-driven routing protocols constantly update their routing information, so the route is available when packets are needed to be forwarded. DSDV (Destination Sequenced Distance Vector routing protocol) [1], WRP (Wireless Routing Protocol) [12][18] and CGSR (Cluster Switch Gateway Routing) [3] are all table-driven routing protocols.

By contrast, the on-demand routing protocol executes a route discovery process only when a mobile node requires a route for sending the packets. The AODV (Ad Hoc On-Demand Distance Vector routing) [15], DSR (Dynamic Source Vector) [9], TORA (Temporally Ordered Routing Algorithm) [13], SSR (Signal Stability Routing) [4] and PAR (Power-Aware Routing) [20] are on-demand routing protocols. Hybrid routing protocol adopts a mixture of the table-driven and on-demand routing protocol. The scope of the table-driven procedure is limited to the mobile node, while an on-demand procedure is utilized outside the mobile node. ZRP (Zone Routing Protocol) [6] is an example of hybrid routing protocol.

2 Related Work

The studies [11] and [2] presented solutions to provide Internet connectivity in ad hoc networks. Lei and Perkin [11] used a modified RIP(Routing Information Protocol) [7] and Broch et al [2] used DSR for ad hoc routing. In [10], Jonsson et al. proposed a method called MIPMANET, which provides MANET MNs with Internet access using tunneling and Mobile IP with CoA(care-of-address). The AODV routing protocol is used within the MANET to obtain routes between MNs and the FA. Sun et al. [21] presented a mechanism by integrating Mobile IP with AODV. Their work also examined the effect of varying beacon intervals on the protocol performance. Ergen

and Puri [5] proposed two protocols, MEWLANA-TD and MEWLANA-RD, to integrate Mobile IP with MANET. These two protocols lead to optimum performance in different environments. Ratanchandani and Kravets [19] proposed a hybrid approach to Internet connectivity for MANET. This study presents several techniques, such as TTL (Time-To-Live) scoping of agent advertisements, eavesdropping and caching agent advertisements. In [22], Tseng et al. integrated the Mobile IP with MANET by implementing two daemons, DSDVd and MIPd, on the application layer to interact with the system kernel via a socket interface.

3 Adaptive Gateway Discovery Algorithm

3.1 Architecture Specification

The gateway has two network interfaces as illustrated in Fig. 2. The wireless interface connects the MANET, while the wired interface connects the Internet. Because of the wired interface to Internet, gateways have no mobility. Gateways act as bridges between MANET and Internet, forwarding data packets between the two networks. Each gateway also serves as a Mobile IP FA to support the Mobile IP service. The hybrid gateway discovery is utilized in the integrated network. Within the MANET, the AODV routing protocol is used because it is one of the best developed routing protocols for MANET.

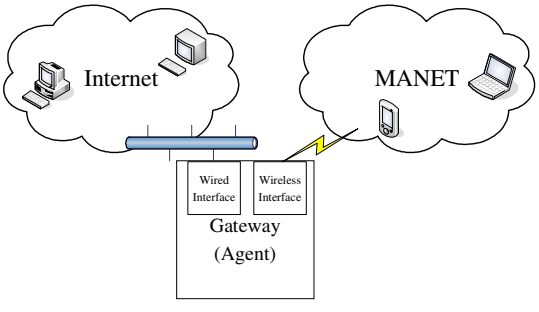


Fig. 2. The gateway interfaces

3.2 Gateway Discovery Approach

The gateway discovery approach utilized in this paper is based on the hybrid gateway discovery system proposed in [19]. The TTL-field in the advertisement message is set to limit the scope of the proactive gateway discovery, assuming that not only gateways can send advertisement message. That is, if a MN in the MANET receives a solicitation message and has a route entry to a gateway in its routing table, then it can send an advertisement message to the originator of the solicitation message. Consequently, the control overhead in discovering gateway and the delay time would be decreased significantly. This paper utilizes the Expanding Ring Search Method

[16] to reduce flooding overhead due to solicitation. The TTL of each solicitation is initially set to 1, and is increased by 2 if the source node does not receive a reply for each interval. When the TTL reaches 7, the solicitation is transmitted by flooding if there no reply is received. Since lengthening the route between MNs and the gateway will decrease the reliability of the route as the network span becomes larger, the maximum flooding TTL is set to 10 to ensure the route's reliability.

3.3 Mobile IP Registration and Route Maintenance on AODV

After receiving an advertisement, a MN must unicast a Registration Request to register its FA (gateway) and the HA. Then, the FA and the HA follow a Mobile IP registration procedure. The HA creates a binding entry for the MN and replies with a Registration Reply back to the MN via the FA. Thus, MNs finish the Mobile IP registration procedure and can access Internet source via gateways.

In the AODV routing protocol, if a link is broken in the route, the node upstream of the break will broadcast a RERR (Route Error) message to notify this broken link to its neighbors. The RERR message contains the node information of the unreachable destination due to the link break. When a node receives the RERR, it marks this route, which has an unreachable destination listed in the RERR message, as invalid. The RERR message is then forwarded continuously until the source receives the message. Then, the source can reinitiate the route discovery process if needed.

Table 1. Variable Simulation Parameters

The number of node density	5, 6, 8
The number of nodes	500
The maximum speed of nodes	2 m/sec
The mobility model of nodes	random waypoint model [8]
The pause time of nodes	5 sec
Simulation time	60 sec

3.4 Estimation and Adaptive Gateway Discovery Algorithm

To manage overhead generation, this paper proposes an adaptive gateway discovery algorithm to be processed on the gateway. The proactive and reactive overhead is then estimated from the information received by the gateway. The gateway then sets the value of ADV_TTL, the TTL of the advertisement message, to limit the generation of control overhead in the network. Table I lists the parameters of our simulation environment, and Fig 3 shows the pseudocode of the proposed adaptive gateway discovery algorithm.

```
Estimation
Switch ( the packet received from GW)
        Case "RERR":
           set the route to the unreachable destination(s) ( listed in the RERR message) as invalid
        Case "Registration Request":
           If the hops between the GW and the originator of the Registration Request <= ADV_TTL
                   estimate the ADV overhead & Mobile IP overhead
                   proactive overhead = proactive overhead + ADV overhead + Mobile IP overhead
           Else
                   If route to the originator of the Registration Request is invalid
                           estimate the gateway discovery overhead
                           reactive overhead = reactive overhead + gateway discovery overhead
                   EndIf
                   estimate the Mobile IP overhead
                   reactive overhead = reactive overhead + Mobile IP overhead
           EndIf
EndSwitch

Adaptive ADV_TTL
While time out
        ratio = reactive overhead / proactive overhead
        If  ratio > O_up_thresh
           ADV_TTL = ADV_TTL + 1
        Elseif  ratio<O_down_thresh
           ADV_TTL = ADV_TTL - 1
        resetup timer , proactive overhead and reactive overhead
        EndIf
EndWhile
```

Fig. 3. The pseudocode of the adaptive gateway discovery algorithm

Estimation of Proactive Overhead. The proactive overhead is the number of advertisement messages sent, given by $O(N_{ADV})$, where N_{ADV} denotes the population of sending advertisement. Hence, the proactive overhead can be expected to be proportional to the node density, expressed as the average number of neighbor per node, and the advertisement transmission zone (ADV_Zone, ADV_TTL^2). Figure 4 shows this equation. Since the MNs register their HAs and the FAs (gateways) after receiving the advertisement message, the gateway would know both the number of registered MNs and the corresponding hops between these MNs and the gateway. If the corresponding hops are smaller than ADV_TTL, then the MNs obtain gateway information using the proactive gateway discovery approach.

Estimation of Reactive Overhead. The reactive overhead can be regarded as the overhead generated by MNs located outside the ADV_Zone, and thus increases with the query rate. Like the proactive overhead, the reactive overhead should be proportional to the node density and the zone of sending solicitation ($Search\_TTL^2$, where Search_TTL denotes the TTL value of each solicitation message). Because the Expanding Ring Search Method is utilized in this paper, the numbers of searches and the Search_TTL value of each search can be computed. Additionally, the gateway may still receive a Registration Request when the Mobile IP registration lifetime of the MNs expires. The RERR (Route Error) message of AODV is used to decide

whether the Registration Request results from the link break or from the expiry of the Mobile IP registration lifetime. If the RERR message is not received before the registration lifetime of the MNs expires, then the Registration Request received by gateway is considered to be re-registered, and the gateway discovery overhead of the MN need not be valuated. Figure 5 displays the reactive overhead with ADV_TTL.

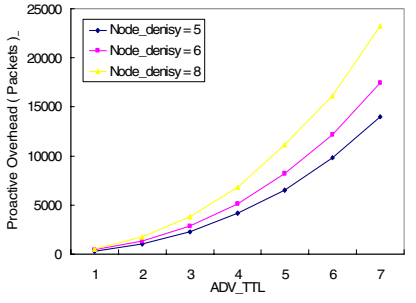


Fig. 4. The proactive overhead vs. ADV_TTL

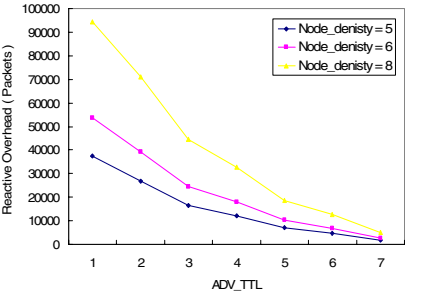


Fig. 5. The reactive overhead vs. ADV_TTL

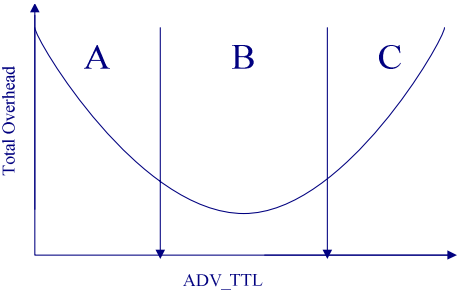


Fig. 6. The total overhead vs. ADV_TTL

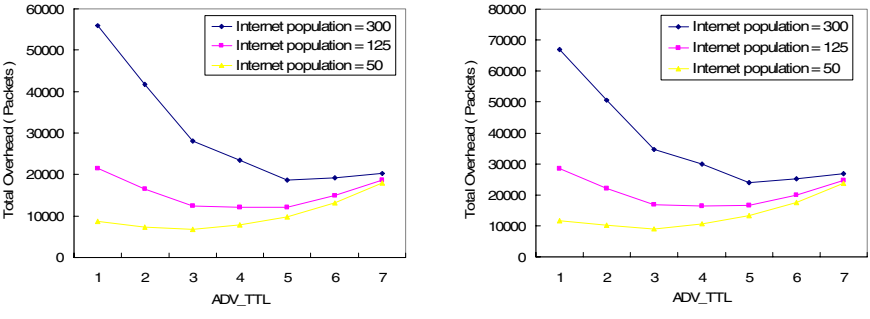
Adaptive Gateway Discovery Algorithm. Based on the simulation in [19], Fig. 6 plots the relationship between the total overhead and ADV_TTL, where the total overhead is the sum of the proactive and reactive overhead. The ideal ADV_TTL should be in region B because the total overhead is minimal in this region. If the ADV_TTL is located in region A, the generated *solicitation* will be too high, resulting in the total overhead to be worse than that in region B. In the other hand, when the ADV_TTL is located in region C, then the generated *advertisement* will be too high, resulting in the total overhead to be worse than that in region B too. To make ADV_TTL locate in region B, an adaptive gateway discovery algorithm using Eq. (1) is proposed. Since too much reactive overhead will be generated when f(ADV_TTL) >> 1, and too much proactive overhead is generated when f (ADV_TTL) << 1, f(ADV_TTL) is compared with predetermined thresholds $O_{up\_threshold}$ and $O_{down\_threshold}$. If f(ADV_TTL) > $O_{up\_threshold}$, then ADV_TTL is increased, and if f(ADV_TTL) < $O_{down\_threshold}$, then ADV_TTL is decreased. Thus, ADV_TTL can be adapted and located within region B.

$$f(ADV\_TTL) = \frac{reactive\ \ overhead}{proactive\ \ overhead} \tag{1}$$

4 Simulation Results

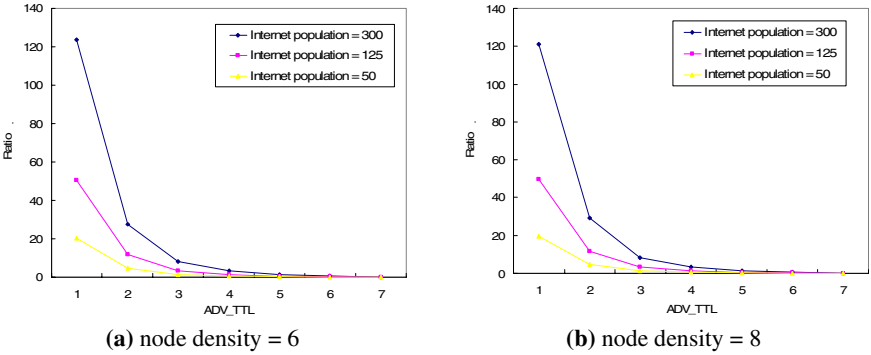
This section focuses on the effect of varying Internet population, which is the number of MNs accessing the Internet. First, the total overhead with an Internet population between 50 and 300 was calculated.

Since more MNs can receive advertisement message from the gateway when ADV_TTL is increased, the proactive overhead is also likely to increase with ADV_TTL, and the reactive overhead decreases with ADV_TTL. Figure 7 shows the total overhead with ADV_TTL between 1 and 7. When ADV_TTL> 7, the range of the advertisement covers almost all the MNs required to access Internet, i.e., the total overhead is the result of using proactive gateway discovery approach.



(a) node density = 6 **(b)** node density = 8

Fig. 7. The total overhead vs ADV_TTL



(a) node density = 6 **(b)** node density = 8

Fig. 8. The ratio of reactive overhead to proactive overhead

According to the analysis of section 3, the MNs advertisement message periodically sent by gateway is received, regardless of whether they intend to access the Internet. Hence, the Internet population is independent of the proactive overhead. Conversely, the Internet population influences the amount of reactive overhead. That is, when the ADV_TTL is large, most of the Internet population obtains the gateway information using proactive gateway discovery. Therefore, the larger the ADV_TTL, the smaller number of MNs using the reactive gateway discovery to access Internet.

Therefore, the total overhead for different Internet populations have the same proactive overhead but different reactive overheads, as shown in Fig. 7 (a)(b). Moreover, if the Internet population is 300, then the best ADV_TTL is 5. If the Internet population becomes 125, the appropriate value of ADV_TTL can be 3, 4, or 5. However, if the Internet population is 50, then the best ADV_TTL is 3.

Finally, the adaptive gateway discovery algorithm can perform well if $O_{up\_threshold}$ and $O_{down\_threshold}$ are properly configured in terms of the ratio of reactive overhead to proactive overhead. In Fig. 8(a), the ratio is calculated at a node density of 6, while in Fig. 8(b), the node density is 8. The Internet population of 300, 125, and 50 are considered, and the best ADV_TTL can be found to be 5, (3, 4, 5) and 3, respectively. Additionally, Figs. 8(a)(b) demonstrate that the acceptable value of $O_{up\_threshold}$ is 5, and that of $O_{down\_threshold}$ is 0.6.

5 Conclusion

This paper considers a network that integrates the Internet with MANET, enabling MNs in the MANET to access the Internet. Since messages between the wireless and wired network must pass through gateways, the hybrid gateway discovery approach, which combines the advantages of proactive and reactive gateway discovery, was utilized. An adaptive gateway discovery algorithm was proposed to adjust the hybrid gateway discovery approach. Based on the information received by gateways, the proactive and reactive overhead of the network is estimated, and the ADV_TTL is adjusted to manage the overhead generation until the appropriate ADV_TTL is found. The simulation results show that the ADV_TTLs change for different Internet populations, and the proposed adaptive gateway discovery algorithm is utilized to obtain the best value of ADV_TTL. This adaptive gateway discovery algorithm can be processed repeatedly to maintain the performance of the network communication as the MANET topology changes.

References

1. P. Bhagwat. and C. Perkins, “Highly Dynamic Destination-Sequenced Distance Vector Routing（DSDV）for Mobile Computers,” in proc. of the ACM SIGCOMM Symposium on Communications, Architectures and Protocols, pp.234–244 , London, UK, Sept. 1994.
2. J. Broch, D. Maltz, and D. Johnson. Supporting, “ Hierarchy and Heterogeneous Interfaces in Multi-Hop Wireless Ad Hoc Networks,” in proc. *of the* I-SPAN’99, pp. 370–375, Perth, Australia, June 1999.
3. C.-C. Chiang, H.-K. Wu, W. Liu, and M. Gerla, “Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel”, in proc. of the IEEE Singapore International Conference on Networks, pp. 197–211, Apr. 1997.

4. R. Dube, et al, "Signal Stability based Adaptive Routing（SSA） for Ad Hoc Mobile Networks," IEEE Personal Communication Magazine, pp. 36–45, Feb. 1997.
5. M. Ergen , A. Puri, " NEWLANA – Mobile IP Enrished Wireless Local Area Network Architecture," in proc. of the Vehicular Technology Conference 2002, pp. 2449–2453, Vancouver, Canada, Sept. 2002.
6. Z. Haas and M. Pearlman, "The Zone Routing Protocol（ZRP）for Ad Hoc Networks," IETF MANET Draft, June 1999.
7. C. Hedrick, "Routing Information Protocol（RIP）," IETF RFC 1058, June 1988.
8. M.-H. Jiang and R.-H. Jan, "An Efficient Multiple Paths Routing algorithm for Ad-hoc Networks," in proc. of the 15th International Conference on Information Networking, pp. 544–549, Beppu, Japan, 2001.
9. David B. Johnson, J. Broch, and David A. Maltz, "Supporting Hierarchy and Heterogeneous Interfaces in Multi-Hop Wireless Ad Hoc Networks." in proc. of the Workshop on Mobile Computing, pp. 370–375, Perth, Australia, June 1999.
10. U. J¨onsson, F. Alriksson, T. Larsson, P. Johansson, and G. Q. M. Jr, "MIPMANET - Mobile IP for Mobile Ad Hoc Networks," in *proc. of the 1st Workshop on MobiHOC' 00*, pp. 75–85, Boston, Massachusetts, Aug. 2000.
11. H. Lei and C. E. Perkins, "Ad Hoc Networking with Mobile IP," in *proc. of the 2nd European Personal Mobile Communications Conference*, pp. 197–202, Bonn, Germany, Oct. 1997.
12. S. Murthy and J. J. Garcia-Luna-Aceves, "A Routing Protocol for Packet Radio Networks," in proc. of ACM MOBICOM'95, pp.86–95, Berkeley, California, Nov. 1995.
13. V. Park and M.Scott Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," in proc. of the IEEE INFOCOM'97, pp. 1405–1313, Kobe, Japan, Apr. 1997.
14. C. E. Perkins, "Mobile IP Design Principle and Practices", Addison Wesley, 1997.
15. C. Perkins and E. M. Royer, "Ad-hoc On-demand Distance Vector Routing," in proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, pp. 90–100, New Orleans, USA , Feb. 1999.
16. C. E. Perkins, E. M. Belding-Royer, and S. R. Das, "Ad Hoc On-demand Distance Vector (AODV) Routing," IETF Internet Draft, June 2002.
17. C. E. Perkins, "IP Mobility Support for IPv4," RFC 3344, Aug. 2002.
18. Jyoti Raju and J. J. Garcia-Luna-Aceves, "A Comparison of On-demand and Table-driven Routing for Ad Hoc Wireless Networks," in proc. of IEEE the ICC, pp. 1702–1706, New Orleans, USA, June 2000.
19. P. Ratanchandani and R. Kravets, "A Hybrid Approach to Internet Connectivity for Mobile Ad Hoc Networks," in proc. of the IEEE Wireless Communications and Networking Conference (WCNC), pp. 1523–1527, New Orleans, USA, Mar. 2003.
20. S. Singh, M. Woo, and C. S. Raghavendra, "Power-Aware Routing in Mobile Ad Hoc Networks," in proc. of the ACM/IEEE MobiCom'98 Conference, pp. 181–190, Dallas, USA, Oct. 1998.
21. Y. Sun, E. M. Belding-Royer, and C. E. Perkins, " Internet Connectivity for Ad hoc Mobile Networks," *International Journal of Wireless Information Networks special issue on Mobile Ad hoc Networks:* Standards, Research, Application, pp. 75–88, Apr. 2002.
22. Y. Tseng, C. Shen and W. Chen, "Integrating Mobile IP with Ad Hoc Networks," IEEE Computer, Vol. 36, No. 5, pp. 48–55, May 2003.

A Sender-Oriented Back-Track Enabled Resource Reservation Scheme

Yi Sun, Jihua Zhou, and Jinglin Shi

Institute of Computing Technology, Chinese Academy of Sciences,
Beijing 100080, P.R. China
Graduation School of the Chinese Academy of Sciences
{sunyi, jhzhou, sjl}@ict.ac.cn

Abstract. Since more and more unicast real-time transactions like VoIP (Voice over IP) are carried on the packet based network, guaranteed good QoS of these transactions becomes an important issue. Traditional RSVP (Resource Reservation Protocol) is aimed at multicast real-time transactions such as VoD (Video on Demand). It does not fit some features of unicast transactions. In this paper, we present a new way to reserve resource on packet based network. Our scheme adds back-track capability into the process of resource reservation thus increasing the successful rate of setting up a Resource Reservation Path. Besides, our method is different from RSVP in that it is sender-oriented. So it can notify the session sender of abnormal events in the reservation process as soon as possible, fulfilling the need of some unicast real-time transactions such as VoIP.

1 Introduction

In recent years, Internet, the representative of packet based network, plays a more and more important role in people's life. And the transactions on Internet are no longer limited to data transactions just as before. Nowadays, many kinds of multimedia transactions recur to Internet to supply services. However, traditional service mode of Internet is "Best Effort", which cannot supply guaranteed QoS (Quality of Service). But multimedia transactions like voice and video are very sensitive to delay, so we must provide enough network resources for such transactions to ensure their good-quality transportation. In order to support real-time multimedia transactions on Internet, IETF (Internet Engineering Task Force) enacted RSVP (Resource Reservation Protocol) [1] in 1997. RSVP is a receiver-oriented reservation protocol. The receiver of session answers for initiating the reservation process and pointing out how many resources should be reserved. The original objective of RSVP is aimed at multicast transactions like VoD [2] and the thought of receiver-oriented just fits the need of such transactions.

In the last decade, VoIP (Voice over IP) as a new force suddenly rises and supplying good-quality voice service becomes one of the objectives and key technologies for next generation of Internet. A lot of effort has been made to enhance the QoS of VoIP [3] [4]. Today the most popular way is to combine Signaling Protocol [5], RTP [6] and RSVP to ensure the QoS of voice communications on IP

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 921 – 930, 2005.

network. However, VoIP is a kind of unicast multimedia transactions, and RSVP is unfit for such transactions somewhat. First, in the voice transaction, it is usually the caller (sender of voice session) pays for the service. Thus, it is reasonable to let the sender of the session decide what quality of service it needs and how many resources are reserved on intermediate nodes. So for such transactions, send-oriented mode is better. Second, in RSVP the reservation process is hop-by-hop along the path from receiver to sender. If an intermediate node fails, it would send error messages to receiver of the voice session. And the sender of the call has to wait for a long time before knowing the failure information. In voice transaction, it is expected that the caller should be notified for the failure information as soon as possible so that he could retry or abort the call in time. Last and most important, in RSVP if one intermediate node fails to reserve, the whole reservation process fails. The reservation process is not able to search other nodes automatically. This means the get-through rate of the phone call is low. (Get-through rate is a most key measurement of QoS for Voice Communication.)

This paper presents a new scheme to reserve resources in packet based network. Our scheme includes back-track capability into the process of resource reservation thus greatly increasing the successful rate of setting up a Resource Reservation Path. Besides, the scheme is sender-oriented. It makes the sender of session decide how many resources should be reserved. And also the sender would be notified of any abnormal events during the reservation process as soon as possible.

The paper is organized as follows: Section 2 presents our scheme in details. Section 3 compares our scheme with traditional RSVP. Section 4 briefly summarizes the paper.

2 Our Scheme

2.1 Background

RSVP defines a series of objects to reserve resource on packet based network. In our scheme we adopt the same formats with RSVP for similar objects and this makes our scheme more compatible with the current equipments such as RSVP Routers. Specially, the most two important objects FLOWSPEC (describes desired QoS of dataflow) and FILTER_SPEC (defines subset of session data packets that should receive desired QoS) in our scheme has identical formats with those in RSVP. The formats of FLOWSPEC and FILTER_SPEC are shown in Reference [1] and [7]. Besides, the formats of EXPLICIT ROUTE object and RECORD ROUTE object in our scheme are illustrated in Reference [8].

2.2 Implementation of Our Scheme

There are four kinds of messages in our scheme: Reserve Request message—Resv_Req, Reserve Confirm message—Resv_Conf, Reserve Error message—Resv_Err and Reserve Retry message—Resv_Retry.

Each of these four messages consists of a common header, followed by a body consisting of a variable number of variable-length “objects”. The format of common header is shown in Fig.1.

"Vers" points out the protocol version number. Current version number is Version 1. "AFT" (Allowed Failure Times) indicates the times that the message is allowed to be back tracked. "Msg Type" designates the type of the message. For Resv_Req message, "Msg Type" is 1; Resv_Conf is 2; Resv_Err is 3 and Resv_Retry is 4. "Check Sum" contains the check sum of the whole message, including the common header and the body. An all-zero value of this field means no check sum is transmitted. "Send TTL" indicates the life time of the message and "Length" field designates the length of the whole message in bytes.

Our scheme makes use of the following 8 kinds of objects: FLOWSPEC, FILTER_SPEC, SESSION, EXPLICIT ROUTE, RECORD ROUTE, ERRORSPEC, CONFIRM and FAILURE_NODE. Each of these objects consists of one or more 32-bit words with a one-word header, with the format shown in Fig.2.

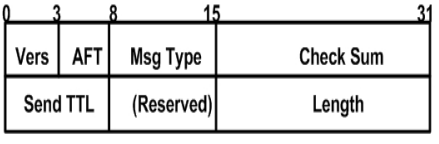


Fig. 1. Format of COMMON HEADER

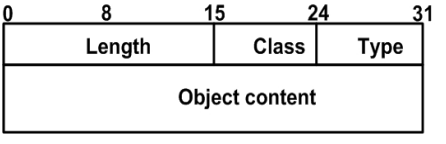


Fig. 2. Format of object

"Length" points out the length of the object in bytes. "Class" identifies the object class; the value of this field for SESSION object is 1, CONFIRM is 2, EXPLICIT ROUTE is 3, RECORD ROUTE is 4, FAILURE_NODE is 5, ERRORSPEC is 6, FLOWSPEC is 7 and FILTER_SPEC is 8. "Type" indicates the object type, unique within every object class. "Object content" presents the content of the object.

When session sender wants to set up a Resource Reservation Path, it would firstly send a Resv_Req message to session receiver. The format of Resv_Req message is as follows,

<Resv_Req message> ::= <COMMON HEADER><SESSION>[<EXPLICIT ROUTE>]
<FLOWSPEC><FILTER_SPEC><RECORD ROUTE> <FAILURE_NODE>

In above definition, the object in brackets is optional. The format of FLOWSPEC is illustrated in RFC2210 [7]. The formats of EXPLICIT ROUTE and RECORD ROUTE are seen in RFC3209 [8]. The format of FILTER_SPEC is described in RFC2205 [1]. And formats of SESSION and FAILURE_NODE objects are as follows.

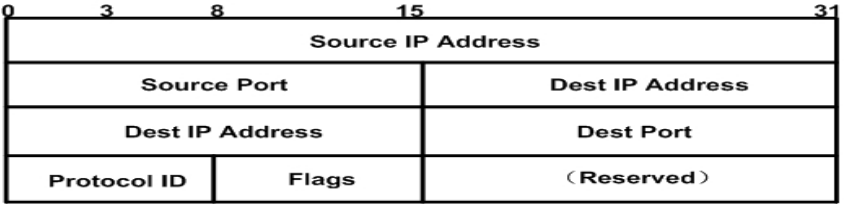


Fig. 3. Format of IPv4 SESSION object content

For SESSION object, the "Class" value is 1 and it is divided into two types: IPv4 ("Type"=1) and IPv6 ("Type"=2). Format of "Object content" for SESSION object is shown in Fig.3.

IPv6 SESSION object shares the same format with that of IPv4 SESSION object, only the corresponding address fields contain 128-bits IPv6 addresses. "Source IP Address" and "Source Port" presents the IP address and port number for the session-sender. "Dest IP Address" and "Dest Port" describes the IP address and port number for the session-receiver. "Protocol ID" indicates the IP Protocol Identifier for the data flow. "Flags" contains some flags for the session.

For FAILURE_NODE object, the "Class" value is 5. Format of "Object content" for FAILURE_NODE is shown in Fig.4.

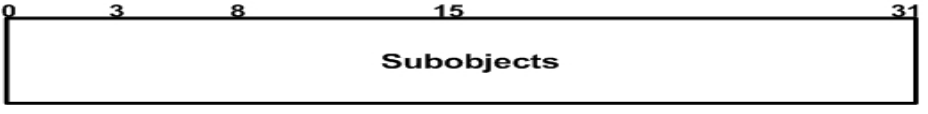


Fig. 4. Format of FAILURE_NODE object content

The content of FAILURE_NODE object consists of a series of variable length subobjects, and each subobject records a failure node. So the subobjects contained in FAILURE_NODE form a failure node list. There are two kinds of subobjects with the format shown in Fig.5.

IPv6 FAILURE_NODE subobject shares the same format with that of IPv4 FAILURE_NODE subobject, only the corresponding address field contains a 128-bits IPv6 address. "Type" field indicates the type of the subobject, for IPv4 subobject the value is 1 and for IPv6 subobject is 2. "Length" points out the length of the subobject in bytes. "IP address" contains the IP address of the failure node. And "Prefix Length" designates the length of the prefix (IPv4 subobject "Prefix Length"=32, IPv6 subobject "Prefix Length"=128).

If the Resv_Req message contains EXPLICIT ROUTE object, the "AFT" field in common header must be set to 0. This means the message must be transmitted along this EXPLICIT ROUTE and not allowed to be back tracked.

Resv_Req message is transmitted along the path that EXPLICT ROUTE object (if exists) defines or route-selection algorism selects. When intermediate node receives the message, it would check the desired QoS information in the message and decide whether it has enough resources to support the dataflow. If it has, the node would reserve proper resources and then record its address into the RECORD ROUTE object. Finally, the node tries to select a next-hop. If it finds a next-hop and the next-hop is not in the failure node list (contained in the FAILURE_NODE object), the node would forward the Resv_Req message to the selected next-hop. Otherwise, if the next-hop is in the failure node list, the node would retry to find another node as next-hop. If the node fails to find a proper next-hop at last, it has to release the resources reserved for the dataflow, clear its address from RECORD ROUTE object and record its address into the failure node list. Then the node starts the back-track check. If the node doesn't have enough resources to support the dataflow, it also starts the back-track check.

To begin back-track check, the node first checks whether the value of "AFT" field equals to 0. If it does, the request cannot be back tracked any longer. Then the node would send a Resv_Err message to the session-sender. The format of Resv_Err message is as follows,

<Resv_Err message> ::= <COMMON HEADER><SESSION><ERRORSPEC><RECORD ROUTE>[<FAILURE_NODE>]

The object in brackets is optional.

For ERRORSPEC object, the "Class" value is 6 and it is divided into two types: IPv4 ("Type"=1) and IPv6 ("Type"=2). Format of "Object content" for ERRORSPEC object is shown in Fig.6.

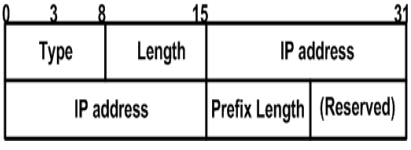


Fig. 5. Format of IPv4 FAILURE_NODE subobject content

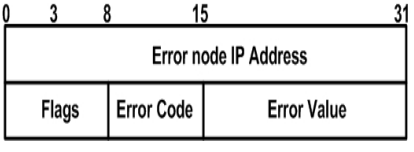


Fig. 6. Format of IPv4 ERRORSPEC object content

IPv6 ERRORSPEC object shares the same format with that of IPv4 ERRORSPEC object; only the corresponding address field contains a 128-bits IPv6 address. "Error node IP Address" indicates the IP address of the current node in which the error was detected. "Flags" contains some flags. "Error Code" and "Error Value" describe the error information.

Resv_Err message is passed along the path that RECORD ROUTE object describes but in the reverse direction and all the nodes would release the resources reserved for the dataflow after receiving the message. When Resv_Err message reaches the session-sender, the reservation process ends with an error.

If the back-track check finds the value of "AFT" field is greater than 0, the reservation request is still allowed to be back tracked. Then the current failure node decreases the value of "AFT" field by 1 and sends a Resv_Retry message to its previous-hop (the IP address of previous-hop node could be got from Record Route object). Format of Resv_Retry message is as follows,

<Resv_Retry message> ::= <COMMON HEADER><SESSION><FLOWSPEC> <FILTER_SPEC><RECORD ROUTE><FAILURE_NODE>

When previous-hop node receives the Resv_Retry message, it would recompute the next-hop according to the information in Routing-Table. If it succeeds to find a next-hop (means the next-hop node is not in the failure node list), the node would reconstruct Resv_Req message based on the content of Resv_Retry message. Resv_Retry message includes all the necessary objects needed by Resv_Req message, so it is very easy to transform Resv_Retry message into Resv_Req message. Then the node sends Resv_Req message to the new next-hop. If no proper next-hop is found,

the node would have to release the resources reserved for the dataflow, clear its address from RECORD ROUTE object and record its address into the failure node list. Then the node starts the back-track check as described above once again.

When Resv_Req message passed through a series of intermediate nodes and reached the session-receiver, a Resource Reservation Path with guaranteed good QoS has been built up. Then, the session-receiver needs to construct CONFIRM object, and sends a Resv_Conf message to session-sender. Format of Resv_Conf message is as follows,

<Resv_Conf message> ::= <COMMON HEADER><SESSION><CONFIRM><RECORD ROUTE>[<FAILURE_NODE>]

The object in brackets is optional.

Format of "Object content" for CONFIRM object is shown in Fig.7.

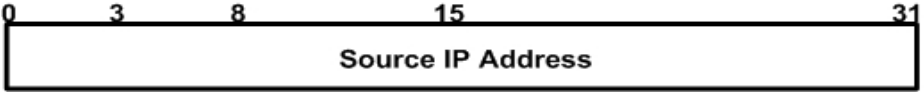


Fig. 7. Format of IPv4 CONFIRM object content

IPv6 CONFIRM object shares the same format with that of IPv4 CONFIRM object; only the corresponding address field contains a 128-bits IPv6 address. "Source IP Address" field contains the IP address of the session-sender.

In order to offer session-sender the detailed information of the reservation process, Resv_Conf message includes RECORD ROUTE object and FAILURE_NODE object, and these objects are copied from the corresponding Resv_Req message.

Resv_Conf message needs not be transmitted along the same path that Resv_Req message transferred, and could choose a totally different path. When session-sender receives Resv_Conf message, the reservation process completes. Then session-sender could make use of the path described in RECORD ROUTE object to deliver its multimedia session. When the session is over, session-sender explicitly sends messages to inform intermediate nodes to release network resources reserved for the dataflow.

3 Analysis

3.1 Successful Rate of Setting Up a Resource Reservation Path

Our scheme adds back-track capability into the process of resource reservation so that we could greatly increase the successful rate of setting up a Resource Reservation Path.

Suppose the probability of a node failing to reserve is p. Thus for an upriver node, it would successfully select a next hop (means the next-hop it selects has enough resources to reserve) with the probability of $1-p$. If back track is not allowed (RSVP does so), the successful rate of setting up a $k+1$ hops Resource Reservation Path is

$(1-p)^k$. Therefore, when p has a pretty large value, it would be very difficult to set up a multi-hop Resource Reservation Path.

To estimate the improvement of the successful rate that back-track process brought, we devised a simple network scene to analyze and simulate. In the scene, a session-sender tried to set up a five-hop path to reach the session-receiver. And at each hop, there are always 10 different candidate nodes to be chosen from. See Fig.8.

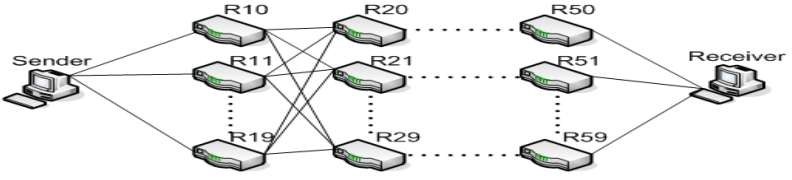


Fig. 8. Network scene 1

Suppose there is one node with insufficient resources in each of the five groups (R10-R19, R20-R29, R30-R39, R40-R49, R50-R59) of intermediate nodes, thus the probability of a node failing to reserve is $p=1/10=0.1$. If back-track is not allowed, the successful rate of setting up a Resource Reservation Path is

$$\text{Rate}_{\text{AFT=0}} = \frac{9\times9\times9\times9\times9}{10\times10\times10\times10\times10} = (1-0.1)^5 = 59.0\% \tag{1}$$

If we allow back track once during the resource reservation process (AFT=1), then the successful rate of setting up a Resource Reservation Path would be

$$\text{Rate}_{\text{AFT=1}} = \frac{9\times9\times9\times9\times9+C_5^1\times9\times9\times9\times9}{10\times10\times10\times10\times10} = 91.8\% \tag{2}$$

If we allow back track twice, the successful rate of setting up a Resource Reservation Path would be

$$\text{Rate}_{\text{AFT=2}} = \frac{9\times9\times9\times9\times9+C_5^1\times9\times9\times9\times9+C_5^2\times9\times9\times9}{10\times10\times10\times10\times10} = 99.1\% \tag{3}$$

Thus it can be seen that including back-track capability into the resource reservation process would greatly enhance the successful rate of setting up a Resource Reservation Path.

According to the same analysis, suppose there are two nodes with insufficient resources in each of the five groups of intermediate nodes, thus the probability of a node failing to reserve is $p=2/10=0.2$. If back-track is not allowed, the successful rate of setting up a Resource Reservation Path is

$$\text{Rate}_{\text{AFT=0}} = \frac{8\times8\times8\times8\times8}{10\times10\times10\times10\times10} = (1-0.2)^5 = 32.8\% \tag{4}$$

If we allow back track once, then the successful rate of setting up a Resource Reservation Path would be

$$Rate_{AFT=1} = \frac{8\times8\times8\times8\times8 + C_5^1\times8\times8\times8\times8\times2\times\frac{8}{9}}{10\times10\times10\times10\times10} = 69.2\% \quad (5)$$

If we allow back track twice, the successful rate of setting up a Resource Reservation Path would be

$$Rate_{AFT=2} = \frac{8\times8\times8\times8\times8 + C_5^1\times2\times\frac{8}{9}\times8\times8\times8\times8 + C_5^2\times2\times\frac{8}{9}\times2\times\frac{8}{9}\times8\times8\times8 + C_5^1\times2\times\frac{1}{9}\times8\times8\times8\times8}{10\times10\times10\times10\times10} = 89.9\% \quad (6)$$

To prove the validity of the computation above, we simulated network scene 1 using C++ language. Under the condition of *p=0.1* or *p=0.2*, the session-sender sent out 10,000 reservation requests and we recorded the times of successful reservation within these 10,000 requests. We repeated the test for 10 times under each condition, and got the result illustrated in Table1 and Table 2.

Table 1. Node failure probability p = 0.1

AFT / test	0	1	2
1	5969	9207	9911
2	5964	9184	9915
3	5892	9167	9929
4	5950	9170	9922
5	5931	9198	9919
6	5862	9216	9907
7	5885	9189	9912
8	5874	9202	9924
9	5952	9181	9900
10	5916	9206	9931
aver	5919	9192	9917
rate(%)	59.2	91.9	99.2

Table 2. Node failure probability p = 0.2

AFT / test	0	1	2
1	3252	6972	8940
2	3295	6945	8947
3	3300	6800	9044
4	3283	6984	9028
5	3303	6947	8971
6	3302	6923	8958
7	3244	6917	8954
8	3219	7022	8995
9	3324	6862	9024
10	3303	7010	9001
aver	3282	6938	8986
rate(%)	32.8	69.4	89.9

As can be seen, the experiment results fit well with the theoretical computation. Our scheme includes back-track capability into the resource reservation process could greatly increase the successful rate of setting up a Resource Reservation Path.

3.2 Failure Event Notice Time

In many unicast transactions such as VoIP, it is expected to notify session-sender of any abnormal event in the shortest time. Unfortunately RSVP is a receiver-oriented

protocol, it is the receiver of the session initiates reservation process, thus reservation failure event would be firstly told the receiver. And then the session-receiver would notice session-sender of the failure event. In VoIP application, the caller (session-sender) sends call-invite message which initiates resource reservation process and then entered the block state waiting for the result. Therefore, it is hoped the reservation failure event should be noticed by the caller as soon as possible. At this point, sender-oriented scheme is better than receiver-oriented one.

Failure Event Notice Time is defined as the interval between the failure event occurs and the session-sender notices the event.

To compare the Failure Event Notice Time of our scheme and RSVP, we simulated another simple network scene containing 20 nodes using NS [9] tool. The 20 nodes were aligned in a row, and a 1Mb capacity 10ms delay-time link existed between every two adjacent nodes. See Fig.9.

Fig. 9. Network scene 2

Along this 20 nodes path, we let each intermediate node in turn fail to reserve from R18 to R1 and computed the ratio of Failure Event Notice Time using RSVP and our method. The result is shown in Fig.10.

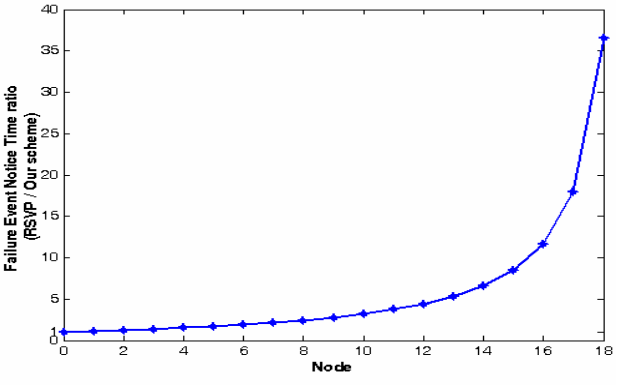


Fig. 10. Failure Event Notice Time

Fig.10. indicates when the failure node is near session-receiver end, there is no big difference between the two methods (the ratio is about 1.1). But when the failure node is near session-sender, the ratio of time using two different methods would exceed 36.5. Average the results of all the 18 intermediate nodes, the mean ratio of time using RSVP and our method is 6.346. Thus, our method greatly shortens the Failure Event Notice Time.

3.3 Overhead

Traditional RSVP is based on the thought of Soft State [10], adjacent nodes must periodically send PATH and RESV messages to maintain the resource reservation. Therefore, the overhead of the protocol [11] is high. Our scheme adopts the thought of Hard State. Hard State means the state of the network entity remains unaltered until explicit operations change it. Hence the Resource Reservation Path needs not to be refreshed and the overhead of the protocol is low.

4 Summary

In this paper, we present a new scheme to reserve resources on Packet Based Network. The scheme adds back-track capability into the reservation process, greatly enhancing the successful rate of setting up a Resource Reservation Path. Besides, it is based on sender-oriented mode, and could make the session-sender notice any abnormal event as early as possible. Thus, our scheme is particularly fit for resource reservation of sender-control unicast transactions like VoIP.

Acknowledgement

This work was supported by the Digital Olympics in Key Technologies R&D Program (Grant No. 2003BA904B06).

Reference

1. R.Braden, L.Zhang, “Resource Reservation Protocol (RSVP)—version 1 Function Specification[R]”, IETF RFC2205, 1997.
2. Goto Yukinori, Nagano Nakaba, Araki Keijiro, “Proposal of VoD system model with RSVP on the Internet”, in IPSJ SIGNotes Distributed Processing System No.088, 2001.
3. Haim Zlatokrilov, Hanoch Levy, “Packet Dispersion and the Quality of Voice over IP Applications in IP Networks”, in Proceedings of INFOCOM2004, 2004.
4. ITU-T, “A high quality low complexity algorithm for packet loss concealment with G.711”, in recommendation G.711 Appendix I (1999).
5. Josef Glasmann, Wolfgang Kellerer, “Service Architectures in H.323 and SIP – A Comparison”, in IEEE Communications Surveys & Tutorials, 2003.
6. H. Schulzrinne, S. Casner, “RTP: A Transport Protocol for Real-Time Applications”, IETF RFC1889, 1996.
7. J.Wroclawski, “The Use of RSVP with IETF Integrated Services”, IETF RFC2210, 1997.
8. D.Awduche, L.Berger, “RSVP-TE: Extensions to RSVP for LSP Tunnels”, IETF RFC3209, 2001.
9. http://www.isi.edu/nsnam/ns
10. S.Raman, S.McCanne,“A Model, Analysis, and Protocol Framework for Soft State-based Communication”, in Proceedings of SIGCOMM’99, 1999.
11. L.Wang, A. Terzis, L.Zhang, “A New Proposal for RSVP Refreshes”, in Proc. IEEE Int. Conference on Network Protocols, 1999.

Available Bandwidth Measurement Schemes over Networks

Fang Qi[1], Jin Zheng[1], Weijia Jia[2], and Guojun Wang[1]

[1] School of Information Science and Engineering, Central South University, Changsha, P.R. China
{csqifang,zhengjin,csgjwang}@mail.csu.edu.cn
[2] Department of Computer Science, City University of Hong Kong, 83 Tat Chee Ave, Kowloon, Hong Kong, SAR China
itjia@cityu.edu.hk

Abstract. In next generation network (NGN), end-to-end QoS is one of the critical issues for real-time multimedia communications and applications. Such applications are sensitive to the availability of bandwidth for a given path. Measuring bandwidth has attracted considerable research efforts in the networking community. This paper intends to the contribution to the available bandwidth measurement for NGN where interoperability and end-to-end QoS are primary objectives. Our discussions of the algorithms focus on the following properties: (1) Efficiency: the applications should not wait too long for data convergence due to traffic that may interfere with the networks; (2) High accuracy: our algorithms should perform error control/cancellation to achieve high measurement accuracy; (3) Interoperability: our algorithms should adaptively apply to different types of networks.

1 Introduction

Next Generation Network (NGN) represents the convergence of multiple independent broadband networks into a single, unified network including high speed voice, video and data services. Providing end-to-end QoS guarantees to real-time multimedia communications and applications is an important objective in designing the next-generation networks. Such applications are sensitive to the availability of bandwidth for a given path.

Detecting link capacity and measuring available bandwidth on the network is critical for the success of streaming applications such as videoconference, dynamic server selection and congestion control transports etc. For instance, streaming applications normally require pre-knowledge of the both metrics in order to make certain decisions (such as admission control and real-time video streaming). Another example is server selection [12]. Clients are typically free to connect to any of the mirror servers. One criterion for choosing the "best" server is the available bandwidth. For congestion control application, the available bandwidth can be used to determine the congestion window size and the slow start threshold in a TCP sender at connection setup [1, 2] or after a congestion period [8, 16]. It can also be used to optimize application specific routing in overlay networks [3]. It is challenging to improve the link capacity

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 931 – 940, 2005.

detection and end-to-end available bandwidth measurement methods which provide service in the differentiated service and guaranteed service. It is also very difficult to apply to heterogeneous (wired and wireless) networks where efficiency, accuracy and interoperability are primarily important for NGN, an active area of future research.

This paper studies a set of algorithms for link capacity detection and end-to-end available bandwidth measurement. Techniques for estimating available bandwidth can be classified into two categories: passive measurement [8] and active probing [4, 10, 14, 15]. Passive measurements use the trace history of existing data transfers. While potentially very efficient and accurate, this approach is limited to the network paths that have recently transmitted user's traffic. Active probing method has been proved to be successful in determining the instantaneous link capacity and available bandwidth. The advantage is that it can probe the path on demand and requires no additional work for the inter-routers. However, it is inaccurate and may introduce some network traffic that can quickly become a significant part of the total traffic on the path. As a result, the traffic may influence the performance of the node on the path to a destination. Therefore, an ideal probing scheme should provide an accurate estimation of the link capacity and available bandwidth for a path effectively while imposing less traffic to the path. Typical methods of active probing scheme are packet pair/train Dispersion (PPTD) [4], Self-Loading Periodic Streams (SLoPS) [10], and Trains of Packet Pairs (TOPP) [15].

2 Basic Definition

In the following, we give the formal definitions for link capacities and available bandwidth. Suppose that each link L_i on the path can transmit data with rate C_i-bps, n is the number of hops in the path, then the link capacities C of the path is defined as:

$$C = \min_{i=1,\ldots,n} C_i \tag{1}$$

However, for a path, the occurrence of available bandwidth may vary with time. A generally accepted opinion is based on the current link utilization. Suppose during a time interval $[t, t+\tau)$, the link utilization on L_i is $u_i (0 \le u_i \le 1)$, then the available bandwidth of L_i is defined as:

$$A_i = C_i(1-u_i) \tag{2}$$

Extending this concept to the entire path, the end-to-end available bandwidth during time interval $[t, t+\tau)$ is the minimum available bandwidth among all the links (i.e., available bandwidth):

$$A = \min_{i=1,\ldots,n} A_i = \min_{i=1,\ldots,n} C_i(1-u_i) \tag{3}$$

To further illustrate the difference of available bandwidth and link capacities, we give a graphical illustration as shown in Fig. 1, where the path includes three hops, the link capacities of the path is C_1, and the available bandwidth of the path is A_3.

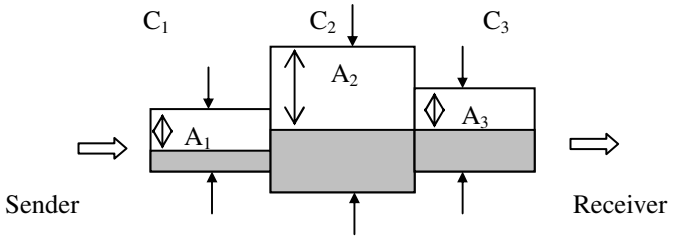


Fig. 1. Capacity and available bandwidth between Sender and Receiver

3 Active Bandwidth Estimation Approaches

This section describes existing bandwidth measurement techniques for estimating capacity and available bandwidth in individual hops and end-to-end paths. We focus on these major techniques: Packet Pair/Train Dispersion (PPTD) probing, Self-Loading Periodic Streams (SLoPS), and Trains of Packet Pairs (TOPP). PPTD and Packet tailgating usually estimates end-to-end capacity, and SLoPS and TOPP estimate end-to-end available bandwidth.

3.1 Packet Pair/Train Dispersion (PPTD) Probing

Packet pair [6] probing is used to detect the end-to-end capacity of a path. The source sends multiple packet pairs to the receiver. Each packet pair consists of two packets of the same size sent back-to-back. The dispersion of a packet pair at a specific link of the path is the time distance between the last bit of each packet. It is based on a fact that the spacing between the packet pairs is determined by the bottleneck link and preserved by other higher-bandwidth links. By measuring the time difference between the round-trip-times (RTT) of the back-to-back packets from one end of the link to the other, those approaches can estimate the bottleneck link capacity.

The principle of the bottleneck spacing effect is when two packets with time separation S are transmitted over a link with a service time $Qb>S$, then as the packets leave the link they will be separated by $\Delta R=Qb$. If $Qb \leq S$, then $\Delta R = S$, indicating that the link could service (i.e. transmit) the packets at the rate they arrived. Using the size of the packets, b and the time separation ΔR, the experienced bandwidth across that link can be estimated as $m=b/\Delta R$ and m is used directly as the available bandwidth estimate.

There are a few potential problems when timing information is obtained from acknowledgement packets instead of directly from the probe packets. First, self-interference of the probe packets and their acknowledgements on shared media links may interfere with the networks. Moreover, these methods are slow in making the measurement decision because they need a converging data to make a decision. Second, although simple in principle, this technique may produce widely varied estimation and result in rather inaccurate estimations. Finally, Packet Pair requires FQ scheduling in routers while the common scheduling discipline in the Internet is First-Come-First-Served which limits its applicability in the Internet.

3.2 Self-loading Periodic Streams (SLoPS)

SLoPS is a recent measurement methodology for measuring end-to-end available bandwidth [5]. If there is an increasing trend, the send rate of the packet train is higher than the available bandwidth. On the other hand, if there is no trend, the send rate is lower than the available bandwidth. The actual estimation procedure is iterative. First a probe packet train is sent at a certain rate. Depending on whether a trend in the one-way delays of that train is detected or not, the send rate for the next train is decreased or increased by a factor of two. When the series of sending rates converges, the iterations end. Pathload is the implementation of the SLoPS methodology.

We model the end-to-end delay of a link/path through probe traffic delay plus some possible queuing delay. Let s be a probe packet (the packet size is also denoted as s). Consider a path from a sender to a receiver that consists of n links (L_1, …, L_n) and the capacity of link L_i is C_i. The end-to-end delay for the packet s from the sender to the receiver is represented as:

$$D = \sum_{i=1}^{n} \left(\frac{s}{C_i} + \frac{q_i}{C_i} + v_i\right) \tag{4}$$

Where v_i (i = 1, 2, …, n) is the constant delay for link L_i which includes the propagation delay and the processing delay (consisting of the time for a router to look up routes in a routing table and the time for forwarding a packet). q_i is the queue size at link L_i before the arrival of s (i.e., not including s itself), thus q_i / C_i is the queuing delay for s. The One-way Delay (OWD) from SND to RCV of packet k is defined as

$$D^k = \sum_{i=1}^{n} \left(\frac{s}{C_i} + \frac{q_i^k}{C_i} + v_i\right) = \sum_{i=1}^{n} \left(\frac{s}{C_i} + d_i^k\right) \tag{5}$$

The OWD difference between two successive packets k and k+1 is

$$\Delta D^k = \sum_{i=1}^{n} \frac{q_i^k}{C_i} = \sum_{i=1}^{n} d_i^k \tag{6}$$

Although SLoPS [10] overcomes the problems such as inaccuracy existing in several other bandwidth probing methods, however, it needs a lot of packet streams to predict the increasing or decreasing the trend of the delay. The measurement time is very long and this may be unsuitable for many real-time applications. Because only the delay trend is taken for adjusting the probe stream rate, many probing information (such as probe delay of each packet) is wasted.

3.3 Trains of Packet Pairs (TOPP)

Melander et al. proposed a measurement methodology to estimate the available bandwidth of a network path [15]. TOPP has probing phase and analysis phase. The probe traffic is generated in the following way. Starting at some rate o^{min}, k well separated *pairs* of equally sized probe packets are sent to the destination host. After those k pairs have been sent, the offered rate o is increased by Δo and another set of k probe pairs are sent. Then o is increased again (by the same amount o) and another set of k probe pairs are sent. This goes on until the offered rate reaches some rate o^{max} which marks the end of the probe phase. Figure 2 illustrates the probe phase. In Fig. 2, each black

dot corresponds to a pair of probe packets [15]. T_s is the time spacing between consecutive pairs, b is the size of the probe packets and Δ t is the time spacing between packets in a pair (such that the offered rate o is achieved). In TOPP's analysis phase, *m* is simply an intermediate value used in the calculation of the final estimate as defined in Section 3.1.

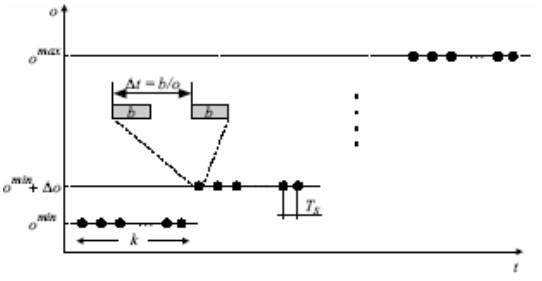


Fig. 2. A TOPP probe sequence, showing the stepwise increase in offered bandwidth over time

TOPP has many benefits because of tractable properties over earlier methods. The measurements in TOPP are performed one-way in order to avoid problems with asymmetric routes and self-interference of the probe traffic on shared media network. Otherwise, it does not require fair queuing policy in the routers like the traditional packet pair technique [10] does. Finally, TOPP [6] provides a theoretical model for the relationship between the available bandwidth and probing packet spacing at both end points. It sends packet pairs to a destination at increasing rates. From the relation between input and output rates of different packet pairs, one can estimate the available bandwidth.

TOPP has some disadvantages. It uses packet pair to estimate the share for new traffics; however, the measurement errors propagated seriously affect the measurement accuracy. Otherwise, TOPP dose not make use of delay correlation information obtainable from packet trains with closely spaced packets.

3.4 Comparison of Bandwidth Estimation Approaches

Packet pair technology [11] such as Bprobe [6] and Pathchar [7] are efficient methods for measuring the bottleneck link capacity. Most of them use the packet pair dispersion technique:

Bprobe [6] uses packet pair dispersion to estimate the capacity of a path. Bprobe processes packet pair measurements with an interesting "union and intersection filtering" technique, in an attempt to discard packet pair measurements affected by cross traffic. Pathrate [7] collects many packet pair measurements using various probing packet sizes. Analyzing the distribution of the resulting measurements reveals all local modes, one of which typically relates to the capacity of the path.

C-probe [6] is an early tool intended to measure the available bandwidth of a network path. It is similar to Packet Pair in that it is based on dispersion of probe packets. The main difference is that C-probe sends a train (or stream) of eight probe packets back-to-back instead of a pair of packets. However, C-probe can only find the

available bandwidth if it coincides with the link bottleneck on a path. By assuming that "almost-fair" queuing occurs during the short packet sequence, it estimates the available bandwidth based on the dispersion of long packet trains at the receiver.

More recently, two new tools have been proposed for available bandwidth estimation: PathChirp [17], IGI (Initial Gap Increasing) [9] and PTR (Packet Transmission Rate) [9]. These tools modify "self-loading" methodology of SLoPS, using different probing packet stream patterns. The main objective in IGI and pathChirp is to achieve similar accuracy with pathload but with shorter measurement latency.

The Initial Gap Increasing (IGI) algorithm uses the information about changes in gap values of a packet train to estimate the competing bandwidth on the tight link of the path. The Packet Transmission Rate (PTR) method uses the average rate of the packet train as an estimate of the available bandwidth. The IGI method loses accuracy if the tight link is not the bottleneck link, or if there is significant competing traffic on links following the tight link. The Packet Transmission Rate (PTR) is much less sensitive to the presence of traffic on links other than the tight link.

PathChirp estimates the available bandwidth along a path by launching a number of packet chirps from sender to receiver and then conducting a statistical analysis at the receiver. The current algorithm of PathChirp for available bandwidth estimation mainly uses information about whether delays are increasing or decreasing in the signatures.

To measure the efficiency of the tools and compare with them, experiment computes the average number of bytes over 25 runs that each tool takes to provide estimates accurate to 10Mbps. The experiment sets the available bandwidth to a constant value using iperf CBR UDP traffic The result in Table 1 indicate pathChirp needs less than 10% of the bytes that pathload uses. In addition to the average number of bytes the too tools use to achieve the desired accuracy, Table 1 provides the 10%-90% values of pathChirp estimates and the average of Pathload's minimum and maximum bounds of available bandwidth. Observe that pathChirp's estimates have a consistent negative bias, implying that its measurements are conservative. These results demonstrate pathChirp is a rapid estimation method and has only a light probing load [17]. The main features, advantages and disadvantages of available bandwidth measurements approaches are listed in Table 2. Also Efficiency, accuracy and interoperability of these algorithms will be described in the table.

Table 1. Efficiency comparison of pathChirp and pathload with iperf CBR crosstrafic

Available Bandwidth active	Pathchirp Efficiency	Pathload Efficiency	Pathchirp Accuracy (10-90%)	Pathload Accuracy (ava. Min-max bounds)
30Mbps	0.41Mb	4.3Mb	14-25 Mbps	16-34 Mbps
50Mbps	0.32Mb	5.5Mb	49-56 Mbps	40-49 Mbps
70Mbps	0.46Mb	9.9Mb	59-68 Mbps	63-70 Mbps

Table 2. Summary of bottleneck link capacity and available bandwidth measurements approaches

Tool	Main features	Advantages	Disadvantages
Bprobe [R.L.Carter et al. 1996]	packet pair measurements with an "union and intersection filtering"	(1)Easy to deploy (2)Simple algorithm	(1)Efficiency is relative low because of self-interference and requiring a converging data
Pathrate [C.Dovrolis. 2001]	Measurements using various probing packet sizes by collecting many packet pairs		(2)Rather inaccurate because of widely varied estimation (3)Interoperability is low because requires FQ scheduling
Cprobe [R.L.Carter et al. 1996]	Measure available bandwidth by sending packet trains		
Pathload [M.Jain. et al. 2002]	Looking for trends in the one-way packet delays of a packet train by sending SLoPS	(1)Non-intrusive (2)Relative high accurate	(1)The average measure time is relative long (2)Unsuitable for many real-time Applications because of sending a lot of packet streams
TOPP[B.Melander et al.2000]	Based on a theoretical model for the relationship between the available bandwidth and probing packet spacing.	(1)Efficiency is high having no self-interference (2)Interoperability is relative high because requires FCFS scheduling	(1)Accuracy is effected when the measurement errors propagated seriously using theoretical model (2)Wasting delay correlation information (3)Interoperability is low with FCFS
IGI[N. Hu.2003]	Using the information about changes in gap values of a packet train to estimate	Similar accuracy with pathload but with shorter measurement time	Accuracy is lost when the tight link is not the bottleneck link
PathChirp [V. Ribeiro et al.2003]	Estimating by launching a number of packet chirps	Similar accuracy with a rapid and light probing load	Have not fully exploit Chirp delay information

4 Proposed Passive Available Bandwidth Estimation Approach

All active measurement tools inject probing traffic in the network and thus are all intrusive to some degree. Specifically, we say that an active measurement tool is intrusive when its average probing traffic rate during the measurement process is significant compared to the available bandwidth in the path.

Apparently, active measurement tools are not suitable for the wireless channel. This is because wireless channels have low reliability, and time varying signal. The dynamics of wireless link bandwidth and SNR at mobile receivers depends on many factors, such as noise, distance, roaming speed, multi-path interference. The signal-to-noise ratio (SNR) γ is a variable affected by fading. And a corresponding measure in a wireless communication environment is the received bit-energy-to-noise ratio γ is in direct proportion to the square of Rayleigh-distributed random process envelop in the flat fading channel model [13].

$$p_{\gamma}(\gamma) = \frac{1}{\gamma} e^{-\gamma/\gamma_0}, \gamma_0 = E(\gamma) \tag{7}$$

For example in DPSK without fading, the bit error rate can be denoted in [13] as

$$\mathrm{BER} = \frac{1}{2}\exp(-\gamma) \tag{8}$$

Then, in DPSK with Rayleigh fading, the average bit error rate can be denoted in [13] as packet propagation time across the channel, respectively.

$$\mathrm{BER\_ave} = \int_0^{\infty} \frac{1}{\gamma} e^{-\gamma/\gamma_0} \frac{1}{2}\exp(-\gamma) d\gamma = \frac{1}{2+\gamma_0} \tag{9}$$

Assume that Tw is time that spending during binary exponential backoff in CSMA/CA MAC protocol. Assume the capacity of the wireless link is C. The relationship between packet error rate (PER) and the bit error rate (BER) depends on the channel coding scheme. Assume that two samples of the process are almost independent (fast fading), or in other words, is no error-correction coding applied and the number of bits in a packet *n*.

$$\mathrm{PER}=1-(1-\mathrm{BER})^n \tag{10}$$

Assume hat tp and ta are the times to transmit a packet and to transmit an ACK. Furthermore, tproc and tprop are the packet processing time at the end-hosts and the channel efficiency can be expressed as follows:

$$\mathrm{effi\_link}=\frac{E(t_p)(1-PER)}{(1-PER)\times E(t_p+t_a+2t_{proc}+2t_{prop})+PER\times E(t_p+t_o+t_{proc}+t_{prop})+E(t_w)} \tag{11}$$

We can estimate the available bandwidth as

$$\mathrm{Avai\_bandwidth\_ave}=\mathrm{effi\_link}\times C \tag{12}$$

The method satisfies the efficiency of scheme because that the estimation scheme is passive measurement, not using any probing messages that may interfere with the networks and the method need not wait too long for data convergence. The accuracy of the method is expected following researching.

5 Conclusions

The primary objectives of end-to-end QoS are to provide the priority including dedicated bandwidth, controlled jitter and latency (required by some real-time and interac-

tive traffic), and improved loss characteristics. This paper studied the detecting link capacity and measuring available bandwidth methods and discusses the efficiency, accuracy and interoperability of these algorithms. These algorithms can satisfy the need of end-to-end QoS to some extent. We are currently investigating some problems and challenges: (1) intrusiveness of bandwidth detection to the networks should be alleviated. Available bandwidth measurement tools including PPTD tools create short traffic bursts of high rate which sometimes higher than the available bandwidth in the path. We consider that the average probing traffic rate during the measurement process is not significant compared to the available bandwidth in the path; (2) accuracy of bandwidth estimation techniques must be improved, especially in high bandwidth paths; (3) various service disciplines for the link access should be considered and (4) interoperability among different types of networks must be handled, especially, the schemes satisfy the need of mobile and grid computing and communications.

Acknowledgements. This effort is sponsored by the National Basic Research Program (973) MOST of China under Grant No. 2003CB317003 and City University of Hong Kong strategic grants 7001709 and 7001587.

References

1. M. Allman and V.Paxson, “On estimating end-to-end network path properties”, In Proceedings of ACM SIGCOMM, pages 263–273, Cambridge, MA, USA, August 1999.
2. M. Aron and P. Druschel, “TCP: Improving startup dynamics by adaptive timers and congestion control”, Technical Report TR98-318, Rice University, Computer Science, 1998.
3. D.G. Andersen, H.Balakrishnan, M. Frans Kaashoek, and R.Morris, “Resilient overlay networks”, Proceedings of ACM Symposium on Operating Systems Principles, Ban, Canada, October 2001.
4. J. C. Bolot, “End-to-end Packet Delay and Loss Behavior in the Internet”, Proc. ACM SIGCOM, Sept. 1993, pp.289-298.
5. Banerjee and A. K. Agrawala, “Estimating Available Capacity of a Network Connection”, Proceedings IEEE International Conference on Networks, Sept. 2001.
6. L.Carter and M.E.Crovella, “Measuring Bottleneck Link Speed in Packet-Switched Networks”. Performance Evaluation, vol.27, no.28,PP.297-318,1996
7. C.Dovrolis, P.Ramanathan, and D.Moore. “What do Packet Dispersion Techniques Measure?”in proc. ACM SIGCOMM, Aug. 2001,pp.905-914.
8. J.C. Hoe. Improving the start-up behavior of a congestion control scheme for TCP. In Proceedings of ACM SIGCOMM, pages 270–280, Stanford, CA, USA, 1996.
9. N. Hu and P. Steenkiste, “Evaluation and Characterization of Available Bandwidth Probing Techniques,” IEEE Journal on Selected Areas in Communications, 2003.
10. M.Jain , C. Dovrolis, “End-to-End Available Bandwidth Methodology Dynamics, and Relation with TCP Throughput.” In Proc.ACM SIGCOMM, August 2002,pp. 295-308.
11. S.Keshav.A Control-Theoretic Approach to Flow Control. In Proceedings ACM SIGCOMM.Sept.1991, pp3-15
12. D. Katabi and J. Wroclawski. A framework for scalable global IPanycast (GIA). In Proceedings of ACM SIGCOMM, Stockholm, Sweden, August 2000.
13. William C. Y. Lee. Mobile Communications Engineering: Theory and Applications. McGraw-Hill Education, 1998

14. K.Lai and M.Baker, “Measuring Link Bandwidth Using a Deterministic Model of Packet Delay”, in Proc. ACM SIGCOM, Sept 2000, pp 283-294
15. B.Melander, M. Bjorkman, and P.Gunningberg, “A new end-to-end probing and analysis method for estimating bandwidth bottlenecks”, Global Internet Symposium, Dec 2000, pp.415-420.
16. S. Mascolo, C.Casetti, M.Gerla, M. Y. Sanadidi, and R. Wang. TCP Westwood: Bandwidth estimation for enhanced transport over wireless links. In Proceedings of Mobile Computing and Networking, pages 287–297, 2001.
17. V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cottrell, “pathChirp: Efficient Available Bandwidth Estimation for Network Paths,” in Proceedings of Passive and Active Measurements (PAM) workshop, Apr.2003.
18. S.Seshan, M.Stemm, R.H.Katz. “SPAND: Shared Passive Network Performance discovery.” Proceedings of 1st Usenix Symposium on Internet Technologies and Systems (USITS’97) Monterey, CA, USA, December 1997.

Chaotic Dynamic Analysis of MPEG-4 Video Traffic and Its Influence on Packet Loss Ratio

Fei Ge[1,2], Yang Cao[2,3], and Yuan-ni Wang[4]

[1] School of Electrical Engineering, Wuhan University, 430072 Wuhan, PR China
gfeei@tom.com
[2] School of Electronic Information, Wuhan University, 430079 Wuhan, PR China
[3] State Key Laboratory of Software Engineering, 430072 Wuhan, PR China
caoyang@whu.edu.cn
[4] Computer Science Department, China Universty of Geoscience , 430074 Wuhan, PR China

Abstract. This paper studies the chaotic dynamic behavior of MPEG-4 video trace and the relationship between the chaotic dynamic characteristic and the QoS parameter. The main research motive is to analysis and to verify the inherent character of MPEG-4 video, and to investigate whether it associates with QoS. The power spectral density estimation of the video trace describes its $1/f^{\beta}$ and periodic characteristics. The principal components analysis of the reconstructed space dimension shows only several principal components can be the representation of all. The correlation dimension proves its fractal characteristic. The video trace is divided into many parts whose largest Lyapunov exponent values, which can be called largest Lyapunov exponent spectrum, are separately calculated using the small data sets method. The largest Lyapunov exponent spectrum shows there exists abundant chaos in MPEG-4 video. Through the comparison of the largest Lyapunov exponent spectrum and the packet loss ratio when transporting the video, the conclusion that there exists certain strong relation between them can be made. The conclusion is significant in QoS constrained video transport.

Keywords: MPEG-4 video traffic, chaotic dynamic analysis, packet loss ratio.

1 Introduction

The network traffic modeling plays an important role in network application field, while the traffic behavior analysis is of help to traffic modeling. As network traffic is expected to offer more and more video services, the video traffic is coming in possession of greater network bandwidth. Although there are several different video code standards, the video community is increasingly using the Moving Pictures Expert Group (MPEG) standards. The MPEG standards achieves high compression ratios by exploiting the reduction of both temporal correlation in *inter*-frame coding by means of motion compensation, and spatial correlation in *intra*-frame coding, using spatial transforms. This produces a high variability in the offered load as Intra frames usually need from 2 to 5 times the number of bits necessary for inter frame coded frames (P and B frames in MPEG terminology). So the correct characterization analysis of

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 941 – 950, 2005.

this type of traffic, which contains inherently dynamic behavior, is increasing its importance. Accurate characters of the transported MPEG traffic are necessary to achieving efficient resources allocation to QoS constraints and to designing online admission control strategies.

In this paper, our focus is mainly on applying the chaotic time series analysis method to characterize the MPEG-4 trace, and on studying the different influence of different character on QoS when transporting video trace on IP network. The basic statistics of the MPEG video trace- Star Wars IV, which can be available from the Telecommunication Networks Group, Technical University of Berlin [3], such as bandwidth distribution, mean, variance, burstiness, autocorrelation, and Fourier spectrum is described in [1], also the marginal distribution of the video bandwidth and the degree of long-range dependence are measured accurately. Because self-similar traffic model can not capture the whole characteristics of the traffic [4] and chaos theory offers an alternative approach to stochastic process, some exploration such as the estimated value of the maximum Lyapunov exponent, the estimation of embedding dimension of video traffic had been done in [2].

Section I is some introduction to the MPEG-4 video trace which is analysised in the paper, and the basic space reconstruction method is mentioned. Section II analysis the main chaotic dynamic characteristics of the video trace. Section III compares the QoS parameter and the largest Lyapunov exponent of the video trace and calculates the correlation coefficient. Section VIII concludes the whole research work.

2 Charactering MPEG4 Traffic Using Chaotic Time Series Method

The video trace we have studied is the high quality (HI) entertainment movie Star War IV outputs. The video trace consists of the number of bytes and the transport time (40ms, corresponding to 25 frames per second) of every frame(including I, P,B Frame type). Table 1 summarizes various statistics[3] of the trace. The complete video trace is shown in Fig. 1.

Table 1. Parameters for Star War IV video trace

Item	Unit	value
compression ratio	YUV:MP4	27.62
Video run time	msec	3599840
Video Frames	-	89998
mean frame size	byte	1376
var frame size	-	816283
CoV of frame size	-	0.66
min frame size	byte	26
max frame size	byte	9370
Mean bit rate	bit/sec	275285
Peak bit rate	bit/sec	1874000

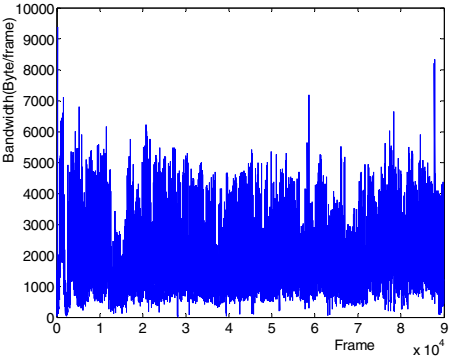


Fig. 1. Time series of entire video sequence, which contains 89998 frames

Space reconstruction can restore the traffic dynamic behavior. The celebrated theorem of Takens guarantees that, for a sufficiently long time series of scalar observations of an n-dimensional dynamical system with a C^2 measurement function, one may recreate the underlying dynamics (up to homeomorphism) with a time delay embedding. Chaotic time series theory offers a new analysis methodology to handle irregular time series, such as traffic measurements. First attempts to apply this approach to the network traffic analysis demonstrated serious difficulties as well as some promising results (see [6] and references therein). In chaotic time series analysis we view the signal $\{x_i\}$ as the one-dimensional projection of a dynamical system operating in a space of vectors $\boldsymbol{Y}_i$ of larger dimension:

$$\boldsymbol{Y}_i = \left(x_i, x_{i+\tau}, \cdots, x_{i+(m-1)\tau}\right). \tag{1}$$

Here m is the dimension of the underlying dynamical system, and τ is a "delay time", or the correlation length of series $\{x_i\}$.

2.1 Power Spectral Density (PSD) Estimate

Power spectral density can be used to differ period signal from stochastic signal according to its peak and wide bandwidth background, also it can detect the frequency component of the video trace. We estimate the Power Spectral Density of the trace with Welch's method[5]. Welch's method for estimating PSD is carried out by dividing the time signal into successive blocks, and averaging squared-magnitude DFTs of the signal blocks. Let $x_m(n)=x(n+mN)$, n=0,1, …,N-1,denote the mth block of the signal $x \in C^{MN}$, with M denoting the number of blocks. Then the Welch PSD estimate is given by

$$\hat{R}_x(\omega_k) = \frac{1}{M}\sum_{m=0}^{M-1} |\, DFT_k(x_m)\,|^2 \triangleq \{\,|\, X_m(\omega_k)^2\,|\,\}_m \tag{2}$$

where "$\{.\}_m$" denotes time averaging across blocks (or frames) of data indexed by m. The PSD of the Star War IV time series is shown in Fig.2(a) and the logarithmic frequency coordinates PSD plot in Fig.2 (b).

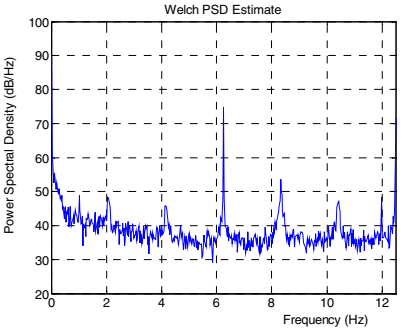


Fig. 2(a). PSD of video trace-Star War IV

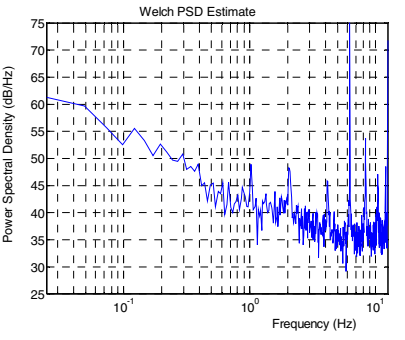


Fig. 2(b). PSD of video trace-Star War IV with logarithmic frequency

From the Fig.2, we can observe the PSD has seven obvious peaks and obvious $1/f^{\beta}$ noise characteristics. So the conclusion that the MPEG4 video traces have at least two internal features can be made. One feature is the periodic behavior what the PSD peaks showed in Fig.2(a). The peaks exist at frequency points 2.1**n* Hz (*n*=0, … ,6), so the period is approximate 0.5s. Other researcher[1] did not mention this feature. Through the analysis of coding process of MPEG-4, we infer that it is the special 12 frames Group of Picture (GoP) that leads to the periodic behavior. The transporting time of one frame is 40 ms, so the transporting time of one GoP is 40*12=480ms. This interval is approximately equal to the period time which PSD shows. And because the type of every GoP's first frame is I type whose compression ratio is low, the type of other frames are P type or B type whose compression ratio are higher, the period behavior is strengthened. Other video traces such as H.263 traces, which has different coding method, hasn't this feature. The other feature showed in Fig.2 (b) is the wide bandwidth quality like $1/f^{\beta}$ noise process. The wide bandwidth characteristic, which can be observed in VBR time series, also implies there may exsist chaotic behavior in MPEG-4 video time series.

2.2 Principal Components Analysis (PCA)

The PCA method consists in applying a linear transformation to the original data space into a feature space, where the data set may be represented by a reduced number of "effective" features and yet retain most of the intrinsic information content of the data. In other words, the data set undergoes a dimensionality reduction. So this method, which is also a well-known technique in multivariate data analysis [7], is an additional method that has been used for clarification of the dimensionality problem. We applied the PCA method to the reconstructed Star War IV traffic data, where m=10 and τ=1. Fig.3 (a) and Fig.3 (b) show the results of the PCA analysis of the MPEG-4 trace and the lbl-pkt-5[11] trace, which contains an hour's worth of all wide-area traffic between the Lawrence Berkeley Laboratory and the rest of the world. Although there are some difference between two figures, the result that the first four principle components possess above 90 percent of the trace is obvious.

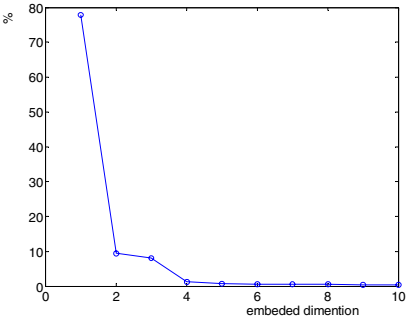


Fig. 3(a). principle component of Star War IV video trace

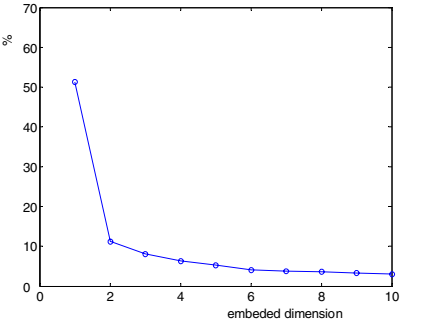


Fig. 3(b). principle component of lbl-pkt-5, a wide area traffic from the Lawrence Berkeley Laboratory

The PCA result offers the optimal choice of embedding parameters that the first four principle components, can reconstruct the underlying dynamic evolution from MPEG-4 time series.

2.3 Estimating the Correlation Dimension

The dimension of a dynamic system is an indication of the number of freedom of that system. The correlation dimension is one of the most important parameters that indicate the number of initial conditions that specify a solution in non-linear chaos dynamic. When the underlying structure of the time series is unknown, it is difficult to robustly estimate the correlation dimension. The Grassberger-Procaccia algorithm (GPA) [8] appears to be the most popular method used to quantify chaos. This is probably due to the simplicity of the algorithm and the fact that the same intermediate calculations are used to estimate both dimension and entropy.

From the result of PCA results, we reconstruct the dynamic system with m=10 and τ=1, and estimate the correlation dimension using GPA. The correlation integral can be estimated by

$$C_n(r) = \frac{1}{N^2} \sum_{i,j=1}^{N} \theta(r - | y_i - y_j |) \tag{3}$$

within the distance between two points given by

$$| y_i - y_j | = \max_{1 \le k \le n} | y_{ik} - y_{jk} | \tag{4}$$

Here θ is Heaviside function

$$\theta(x) = \begin{cases} 0, & x \le 0 \\ 1, & x > 0 \end{cases} \tag{5}$$

The value $C_n(r)$ is the empirical probability that a randomly chosen pair of points $(\boldsymbol{Y}_i : \boldsymbol{Y}_j)$ will be separated by a distance less or equal to r.

To estimate the embedding dimension D_{GP}, one computes $C_n(r)$ for r ranging from 0 to the largest possible value and for m increasing from 1 up to the largest possible value.

$$D_{GP} \approx \frac{\ln C_n(r)}{\ln r} \tag{6}$$

Fig.4 shows the correlation dimension estimate of the MPEG-4 video phase space. The correlation dimension, *lnC(r)/lnr*, is about 4, which is according to the PCA result.

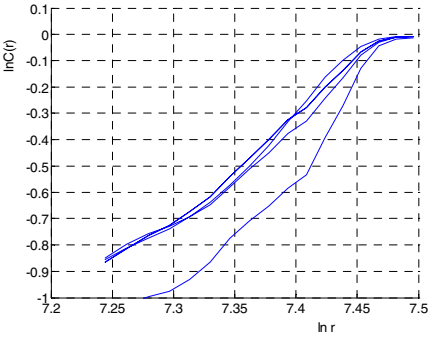


Fig. 4. The correlation estimate of video trace

2.4 Largest Lyapunov Exponent Estimation

The practical significance of the GPA is questionable, and the Lyapunov exponents may provide a more useful characterization of chaotic systems[9]. The Lyapunov exponent (LE) gives the rate of exponential divergence from perturbed initial conditions. The usual test for chaos is calculation of the largest Lyapunov exponent. A positive largest Lyapunov exponent indicates chaos.

There are several methods to calculate the Lyapunov exponent, mainly are Wolf method and Jocobian method. When one has access to the equations generating the chaos, this is relatively easy to do. When one only has access to an experimental data record, such a calculation is difficult. In most case, the largest Lyapunov exponent is sufficient to character the chaos dynamics, we use the small data sets method[9] for reference to calculate the largest Lyapunov exponent of the video trace. But our method is not completely the same with M.T.Rosenstein[9], and takes a more effective method that makes some progress.

Because the video time series is serious long(89998 frames), here we adopt an improved method to estimate the largest Lyapunov exponent of the time serious. After reconstructed the phase space using the embedded dimension m=10 andτ=1, we obtain 89988 points of 10 dimension space. Then we divide 89988 vectors into 90 trace blocks, 1000 space points which is enough when using the small data sets method[9] in every block. Then we separately calculate the 90 largest Lyapunov exponents of every block, and average the 90 values to obtain the largest Lyapunov exponent of the whole video time series. Several reasons to make such improvement are: the chaos dynamic behavior of every block of the MPEG4 should also be emphasized due to the

continuously changeable transport enviornment; the character of every block should be according to the character of the whole except a little departure; the small data set method is quite suitable to calculate the data vector of every block. The following describes the detail steps to analysis the largest Lyapunov exponent.

The first step of our approach is to reconstruct the dynamic space from the single video time series $\{x_i\}$. The reconstructed trajectory, $\boldsymbol{Y}$, can be expressed as a matrix where each row is a phase-space vector. That is,

$$\boldsymbol{Y}=[\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_M]^T \tag{7}$$

where $\boldsymbol{Y}_i$ is the state of the system at discrete time i. For the *89998*-point MPEG4 time series, $\{x_1, x_2, \ldots, x_{89998}\}$, each $\boldsymbol{Y}i$ is given by

$$\boldsymbol{Y}_i = [x_i \;\; x_{i+d} \;\; \ldots \;\; x_{i+(m-1)d}] \tag{8}$$

where d is the *lag* or *reconstruction delay*, and m is the *embedding dimension*. From above PCA and correlation dimension estimate, we set m=10, which accords to Taken's theorem. We set d with experiential value 1. Thus, $\boldsymbol{Y}$ is a $M \times m$ matrix, and the constants m, M, d, and N are related as

$$M = N - (m - 1)d\text{=}89998\text{-}(10\text{-}1)\times 1\text{=}89989 \tag{9}$$

The second step is to divide the 89989 space vectors into 90 blocks, that is, every block is composed of 1000 space vectors, which is suitable for small data sets method[9]. Then we obtain the 90 blocks

$$\boldsymbol{Y}_j^k \quad (k\text{=}1, 2, \ldots, 90, \quad i\text{=}(k\text{-}1)\times 1000+j) \tag{10}$$

The third step is to calculate the 90 $\boldsymbol{Y}_j^k$ largest Lyapunov exponents. Fig. 5 shows the 90 results of largest Lyapunov Exponent.

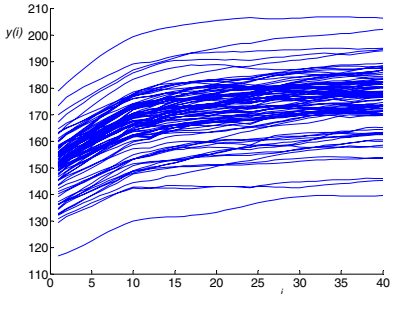


Fig. 5. The 90 largest Lyapunov exponent, the former line trend represents the largest Lyapunov exponent value

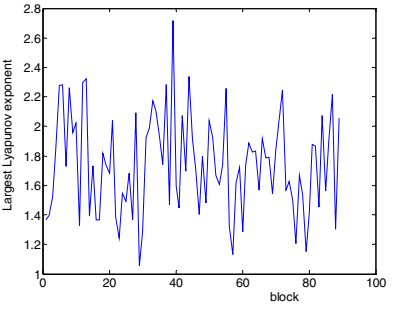


Fig. 6. The largest Lyapunov spectrum

From Fig.5, we can discover that the largest Lyapunov exponent of every block is above zero and equal with each other. Obviously the dynamic behavior of the MPEG-4 trace is complex.

The last step is calculate the 90 values using the least square approximation. We name the result as the largest Lyapunov spectrum. Fig. 6 shows the largest Lyapunov exponent spectrum against the block. The average value of the 90 largest Lyapunov exponents is 1.7416.

3 Packet Loss Ratio Analysis

Packet loss ratio can influence the video quality, and the Lyapunov exponents may provide a more useful characterization of chaotic systems. Here we mainly study the relation between the largest Lyapunov exponent and the packet loss ratio. Since the Lyapunov exponent gives the rate of exponential divergence from perturbed initial conditions, the intuition tells us that there should have certain relation between the largest Lyapunov exponent and the the packet loss ratio. We can also infer that if the largest Lyapunov exponent is larger, the packet loss rate is larger when the router buffer is finite.

To obtain the packet loss ratio of every block trace when transporting the Star War IV video trace in IP network, we use ns2 to simulate the transport process. The scenario which is made up 4 nodes-one video source, two taildropped routers and one video receiver, and three links is shown in Fig.7. The rate of link L1 is set to 10M bps, link L2 0.5M bps and link L3 10M bps. We first convert the Star War IV video trace into certain format that ns simulator can read. Then we obtain the packet number are forwarded and dropped by router node 1. So we can obtain the packet drop rate.

To better compare the largest Lyapunov exponent and the packet loss rate, we normalize the value by getting rid of the mean value and then multiplying a suitable proportion. The part result of the trace showed in Fig. 8. Fig.8(a) shows the fore 40 blocks and Fig.8(b) shows the result from the 45th block to 80th block. From Fig.8, we can see that the trend of the largest Lyapunov exponent and the loss rate is approximately concurrent in most cases, which means when the largest Lyapunov exponent grows, the packet loss ratio grows, while when the largest Lyapunov exponent decreases, the packet loss ratio decreases. The experiment results verify our surmise. The Lyapunov exponent gives the rate of exponential divergence, which can result in burst traffic. Also the degree of burst can lead to different occupancy rate of the router buffer which can help to packet loss ratio. The conclusion that the chaotic characteristic and the packet loss ratio is correlative can be made. As far as the Star War IV is concerned, the correlation coefficient of the largest Lyapunov exponent spectrum and packet loss ratio is 0.9713. This conclusion can be significative when designing QoS constrained video transport.

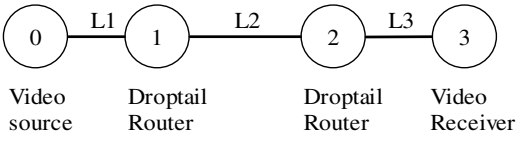


Fig. 7. The simulation scenario composed with four nodes in ns simulator environment

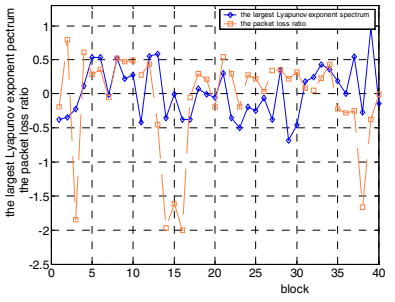


Fig. 8(a). The former 40 values of the largest Lyapunov spectrum and the packet loss ratio

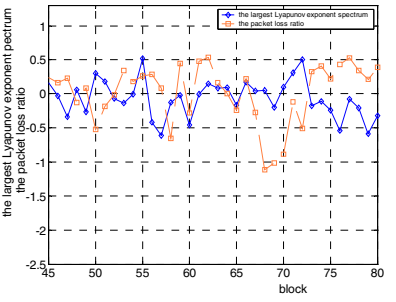


Fig. 8(b). The 35 values of the largest Lyapunov spectrum and the packet loss ratio between 45th and 80th block

4 Conclusion

There exist abundant dynamic behavior in MPEG-4 video. Its power spectral density has the $1/f^{\beta}$ characteristic, also has the solely periodic feature. Principal components analysis and the correlation dimension estimate show that although the MPEG-4 video has complex dynamic, the main factors only contain about 4 factors. This paper presents and calculates the video trace largest Llyapunov exponent spectrum. Through the ns simulate experiment, this paper reveal some inherent relation between the largest Lyapunov exponent spectrum and one QoS parameter-packet loss ratio of MPEG-4 video.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant Number 60132030.

References

1. Garrett Mark W and Willinger Walter. Analysis, modeling and generation of self-similar VBR video traffic. Proceedings of the conference on Communications architectures, protocols and applications, 1994, pp.269-280.
2. Alkhatib, Ahmad, Krunz and Marwan. Application of chaos theory to the modeling of compressed video. IEEE International Conference on Communications, v 2, 2000, pp. 836-840.
3. Fitzek and. Reisslein M, MPEG-4 and H.263 video traces for network performance evaluation (extended version). Technical Report: TKN-00-06, TU Berlin Department of Electrical Engineering, Telecommunication Networks Group, October 2000.http://www-tkn.ee.tu-berlin.de/research/trace/trace.html.

4. Han Liangxiu and Cen Zhiwei. A new multi-fractal network traffic model. Chaos, Solitons and Fractals, 2002;5 :1507-1523.
5. Welch. The use of fast fourier transform for the estimation of power spectra:A method based on time averaging over short, modified periodograms. IEEE Trans.Audio Electroacoustics, Vol. AU-15 (June 1967), pp. 70-73.
6. Kugiumtzis D and Boudourides MA. Chaotic analysis of internet ping data: just a random number generator? Contributed paper on the SOEIS meeting at Bielefeld,1998,March, 27-28.
7. Jolli.e IT. Principal component analysis. New York: Springer; 1986.
8. Grassberger P and Procaccia I. Characterization of strange attractors. Phys. Rev. Lett. 50 (1983) pp.346.
9. Michael T. Rosenstein, James J. Collins and Carlo J. De Luca. A practical method for calculating largest Lyapunov exponents from small data sets. Physica D 65(1993): 117-134.
10. Hao Bai-lin. Starting with parabolas-an introduction to chaotic dynamics. Shanghai Scientific and Technological Education Publishing House,1993.
11. http://ita.ee.lbl.gov/html/contrib/LBL-PKT.html

A Simple Streaming Media Transport Protocols Based on IPv6 QoS Mechanism

Yan Wei[1], Cheng Yuan[2], and Ren Maosheng[1]

[1] Network Laboratory Department of Computer Science Peking University, Beijing 100871, China
[2] IBM China Software Development Lab, China
{yanwei, chengyuan, rms}@net.pku.edu.cn

Abstract. The rapid growth of internet promotes evolution of network applications. As different network application has different set of requirements for network technology, many protocols have been developed. Most of these protocols overlap and are redundant in function, which result in lower performance. This paper presents an IPv6 QoS-based simple streaming transport protocol (SSTP), which aims at next generation networks and takes into account of the feature of the multimedia stream transmission. The protocol has been implemented and deployed in a multimedia stream playing system. This paper not only introduces the background and design methodology of SSTP, but also describes SSTP in detail. At last, this article compares the performance of SSTP and RTP/RTCP, and gives the consideration about improvement of SSTP.

1 Introduction

Unlike today's internet, the Next Generation Internet (NGI) not only offers more bandwidth, but also provides Quality of Service(Qos) to some degree. It is of practical importance to research novel streaming media transport protocols for NGI based on IP QoS. In this paper, a Simple Streaming media Transport Protocol (SSTP) is designed and implemented for NGI and its functionality and capability are demonstrated by a media streaming demo system in IPV6 test-bed.

The rest of this paper is organized as follows: The characteristics of streaming media and previous streaming media transport protocols are introduced in Section 2. We discuss the design principle of SSTP, describe its functionality and offer the essential technologies to implement it in Section 3. In Section 4, according to analytical performance and actual demo results, we propose the method to improve SSTP. The development of future streaming media transport technologies are discussed in Section 5.

2 The Characteristics of Streaming Media and Streaming Media Transport Protocols

There are two main patterns of transmitting audio and video media flow on the Internet: downloading file to local user and playing file directly on the Internet. Due

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 951 – 960, 2005.

to the limitation of bandwidth and congestion in network, it usually requires a long time to download a media file entirely. While streaming transmission, a means to deliver audio and video data continuously in real-time to user, only requires several seconds to startup for viewing. While the media file is being played on local computer, the rest of the file is being downloaded simultaneously from a remote media server. The obvious advantage of streaming transmission is the great reduction of startup delay. More importantly, it doesn't require large cache capacity.

2.1 Characteristics of Streaming Media

The streaming transmission based media is called streaming media, which plays a media file while downloading it. In contrast to the dominating protocol in the current internet –TCP, streaming media has following characteristics:

- The consumed bandwidth is relatively constant. Both CBR (Constant Bit Rate) audio streaming and VBR (Variable Bit Rate) video streaming consume the relative bandwidth in the long term of time.
- Sensitive to time. The streaming media, which plays media file while downloading it, requires that the data must be delivered to user in a given time slice, the delayed data is useless to local media player.
- The codec of streaming media has some tolerant capability for packet loss, but it requires the distribution of packet loss to be uniform.

2.2 Streaming Media Transport Protocol

Media files are stored on severs and users request them for viewing. This process is called a session, which is from startup request to end of playing media. A session control protocol — RTSP (Real-Time Streaming Protocol) is applied widely now, which is used for setting up, transmitting and controlling one or more media flows in a streaming media session.

Streaming media transport protocol is designed to offer functionality of transmitting media flow on network, alleviating or eliminating the limitation of internet itself. The goal of this protocol design is to provide a real-time, sequential, low-delay, low-loss and low-jitter channel for streaming media. RTP, as a media transport protocol, is widely used now.

RTP is proposed by IETF working group. RTP is originally designed to meet the needs of multi-participant multimedia conferences. RTP is defined to offer time information and data flow synchronization under the communication pattern of one-to-one or one-to-many.

RTP actually includes two sub-protocols, one is RTP used to deliver data and the other is RTCP used for QoS statistical information and control feedback information. So two pairs of port numbers are used in a RTP session, one is used by RTP to transmit data and the other is used by RTCP to deliver control information.

Other than the RTP protocol, Shanwei Cen[3]proposed the SCP protocol considering the unsuitability of TCP in streaming media transmission. Christopher K. Hess and Roy H. Campbell proposed the MSP protocol; Biswaroop.Mukherjee proposed the TLTCP protocol in his thesis. But none of them is perfect, some is not TCP-friendly, some wastes too much bandwidth and some doesn't support IP multicast.

3 Simple Streaming Media Transport Protocol

Present internet is a best-effort switch network, hardly providing any QoS function. So it can't guarantee the bandwidth of a flow and an end-to-end delay. The prevalent RTP/RTCP protocol also has shortcoming. RTP is originally designed to meet the needs of multi-participant multimedia conferences which is different from the model of client/server. Therefore, when using RTP to transmitting media data, each packet header contains some redundant information, thus wasting bandwidth unnecessarily. What is more, because IP layer will provide end-to-end QoS guarantee in next generation internet, RTCP is useless in NGI.

We design and implement a novel Simple Streaming media Transport Protocol (SSTP) according to the characteristic of NGI. To improve the transmission performance, we reduce much control information which is used presently but is unnecessary for the future.

3.1 Design Principle

According to the characteristic of NGI, we affirm that the transmission pattern of streaming media will change entirely: it is not required for transport protocol to be adaptive to the fluctuation of flow in network. After the request to network layer for the reservation of end-to-end bandwidth is satisfied, transport protocol can use its own channel freely. In this paper, we design and implement the Simple Streaming media Transport Protocol (SSTP), a novel media transport protocol specially for next generation internet.

SSTP is designed based on the assumption that IP layer support Qos. That is to say, for admitted flow, and given bandwidth, packet loss ratio and packet delay are guaranteed in network layer. We describe the SSTP design principle as follows:

- Make full use of the Qos function in network layer.
- In order to alleviate the load of network and improve the performance of end systems, the format of a packet and the states of protocol should be simplified.
- Provide the function of sorting, checking and synchronizing for streaming media.

3.2 SSTP Protocol

Based on above principles, SSTP protocol is directly designed on IPv6, and it takes advantage of the flowlabel field of IPv6 header to identify different media streams. It has two advantages compared to the RTP based on UDP: first, the decrease of protocol layers can improve the processing speed of the end systems (including both server and client); second, this decrease cuts down the length of the data packet header, it can improve the utility of the bandwidth, which is very important for media stream transport protocol for its high demand for bandwidth.

Based on IPv6. The header of the SSTP protocol abuts on the header of IP protocol. In order to avoid conflicting with other protocols at the time, the value of *Next Header* field in IPv6 header is assigned 200 for the SSTP protocol.

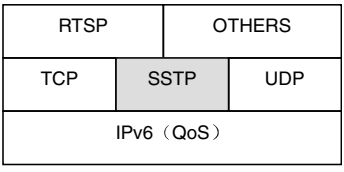


Fig. 1. Position of SSTP in protocol layers

It is prescribed in IPv6 standard that the MTU of all links in Internet is at least 1280 bytes. In order to avoid fragmenting in IP layer, it is suggested in the SSTP protocol that the length of every data packet is not bigger than 1280 -40 = 1240 bytes (40 bytes for the length of IPv6 header). Considering the transport efficiency, the length of SSTP data packet should be near the maximum value as soon as possible (the maximum value is MTU of link - 40).

The flowlabel of IPv6 is used in SSTP protocol. The sender set different flowlabels for different audio streams and/or video streams of the same session; the receiver distinguishes different media streams by the flowlabel of the data packet received.

There is no checksum in IPv6 header. The upper layer protocol SSTP uses pseudo headers to calculate checksum in terms of prescript, the value of Next Header field is 200.

SSTP Message. Format SSTP message is divided into two classes according to its function: data message and control message. Data message is used to transport media stream; control message is used to transport controlling information.

The header of SSTP message is 8 or 12 bytes in length. They are separately used for byte-stream mode and frame mode. There are some fields in the header, including Mode, Extend, Control, Event, Timestamp, Checksum, Sequence Number and Payload. The Event and Timestamp fields are used to control frame boundary and synchronization at frame mode; the calculation of checksum include pseudo header, SSTP header and SSTP payload.

When transporting media stream, control message is used only if a sender (Server) definitely require a receiver (Client) to send it. The control message provide the feedback path to sender for receiver, the feedback information usually includes loss rate and bandwidth estimation at present.

Function Description of SSTP. Two transport modes are applied in SSTP: byte-stream mode and frame mode. In byte-stream mode, the server is required to possess the media stream file to be transported. All audio streams and video streams required by the client are included in the file. As a supplement to the byte-stream mode, the frame mode is mainly used in live TV broadcast. We carry out the byte-stream mode at present. In byte-stream mode, the entire file is treated as a byte stream, regardless of the format and number of media streams in the file. The server identifies the media stream file by the client's requirement and gets the header info of the media stream (such as the natural play time and the size of media data etc.). Based on this information, the server first calculates the bandwidth for transporting the file and identifies the flowlabel, then applies resource reservation to network layer QoS manager. Second, the server calculates the size of available SSTP data packet payload,

based on calculation results (bandwidth, the size of SSTP data packet available payload), it can confirm the interval between two continuous data packets. Finally, the server packs the data to be sent into packets at fixed size, fills flowlabel, checksum and sequence number in the header and sends at the special interval. The client gets the data packets from the server by the same flowlabel and protocol identification. The data packets are discarded if their checksums are wrong. Then the rest packets are reordered by their sequence numbers. A new byte stream is formed and delivered to upper layer, the codec in upper layer decodes the byte stream and plays it.

The byte-stream mode has many advantages over current RTP. First, the server doesn't need to read and identify every frame, the efficiency at server is improved remarkably, and a server can serve for more clients. Second, the transport rate of data packets is absolutely constant, it is convenient for processing QoS (i.e. to alleviate the work of shaping at ER), meanwhile, it is easy to reserve resources Third, filling much data into a packet as soon as possible decreases the header payload, which improves the utility of the bandwidth Finally, every media stream file has only one flow in the network, not several audio/video streams. Therefore, the payload of the QOS router for resource reservation and the payload of end system are decreased.

3.3 Play System

The above SSTP is only a protocol at the transport layer. Its main function is transporting media stream. In practical media stream transport system, it must cooperate with RTSP. The client finds the URI of the requested media stream file, then uses RTSP protocol to communicate with media server. During this process, the server requests QOS manager in the network layer to apply to resource reservation. After the RTSP session is established successfully, the client sends "PLAY", the server begins to transport media stream through SSTP (Figure 2 show this process).The Control message in SSTP protocol is not used at the present, because, theoretically, QOS at the network layer sustains resource reservation from end to end. It is not necessary to get the receiver's feedback except to get some statistic information.

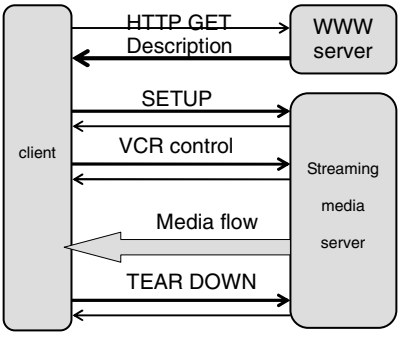


Fig. 2. Communication between media client and server

Media stream play system adopts C/S model. It sustains two application modes: VOD mode and Broadcast Mode, corresponding to bytestream mode and frame mode of SSTP protocol respectively. We designed and implemented a media stream transport system *SimpleStreamer* based on an open source code library called *Live*, the protocol of transporting media stream in this system is SSTP. The system is shown in figure 3.

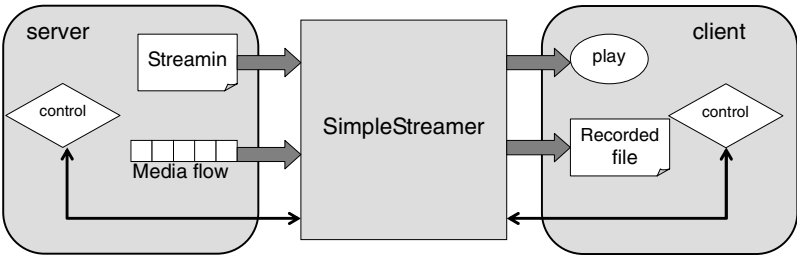


Fig. 3. Media stream play system SimpleStreamer based on SSTP

SimpleStreamer integrated the interface of IP QOS. It is simple, scalable, unicast compatible, and easy to manipulate. The client of the system can demand quality of service based on his/her own requirement.

4 Performance Evaluation and Comparison

We implement SSTP and SimpleStreamer player using commodity Linux environment. In this section, performance evaluation parameters and theoretic analysis results are presented, followed by the experimental performance analysis of SSTP and Simple Streamer based on the practical measurements.

4.1 Evaluation Parameter Analysis

Following parameters are usually concerned for analysis of streaming transport protocol

- Extra bandwidth overhead for protocol operations i.e. bandwidth occupied by appended packet head and control signaling when the same amount of *net* streaming media is delivered.
- CPU usage of end systems. The complexity of the protocol operation is measured especially at the streaming media server. The CPU cycles taken by the protocol is expected to be as few as possible.
- Delay and jitter of data packets.
- Loss rate of data packets. Packet loss to a centain extent is tolerable for the media stream, however, if large amount of packets are lost in a short period, the playback quality will degrade severely.

- Whether the protocol is TCP friendly. As TCP streams constitute a large part of the traffic over the Internet, the ability to be TCP friendly is also an important criterion for streaming transport protocols.

As the infrastructure of the Next Generation Internet has changed a lot, the standard for performance evaluation of streaming media transport protocols will also be different. For a streaming media transport protocol with underlying end-to-end QoS mechanism in the IP layer, the delay jitter, loss rate and TCP friendliness mentioned above has been guaranteed and need no further consideration theoretically, which means different protocols will perform consistently in these aspects. As a result, the performance evaluation of streaming protocols for NGI will concentrate on bandwidth and CPU usage.

4.2 Performance Comparison of SSTP and RTP

Comparison of Bandwidth Overhead and Net Bandwidth. The bandwidth overhead of a specific streaming transport protocol is determined by several factors including MTU, media frame rate, packet header length, packing method, protocol control messages and the length/frequency of the ACK messages. Taking minimum MTU=1280 bytes for IPv6 as an example, the bandwidth overhead of SSTP is 3.9% compared to 9.8% for TCP. When the frame length variation is considered, bandwidth overhead of RTP is 9.9%±2.5%. SSTP is obviously more efficient than RTP.

End System CPU Usage Comparison. The end system CPU usage is determined by the complexity of protocol operations and the number of RAM access for packets retrieval. The CPU usage is expected to be as low as possible, especially that of the streaming servers for which the CPU capacity is already a big issue.

In a VOD system, media files are processed and stored in the storage systems of media servers. The load per second of the media server running RTP is at least M×(Fread+Fparse+Fadd) heavier than those running SSTP. Fread is the computation amount in the process from distinguishing the starting tag of the frame to reading in the complete description information of the frame head. Fparse is the computation required for obtaining the frame length, frame playback time and other related information according to the description information conveyed in the frame head. Fadd is the computation complexity for calculating the appended frame head. M usually falls between 23 and 30. F* is closely related to the specific coding format of the media file [8].

4.3 Comparing SSTP's Performance to RTP's Performance

Theoretically, service requirements such as loss, delay jitter and TCP friendliness have already been fulfilled by the underlying networking infrastructure with end-to-end QoS mechanism. However, as NGI technology is still in its infancy, the practical performance needs further verifying. Thus we measured and compared the practical performance of SSTP and RTP on an IPv6 testbed. The measurement results are presented here in detail.

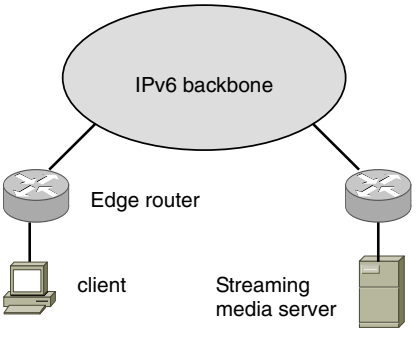


Fig. 4. SSTP and media play system

Experiment Environment and Methods. Fig.4 depicts the topology of our experimental network. The streaming server and client (each with a 2.6GHz Pentium4 CPU, 1GB DDR SDRAM, 120GB Hard disk) support both SSTP and RTP protocols.

In this experiment, jitter and loss rate of the two different transport protocols are measured under the situations of transmitting video/audio streams (6 times) and video streams (9 times) respectively.

Experimental Results Analysis. The loss rate and jitter of datagram transmitted by SSTP is more then that transmitted by RTP in networks in which there is little QoS mechanism. The loss rate and jitter of a datagram transmitted by SSTP is also more then that of one transmitted by RTP in networks which support the end-to-end QoS.

From performance graphics above we notice that the loss rate and jitter of datagram have been rapidly reduced when the IP QoS mechanism is deployed, while that of datagram transmitted by RTP have not been reduced much. It is that SSTP benefits from guaranteed bandwidth from IP QoS mechanism.

From the analysis above, we also conclude that the performance of SSTP is not better than that of RTP, because RTP has a partnership, RTCP, which controls traffics when there is congestion. SSTP is based on IP QoS, and therefore we can simplify the control mechanism of the protocol.

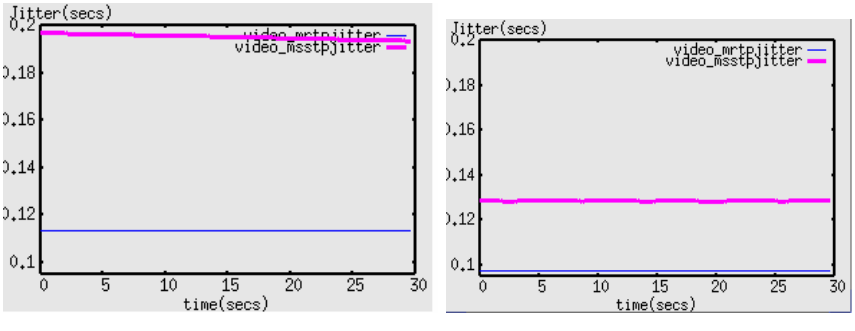


Fig. 5. (a) jitter without QoS (b) jitter with QoS

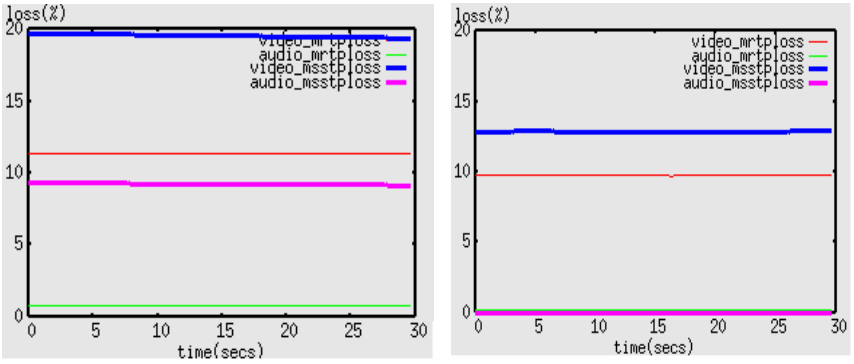


Fig. 6. (a) loss rate without QoS (b) loss rate with QoS

5 Future Work

The SSTP protocol based on IP QoS mechanism presented in this paper is the first step of design and implementation. Although the results of experiments do not live up to our expectations, but the actual performance is approximately in accordance with theoretic analysis, which demonstrates that the SSTP protocol with underlying QoS support succeeds in delivering media streams efficiently and promptly.

According to the above analysis results, we plan to improve SSTP from the following aspects. First, we will add at least control signaling to SSTP through successive experiments to improve the playback quality, save network bandwidth and decrease the CPU usage. Then, we will devise an approach to control the buffer size of the receivers, at the same time maintaining a certain level of playback quality for memory limited networking devices. Finally, we will integrate an end-to-end control signaling with the IP QoS mechanism in order to shorten the setup response time perceived by the clients through resource reservation and to grant clients the ability to adjust the bandwidth allocated to them through previous requests.

References

[1] S. Cen, C. Pu and J. Walpole, "*Flow and Congestion Control for Internet Media Streaming Applications*", in Proc. Multimedia Computing and Networking, January 1998.

[2] Shulzrinne, H., Casner, S., Frederick, R. and V. Jacobson, "RTP: A Transport Protocol for real-time applications", RFC 1889, January 1996

[3] Reynolds, J. and J. Postel, "Assigned Numbers, STD 2", RFC 1700, October 1994

[4] Live Networks Inc., "LIVE.COM Streaming Media", www.live.com

[5] Chengyuan, "SSTP——a streaming media protocol on basis of QoS", master thesis June, 2003

[6] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998

[7] M. Handley and V. Jacobson, "SDP: Session Description Protocol", RFC2327, April 1998
[8] Jian Lu, “Signal Processing for Internet Video Streaming: A Review”, Apple Computer Inc. streamingmedialand.com/sp4streaming2.pdf
[9] DivXNetworks, Inc. “DIVX Player”, www.divx.com
[10] Rob Koenen, “Overview of the MPEG-4 Standard”, ISO/IEC JTC1/SC29/WG11 N4668, March 2002
[11] MP4 Forum, “MPEG-4 – The Media Standard The landscape of advanced multimedia coding”, Nov 2002
[12] Microsoft Corporation, “Advanced Systems Format (ASF) Document Revision 01.20.00e”, Dec. 6, 2002
[13] R. Steinmetz and K. Nahrstedt, “Multimedia: Computing, Communications, and Applications”, Prentice Hall PTR, New Jersey, 1995.
[14] K. Almeroth, “The evolution of multicast: From the MBone to inter-domain multicast to Internet2 deployment”, IEEE Network, January/February 2000.
[15] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S. and S. Jamin , “Resource ReserVation Protocol -- Version 1 Functional Specification”, RFC 2205, September 1997.

An Aided Congestion Avoidance Mechanism for TCP Vegas*

Cheng-Yuan Ho[1], Chen-Hua Shih[1], Yaw-Chung Chen[1], and Yi-Cheng Chan[2]

[1] Department of Computer Science and Information Engineering,
National Chiao Tung University, No. 1001, Ta Hsueh Road, Hsinchu City, 300, Taiwan
{cyho, shihch, ycchen}@csie.nctu.edu.tw
[2] Department of Computer Science and Information Engineering,
National Changhua University of Education, No. 1, Jin-De Road, Changhua, Taiwan
ycchan@cc.ncue.edu.tw

Abstract. TCP Vegas detects network congestion in the early stage and successfully prevents periodic packet loss that usually occurs in TCP Reno. It has been demonstrated that TCP Vegas achieves much higher performance than TCP Reno in many aspects. However, TCP Vegas cannot prevent unnecessary throughput degradation when congestion occurs in the backward path, it passes through multiple congested links, or it reroutes through a path with longer round-trip time (RTT). In this paper, we propose an aided congestion avoidance mechanism for TCP Vegas, called Aid-Vegas, which uses the relative one-way delay of each packet along the forward path to distinguish whether congestion occurs or not. Through the results of simulation, we demonstrate that Aid-Vegas can solve the problems of rerouting and backward congestion, enhance the fairness among the competitive connections, and improve the throughput when multiple congested links are encountered.

1 Introduction

With the fast growth of Internet traffic, how to efficiently utilize network resources is essential to a successful congestion control. Transmission Control Protocol (TCP) is a widely used end-to-end transport protocol on the Internet; it has several implementation versions (i.e., Tahoe, Reno, Vegas...) which intend to improve network utilization. Among these TCP versions, Vegas can achieve much higher throughput than that of others [1].

TCP Vegas uses the difference between the expected and actual throughput to estimate the available bandwidth in the network, control the throughput, and avoid congestion. The idea is that when the network is not congested, the actual throughput will be close to the expected throughput. Otherwise, it will be much smaller than expected. TCP Vegas uses the congestion window size and measured RTT to estimate the amount of data in the network pipe, maintains

* This work was sponsored in part by National Science Council under grant no. NSC 93-2752-E-009-006-PAE.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 961–971, 2005.

extra data with amount between the lower threshold (α) and the upper threshold (β), gauges the congestion level in the network, and updates the congestion window size accordingly. As a result, Vegas is able to detect network congestion in the early stage and successfully prevents periodic packet loss that usually occurs in Reno. Furthermore, Vegas keeps an appropriate amount of extra data in the network to avoid congestion as well as to maintain high throughput. Many studies have demonstrated that Vegas outperforms Reno in the aspects of overall network utilization, stability, fairness, and throughput [1], [2], [3], [4]. However, it suffers some problems that inhere in its congestion avoidance scheme, including those issues of rerouting [3], [5], fairness [4], network asymmetry [5], [6], [7], [8], [9], and path with multiple congested links [10]. All these problems may be obstacles for Vegas to achieve a success.

In this work, we propose an aided congestion avoidance mechanism for TCP Vegas (abbreviated as Aid-Vegas hereafter). By using the relative one-way delay of each packet along the forward path to distinguish whether congestion occurs or not, Aid-Vegas may solve the problems of rerouting and backward congestion, enhance the fairness among the competitive connections, and improve the throughput when passing through multiple congested links. We demonstrate the effectiveness of Aid-Vegas based on the results of simulation.

The rest of this paper is organized as follows. Section 2 describes Vegas and its problems. Section 3 outlines prior related work. Section 4 discusses the Aid-Vegas. The simulation results are presented in Section 5. Finally, Section 6 makes some concluding remarks.

2 TCP Vegas and Its Problems

In this section, we review the congestion avoidance mechanism of TCP Vegas and describe its problems in detail. The detailed description of Vegas can be found in [1].

2.1 Congestion Avoidance Mechanism of TCP Vegas

Different from Tahoe and Reno, which detect network congestion based on packet losses, Vegas estimates *a proper amount of extra data*, called Δ for short, to be kept in the network pipe and controls the congestion window size accordingly during the congestion avoidance phase. It records the RTT and sets BaseRTT to the minimum of ever measured RTTs. The Δ is between two thresholds α and β, as shown in the following:

$$\alpha \leq (Expected - Actual) \times BaseRTT \leq \beta, \tag{1}$$

where *Expected* throughput is the current congestion window size divided by BaseRTT, and *Actual* throughput is the current congestion window size divided by the newly measured RTT. Both throughput and congestion window size are kept constant when Δ is between α and β. If Δ is greater than β, it is taken as

a sign for incipient congestion, thus the congestion window size will be reduced. On the other hand, if Δ is smaller than α, the connection may be underutilizing the available bandwidth. Hence, the congestion window size will be increased.

2.2 Problems in Vegas

There are several problems in Vegas that may have a serious impact on the performance during the congestion avoidance phase. We summarize these problems as follows.

Network Asymmetry: By adjusting source congestion window size, Vegas keeps an estimated extra data on the bottleneck to avoid congestion as well as to maintain high throughput. However, a roughly measured RTT may lead to an improper adjustment of congestion window size. If the network congestion occurs in the direction of ACK packets (backward path), it may underestimate the actual rate and cause an unnecessary decreasing of the congestion window size. Ideally, congestion in the backward path should not affect the network throughput in the forward path, which is the data transfer direction. Obviously, the control mechanism must be able to distinguish the direction of congestion and adjust the congestion window size only if necessary.

Rerouting: Vegas estimates the BaseRTT and RTT to compute the expected and actual throughput respectively and then adjusts its window size accordingly. This idea works well in usual situations. However, rerouting may cause a change of the fixed delay, which is the sum of propagation delay and packet processing time along the round-trip path, and result in substantial throughput degradation. When the route of a connection is changed, if the new route features shorter fixed delay, it will not cause any serious problem for Vegas because most likely some packets will experience shorter RTT, and BaseRTT will be updated eventually. On the other hand, if the new route for the connection has a longer fixed delay, it would be unable to tell whether the increased RTT is due to network congestion or route change. The source host may misinterpret the increased RTT as a signal of congestion in the network and decrease its window size. This is just the opposite of what the source should do.

Unfairness: Different from TCP Reno, Vegas is not biased against the connections with longer RTT [3], [4]. However, there is still unfairness which comes with the nature of Vegas. Since Vegas attempts to maintain Δ between two thresholds α and β by varying its congestion window size, but the range between α and β features uncertainty that affects the achievable throughput of connections. Furthermore, Vegas may keep different extra data in the bottleneck for connections with the same round-trip path. As a result, it prohibits better fairness among the competing connections.

Multiple Congested Links: In the paper about Vegas [1], it is assumed that only one bottleneck in the connection so the extra data is just queued in one router. However, when Vegas passes through multiple congested links, it could not tell whether the packets are queued in a single or multiple routers because

Vegas uses the RTT to estimate the backlog in the path. As a result, it tends to decrease congestion window size and hence degrade the throughput of multi-hop connections due to reducing the sending rate unnecessarily [10].

3 Related Works

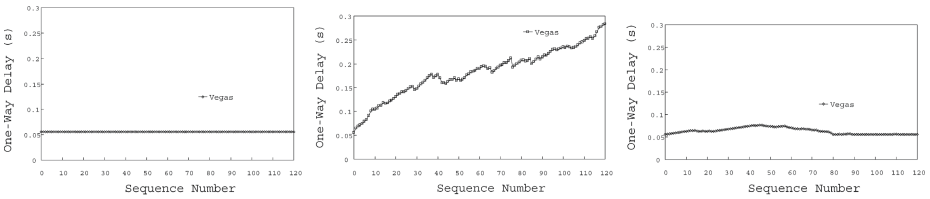
Congestion control for TCP is a very active research area; solutions to TCP congestion control address the problem either at the intermediate routers in the network [5], [11] or at the endpoints of the connection [6], [7], [8], [9], [12].

Router-based support for Vegas congestion avoidance mechanism can be provided as RoVegas [5], a solution which uses a normal TCP packet (data or ACK) with AQT (accumulate queuing time) option in its IP header as a probing packet to collect the queuing time along the path. As an alternative to packet dropping, an ECN (Explicit Congestion Notification) [11] bit can be set in the packet header for prompting the source to slow down. However, current TCP and router implementations do not support these two methods.

Several end-to-end congestion control approaches have been proposed to improve TCP performance for asymmetric networks. These approaches obtain either the forward trip time [7] or the actual flow rate on the forward path [8] depending on TCP timestamps option. Although solutions such as ACC (ack congestion control), AF (ack filtering), SA (sender adaptation), and AR (ack reconstruction) have improved the Reno's performance under asymmetric networks [12], these are not effective for handling asymmetry problems of Vegas [8]. By using the relative delay estimation along the forward path, TCP Santa Cruz [9] is able to identify the direction of congestion. However, it is not for rate-based Vegas. Enhanced Vegas [6] is a mechanism that works under asymmetric networks and uses TCP timestamps option to estimate queuing delay on the forward and backward path separately without clock synchronization. Nevertheless, clock skew issue such as the convergence speed of the clock ratio is still a problem of Enhanced Vegas.

4 Aid-Vegas

Vegas estimates a proper amount of extra data to be kept in the network pipe and controls the congestion window size accordingly. It works well during the congestion avoidance phase when no other competing sources exist. However, the aforementioned problems such as rerouting, unfairness, and multiple congested links may be encountered. Also it leads to unnecessary throughout degradation when the congestion occurs on the backward path. In other words, Vegas does a good job on its increasing part, but it may mistakenly decrease the congestion window easily when there are some variations in the network. In this work, we propose Aid-Vegas, which preserves the advantages of TCP Vegas, to deal with these problems. The detail of the proposed mechanism is explained as follows.



(a) Steady one-way delays (b) Increase trend (c) Short-term intervals

Fig. 1. The relative one-way delay of packets for Vegas

4.1 Mechanism Description

Suppose that Src (source) transmits packets to Dest (destination), Src timestamps each packet i with a timestamp t_i prior to its transmission, and a_i is the arrival time of the i^{th} packet at Dest. Dest computes the *relative one-way delay* of each packet as $D_i = a_i - t_i$. As will become clear later, the measurement methodology does not need synchronized clocks, because we are only interested in the relative magnitude of *one-way delays*. If a Vegas source keeps a constant throughput and a stable Δ between α and β during the congestion avoidance phase, so the relative one-way delay of each packet is steady as shown in Fig. 1 (a). We could see two situations when other sources want to compete the resource with it [13], [14]. One is that while the total flow rates are larger than the link capacity consistently, the relative one-way delays of Vegas would experience an increasing trend as shown in Fig. 1 (b). The other is the short-term relative one-way delays are increasing, as shown in Fig. 1 (c) when transient traffic of other sources passes by. As a result, Aid-Vegas utilizes one-way delays relationship to decide whether congestion happens in the forward path. The detailed mechanism is described as follows.

In the destination side, after computing D_i for each packet, Dest tells the result of D_i comparing with D_{i-1} to Src by using two reserved bits in TCP header, called ROD bits. In case $|D_i - D_{i-1}| \leq \tau$, where τ is a predefined time interval, or it is the first packet of a connection (i.e., $i = 1$), the ROD value is 00_{binary}. If $(D_i - D_{i-1}) > \tau$, the ROD value is 10_{binary}. On the other hand, the ROD value is 01_{binary} when $(D_{i-1} - D_i) > \tau$. In the source host, we set the trend D_t, whose default value is 0 in every RTT, to represent a congestion in the forward path. Whenever Src receives an ACK, it sums ROD value to D_t, and the values for 00_{binary}, 01_{binary} and 10_{binary} of ROD are set to 0, 1, and -1 respectively. The detailed description of Src and Dest's behaviors with variations of one-way delays is shown in Table 1.

Table 1. Src and Dest's behaviors with variations of one-way delays

variation	Dest	Src
$\|D_i - D_{i-1}\| \leq \tau$ or $i = 1$	ROD $= 00_{binary}$	$D_t = D_t + 0$
$(D_{i-1} - D_i) > \tau$	ROD $= 01_{binary}$	$D_t = D_t + 1$
$(D_i - D_{i-1}) > \tau$	ROD $= 10_{binary}$	$D_t = D_t - 1$

Table 2. 9 situations and congestion window size adjustment

	9 situations								
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
Src	+	+	+	◇	◇	◇	−	−	−
Dest	+	◇	−	+	◇	−	+	◇	−
	Adjustment								
CWD	+1	+1	+1	+1	±0	±0	+1	±0	−1

When this calculation is performed once per RTT, Src adjusts the congestion window size according to both values of Δ and D_t. If D_t is positive, it indicates that there is no congestion in the network, so Src increases the congestion window. Congestion may occur in the forward path and the throughput may be reduced when D_t is negative. The flow rates in the network are balanced while D_t equals 0, so Src doesn't adjust anything. Accordingly, the combinations of Src and Dest's notifications for adjusting the congestion window size results in 9 situations. Table 2 shows these 9 situations and how Src adjusts the congestion window size, where +, ◇, and - represent $\Delta < \alpha$, $\alpha \leq \Delta \leq \beta$, and $\beta < \Delta$ in Src column, and $D_t > 0$, $D_t = 0$, and $D_t < 0$ in Dest column respectively, and the signification of CWD is the congestion window size. We reserve the increasing part (i.e., (1)∼(3) notifications in Table 2) of Vegas and modifies both the steady and decreasing parts (i.e., the others in Table 2).

Since the idea is simple, we omit the pseudo codes in this section. The proposed scheme can improve the performance of TCP Vegas according to the simulation results in the following section.

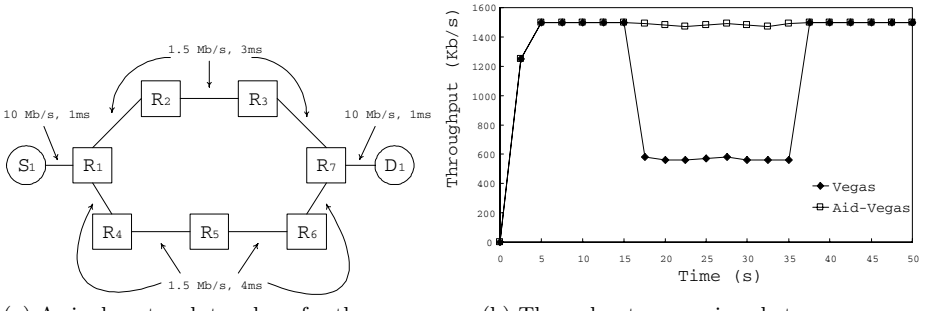
5 Simulation Results

In this section, we compare the performance of Aid-Vegas with Vegas by using the network simulator *ns2* [16], version 2.26. We show the simulation results for backward congestion, rerouting, fairness among the competing connections, and multiple congested links. The FIFO service discipline is assumed. Several VBR sources construct both forward and backward traffic, these are distributed ON-OFF sources with the Pareto model. During ON periods, some VBR sources generate backward traffic and send data at 3.0 Mb/s, while the others send data at 0.8 Mb/s or 1.0 Mb/s. Unless otherwise stated, the size of each FIFO queue used in routers is 50 packets, the size of data packet is 1 Kbytes, and the size of ACK is 40 bytes for both Vegas and Aid-Vegas. To ease the comparison, we assume that the sources are back logged.

5.1 Rerouting

From Section 2, we could see that there is no serious problem for Vegas when it reroutes with a shorter RTT, so does Aid-Vegas because it is based on Vegas.

However, if packets are rerouted with a longer RTT, Vegas may suffer throughput degradation, but not for Aid-Vegas. The reason is that Vegas could not differentiate whether the increased RTT is due to route change or network congestion. On the other hand, Aid-Vegas uses the relative one-way delay to distinguish the congestion, so rerouting to a longer RTT doesn't affect the throughput. In the following, we show the simulation result of both Vegas and Aid-Vegas when packets are rerouted to a longer path with larger RTT.



(a) A single network topology for the simulation of Vegas and Aid-Vegas.

(b) Throughput comparison between Vegas and Aid-Vegas with rerouting.

Fig. 2. The simulation topology and results of rerouting

Figure 2 (a) shows the first network topology. A source S_1 of either Vegas or Aid-Vegas sends data packet to its destination D_1. The bandwidth and propagation delay are 10 Mb/s and 1 ms for each full-duplex access link, 1.5 Mb/s and 3 ms for the full-duplex trunk link from R_1 to R_2, from R_2 to R_3 and from R_3 to R_7, and 1.5 Mb/s and 4 ms for the full-duplex trunk link from R_1 to R_4, from R_4 to R_5, from R_5 to R_6 and from R_6 to R_7. At the beginning, the packets are routed through S_1, R_1, R_2, R_3, R_7, and D_1 in order. At 15^{th} second, the connection link from R_2 to R_3 is broken and then recovered at 35^{th} second. Therefore, the packets pass through the other path from 15^{th} till 35^{th} second. As shown in Fig. 2 (b), when the packets are routed through the path with shorter RTT, Vegas achieves high throughput and stabilizes at 1.5 Mb/s. However, the performance of Vegas degrades dramatically when the packets are rerouted through the other path. On the other hand, Aid-Vegas always maintains a steady throughput regardless of the route change.

5.2 Network Asymmetry

In this subsection, we are interested in the throughput of different mechanisms when the congestion is caused by additional backward traffic. Since TCP RoVegas [5] and Enhanced Vegas [6] improve the performance of Vegas when the congestion occurs in the backward path, we also compare Aid-Vegas with them. Therefore, we will see the performance of four mechanisms in this part.

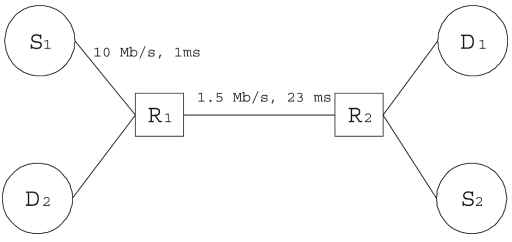
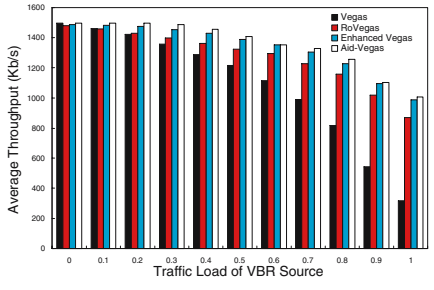
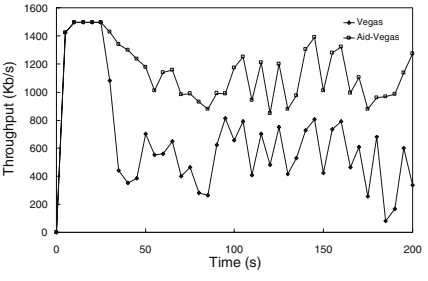


Fig. 3. A single bottleneck network topology for investigating throughput of different mechanisms when the congestion occurs in the backward path



(a) Average throughput versus different backward traffic loads for four schemes in the single bottleneck topology.



(b) Throughput comparison between Vegas and Aid-Vegas with the backward traffic load is 0.9.

Fig. 4. The simulation results of network asymmetry

The second network topology for the simulations is shown in Fig. 3. Sources, destinations, and routers are represented as S_i, D_i, and R_i respectively. A source and destination with the same suffix value represent a traffic pair. The bandwidth and propagation delay are 10 Mb/s and 1 ms for each full-duplex access link, 1.5 Mb/s and 23 ms for the full-duplex trunk link. A source of Vegas, RoVegas, Enhanced Vegas, or Aid-Vegas is attached to S_1 and a VBR source is attached to S_2. The S_1 starts sending data at 0 second, while S_2 starts at 25^{th} second.

From the results shown in Fig. 4 (a), we can observe that when different backward VBR traffic loads varies from 0 to 1, the average throughput of Aid-Vegas is higher than those of other mechanisms. For example, as the backward traffic load is 1, Aid-Vegas achieves a 3.2 times higher average throughput in comparison with that of Vegas. In addition, Fig. 4 (b) depicts the throughput comparison between Vegas and Aid-Vegas with a VBR source which has 1.35 Mb/s averaged sending rate. Obviously, we have demonstrated that Aid-Vegas significantly improves the connection throughput when the backward path is congested.

5.3 Fairness Improvement

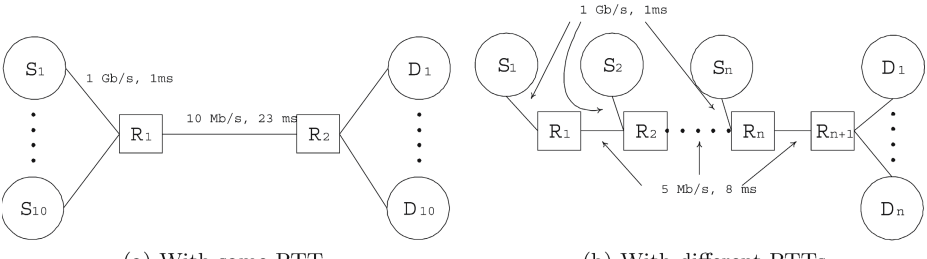
Vegas experiences unfairness because it attempts to maintain Δ between two thresholds α and β by adjusting its congestion window size, but the range bet-

ween α and β includes uncertainty to the achievable throughput of connections. Here, we show the comparison between Vegas and Aid-Vegas with multiple users competing network resources.

First, we are interested in those sources with same RTT. The network topology for simulations is shown in Fig. 5 (a), where the bandwidth and propagation delay are 1 Gb/s and 1 ms for full-duplex access link, and 10 Mb/s and 23 ms for full-duplex trunk link respectively. The sources are either Vegas or Aid-Vegas. We consider two cases here. One is that ten sources start at the same time, and the other is that ten connections from S_1 to S_{10} successively join the network one by one every 30 seconds. In addition, we don't change the default value of α and β. We use the fairness index [15] to represent the result of these two cases, as shown in Table 3, from which we observe that the fairness index of Aid-Vegas is higher than that of Vegas, especially when the sources start at the same time.

Table 3. The fairness index for Vegas and Aid-Vegas

	Vegas	Aid-Vegas
start at the same time	0.967	0.999
start at the different time	0.932	0.985



(a) With same RTT. (b) With different RTTs.

Fig. 5. The network topology for ten sources

Table 4. The fairness index for Vegas and Aid-Vegas

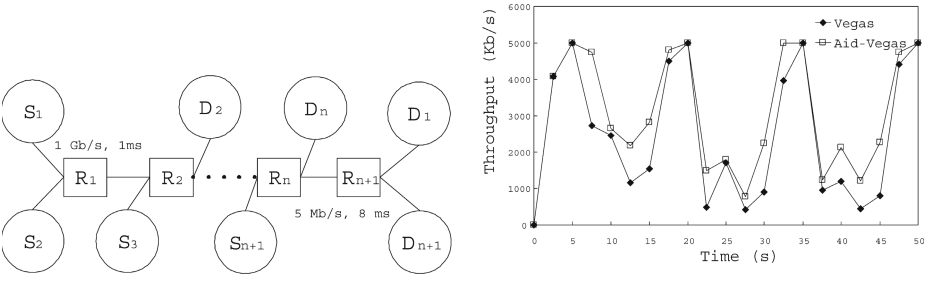
	Vegas	Aid-Vegas
$i = 2$	0.973	1.000
$i = 3$	0.943	0.998

Second, we simulate the source with different RTTs, and the network topology is shown in Fig. 5 (b). The bandwidth and propagation delay are 1 Gb/s and 1 ms for full-duplex access link, and 5 Mb/s and 8 ms for full-duplex trunk link respectively. The sources are either Vegas or Aid-Vegas. Table 4 depicts the fairness index for Vegas and Aid-Vegas with $i = 2$ and 3. We observe that the fairness index of Aid-Vegas is bigger than that of Vegas from Table 4.

From these simulation results, we can observe that the Aid-Vegas is more suitable than Vegas for multiple sources with same mechanism.

5.4 Multiple Congested Links

Vegas adjusts the congestion window size according to the comparison result of Δ with α and β. However, packet routing though multiple congested links causes the RTT increased. As a result, Vegas mistakenly judges that congestion occurs and decreases the congestion window size. On the other hand, Aid-Vegas uses the relative one-way delay of each packet to distinguish congestion. If the traffic of all flows is steady, the curve of relative one-way delays will be short-term intervals. Figure 6 (a) shows the general network topology, where the bandwidth and propagation delay are 1 Gb/s and 1 ms for the full-duplex access link, and 5 Mb/s and 8 ms for the full-duplex trunk link respectively . The S_1 is either Vegas or Aid-Vegas source, and the other sources are VBR sources with 0.8 Mb/s or 1.0 Mb/s during ON periods. Figure 6 (b) depicts the throughput of Vegas and Aid-Vegas when there are two VBR sources (i.e., n = 2) in the network. The throughput is 0.8 Mb/s and 1.0 Mb/s for S_2 and S_3 respectively. S_2 is ON from 7^{th} to 16^{th} second, S_3 is ON from 21^{th} to 30^{th} second, and both are ON from 37^{th} to 45^{th} second. From Fig. 6 (b), we can observe that the performance of Aid-Vegas is higher than Vegas.



(a) Network topology. (b) Throughputs.

Fig. 6. The simulation with multiple congested links

6 Conclusions

In this research, we propose Aid-Vegas for TCP Vegas. Comparing with other previous studies, Aid-Vegas provides a more effective way to solve the problems of rerouting and backward congestion, to enhance the fairness among the competing connections, and to improve the throughput when passing through multiple congested links. Through simulation, we demonstrate the effectiveness of Aid-Vegas. We will focus on its coexistence with other TCP implementations as well as its performance over wireless networks in our future work.

References

1. L. S. Brakmo and L. L. Peterson, 'TCP Vegas: End to End Congestion Avoidance on a Global Internet', *IEEE J. Select. Areas Commun., vol. 13, pp. 1465-1480, Oct. 1995.*
2. J. S. Ahn, P. B. Danzig, Z. Liu, and L. Yan, 'Evaluation of TCP Vegas: Emulation and Experiment', *ACM SIGCOMM'95, vol. 25, pp. 185-195, Aug. 1995.*
3. J. Mo, R. J. La, V. Anantharam, and J. Walrand, 'Analysis and Comparison of TCP Reno and Vegas', *IEEE INFORCOM'99, vol. 3, pp. 1556-1563, Mar. 1999.*
4. G. Hasegawa, M. Murata, and H. Miyahara, 'Fairness and Stability of Congestion Control Mechanism of TCP', *Telecommunication Systems Journal, pp. 167-184, Nov. 2000.*
5. Y. C. Chan, C. T. Chan, Y. C. Chen, and C. Y. Ho, 'Performance Improvement of Congestion Avoidance Mechanism for TCP Vegas', *IEEE ICPADS'2004, pp. 605-612, Jul. 2004.*
6. Y. C. Chan, C. T. Chan, and Y.C. Chen, 'An Enhanced Congestion Avoidance Mechanism for TCP Vegas', *IEEE Commun. Lett., vol. 7, issue 7, pp. 343-345, Jul. 2003.*
7. O. Elloumi, H. Afifi, and M. Hamdi, 'Improving Congestion Avoidance Algorithms for Asymmetric Networks', *in Conf. Rec. 1997 IEEE Int. Conf. Communications, pp. 1417-1421.*
8. C. P. Fu and S. C. Liew, 'A Remedy for Performance Degradation of TCP Vegas in Asymmetric Networks', *IEEE Commun. Lett., vol. 7, pp. 42-44, Jan. 2003.*
9. C. Parsa and J. J. Garcia-Luna-Aceves, 'Improving TCP Congestion Control over Internet with Heterogeneous Transmission Media', *in Conf. Rec. 1999 IEEE Int. Conf. Network Protocols, pp. 213-221.*
10. H. Cunqing and T.-S. P. Yum, 'The Fairness of TCP Vegas in Networks with Multiple Congested Gateways', *High Speed Networks and Multimedia Communications 5th IEEE International Conference on, pp. 115-121, July 2002.*
11. V. Jacobson and S. Floyd, 'TCP and Explicit Congestion Notification', *In Computer Communication Review, vol. 24, No. 5, pp. 8-23, Oct. 1994.*
12. H. Balakrishnan and V. N. Padmanabhan, 'How Network Asymmetry Affects TCP', *IEEE Commun. Mag., vol. 39, pp. 60-67, Apr. 2001.*
13. M. Jain and C. Dovrolis, 'End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput', *Networking, IEEE/ACM Transactions on, vol. 11, issue 4, pp. 537-549, Aug. 2003.*
14. M. Jain and C. Dovrolis, 'Pathload: A Measurement Tool for End-to-End Available Bandwidth', *In Passive and Active Measurements, Fort Collins, CO, Mar. 2002.*
15. R. Jain, D. Chiu, and W. Hawe, 'A Quantitative Measure Of Fairness and Discrimination for Resource Allocation in Shared Computer Systems', *DEC Research Report TR-301, Sep. 1984.*
16. http://www.isi.edu/nsnam/ns/

On the Design of Provably Secure Identity-Based Authentication and Key Exchange Protocol for Heterogeneous Wireless Access

Jun Jiang, Chen He, and Ling-ge Jiang

Institute of Modern Communications, Shanghai Jiaotong Univ.,
200030 Shanghai, China
jiangjunok@hotmail.com,
{chenhe, lgjiang}@sjtu.edu.cn

Abstract. Heavily based on the provable security model of Canetti and Krawczyk (CK-model), an identity-based authentication and key exchange (AKE) protocol which uses pairings is proposed for securing heterogeneous wireless access in this paper. By using the CK-model approach, an ideal and secure key exchange protocol was first proposed. Then a full-fledged authenticator is built to provide authentication of the ideal protocol. This completes a practical AKE protocol for heterogeneous environment while carrying the security proof. Analysis shows that our protocol is secure with partial forward secrecy, and efficient for considering the asymmetric wireless environment.

1 Introduction

The next generation wireless communication networks will be characterized by an integration of heterogeneous wireless access technologies and multi-providers. To achieve the user seamless roaming across the heterogeneous wireless networks, efficient AKE protocol that supports secure heterogeneous access is critical. The typical AKE protocol for heterogeneous wireless access involves three different entities to perform identity authentication and session key agreement.

In the past, the public key based AKE protocol was considered not suitable for the wireless communications because of its heavy computation overhead on the mobile terminal and limited radio communication bandwidth. However, it is worth noting that this constraint is likely to erode with the rapid progress of very large scale integration (VLSI) technology, which can be used to design more powerful mobile terminal. Currently, public key technology is paid much more attention to develop wireless AKE protocol because of its security, scalability, and flexibility.

Identity-based public key cryptosystem first presented by Shamir [1] in 1984, unlike the traditional certificate-based public key system, provides a train of thought to design secure and efficient AKE protocol. This certificateless based public key system does not only possess the secure merits of public key encryption, but also reduce the complexity of traditional certificate-based public key system.

In 2000, Joux [2] proposed a tripartite key exchange protocol based on pairings on an elliptic curve. Al-Riyami and Paterson [3] later improved this tripartite protocol to

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 972 – 981, 2005.

provide authentication by adopting certificates. Yet, both of these two protocols do not adopt the identity-based cryptosystem. Since the first feasible solutions by using identity-based encryption in combine with the pairing on elliptic curves were given by Boneh and Franklin in 2001 [4], some identity-based tripartite AKE protocols were presented in [5,6,7]. However, these tripartite AKE protocols are all role symmetric that both entities have the same computational complexity. Hence, they are not fit for the asymmetric wireless environment.

In this paper, we develop a novel identity-based AKE protocol appropriate for heterogeneous wireless access. Formally, our protocol is designed according to the CK-model, which was presented by Canetti & Krawczyk at Eurocrypto 2001 [8]. The model is achieved on the basis of Bellare-Rogaway model given at Crypto 1993 [9] and Bellare-Canetti-Krawczyk model presented in 1998 [10]. The main features of the CK-model are the *indistinguishability* approach [11] and the adversarial model of [10]. The CK-model relies on proving the security of protocols in an ideal world model, and then using a secure transformation mechanism to convert them into a real world model. Based on the CK-model, the designed AKE protocol possesses the feature of provable security. In addition, our protocol is efficient and appropriate for heterogeneous environment.

The rest of this paper is organized as follows. In section 2 we briefly describe the background of the CK-model and bilinear pairing techniques. In section 3, a novel secure key exchange protocol in the "ideal settings" is built. According to the modular method, Section 4 gives a mixed authenticator through integrating several existed secure MT-authenticators for our heterogeneous access scenario. In Section 5, a practical secure key exchange protocol with "explicit" mutual authentication is constructed by applying the new authenticator. Section 6 concludes the paper.

2 Preliminary

2.1 Overview of the CK-Model

This section briefly describes the CK-model approach and one can read [8] and [10] for further details. In CK-model, two adversarial models are defined to deal with how the adversary activates parties, namely the *authenticated-links adversarial model* (AM) and *unauthenticated-links adversarial model* (UM). In the AM, the adversary is able to invoke protocol runs, masquerade as protocol principals, and find used session keys. However, the adversary is unable to fabricate or replay messages which appear to come from uncorrupted parties. The UM corresponds to the "real world" where the adversary completely controls the network in use. The only difference between AM and UM is the amount of control adversary has over the communications lines between parties. The security definition of protocol in AM is described below based on *indistinguishability*.

Definition 1. *A KE protocol π is called session key (SK-) secure in the AM if the following properties are satisfied for any AM-adversary $\mathcal{A}$.*

1. *If two uncorrupted parties complete matching sessions then they both output the same key;*
2. *The probability that $\mathcal{A}$ guesses correctly the bit b (can distinguish the key from a random string) is no more than 1/2 plus a negligible fraction in the security parameter.*

The definition of SK-secure protocols in UM is done analogously. Canetti and Krawczyk [8] proved that a SK-secure protocol in AM is converted to an SK-secure protocol in UM if an *authenticator* is used. The systematically engineering method by using the CK-model to design a SK-secure UM protocol usually includes the following three steps [12]:

(1) Design or reuse a SK-secure protocol in the AM.
(2) Design or reuse a provably secure authenticator for transforming the AM protocol.
(3) Apply the authenticator to the AM protocol to produce an automatically secure UM protocol.

2.2 Bilinear Paring

In this section, we simply review the basic properties of bilinear pairing. For the further details one can refer to [1,4]. Let G_1 and G_2 denote two groups of the same prime order q, where G_1, denotes the group of points on an elliptic curve; and G_2, denotes a subgroup of the multiplicative group of a finite field. Let e denote a general bilinear map, i.e., e: $G_1 \times G_1 \rightarrow G_2$, which has the following three properties:

(1) Bilinear: If $P, P_1, P_2, Q, Q_1, Q_2 \in G_1$ and $a \in \mathbf{Z}^*_q$, q is a prime order, then,
$e(P_1+ P_2, Q)= e(P_1, Q)e(P_2, Q)$;
$e(P, Q_1+Q_2)= e(P, Q_1)\ e(P, Q_2)$;
$e(aP, bQ)= e(bP, aQ)= e(P, Q)^{ab}$.
(2) Non-degenerate: If P is a generator of G_1, then $e(P,P)$ is a generator of G_2, namely $e(P, P) \neq 1$.
(3) Computable: There is an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in G_1$ in polynomial time.

Bilinear Diffie-Hellman Problem(BDHP): Let G_1, G_2 be two groups of prime order q. Let $e : G_1 \times G_1 \rightarrow G_2$ be a bilinear map and let P be a generator of G_1. The BDHP in $(G_1,G_2,\ e)$ is as follows: For $a,\ b,\ c \in \mathrm{Z}^*_q$, given $(P,\ aP,\ bP,\ cP)$, compute $W=e(P,P)^{abc} \in G_2$.

BDH assumption: There exists no algorithm running in expected polynomial time, which can solve the BDH problem in $\{G_1, G_2, e\}$ with *non-negligible* probability**.**

3 Proposed Secure Protocol in the AM

3.1 Heterogeneous Wireless Access Model

The typical heterogeneous wireless environment involves different operators which may own different wireless access networks, and users with multi-mode mobile terminal subscribed to one network operator, namely the home network. Considering the

case that the user roams into a foreign network and wants to use the network to access Internet or other external data networks. Since the user and foreign network do not share a preexist secret, it is required to setup a dynamic secure association (SA) to secure the communication link. After building the dynamic SA, the user and foreign network server share a secret session key, therefore the fragile radio link could be protected.

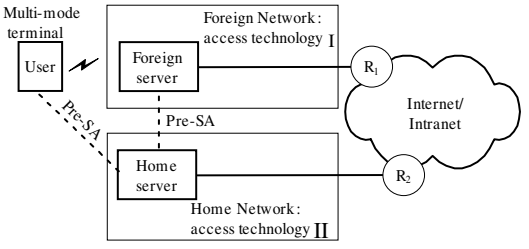


Fig. 1. Simplified heterogeneous wireless network model

Figure 1 shows the simplified heterogeneous wireless access model. It is assumed that the roaming agreement and SA is preexisted between the foreign network and home domain. Also assumed that the home network and its subscriber have long-term agreements used to authenticate each other. On the user side, a tamper-proof smart card can be employed to store the user's public key, identity information and public system parameters distributed from his home network. Foreign server acts as access server providing user local access, and home server maintains the user's basic information used for access authentication and services authorization. R_1 and R_2 in Figure 1 denote the router.

3.2 System Setup

In our protocol, let U denote the user, H be the home server and F represent the foreign server. Two trusty key generation center, say KGC_1 and KGC_2 for home network and foreign network respectively are deployed in our protocol. They each have a public/private key pair: $(P, s_1P \in G_1, s_1 \in \mathbf{Z}^*_q)$ and $(P, s_2P \in G_1, s_2 \in \mathbf{Z}^*_q)$, where P and G_1 are defined in previous section and globally agreed. When the user subscribes to his home domain, the KGC_1 needs to issue the private key $S_U=s_1Q_U$ for the user in a secure manner, where $Q_U=H_1(ID_U) \in G_1$, H_1: $\{0,1\}^n \rightarrow G_1$ is a one-way hash function mapping the identity onto his public key, and ID_U denotes the identity of U. Also, the KGC_1 issues the private key $S_H=s_1Q_H$ for H where $Q_H=H_1(ID_H)$, and the KGC_2 issues the private key $S_F=s_2Q_F$ for F where $Q_H=H_1(ID_H)$ before protocol's execution. Let k be the security parameter. Also let H_2 be hash function used for generating final session key between U and F in our proposal. All these hash functions are all modeled as random oracles. $x \| y$ denotes the concatenation of x and y. The session identifier *sid* is fundamentally critical to the security of our protocol. Its role aims at maintaining a particular session of the communication parties among all concurrent sessions such that the probability of the appearance of the same identifier twice is *negligible*.

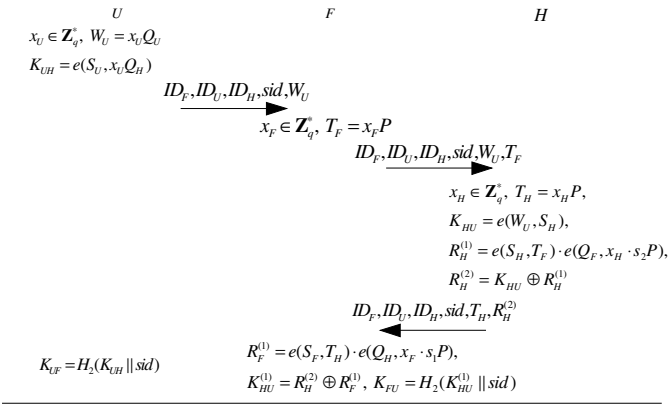


Fig. 2. IDB-HWKE protocol in the AM

3.3 Identity-Based KE Protocol for Heterogeneous Wireless Access (IDB-HWKE)

Our protocol aims at exchanging a common session key between *U* and *F* through involving *H* when user roams into a foreign network and wants to access it. Figure 2 illustrates the secure IDB-HWKE protocol. Formally, we give the protocol's security in Theorem 1.

Theorem 1. *Let H_1 and H_2 be random oracles. Then protocol IDB-HWKE is SK-secure in the AM if the BDH Assumption holds.*

The security proof follows the general technique in the security proof in [8], and we omit the details here because of the space limitation.

4 Mixed Use of MT-Authenticators

In light of the CK-model, a full-fledged authenticator should be built depending on the number of message flows in the AM. The full-fledged authenticator to transform the message flows in IDB-HWKE protocol can be achieved according to the following Theorem 2 [10].

Theorem 2 ([10]). *Let λ be an MT-authenticator (e.g. λ emulates message transmission MT in unauthenticated networks), and let C_λ be a complier constructed based on λ. Then C_λ is an authenticator.*

Theorem 2 implies authentication of a single message and puts no restriction on how the other messages of the same AM protocol are authenticated, thus is independent of the number of messages or participants. Since protocol IDB-HWKE contains three message flows among *U*, *F* and *H*, it is a natural way to build secure protocol in UM through combining three MT-authenticators to convert different messages in the AM.

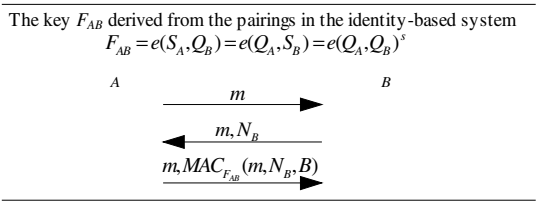


Fig. 3. MAC based MT-authenticator using identity-based static key

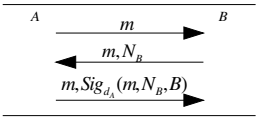


Fig. 4. Signature based MT-authenticator λ_{Sig}^{Nonce} by using nonce

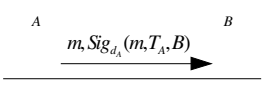


Fig. 5. One-flow signature based MT-authenticator using time stamps

For the identity-based infrastructure is already deployed to design the key exchange protocol in AM, it is practical and convenient to reuse them to construct authenticator. Here, we utilize the existed securely identity-based statically keyed MT-authenticator given in [13] to transform messages from AM to UM between *U* and *H*. Also for the message flow delivered from *H* to *F*, we employ the signature based MT-authenticator λ_{Sig}^{Nonce} with nonce [8]. A MAC based MT-authenticator λ_{MAC} by using identity-based static key and signature based MT-authenticator λ_{Sig}^{Nonce} with nonce are respectively proved secure in [13, 8] and shown in Figure 3 and 4.

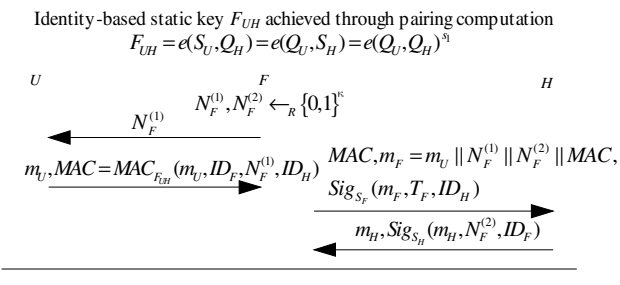


Fig. 6. Full-fledged authenticator by mixed using of MT-authenticators

Figure 5 illustrates a signature based MT-authenticator λ_{Sig}^{Time} [14] by using time stamps to provide message freshness. When applied this MT-authenticator to compile the message sent by F to H, extra message flow to transmit the random number is not required, it thus saves the link bandwidth. By using the time stamp T_A, we need a secure time server TS which is modeled as a universal time oracle available to all parties. The security proof of λ_{Sig}^{Time} is given in [14].

Through combining the MT-authenticators described above, we automatically obtain the full-fledged authenticator $C_{\lambda_{Sig}}^{\lambda_{MAC}}$ represented in Figure 6 appropriate for transforming IDB-HWKE protocol to UM one.

5 Secure Protocol in the UM

5.1 Identity-Based Key Exchange Protocol with Mutual Authentication

By using the new derived authenticator $C_{\lambda_{Sig}}^{\lambda_{MAC}}$, the IDB-HWKE protocol can be compiled to the secure UM protocol with security proof according to the Theorem 3 [8].

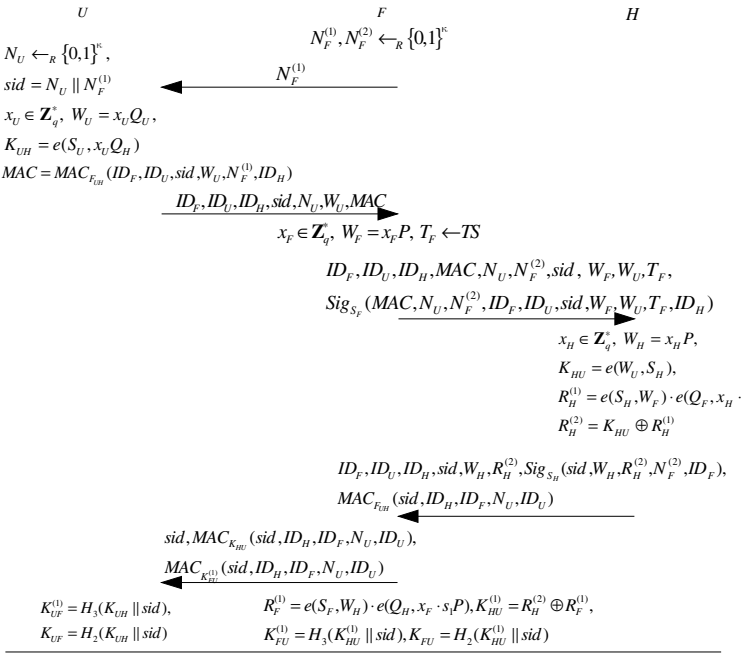


Fig. 7. Identity-based AKE protocol in the UM for heterogeneous wireless access

Theorem 3 ([8]). *Let π be a SK-secure key exchange protocol in the AM. Let λ be an MT-authenticator and let C_λ be an authenticator constructed based on λ. Then $\pi' = C_\lambda(\pi)$ is a SK-secure key exchange protocol in the UM.*

Generally, the automatically achieved UM protocol only realizes the "explicit" single-side entity authentication of U by H. Here we add on an additional authenticated message containing the identities and random nonce N_U from F to U to offer "explicit" mutual authentication. Note that hash function H_3 is also modeled as random oracle aiming at calculating a MACed key. Figure 7 illustrates the achieved AKE protocol in UM. In the protocol, $T_F \leftarrow TS$ denotes that F gets a time from the time server TS to form the time stamp. Note that the added messages $MAC_{K_{HU}}(sid, ID_H, ID_F, N_U, ID_U)$ and $MAC_{K_{FU}^{(1)}}(sid, ID_H, ID_F, N_U, ID_U)$ are used to provide key confirmation by U, thus to implement "explicit" mutual authentication, and these messages does not impair the total protocol's security. The session identifier s is a concatenation of N_U and $N_F^{(1)}$.

5.2 Further Security Analysis

We further discuss the protocol's forward secrecy heuristically in this section. When compromise only the long-term secret key S_U, or S_F, it will not do harm to the security in the past. If S_H is compromised, the past communication will be compromised because the past key K_{HU} can be calculated through S_H and W_U eavesdropped in that session. Also compromise of both S_U and S_F still leads to the compromise of the past communications. Compromise of the KGC_1's private key s_1 will lead to the compromise of the past communications because the secret K_{HU} can be calculated as $e(W_U, s_1Q_H)$. This means that our protocol does not provide KGC forward secrecy. If any temporary secret x_U, x_F, or x_H is compromised, it reveals none of the long-term private keys as well as the agreed session key between U and F. Hence, our protocol provides partial forward secrecy.

5.3 Performance Analysis

Since in wireless communications, the mobile terminal is restricted to perform complicated time-consuming operations because of the battery power, clock frequency and storage space. So, we mainly consider a computational load on user side of the proposed scheme.

Table 1. Computational overhead among the U, F, and H

Operation	U	F	H
The number of MAC operation	3	1	2
The number of hash operation	2	2	0
Point multiplication	1	2	2
Evaluations of the pairing	$1(+1)^{(-)}$	2	3(+1)
The number of PRF operation	2	3	2
Signature / Verification	0	1/1	1/1

$^{(-)}$ The number in bracket denotes the number of the precomputation

Because of the asymmetric feature in wireless environment, our protocol is designed to coincide with this feature that a majority of time-consuming operations are

shifted into the fixed servers. In our protocol, U is required to perform 2 evaluations of the pairing, and one of the pairing calculations for generating the static key F_{UH} can be precomputed. Table 1 shows the asymmetric feature of our scheme.

We also compare our scheme with recently proposed identity-based tripartite authenticated key agreement protocols [5, 6, 7]. Because of the role symmetric feature in [5, 6, 7], we focus on any entity in each protocol, and compare them with the user in our scheme in terms of the computational overhead in Table 2. From Table 2, one can find that our protocol is much more attractive for low power user terminal in terms of time-consuming operations, e.g. pairing computation and point multiplication.ïï

Table 2. Comparison of computational load among protocols

Operations	Our protocol	Zhang-Liu-Kim(--) [6]	Shim [7]	Nalla-Reddy (ID-AK-3) [5]
Evaluations of the pairing	1(+1) (-)	8	3	4
Point multiplication	1	6	5	4
Number of hash operation	2	3	4	3
Exponentiations	0	8	1	0
Number of MAC operation	3	0	0	0

(-) The number in bracket denotes the number of the precomputation

(--) Note that one round of Zhang et al.'s protocol can generate eight session keys using eight paring operations

6 Conclusion

According to the CK-model, we have designed an identity-based AKE with pairings for heterogeneous wireless access. Since the provable security CK-model is adopted, our protocol can automatically carry the security proof. Instead of simply applying the authenticators to each AM protocol message, our protocol provides "explicit" mutual authentication between the user and network. In addition, we further analyze the security of the protocol heuristically, and designate that the protocol does not provide perfect forward secrecy. Compared with recently proposed provably secure tripartite key agreement protocol, our scheme is computational asymmetry and more attractive for heterogeneous wireless access.

Acknowledgement. This work was supported in part by the National Natural Science Foundation of China under Grant No.60372076.

References

1. Shamir, A.: Identity-based cryptosystems and signature schemes. In Advances in Cryptology - Crypto'84. Lecture Notes in Computer Science, Vol. 196. Springer-Verlag (1984) 47-53
2. Joux, A.: A one round protocol for tripartite Diffie-Hellman. Proc. of Algorithm Number Theory Symposium - ANTS-IV, Lecture Notes in Computer Science Vol. 1838. Springer-Verlag (2000) 385-394

3. Al-Riyami, S. S. and Paterson, K. G.: Tripartite Authenticated Key Agreement Protocols from Pairings. IMA Conference on Cryptography and Coding, Lecture Notes in Computer Science, Vol. 2898. Springer-Verlag (2003) 332-359
4. Boneh, D. and Franklin, M.: Identity Based Encryption From the Weil Pairing. Advances in Cryptology – Crypto'01, Lecture Notes in Computer Science, Vol. 2139. Springer-Verlag (2001) 213-229
5. Nalla, D. and Reddy, K.C.: ID-based tripartite Authenticated Key Agreement Protocols from pairings. Cryptology ePrint Archive, Report. 2003/004
6. Zhang, F., Liu, S. and Kim, K.: ID-Based One Round Authenticated Tripartite Key Agreement Protocol with Pairings. Cryptology ePrint Archive, Report 2002/122
7. Shim, K.: A Man-in-the-middle Attack on Nalla-Reddy's ID-based Tripartite Authenticated Key Agreement Protocol. Cryptology ePrint Archive, Report, 2003/115
8. Canetti, R. and Krawczyk, H.: Analysis of key-exchange protocols and their use for building secure channels. In *Advances in Cryptology – Eurocrypt 2001*, Lecture Notes in Computer Science, Vol. 2045. Springer-Verlag (2001) 453-474
9. Bellare, M. and Rogaway, P.: Entity authentication and key distribution. In *Advances in Cryptology – Crypto'93*, Lecture Notes in Computer Science, Vol. 773. Springer-Verlag (1994) 232-249. Full version at http://www.cse.ucsd.edu/users/mihir/papers/eakd.pdf
10. Bellare, M., Canetti, R. and Krawczyk, H.: A modular approach to the design and analysis of authentication and key exchange protocols. In *Proceedings of the 30th Annual Symposium on the Theory of Computing, ACM*, pp. 412-428, 1998. Full version at http://www.cse.ucsd.edu/users/mihir/papers/modular.pdf
11. Goldwasser, S. and Micali, S.: Probabilistic encryption. *Journal of Computer and System Science*, 28(2): 270-299, 1984
12. Yiu Shing Terry Tin, Boyd, C. and Juan Manuel Gonz´alez Nieto.: Provably Secure Mobile Key Exchange: Applying the Canetti-Krawczyk Approach. In *Proceedings of ACISP 2003*, Lecture Notes in Computer Science, Vol. 2727. Springer-Verlag (2003) 166-179
13. Boyd, C., Mao, W. B. and Paterson, K.: Key Agreement using Statically Keyed Authenticators. *Applied Cryptography and Network Security: Second International Conference, ACNS 2004*, Lecture Notes in Computer Science, Vol. 3089. Springer-Verlag (2004) 248-262
14. Yiu Shing Terry Tin, Harikrishna Vasanta, Boyd, C. and Juan Manuel Gonz´alez Nieto.: Protocols with Security Proofs for Mobile Applications. In *Proceedings of ACISP 2004*, Lecture Notes in Computer Science, Vol. 3108. Springer-Verlag (2004) 358-369. Full version of this paper is available at http://sky.fit.qut.edu.au/~boydc/papers/

Efficient Identity Based Proxy-Signcryption Schemes with Forward Security and Public Verifiability

Meng Wang[1], Hui Li[2], and Zhijing Liu[1]

[1] School of Computer Science and Technology
[2] School of Telecommunications Engineering,
Xidian University, Xi'an 710071, P.R. China
wangmeng@xidian.edu.cn, lihui@mail.xidian.edu.cn

Abstract. Li and Chen proposed an identity based proxy-signcryption scheme which is based on the Libert and Quisquater's identity based signcryption scheme recently. However, we demonstrate that their scheme does not satisfy the strong unforgeability and forward security in the strict sense. Furthermore, establishing a secure channel has much influence on their scheme. Based on the new ID-based signcryption scheme proposed by Chow *et al*, an efficient forward secure and public verifiable identity based proxy-signcryption scheme without the secure channel is proposed. The proposed scheme overcomes the weakness of Li-Chen scheme. The security and performance of the scheme are also analyzed.

1 Introduction

The idea of an identity-based encryption (IBE) scheme is that an arbitrary string such as a user's email address can serve as a public key. Signcryption, first proposed by Zheng [1], is a new cryptographic primitive which simultaneously fulfill both the functions of signature and encryption in a single logical step, and with a computational cost significantly lower than that required by the traditional signature-then-encryption approach. A proxy signature scheme allows one user Alice, called original signer, to delegate her signing capability to another user Bob, called proxy signer.

The basic idea of ID-Based proxy-signcryption schemes is as follows. The original signcrypter Alice sends a specific message with its signature to the proxy signcrypter Bob, who then uses this information to construct a proxy private key. With the proxy private key, Bob can generate proxy signcryption by employing a specified standard ID-Based signcryption scheme. When a proxy signcryption is given, a verifier first computes the proxy public key from some public information, and then checks its validity according to the corresponding standard ID-Based signcryption verification procedure.

Following the definitions from [2] and [3], a strong ID-based proxy-signcryption scheme should satisfy the following properties.

- **Forward Security (FwSec):** An attacker cannot reveal the messages signcrypted before even with the knowledge of the sender's private key.
- **Public Verifiability (PubVer):** The origin of the ciphertext can be verified by a third party without knowing the recipient's private key.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 982 – 991, 2005.

- **Provable Security (ProvSec):** An ID-based signcryption scheme is said to be provably secure if it satisfies the property of semantical security and is secure against an existential forgery for adaptive chosen-message-and-identity attacks [2].
- **Strong unforgeability (StrUnfg):** The original signer and other third parties who are not designated as a proxy signer cannot create a valid proxy signature.
- **Verifiability (Ver):** The original signer's delegation on the signed message is verifiable using publicly available parameters.
- **Strong identifiability (StrIde):** Anyone can determine the identity of the corresponding proxy signer from a proxy signature.
- **Strong undeniability (StrUnd):** Once a proxy signer creates a valid proxy signature for an original signer, he cannot repudiate his signature creation against anyone else.
- **Prevention of misuse (PreMis):** The proxy cannot use the proxy key for other purposes than generating a valid proxy signature.

Recently, Li and Chen proposed an identity based proxy signcryption scheme[4]. However, their scheme is based on the Libert and Quisquater's ID-based signcryption scheme [5] which does not satisfy the above requirements at the same time. In addition, establishing a secure channel for the delivery of the proxy-signcryption key has much influence on their scheme.

We demonstrate that Li-Chen scheme is not a proxy-protected one and provide a revised version which satisfy the strong unforgeability without the secure channel. In this paper, we also propose an identity proxy-signcryption scheme which can satisfy all the above requirements without the secure channel. In fact, our proposed scheme is based on the ID-based signcryption scheme proposed by Chow *et al* [2] which is modified from the Libert and Quisquater's scheme.

The rest of this paper is organized as follows. Section 2 contains some formal definitions of bilinear mapping. Li-Chen scheme is briefly reviewed and analyzed in section 3, we also provide a revised version in section 3.Our scheme is presented in Section 4.We analyze the security and performance of our scheme and show that it satisfies all the requirements above. In Section 5, we compare the functions and efficiency of each scheme. Section 6 concludes the paper.

2 Bilinear Mapping and Bilinear Diffie-Hellman Problems

Let $(G_1,+)$ and $(G_2,\cdot)$ be two cyclic groups of prime order q. The bilinear pairing is given as $\hat{e}: G_1 \times G_1 \to G_2$,which satisfy the following properties:

1. Bilinear: For $P,Q,R \in G_1$, there exists:
   $\hat{e}(P+Q,R)=\hat{e}(P,R)\hat{e}(Q,R)$, and $\hat{e}(P,Q+R)=\hat{e}(P,Q)\hat{e}(P,R)$.
2. Non-degenerate: There exists $P,Q \in G_1$ such that $\hat{e}(P,Q) \neq 1$.
3. Computable: There exists an efficient algorithm to compute $\hat{e}(P,Q) \forall P,Q \in G_1$.

Now we describe some mathematical problems in G_1.

- **Discrete Logarithm Problem (DLP):** Given two group elements P and Q, find an integer n, such that $Q = nP$ whenever such an integer exists.
- **Decision Diffie-Hellman Problem (DDHP):** For $a,b,c \in Z_q^*$, given P, aP, bP, cP, decide whether $c \equiv ab \mod q$.
- **Computational Diffie-Hellman Problem (CDHP):** For $a,b,c \in Z_q^*$, given P, aP, bP, compute abP.

In this paper, we consider variants of DDHP and CDHP, in which $c^{-1}P$ is also given as input. We refer these variants as MDDHP (Modified DDHP) and MCDHP (Modified CDHP). No known existing algorithm can solve them. Note that the private signcryption key and the private decryption key are separated in our scheme.

3 Review and Cryptanalysis of Li-Chen Scheme

3.1 Brief Review of Li-Chen Scheme

The Private Key Generator (PKG) publishes system's public parameters:

$$\{ G_1, G_2, n, k, \hat{e}, P, P_{pub}, H, H_1, H_2, H_3, E, D \}$$

Then Alice chooses $x \leftarrow_R Z_q^*$ and computes $U = xP$, $d_{ap} = H(m_w \| U) d_{ID_A} + xP_{pub}$, sends (m_w, U, d_{ap}) as the delegation to the proxy signcrypter securely.

To signcrypt a message $m \in \{0,1\}^*$, a proxy signcrypter chooses $x' \leftarrow_R Z_q^*$ and computes

$$\begin{aligned}
Q_{ID_B} &= H_1(ID_B) \\
k_1 &= \hat{e}(P, P_{pub})^{x'} \\
k_2 &= H_3(\hat{e}(P_{pub}, Q_{ID_B})^{x'}) \\
c &= E_{k_2}(m) \\
r &= H_2(c, k_1) \\
S &= x' P_{pub} - (r d_{ID_p} + d_{ap})
\end{aligned}$$

When receiving (m_w, U, c, r, S), Bob performs the following tasks:

computes

$$\begin{aligned}
k_1' &= \hat{e}(P,S)\hat{e}(P_{pub}, Q_{ID_P})^r \hat{e}(P_{pub}, Q_{ID_A})^{H(m_w \| U)} \hat{e}(U, P_{pub}) \\
k_2' &= H_3(\hat{e}(S, Q_{ID_B})\hat{e}(Q_{ID_P}, d_{ID_B})^r \hat{e}(Q_{ID_A}, d_{ID_B})^{H(m_w \| U)} \hat{e}(U, d_{ID_B})) \\
m &= D_{k_2'}(c)
\end{aligned}$$

If $r \neq H_2(c, k_1')$ returns $\perp$ else accepts m.

3.2 Cryptanalysis of Li-Chen Scheme

As mentioned above, the scheme needs a secure channel in the proxy delivery step. Otherwise, anyone who obtained the proxy key d_{ap} and d_{ID_P} can compute

$$H_3(\hat{e}(S, Q_{ID_B})\hat{e}(d_{ID_P}, Q_{ID_B})^r \hat{e}(d_{ap}, Q_{ID_B}))$$

without the knowledge of d_{ID_B} ,so the scheme is not a forward secure one in a strict sense. When Li analyze the security of their scheme, they only consider the situation that first the original delegates the signcrypting capability and then the proxy signcrypted the delegated message. However, we found the fact that the order can be reversed. Here, some possible attacks are proposed to show that this scheme is not a proxy-protected one.

Attack 1. Note that the scheme is derived from the Libert and Quisquater's ID-based signcryption scheme. The signcryption key d_{ap} is only simply added to the signature S .So the original signcrypter can remove d_{ap} from S to obtain a plain Libert and Quisquater ID-based signcryption of the proxy signcrypter easily.

Attack 2. Now we take a closer look at the equation:

$$\begin{aligned} S &= x' P_{pub} - (r d_{ID_P} + d_{ap}) \\ &= S_P - d_{ap} \end{aligned}$$

Where S_P is a valid Libert and Quisquater's ID-based signcryption on the message m by the proxy signcrypter and the second term d_{ap} is generated by the original signcrypter. Using this method, every Libert and Quisquater's ID-based signcryption can be converted to a proxy-signcryption one in which the signer is regarded as the proxy signer by the verifier.

Attack 3. If the original signcrypter is a dishonest one or once the signcryption key d_{ap} is revealed. The attacker C can perform the following operation to change the original signer from A to C regardless of the proxy signcrypter's will.

Step 1: The proxy signcrypter P sends (m_w, U, c, r, S) to Bob in which the original signcrypter is Alice.

Step 2: C intercepts (m_w, U, c, r, S), and computes:

$$x' \leftarrow_R Z_q^* \ , \ U' = x'P, \ d_{cp} = H(m_w' \| U')d_{ID_C} + x' P_{pub},$$

in which the warrant m_w' includes the information of original signcrypter C and the proxy P.

Step 3: C computes $S' = S + d_{ap} - d_{cp}$, and sends (m_w', U', c, r, S')to Bob.

Step 4: Bob computes $k_1' = \hat{e}(P,S)\hat{e}(P_{pub}, Q_{ID_P})^r \hat{e}(P_{pub}, Q_{ID_C})^{H(m_w' \| U')} \hat{e}(U', P_{pub})$. If $r = H_2(c, k_1')$, Bob accepts it and regards C as the original signcrypter.

3.3 Revision of Li-Chen Scheme

The weakness of the Li-Chen scheme results from the fact that the proxy signcryption key is not modified by the proxy signcrypter before using it and is only simply added in the signcryption stage. We revise the scheme as follows:

[Generation of the proxy key]

Alice chooses $x \leftarrow_R Z_q^*$ and computes: $U = xP$, $d_{ap} = H(m_w \| U)d_{ID_A} + xP_{pub}$, and then sends (m_w, U, d_{ap}) directly to the proxy signcrypter P.

After accepting (m_w, U, d_{ap}) by calculating $\hat{e}(P, d_{ap}) = \hat{e}(P_{pub}, Q_{ID_A})^{H(m_w \| U)} \hat{e}(U, P_{pub})$. P computes $d_P = H(m_w \| U)d_{ID_P} + d_{ap}$, and takes d_P as the proxy signcryption key.

[Proxy Signcryption]

P computes $S = x' P_{pub} - (rd_{ID_P} + d_P)$ and sends (m_w, U, c, r, S) to Bob.

[Unsigncryption]

When receiving (m_w, U, c, r, S), Bob calculates

$$k_1' = \hat{e}(P, S)\hat{e}(P_{pub}, Q_{ID_P})^{H(m_w \| U)+r} \hat{e}(P_{pub}, Q_{ID_A})^{H(m_w \| U)} \hat{e}(U, P_{pub})$$
$$k_2' = H_3(\hat{e}(S, Q_{ID_B})\hat{e}(Q_{ID_P}, d_{ID_B})^{H(m_w \| U)+r} \hat{e}(Q_{ID_A}, d_{ID_B})^{H(m_w \| U)} \hat{e}(U, d_{ID_B}))$$

3.4 Security Analysis

Consistency. The consistency can be easily verified by the following equations:

$$\begin{aligned}
k_1' &= \hat{e}(P, S)\hat{e}(P_{pub}, Q_{ID_P})^{H(m_w \| U)+r} \hat{e}(P_{pub}, Q_{ID_A})^{H(m_w \| U)} \hat{e}(U, P_{pub}) \\
&= \hat{e}(P, S)\hat{e}(P, rd_{ID_P})\hat{e}(P, H(m_w \| U)d_{ID_P} + H(m_w \| U)d_{ID_A} + xP_{pub}) \\
&= \hat{e}(P, x' P_{pub} - rd_{ID_P} - d_P)\hat{e}(P, rd_{ID_P})\hat{e}(P, d_P) \\
&= \hat{e}(P, P_{pub})^{x'} = k_1 \\
k_2' &= H_3(\hat{e}(S, Q_{ID_B})\hat{e}(Q_{ID_P}, d_{ID_B})^{H(m_w \| U)+r} \hat{e}(Q_{ID_A}, d_{ID_B})^{H(m_w \| U)} \hat{e}(U, d_{ID_B})) \\
&= H_3(\hat{e}(S, Q_{ID_B})\hat{e}(rd_{ID_P}, Q_{ID_B})\hat{e}(H(m_w \| U)d_{ID_P} + H(m_w \| U)d_{ID_A} + xP_{pub}, Q_{ID_B})) \\
&= H_3(\hat{e}(x' P_{pub} - rd_{ID_P} - d_P, Q_{ID_B})\hat{e}(rd_{ID_P}, Q_{ID_B})\hat{e}(d_P, Q_{ID_B})) \\
&= H_3(\hat{e}(P_{pub}, Q_{ID_B})^{x'} = k_2
\end{aligned}$$

Strong unforgeability. Note that we do not require a secure channel for the delivery of the signed warrant. In order to show that the revised scheme is secure under attack 1 to 3, we must show that the original signcrypter and the proxy signcrypter cannot create the proxy signcryption by the reversing order method. Now we consider the following equations:

$$S = x' P_{pub} - (rd_{ID_P} + d_P)$$
$$= \underbrace{x' P_{pub} - rd_{ID_P}}_{1} - \underbrace{(H(m_w \| U) d_{ID_P} + d_{ap})}_{2}$$

There is a plain Libert and Quisquater's ID-based signcryption in term(1). The scheme is secure under Attack1 because the original signcrypter cannot obtain term(1) even after removing d_{ap} from S, so the revised scheme is secure under Attack 1. In addition, in order that an attacker creates the proxy-signcryption generated by our revised scheme from the valid Libert and Quisquater's ID-based signcryption in the form of term(1), the attacker should be able to compute term(2). However, in order to do that he must compute d_{ID_P} from term(2) for the first. We know that the difficulty is equal to solving DLP, and it is still infeasible to do that. Since the attacker C who wants to change the original signcrypter from A to C will have to solve the same problem, our revised scheme is secure under Attack 2 and Attack 3. So our revised scheme is a proxy-protected one which satisfies the strong unforgeability.

3.5 Performance Analysis of the Revised Scheme

We assume that the computation of pairings is the most time consuming. Under this assumption, when user often communicate with each other, those parings

$$\hat{e}(P, P_{pub}), \hat{e}(P_{pub}, Q_{ID_P}), \hat{e}(P_{pub}, Q_{ID_A}), \hat{e}(U, P_{pub}), \hat{e}(Q_{ID_P}, d_{ID_B}), \hat{e}(Q_{ID_A}, d_{ID_B}), \hat{e}(U, d_{ID_B})$$

can be pre-computed similar to Li-Chen scheme. Furthermore, because establishing a secure channel for the delivery of proxy signcryption key has much influence on the efficiency of the scheme, so our revised scheme without the secure channel will be more efficient than the original one.

4 Our Proposed Scheme

[Setup]

Given a security parameter k, the PKG chooses groups G_1 and G_2 of prime order q, a generator P of G_1, a bilinear map $\hat{e}: G_1 \times G_1 \to G_2$ and hash functions :

$$H: \{0,1\}^* \to Z_q^*,\ H_1: \{0,1\}^* \to G_1,\ H_2: G_2 \to \{0,1\}^n,\ H_3: \{0,1\}^n \times G_2 \to Z_q^*.$$

It chooses master key $s \in Z_q^*$ and computes $P_{pub} = sP$ .It also chooses secure symmetric ciper (E, D) which takes a plaintext ciphertext of length n respectively, and also a key of length n. The PKG publishes system's public parameters

$$\{ G_1, G_2, n, q, \hat{e}, P, P_{pub}, H, H_1, H_2, H_3, E, D \}$$

and keeps the master key S secret.

[**Extraction**]

Given an identity ID, the PKG computes $Q_{ID} = H_1(ID)$, the private signcryption key $S_{ID} = s^{-1}Q_{ID}$ and the private decryption key $D_{ID} = sQ_{ID}$.

[**Generation of the proxy key**]

Alice chooses $x \leftarrow_R Z_q^*$, and computes $U = xP$, $d_{ap} = H(m_w \| U)D_{ID_A} + xP_{pub}$, then sends (m_w, U, d_{ap}) as the delegation to the proxy signcrypter. There is an explicit description of relative rights and information of the original signcrypter and the proxy signcrypter in the warrant m_w.

The proxy signcrypter accepts d_{ap} as a valid proxy signcryption key only if the equation is satisfied: $\hat{e}(P, d_{ap}) = \hat{e}(P_{pub}, Q_{ID_A})^{H(m_w \| U)} \hat{e}(P_{pub}, U)$.

If it is finished successfully,the proxy signcrypter can signcrypt any message which conforms to the warrant on the behalf of the original signcrypter.

[**Proxy Signcryption**]

To signcrypt a message $m \in \{0,1\}^n$, a proxy signcrypter chooses $x' \leftarrow_R Z_q^*$, and computes

$$
\begin{aligned}
&Q_{ID_B} = H_1(ID_B)\\
&k_1 = \hat{e}(P, Q_{ID_P})^{x'}\\
&k_2 = H_2(\hat{e}(Q_{ID_P}, Q_{ID_B})^{x'})\\
&c = E_{k_2}(m)\\
&r = H_3(c, k_1)\\
&S = (x' - r - d_{ap})S_{ID_P}
\end{aligned}
$$

[**Unsigncryption**]

When receiving (m_w, U, c, r, S), Bob performs the following tasks:

computes

$$
\begin{aligned}
&Q_{ID_A} = H_1(ID_A)\\
&Q_{ID_P} = H_1(ID_P)\\
&k_1' = \hat{e}(P_{pub}, S)\hat{e}(P, Q_{ID_P})^r \hat{e}(P_{pub}, Q_{ID_A}Q_{ID_P})^{H(m_w \| U)} \hat{e}(P_{pub}, UQ_{ID_P})\\
&k_2' = H_2(\hat{e}(D_{ID_B}, S)\hat{e}(Q_{ID_B}, Q_{ID_P})^r \hat{e}(D_{ID_B}, Q_{ID_A}Q_{ID_P})^{H(m_w \| U)} \hat{e}(D_{ID_B}, UQ_{ID_P}))\\
&m = D_{k_2'}(c)
\end{aligned}
$$

If $r \neq H_3(c, k_1')$ returns $\perp$ else accepts m.

4.1 Security Analysis

Consistency. The consistency can be easily verified by the following equations:

$$k_1' = \hat{e}(P_{pub}, S)\hat{e}(P, Q_{ID_P})^r \hat{e}(P_{pub}, Q_{ID_A}Q_{ID_P})^{H(m_w \| U)} \hat{e}(P_{pub}, UQ_{ID_P})$$

$$= \hat{e}(P_{pub}, S)\hat{e}(P_{pub}, rS_{ID_P})\hat{e}(P_{pub}, (H(m_w \| U)D_{ID_A} + xP_{pub})S_{ID_P})$$
$$= \hat{e}(P_{pub}, (x' - r - d_{ap})S_{ID_P})\hat{e}(P_{pub}, rS_{ID_P})\hat{e}(P_{pub}, d_{ap}S_{ID_P})$$
$$= \hat{e}(P, Q_{ID_P})^{x'} = k_1$$

$$k_2' = H_2(\hat{e}(D_{ID_B}, S)\hat{e}(Q_{ID_B}, Q_{ID_P})^r \hat{e}(D_{ID_B}, Q_{ID_A}Q_{ID_P})^{H(m_w \| U)} \hat{e}(D_{ID_B}, UQ_{ID_P}))$$
$$= H_2(\hat{e}(D_{ID_B}, S)\hat{e}(D_{ID_B}, rS_{ID_P})\hat{e}(D_{ID_B}, (H(m_w \| U)D_{ID_A} + xP_{pub})S_{ID_P}))$$
$$= H_2(\hat{e}(D_{ID_B}, (x' - r - d_{ap})S_{ID_P})\hat{e}(D_{ID_B}, rS_{ID_P})\hat{e}(D_{ID_B}, d_{ap}S_{ID_P}))$$
$$= H_2(\hat{e}(Q_{ID_B}, Q_{ID_P})^{x'}) = k_2$$

Forward Security. Unsigncryption requires the knowledge of $\hat{e}(Q_{ID_B}, Q_{ID_P})^{x'}$. For an attacker, it is difficult to get x' from k_1 since it is difficult to invert the bilinear mapping. In addition, it is difficult to compute x' from S even with the knowledge of r and d_{ap} and S_{ID_P} since it is difficult to compute DLP. Thus, our proposed scheme is forward secure in the strict sense.

Public Verifiability. Any third party can be convinced of the message's origin by recovering k_1' and checking if the equality $r = H_3(c, k_1')$ holds. Thus, the origin of ciphertext can be verified without the help of recipient in our proposed scheme.

Provably Security. Following the ideas in [5] and [6], our scheme is both IND-IDSC-CCIS2 and EUF-IDSC-CMIA2 secure.

Strong Unforgeability. In order to show that the revised scheme is secure under attack 1 to 3, we must show that the original signcrypter and the proxy signcrypter cannot create the proxy signcryption by the reversing order method. Now we consider the following equations:

$$S = (x' - r - d_{ap})S_{ID_P}$$
$$= \underbrace{(x' - r)S_{ID_P}}_{1} - \underbrace{d_{ap}S_{ID_P}}_{2}$$

There is a plain Chow *et al*'s ID-based signcryption in term(1).Any attacker who want to remove term(2) from the valid S to get term(1) will have to compute term(2) for the first. However, it is difficult to computer term(2) without the knowledge of S_{ID_P}. Furthermore, the attacker C who wants to change the original signcrypter from A to C will have to computer term(2) for himself too. We know that it is difficult to compute S_{ID_P} from term(2) since it is difficult to compute DLP. Thus, our scheme satisfies the strong unforgeability and is secure under Attack 1 to 3.

4.2 Performance Analysis

Follow the ideas in [4], we assume that the computation of pairings is the most time consuming. Under this assumption, when users often have to communicate with each other, all those pairings can be pre-computed. In this case the most expensive operations are four exponentiations in G_2 and two pairing evaluations. Since the secure channel is not necessary, the proposed scheme is more efficient than Li-Chen scheme.

5 Comparison

Table 1 shows the comparison of Li-Chen scheme, the revised version and our proposed scheme in some security requirements mentioned above and the efficiency.

Table 1. Comparison of features and efficiency of existing proxy-signcryption schemes

Schemes	FwSec	PubVer	StrUnfg	Pa	Ex	Sc
Li-Chen scheme	N	Y	N	2(+8)	4(+2)	Y
Revised Version	N	Y	Y	2(+8)	4(+2)	N
Our Scheme	Y	Y	Y	2(+8)	4(+2)	N

1. Paring(Pa): The total number of pairing computations required in the stage of proxy signcryption and unsigncryption. In the table, we represent this total in the form of x(+y) where y is the number of operations which can be pre-computed and cached for subsequent uses if people often communicate with their partners.
2. Exponentiation(Ex): The total number of exponentiations required. As mentioned above, the total is in the form of x(+y) where y is the number of operations which can be pre-computed.
3. Secure Channel(Sc): The scheme needs a secure channel or not.

6 Conclusions

We analyze Li-Chen scheme and show that it is not a proxy protected one and does not satisfy the forward security in the strict sense. We also provide an efficient revised version of their scheme which is a proxy-protected scheme. With the assumption that the MCDHP and MDDHP are hard to compute, we propose an efficient ID-based proxy-signcryption scheme which satisfies both forward security and public verifiability without the secure channel. The proposed scheme is secure and the performance is superior to Li-Chen scheme.

Acknowledgments

This research was supported by the National Natural Science Foundation of China under grant number 60173056.

References

1. Yuliang Zheng.: Digital Signcryption or How to Achieve Cost (Signature & Encryption) << Cost (Signature) + Cost (Encryption). CRYPTO 1997, Lecture Notes in Computer Science, Vol.1294, pp.165-179.Springer-Verlag, 1997.
2. Sherman S.M.Chow, S.M.Yiu, Lucas C.K.Hui, and K.P.Chow.: Efficient Forward and Provably Secure ID-Based Signcryption Scheme with Public Verifiability and Public Ciphertext Authenticity. ICISC 2003, Lecture Notes in Computer Science, Vol.2971, pp.352-369. Springer-Verlag, 2003.
3. B.Lee, H.Kim, K.Kim.: Strong Proxy Signature and its Applications, SCIS2001, Lecture Notes in Computer Science , Vol 2119, pp 603-608, 2001.
4. Xiangxue Li, Kefei Chen.: Identity Based Proxy-Signcryption Scheme from Pairings. Proceedings of the IEEE International Conference on Services Computing (SCC 2004), pp.494-497, 2004.
5. B.Libert and J.Quisquater.: New Identity Based Signcryption Schemes from Pairings. In IEEE Information Theory Workshop, pp.155-158, 2003.
6. Xavier Boyen. Multipurpose Identity-Based Signcryption.: A Swiss Army Knife for Identity-Based Cryptography. CRYPTO 2003, Lecture Notes in Computer Science, Vol.2729, pp.382-398. Springer Verlag, 2003.

PKM: A Pairwise Key Management Scheme for Wireless Sensor Networks*

F. An[1], X. Cheng[2], J.M. Rivera[2,3], J. Li[4], and Z. Cheng[5]

[1] Institute of Computing Technology, Chinese Academy of Sciences,
P.O. Box 2704, Beijing 100080, China
anfengguang@software.ict.ac.cn
[2] Department of Computer Science, The George Washington University,
801 22nd St. NW, Washington, DC 20052, USA
cheng@gwu.edu
[3] United States Army, Washington, DC, USA
jose.rivera@us.army.mil
[4] Department of Systems and Computer Science,
Howard University, Washington, DC, USA
lij@scs.howard.edu
[5] National Taxation Bureau of Rizhao City,
Shandong Province 276826, China
green_sd@163.com

Abstract. Sensor networks are characterized by strict resource limitations and large scalability. Many sensor network applications require secure communication, a crucial component, especially in harsh environments. Symmetric key cryptography is very attractive in sensor networks due to its efficiency, but establishing a shared key for communicating parties is very challenging. The low computational capability and small storage budget within sensors render many popular public-key based key distribution and management mechanisms impractical. In this paper, we propose and analyze a truly in-situ key management scheme for large scale sensor networks, called: Public Key Management (PKM). In this scheme, we deploy service and worker sensors. The service sensors contain a key space, while worker sensors are deployed blind, with no predeployment knowledge. Worker sensors obtain security information from service sensors through a secure channel after deployment. After obtaining security information, worker sensors compute shared keys with their neighbors. For security reasons, service sensors erase stored key space information after deployment. During this procedure, PKM shifts a large amount of computational overhead from worker sensors to service sensors, thus conserving worker sensors' resources. PKM's performance, in terms of storage, computational overhead and resiliency, is very good.

Keywords: Sensor networks, security, key management, key distribution.

* The research of Dr. Xiuzhen Cheng is supported by NSF CAREER Award No. CNS-0347674.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 992–1001, 2005.

1 Introduction

Today's smart sensors are plagued by strict resource limitations (battery, memory, CPU, etc.). For example, the MICA2 Berkeley mote has an 8-bit, 7.3828MHz Atmega 128L processor with 4KB SRAM and 128KB ROM [1]. However, these sensors are usually deployed in large scale with high density for monitoring and control. These endemic characteristics create challenging problems in establishing secure communication in sensor networks.

Many sensor network applications require security assurance. Due to its efficiency, symmetric key cryptography is very attractive in sensor networks. However, establishing shared keys between neighboring sensors cannot be considered trivial. The well-established Public Key Infrastructure (PKI) key distribution and management schemes are not applicable due to high computational overhead and large memory requirement. As reported by Carman *et. al* [4], a middle-ranged processor such as the Motorola MC68328 "DragonBall" consumes 42mJ (840mJ) for RSA encryption (digital signature) and 0.104mJ for AES, when the key size for both cases is 1024 bits.

A few key pre-distribution protocols for shared key construction have been proposed in literature [6,7,9] but they may not scale well to large sensor networks or may require strict deployment knowledge for better scalability. Further, all these schemes require some kind of security information to be pre-loaded to the memory of a sensor, assuring storage space waste, since some of the information may never be used during the lifetime of the sensor. In this paper we explore in-situ key computation, instead of security information pre-distribution. To achieve this goal we are willing to sacrifice the "Key Space Carriers", the service sensors. In the initial phase (pre-deployment), worker sensors do not need security information. They actualize their security posture after deployment by interacting with the service sensors. A secure channel is established for secure information transfer between a service sensor and a worker sensor. Once this interaction is completed, service sensors will erase the stored key space. This procedure exploits the asymmetric feature in the computational overhead of Rabin's crypto-system, thus shifting most of the computational burden to the service sensors.

This paper is organized as follows: we briefly overview related works in Section 2, our PKM scheme is proposed in Section 3, public key assisted secure channel establishment protocol is presented in Section 4, we analyze our protocol in Section 5 and conclude our paper in Section 6.

2 Related Work

In this section, we briefly survey the following shared key establishment schemes: random key pre-distribution, symmetric matrix based (Blom's method based) key computation and deployment knowledge based key management.

The pioneer work on random key pre-distribution for sensor networks is proposed by Eschenauer and Gligor in [9]. A large key pool K is computed offline

and each sensor picks k keys randomly from K without replacement before deployment. These k keys form the key ring of the sensor. After deployment, a sensor establishes a shared key with a neighbor if their key rings have at least one key in common. The security of random key pre-distribution is enforced by [5] in which $q > 1$ common keys are required to establish a shared key. These q keys are hashed into one key to achieve better resiliency to sensor capture.

Du *et. al* [6] is the first to apply Blom's scheme for shared key establishment in sensor networks. Blom's scheme is based on the computation of a symmetric matrix which provides a key space for all sensors that possess a public and a private share of the key space. In [6], ω key spaces instead of one key space is precomputed and each sensor stores the private/public shares of τ key spaces. These τ key spaces are randomly selected from the ω key spaces without replacement. If two sensors share information from one common key space, they can establish a shared key after exchanging their public shares. This scheme combines the idea of random key pre-distribution in [9] with Blom's method. We will elaborate Blom's scheme in the next section.

To improve scalability, a deployment knowledge based key management approach is proposed in [7]. In this scheme, multiple deployment points are identified in the sensor network and for each deployment point, a key space is precomputed. Neighboring deployment points have a number of keys in common. In other words, their key spaces consist of common keys. All sensors are grouped before deployment and each group corresponds to one deployment point. Each sensor randomly picks k keys from the key space of its group. After deployment, sensors in close neighborhood have a high probability of sharing a common key. This scheme places strong requirements on deployment, but achieves better scalability compared with those proposed in [6,9].

A geographic information based key management protocol is designed in [10]. A general framework for establishing pairwise keys in sensor networks is studied in [12], which is based on the polynomial-based key pre-distribution protocol proposed by [3]. A location-aware deployment model for shared key establishment is presented in [11].

Our work is different from all those mentioned above in that it is truly an in-situ key management scheme for sensor networks. We do not require any key-related information to be pre-distributed to worker sensors. Instead, we randomly deploy service sensors that convey security information to the worker sensors in the neighborhood. This protocol has better scalability, with no requirement of deployment knowledge.

3 PKM: The Pairwise Key Management Scheme

In this section, we propose PKM, a key management protocol for establishing pairwise keys between neighboring sensors. This protocol is based on Blom's λ-secure key management scheme [2], which has been well-tailored for light-weight sensor networks by [6]. In the following, we will give an overview on Blom's scheme based on [6].

3.1 Blom's Key Management Scheme

Let G be a $(\lambda+1)\times M$ matrix over a finite field $GF(q)$, where q is a large prime. The connotation of M will become clear later. G is public, with each column called a *public share*. Let D be any random $(\lambda+1)\times(\lambda+1)$ symmetric matrix. D must be kept private, which is known to the network service provider only. The transpose of $D\cdot G$ is denoted by A. That is, $A=(D\cdot G)^T$. A is private too, with each row called a *private share*. Since D is symmetric, $A\cdot G$ is symmetric too. If we let $K=(k_{ij})=A\cdot G$, we have $k_{ij}=k_{ji}$, where k_{ij} is the element at the ith row and the jth column of matrix K, $i,j=1,2,\cdots,M$.

The basic idea of Blom's scheme is to use k_{ij} as the secret key shared by sensor i and sensor j. D and G jointly define a *key space* (D,G). Any public share in G has a unique private share in A, which form a so-called *crypto pair*. For example, the ith column of a G, and the ith row of A form a crypto pair and the unique private share of the ith column of G, a public share, is the ith row of A. Two sensors whose crypto pairs are obtained from the same key space can compute a shared key after exchanging their public shares. From this analysis, it is clear that M is the number of sensors that can compute their pairwise keys based on the same key space.

In summary, Blom's scheme states the following protocol for sensors i and j to compute k_{ij} and k_{ji}, based on the same key space:

- Each sensor stores a unique crypto pair. Without loss of generality, we assume sensor i gets the ith column of G and the ith row of A, denoted by g_{ki} and a_{ik}, where $k=1,2,\cdots,\lambda+1$, respectively. Similarly, sensor j gets the jth column of G and the jth row of A, denoted by g_{kj} and a_{jk}, where $k=1,2,\cdots,\lambda+1$, respectively.
- Sensor i and sensor j exchange their stored public share drawn from their crypto pairs as plain texts.
- Sensor i computes k_{ij} as follows:

$$k_{ij}=\sum_{k=1}^{\lambda+1} a_{ik}\cdot g_{kj};$$

Similarly, sensor j computes k_{ji} by

$$k_{ji}=\sum_{k=1}^{\lambda+1} a_{jk}\cdot g_{ki}.$$

Blom's key management scheme ensures the so-called λ-secure property, which means that the network should be perfectly secure as long as no more than λ sensors are compromised. This requires that any $\lambda+1$ columns of G must be linearly independent. A good candidate of G can be the Vandermonde matrix:

$$G=\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ s & s^2 & s^3 & \cdots & s^M \\ s^2 & (s^2)^2 & (s^3)^2 & \cdots & (s^M)^2 \\ & & & \vdots & \\ s^\lambda & (s^2)^\lambda & (s^3)^\lambda & \cdots & (s^M)^\lambda \end{bmatrix},$$

where s is the primitive element of $GF(q)$ for some large prime number q with $M < q$. Note that only the second element of each column needs to be disseminated as a public share, from which other elements can be easily restored.

3.2 PKM: The Pairwise Key Management Protocol

Let each service sensor I carry a key space $(D, G)_I$ before deployment. Service sensors can announce their existence through beacon broadcasting after deployment, or we can allow worker sensors to query their neighborhood (can be multi hop) for service sensors. The topic on how to identify service sensors in the neighborhood is beyond the scope of this paper. We assume that there exists an idealized protocol, such that each worker sensor can be connected through single or multi hop to at least one service sensor.

In order for a worker sensor to obtain a column of G and the corresponding row of A from a service sensor, we need a secure channel between the worker sensor and the service sensor. A public-key encryption based protocol for secure channel establishment will be proposed in the next section. For now, we assume there exists a shared key K_s between a worker sensor and its service sensor.

Our pairwise key establishment protocol can be stated as follows.

- Each worker sensor sends a request to the service sensor, asking for a crypto pair containing a public and a private share. This message can be secured by K_s.
- Upon receiving a request from sensor i, service sensor I selects an unused crypto pair and then transmits it to i. This message must be encrypted by K_s.
- Two neighboring sensors exchange their public shares obtained from the same key space (through the same service sensor) to compute their pairwise key based on Blom's scheme.

This procedure can be further secured with the introduction of nonces to avoid replay attacks.

Note that in this protocol we do not ensure a globally unique id for each service sensor. But we do require that service sensors connected to a common worker sensor be uniquely identified in order for the sensor to tell the key spaces apart. Further, the Gs for different service sensors can be the same, as long as the key spaces associated with a single worker sensor can be clearly identified; because of this, this protocol scales well to very large sensor networks.

4 Public Key Assisted Secure Channel Establishment

It is clear that the applicability of our PKM protocol depends on the availability of a secure channel between a worker sensor and the corresponding service sensor since the worker sensor needs its private share for pairwise key computation. In this section, we propose a public key assisted key exchange protocol to establish a secret key K_s between a worker sensor and a service sensor.

Our simple idea is based on the following observations. First, pairwise key establishment is a bootstrapping protocol that ensures a newly deployed sensor network to initiate a secure infrastructure. Thus, we can simply assume all worker sensors and service sensors are trust-worthy if they are deployed together. Rekeying and key establishment for future deployment will be studied in future research. Second, there is no *a priori* security information between a worker sensor and a service sensor. For the sensor network under consideration, public-key based key exchange for secret key establishment is the only choice for us. Third, since worker sensors are supposed to operate for an un-deterministically long time while service sensors can die after deployment, cryptographic algorithms that shift large amount of the computational overhead to the service sensors are preferred.

Our public-key assisted secret key exchange is based on the Rabin cryptosystem [14]. Rabin's scheme has asymmetric computational cost. Its encryption operation is several hundred times faster than RSA, but it's decryption time is comparable to that of RSA. The security of Rabin's scheme is based on the factorization of large numbers, thus, it is also comparable to that of RSA.

4.1 Rabin's Scheme

Rabin's scheme is an asymmetric cryptosystem where we need to compute both a public and a private key.

Key Generation: Choose two large distinct primes p and q such that $p \equiv q \equiv 3$ *mod* 4. (p, q) is the private key while $n = p \cdot q$ is the public key.

Encryption: For the encryption, only the public key n is needed. Let p be the plain text that is represented as an integer in Z_n. Then the cipher text $c = p^2$ *mod* n.

Decryption: Since $p \equiv q \equiv 3 \bmod 4$, we have

$$m_p = c^{\frac{p+1}{4}} \bmod p$$

and

$$m_q = c^{\frac{q+1}{4}} \bmod q.$$

By applying the extended Euclidean algorithm, y_p and y_q can be computed such that $y_p \cdot p + y_q \cdot q = 1$.

From the Chinese Remainder Theorem, four square roots $+r, -r, +s, -s$ can be obtained:

$$r = (y_p \cdot p \cdot m_q + y_q \cdot q \cdot m_p) \bmod n \quad (1)$$

$$-r = n - r \quad (2)$$

$$s = (y_p \cdot p \cdot m_q - y_q \cdot q \cdot m_p) \bmod n \quad (3)$$

$$-s = n - s \quad (4)$$

Note that encryption in Rabin's scheme requires only one squaring, which takes less time and energy compared to RSA, which requires multiple squarings and multiplications. Also note that decryption in Rabin's scheme produces three false results in addition to the correct plain text. This can be easily overcome in practice by adding pre specified redundancy to the plain text before encryption.

Furthermore, a careful reader may wonder why we did not choose RSA with a small prime number such as 3 as the public encryption exponent. RSA is preferable since it has gone through extensive cryptanalysis. But unfortunately a small encryption exponent is not secure when the same message needs to be sent out to multiple destinations or the plain text is too short [13]. This may be the case in our application scenario since a request may be short and it may be delivered to multiple service nodes.

4.2 SKE: Secret Key Exchange Protocol

Based on Rabin's scheme described in Subsection 4.1, we propose the following secret key exchange protocol between worker sensors and service sensors.

- For each service sensor, computes two large distinct primes p and q such that $p \equiv q \equiv 3 \ mod \ 4$. $n = p \cdot q$ is the public key and (p, q) is the private key. Broadcasts n to all associated worker sensors. Note that p and q can be computed off-line by a supercomputer for all service nodes.
- For each associated service sensor I, a worker sensor picks K_s, computes $E_n(K_s||R) = K_s^2 \ mod \ n$, where R is a predefined bit pattern to resolve the ambiguity in Rabin's decryption, and transmits $E_n(K_s)$ to I. K_s is the shared key between the worker sensor and service sensor I.
- Upon receiving the $E_n(K_s)$ from a worker sensor, the service sensor computes $D_{(p,q)}(E_n(K_s))$ based on Rabin's decryption algorithm.

Note that this protocol only requires executing the encryption algorithm once by the worker sensor and no decryption is involved. Furthermore, encryption in Rabin's scheme is extremely simple. This can save energy to extend the operational time of worker sensors.

5 Analysis

In this section, we briefly analyze PKM along the lines of connectivity, overhead and security. This study is motivated by [6].

5.1 Connectivity Analysis

The graph formed by all secure links is called a *key-sharing* graph, denoted by G. For a network to function properly, G must be connected. If a sensor establishes shared keys with all neighbors, then the induced graph G is connected if the original topology is connected.

In reality it may not be efficient to require all links to be active all the time, mainly due to the contention and delay caused by the MAC layer. This is especially true in a dense sensor network where each sensor has tens of immediate neighbors. Usually it suffices if a subset of links are secured and all secure links form a connected graph. In other words, a worker sensor establishes secure links with a subset of neighbors and communicates with other neighbors through multi hop transmission. But what is the expected degree of each worker sensor in the key-sharing graph G such that G is connected? Or connected with high probability? Erdős and Rényi's random graph theory can help us to answer this question.

Let d be the expected degree for a worker sensor in G. The connectivity theory of Erdős and Rényi [8] states that G is connected with a probability P_c for a network with N nodes when N is large if

$$d = \frac{N-1}{N}[ln(N) - ln(-ln(Pc))].$$

Actually d should be sufficiently large such that G is connected with high probability. This analysis can provide a guideline when applying our PKM key management protocol to a newly deployed sensor network.

5.2 Overhead Analysis

In this Subsection, we will study the memory usage, and the communication and computation overheads of a worker sensor in our PKM scheme.

The storage budget allocated for shared key establishment in a sensor node impacts the security level of our PKM scheme. Blom's symmetric key computation algorithm is λ-secure. Thus, the larger the λ, the better the security. However, the total memory budget and the number of crypto pairs a worker sensor must store for ensuring global connectivity places constraints on the size of λ.

A worker sensor needs to locate service sensors in the neighborhood, establish secure channels with them and then obtain crypto pairs for shared key computation. Service sensors will broadcast their existence, this way worker sensors do not need to query their neighborhood. It is clear that all these communications are confined locally.

Computing a shared key requires 2λ modular multiplications, with λ of them for restoring all the elements in the public share and the other λ of them for computing the shared key. A worker sensor also needs λ modular addition for a shared key computation.

5.3 Security Analysis

In this subsection, we study the resiliency of PKM to sensor capture through probability analysis. We assume that if a sensor is captured, all pairwise keys shared by this sensor and others will be compromised. Further, we assume all

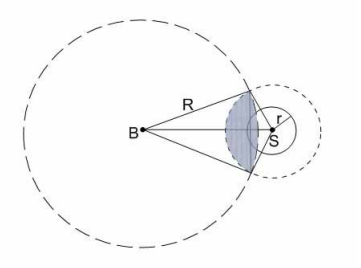


Fig. 1. The overlapping area is colored gray. In this example, $t = 2$. B is the center of the disk all x compromised nodes reside. S is the service sensor whose key space information may be released by compromised nodes

sensors are distributed randomly and uniformly in a two-dimensional area A. Note that we only consider the simple scenario when a service sensor provides security information to all worker sensors within t hops away and a worker sensor asks for a crypto pair from each service sensor within t hops away. Let r be the transmission range.

Assume that x number of nodes have been captured. Since Blom's scheme is λ-secure, we assume $x > \lambda$. There exists two scenarios we need to consider:

- **Case A:** The x compromised nodes are independently and randomly distributed in the whole area A;
- **Case B:** The x compromised nodes are independently and randomly distributed in a small area B.

We denote the areas of A and B by A and B, respectively. Let P be the probability that any key space is compromised. In other words, P is the probability that more than λ number of compromised sensors have obtained information from a service sensor S carrying the key space. Let p be the probability that each compromised node carries information about S. We have:

$$P = \sum_{j=\lambda+1}^{x} \binom{x}{j} p^j (1-p)^{x-j}. \tag{5}$$

Next, we will study p for each case:

Case A: Since S provides information to all worker sensors within t hops away, it is possible that a sensor within the disk area of radius $t\ r$ centered at S contains information about S. Thus $p \leq \frac{\pi(tr)^2}{A}$.

Case B: For simplicity, we assume area B is a circle with radius $R > 2\ t\ r$ centered at location B. Let y be the Euclidean distance between S and B. When $y > R + t\ r$, no information on S is released by the x captured nodes. Therefore $p = 0$. Otherwise, $p \leq \frac{\theta}{B}$, where θ is the overlapping area, as shown by the gray area in Fig. 1.

6 Conclusion

In this paper, we have proposed and analyzed "PKM", an in-situ key management protocol for sensor networks. PKM is based on the idea of sacrificing a number of service sensors so a large amount of computational and storage overhead can be shifted away from worker sensors. Since worker sensors require no information to be pre-loaded, PKM scales well to large sensor networks. As a future research, we will analyze the performance of PKM with more general assumptions.

References

1. http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/6020-0042-06_B_MICA2.pdf.
2. R. Blom, An optimal class of symmetric key generation systems, *Advances in Cryptology: Proceedings of EUROCRYPT 84 (Thomas Beth, Norbert Cot, and Ingemar Ingemarsson, eds.), Lecture Notes in Computer Science*, Springer-Verlag, vol. 208, pp.335-338, 1985.
3. C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, Prefectly-Secure Key Distribution for Dynamic Conferences, *Advances in Cryptoglogy - CRYPTO'92, LNCS 740*, pp. 471-486, 1993.
4. D. W. Carman, P. S. Kruus, and B. J. Matt, Constraints and Approaches for Distributed Sensor Network Security, NAI Labs Technical Report No. 00-010, September, 2000.
5. H. Chan, A. Perrig, and D. Song, Random Key Predistribution Schemes for Sensor Networks, *IEEE SP 2003*.
6. W. Du, J. Deng, Y.S. Han, and P.K. Varshney, A pairwise key pre-distribution scheme for wireless sensor networks, *CCS'03*, pp. 42-51, October 27-31, 2003, Washington DC, USA.
7. W. Du, J. Deng, Y.S. Han, S. Chen, and P.K. Varshney, A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge, *IEEE INFOCOM 2004*.
8. Erdős and Rényi, On Random Graphs I, *Publ. Math. Debrecen*, 6:290-297, 1959.
9. L. Eschenauer and V.D. Gligor, A Key-Management Scheme for Distributed Sensor Networks, *CCS'02*, pp.41-47, November 18-22, 2002, Washington DC, USA.
10. S. C.-H Huang, M.X. Cheng, and D.-Z. Du, GeoSENS: Geo-based SEnsor Network Secure Communication Protocol, manuscript, 2004.
11. D. Liu and P. Ning, Location-Based Pairwise Key Establishments for Static Sensor Networks, *Proc. 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 72-82, 2003.
12. D. Liu and P. Ning, Establishing Pairwise Keys in Distributed Sensor Networks, *ACM CCS'03*, pp. 52-60, 2003.
13. A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, Handbook of Applied Cryptography, CRC Press, 1997.
14. M.O. Rabin, Digitalized signatures and public key functions as intractable as factorization, MIT/LCS/TR-212, MIT, 1979.

Secure Group Instant Messaging Using Cryptographic Primitives*

Amandeep Thukral and Xukai Zou

Purdue University School of Science at Indianapolis, Indianapolis, IN 46202, USA
{athukral,xkzou}@cs.iupui.edu

Abstract. Instant Messaging (IM) services over the last few years have evolved from a casual communication tool to a formal business communication tool. Security requirements change drastically when instant messaging systems are employed in the corporate world as sensitive data needs to be transmitted. Many security enhancements have been proposed for instant messaging from the perspective of peer-to-peer talk. In this paper, we study the extension of secure instant messaging for group settings. We also propose a scheme, IBECRT, which uses ID-based encryption and the Chinese Remainder Theorem. The scheme does not require a central trusted entity for key distribution and achieves uniform work-load distribution. Additionally, the protocol has the following properties: hiding the users' identity in a conference, authentication of senders, and integrity protection of the messages exchanged.

1 Introduction

Instant Messaging (IM) or collaborative software refers to a type of communication which enables users to communicate in real time over the Internet. Traditionally IM was used as a popular means of communication amongst users for casual purposes. Recently its use has been extended to group communication and secure conferencing among multiple users. IM may soon be an indispensable communication tool for business purposes at work places [1]. It offers advantages of both telephone and email communications.

According to a survey by Osterman Research, one or the other IM service was being used by 90% of the enterprises surveyed [2]. Research firm IDC estimates that currently over 65 million people worldwide use at least one of the major consumer instant messaging services (like AOL, Yahoo, MSN or ICQ) at work and this number would increase up to 260 million by the year 2006.

Most of these existing IM services were designed giving scalability priority over security. Resig et.al [3] have proposed a framework for mining useful information from instant messaging services. Although the paper focuses on the use of this method as a means of counter-terrorism, similar methods can be used for mining information in business communications using IM, thus exposing possible confidential information. Extending the current IM services for secure business communication would require strong security capabilities to be incorporated, such that the confidentiality of the information is preserved.

* This work was partially supported by the U.S. NSF grant CCR-0311577.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 1002–1011, 2005.

A lot of work has been done in providing security to existing instant messaging services. AOL released AIM 5.2 build 3139, which provided message encryption with the aim of providing confidentiality to user communication. Some commercial secure IM implementations have also been developed [4,5]. Jabber is an XML protocol developed by Jabber Software Foundation, an open-source community. Like most other implementations, it is based on a client-server architecture and employs SSL encryption. Since SSL was designed to provide confidentiality for applications by establishing a secure connection between a client and a server, extending it to IM would involve multiple SSL connections between a centralized server and the users. Using the SSL approach for the client-server architecture also raises privacy concerns as the server is involved in the communication process [2].

Most of the above mentioned secure instant messaging services have been developed with a view of peer-to-peer talk. Instant messaging services have also provided extensions for group communication. However, as we know, little work has been done to extend the concept of secure instant messaging for group communication. In this paper, we study the extension of secure instant messaging for group settings. A contributory key management protocol, IBECRT, for secure group instant messaging is also proposed. Our proposed protocol uses Identity based encryption(IBE) to provide user authentication and uses the Chinese Remainder Theorem (CRT) to provide user anonymity for group communication.

The system consists of a central server that performs the registration and key initiation tasks. However, the server is not involved in the key agreement process for either a peer-to-peer talk or a secure conference. Thus, the system is able to preserve confidentiality against the central server. The proposed key management scheme is contributory in nature and hence does not put the burden of key generation and distribution on a single entity. Group rekeying after a member leave is often a difficult problem. The proposed protocol does the rekeying efficiently as it requires only one round of broadcast to obtain the new key when a member leaves a secure conference. The new protocol also hides the identities of the members of a secure conference from outside users. This anonymity is especially very important to improve security of the instant messaging system.

The rest of the paper is organized as follows. The desired features of any protocol for secure group instant messaging are discussed in section 2. A review of related work and existing IM systems is presented in section 3. The proposed protocol is described in section 4 and a comparison with other protocols is presented in section 5. Section 6 presents our conclusions.

2 Desired Features for Secure Group Instant Messaging

The problem of extending instant messaging for group settings is unique and quite challenging. We observe that a lot of issues pertaining to secure group communication (SGC), like forward and backward secrecy [6], are also applicable to group instant messaging. SGC refers to 'secure communication amongst a group of members (two or more) in a way that anyone who is not a member of

the group is unable to glean any information'. Secure group communication has been a critical area and has inspired a lot of research [6,7,8]. Instant messaging, however, puts extra constraints on the requirements and thus it is necessary to identify the desired properties.

1. Efficiency- Any scheme for group instant messaging must be efficient in terms of computation and communication. Often tensions permeate in IM due to collision of conventions between verbal and written communication [9]. As such, the response time of the system in terms of the time to encrypt and decrypt the messages must be as low as possible and the messages must be delivered in a timely fashion.
2. Confidentiality- To messages should be integrity protected not only against an intruder but also against the server to achieve the desired security requirements.
3. Equal distribution of work load- As a result of the stringent timeliness requirement in IM, the desired system should equally distribute the work load of key management amongst all users. A contributory key agreement protocol is one where the secret group key for communication is established as a function of some information provided by all the participants in the group. A contributory key agreement protocol helps in equally distributing the computation and communication load amongst all the members of the group and such a scheme is most appropriate for small sized groups [10]. These features are highly desirable in group instant messaging.
4. Remote Management of User Profiles- A user's contact list (friend list) and preferences should be maintained by a central server so as to facilitate easy and secure use of the IM system from any computer where the client is installed.

3 Related Work

3.1 Existing IM Systems

Secure instant messaging is a relatively new field of research. Kikuchi et.al have proposed an Instant Messaging protocol [2] based on a modified Diffie Hellman scheme for key management. The proposed protocol provides confidentiality against a malicious administrator and allows for group instant messaging amongst multiple users. Being non-contributory in nature, the load of key management falls on the initiator of the group chat, who is responsible for the generation and distribution of the shared session key for chat. Also, if the initiator leaves but the rest of the members still want to continue the group conversation, the entire group needs to be reconstructed from scratch by the remaining members.

3.2 ID-Based Cryptography

Identity Based Encryption (IBE) schemes were first proposed by Adi Shamir in 1984 [11]. In the paper, Shamir presented a new model for asymmetric cryptography which aimed at removing the need of the bulky Public Key Infrastructure

by using a characteristic (like an email address) that uniquely identifies a user as its public key.

A lot of schemes were proposed thereafter, but almost all of them were computationally so expensive that they did not offer any significant advantage over the existing Public Key Infrastructure. The first practical scheme for IBE based on Weil Pairing was proposed by Boneh and Franklin [12].

The initial idea behind development of IBE was to develop a public key encryption mechanism without the use of a complicated PKI. But since public key cryptosystems are several orders of magnitude slower than secret-key cryptosystems, the extension of IBE to secret key cryptosystems was natural.

3.3 Key Agreement Protocols

Various protocols for key agreement for secure group communication have been proposed in literature [13] [14]. Anton et al. in [13] have discussed a number of contributory key agreement protocols for ad-hoc networks. Group Diffie Hellman (GDH) protocol suite have been studied in [14]. The GDH suite has efficient protocols for group communication, but they all have the drawback that they require member serialization i.e. the members must be serialized or structured in some way so as to achieve the common group key. Also, the last member acts as the Group Controller and thus has to perform extra computation.

4 IBECRT: A Key Agreement Protocol for Group Instant Messaging

In this section we discuss the details of our proposed key agreement scheme for group instant messaging. The protocol uses the concepts of ID-based encryption(IBE) and the Chinese Remainder Theorem (CRT) in order to achieve group key agreement in a way that hides the identities of the members of the group. The framework includes a trusted central server which has a public key P_s. Before we describe the details of the protocol, some important definitions pertaining to the discussion are mentioned below.

4.1 Primary Definitions

- *username*- Each user in the system is identified with a username that it selects at the time of registration.
- *ID*- The ID is also unique to every user in the system and is the MD5 hash value of the username.
- *Co-prime set*- The central server maintains a set $S = \{N_1, N_2, ..., N_m, N_0\}$ of co-primes, where N_1 to N_m correspond to the users in the system and N_0 is a global prime value which is used for authentication purposes. Denote $N = \max\{N_1, N_2, ..., N_m, N_0\}$.
- *Contact list (CL)*- The central server also maintains a contact list (a list of friends) for each user. The list includes the IDs, the coresponding co-prime values and the current status (*online, offline or away*) for the users in the list.

The discussion of the proposed protocol is divided into four phases: registration and key initiation, conference key management, member join and leave.

4.2 Registration and Key Initiation

Registration. Each new user in the system registers with the central server by chosing a username and a password. Once selcted, the server generates the ID corresponding to the username, which serves as the public key for the user. The central server also generates the corresponding private decryption key and adds a co-prime element in the co-prime set for the user. Hence, each user U_i has a public share $\{ID_i, N_i\}$.

Signing In. This is where the user signs in and becomes *online*. The process is initiated by the user, who signs in using the client application. The client encrypts the ID and the password of the user with P_s and sends it to server. The server authenticates the user based on the password. On successful authentication, the server sends back a message consisting of the decryption key corresponding to the user's ID and its contact list. The message is encrypted using a key k derived from the hash of the user's ID and *password*, i.e. $k = h(ID_i, password)$.

The messages exchanged can be depicted by the figure below.

Signing In:

User	$\xrightarrow{E_{P_s}(ID_i, password)}$	*Server*
User	$\xleftarrow{E_k(ID_i^{-1}, CL)}$	*Server*

Fig. 1. Messages exchanged in the sign in process

4.3 Conference Key Management

We now discuss the steps that are to be executed by the users in order to achieve a shared secret key for a secure conference. The term session refers to one such conference and a user can participate in multiple such conferences. Each conference has a conference key which is computed from the share contributed by the members. Let us assume a conference consisting of $\{U_1, U_2, ..., U_m\}$. The discussion has been split into two parts: initiaion phase and key agreement phase.

- Initiation Phase. The initiation phase is like a setup process for a secure conference, where the initiator invites users in his list to join a secure conference. The invitation messages are integrity protected using a random s_0 selected by the initiator. Also a list consisting of the IDs of the users is also sent to the members. The process is explained in the following steps.
  - Step 1. The initiator of the conference, say U_1, selects a random $s_0 \in \mathbb{Z}_N$. It then encrypts s_0 under the IDs of the users of the conference, as shown.

$$R_{2,0} = E_{ID_2}(s_0)$$
$$R_{3,0} = E_{ID_3}(s_0)$$
$$.$$
$$.$$
$$R_{m,0} = E_{ID_m}(s_0)$$

- Step 2. Next U_1 solves the following set of congruences to obtain the CRT value X, for all the users in the conference.

$$X = R_{2,0} \bmod N_2$$
$$X = R_{3,0} \bmod N_3$$
$$.$$
$$.$$
$$X = R_{m,0} \bmod N_m$$
$$X = E_{s_0}(s_0) \bmod N_0$$

  To hide the identities of the conference members from outsiders, U_1 encrypts the list of members under s_0, as shown.

$$L = E_{s_0}(ID_1, ID_2, ..., ID_m)$$

  It then sends X and L to the conference members.
- Step 3. The users receive the CRT value X and the list L. Each user U_i obtains $R_{i,0}$ by solving X mod N_i. It obtains s_0 by decrypting $R_{i,0}$ using its decryption key ID_i^{-1}.
  To verify that the user is in fact intended to be a member of the conference, it performs the following two computations.

$$t = X \bmod N_0 \qquad t' = E_{s_0}(s_0) \bmod N_0$$

  If $t = t'$, it is implied that X was not modified during the transmission and the user is a valid member in the conference. This step also ensures that the conference initiation messages are integrity protected.
  The legitimate users then decrypt L using s_0 and obtain the list of the users in the conference. The users in the conference then carry out the following step to obtain the common conference key.

– Key Agreement Phase. Once the initiation task is over, the following steps are carried out by all the members to achieve a shared conference key.

- Step 4. Each conference member U_i selects a random $s_i \in \mathbb{Z}_N$, which will be its share of the conference key. It then encrypts s_i under the IDs of all the members of the conference.

$$R_{j,i} = E_{ID_j}(s_i)$$

  for $j = 1, 2, ..., i-1, i+1, ..., m$.
  For example, user U_2 computes the follwing.

$$R_{1,2} = E_{ID_1}(s_2)$$
$$R_{3,2} = E_{ID_3}(s_2)$$
$$.$$
$$.$$
$$R_{m,2} = E_{ID_m}(s_2)$$

- Step 5. Each user U_i then solves the following congruences to obtain the CRT value X_i.

$$X_i = R_{j,i} \bmod N_j$$
$$X_i = E_{s_i}(s_i) \bmod N_0$$

Again taking U_2 as an example, it solves the following congruences.

$$X_2 = R_{1,2} \bmod N_1$$
$$X_2 = R_{3,2} \bmod N_3$$
$$.$$
$$.$$
$$X_2 = R_{m,2} \bmod N_m$$
$$X_2 = E_{s_2}(s_2) \bmod N_0$$

It then sends X_2 to the conference members.

- Step 6. On receiving the CRT values from all other members in the conference, each user U_i obtains the share from all other members through the following computations.

$$R_{i,j} = X_j \bmod N_i \qquad s_j = D_{ID_i^{-1}}(R_{i,j})$$

As an example, let us assume user U_1 obtains the CRT value X_2 from U_2 which we computed before. It can then obtain the secret share s_2 as shown below.

$$R_{1,2} = X_2 \bmod N_1 \qquad s_2 = D_{ID_1^{-1}}(R_{1,2})$$

Similarly it obtains the secret shares from all other members of the conference. Also, as in the key initiation phase, the messages are integrity protected through the use of N_0 while solving the congruences. U_i can verify that the secret shares were not modified during transmission as shown below.

$$t = X_j \bmod N_0 \qquad t' = E_{s_j}(s_j)$$

U_i compares t and t' to make sure that $t = t'$ i.e. the secret has not been modified during transmission.

- Step 7. Each user can then compute the conference key as follows.

$$CK = h(s_1||s_2||...||s_m)$$

As we can see, using the Chinese Remainder Theorem, the users are able to achieve a common conference key and at the same time hide the identities of the members involved in the conference. Also since the key is obtained with contribution from all individual members, even if the initiator leaves the conference, the remaining members can generate a new conference key to communicate.

4.4 Member Join

We now explain the steps needed to be carried out when a user is to become a member in an existing conference. Let us assume that there exists a conference initiated by U_1 consisting of the users $\{U_1, U_2, U_3, U_4\}$, and another user, say U_5, wishes to join the conference. Admission control policies could be used so

as to decide upon whether or not U_5 should be allowed to be a member or not depending upon the application of the IM system.

Assuming that user U_5 has been allowed to become the member of the conference, the members perform the following operations to obtain the new key.

- Initiator U_1 updates the member list L by adding ID_5 in it. It sends the updated list and the hash of the current key, $h(CK)$ to U_5.
- U_5 then carries out Step 4 and Step 5 of the Key Agreement Phase and sends its secret s_5 across the conference members. Existing members obtain s_5 in a similar way as they obtained other secret shares before and can then compute the new conference key as follows.

$$CK_{new} = h(h(CK)||s_5)$$

4.5 Member Leave

In case a member leaves an existing conference, the key must be changed such that the leaving member is not able to decrypt any further messages. Taking the example mentioned above, let us assume that the member U_3 decides to leave the conference. The following operations would need to be performed in order to recompute the new conference key.

- Step1. Each user U_i selects a new random secret $s_i' \in \mathbb{Z}_N$, and solves the CRT in the same way as done before, with the exception that N_3 is left out of the computation. It then sends s_i' across the members.
- Step2. The new conference key is computed by the remaining members using members by taking a hash of the new secret shares of the individual members.

Note that Step 1 could have been performed by just one remaining member of the conference and the new key could have been computed in a way similar to member join. But since we argue that the computation and communication load must be equally balanced amongst the users, all members perform the above operations to achieve the new key. Also since U_3 was left out of the computation, even if it can obtain the broadcast message, it cannot obtain the new conference key.

5 Discussion

The protocol described in the previous section meets the requirements specified in section 2. The protocol uses the concept of ID-Based encryption schemes together with the Chinese Remainder Theorem to achieve a contributory key agreement scheme. Every member of a secure conference contributes towards the shared conference key. The shared key is achieved in a way that any user not included in a conference cannot know about the members in the conference. This anonymity is especially important to improve security. By limiting the knowledge about the existing members in a secure conference, a malicious user is unable to identify the specific members and attack them directly.

The proposed protocol does not require the members to be serialized for proper execution. Every user in a secure conference is treated equally and performs the same amount of work to achieve the common conference key. By encrypting the secret share with itself prior to broadcasting, the members of a conference can verify if the CRT value was changed during transmission.

The proposed scheme uses the ID-Based encryption protocol proposed by Boneh and Franklin [12], which is based on pairings over elliptic curves. The protocol has proved to have chosen ciphertext security in the random oracle model. The Chinese Remainder Theorem, when applied in a similar way for access control scheme by Zou et al. [15], was found vulnerable to a class of attacks called *GCD based attacks* [16]. Our proposed scheme defeats these attacks by encrypting the coefficients of the congruences in the CRT using a public key cryptosystem under the IDs of the members. Thus IBECRT scheme is secure because of the difficulty of partitioning the product into specific factors and in the specific order along with the security of the underlying cryptosystems.

As for the performance of IBECRT scheme, we consider three complexities: space, time and communication complexity. The space complexity accounts for the space required by each user to represent and store ID_i, s_i, X_i and N_i, which require large integers. The time complexity accounts for the complexity of the CRT algorithm, but ignores the time consumed on key generation, encryption and decryption, which will depend on the specific algorithm selected. The communication complexity represents the key material, including the CRT parameters, that are exchanged amongst the members of a conference. The complexities are summarized in the following table.

Table 1. Complexity calculations for IBECRT

Criteria	Complexity
$Space$	$O(m^2l)$
$Time$	$O(M(ml)log(m)) + O(mM(l)log(l))$
$Communication$	$O(ml)$

Note: m: the number of users in a secure conference; l: the length of a large integer in bits; $M(n)$: the time to multiply two n-bit integers in bit operations; $O(n)$ is measured in bits and not in bytes.

6 Conclusions

In this paper, we studied the issues relating to extension of instant messaging for group settings. We have proposed a contributory key agreement protocol based on the Chinese Remainder Theorem and ID-based encryption. The proposed scheme has highly desirable properties such as distributed key agreement, mutual authentication and conference anonymity.

Acknowledgements

We thank Mr. Sriram Srinivasan for the useful discussions at the initial stages of the work.

References

1. Security, S.E.: Secure Instant Messaging. White Paper on Secure Instant Messaging (2002)
2. Kikuchi, H., Tada, M., Nakanishi, S.: Secure instant messaging protocol preserving confidentiality against administrator. Advanced Information Networking and Applications (2004) 27–30
3. Teredesai, A., Resig, J.: A Framework for mining instant messaging services. Workshop on Link Analysis, Counter-terrorism, and Privacy at Fourth SIAM Conference, April 2004 (2004)
4. : Ipswitch Instant Messaging Guide. http://www.ipswitch.com/support/ICS/index.asp (2003)
5. : Jabber Inc. Enterprise Instant Messaging: Essential Infrastructure. http://www.jabber.com/index.cgi?CONTENT_ID=55 (2003)
6. Zou, X., Ramamurthy, B., Magliveras, S.S., eds.: Secure Group Communications over Data Networks. ISBN: 0-387-22970-1, Springer, New York, NY (2004)
7. Burmester, M., Desmedt, Y.: A secure and efficient conference key distribution system. EUROCRYPT'94, LNCS, Springer, Berlin **950** (1995) 275–286
8. Banerjee, S., Bhattacharjee, B.: Scalable secure group communication over IP multicast. IEEE Journal on Selected Areas in Communications **20** (2002) 1151–1527
9. Voida, A., Newstetter, W., Mynatt, E.: When conventions collide: The tensions of instant messaging attributed. CHI 2002 (2002) 187–194
10. Amir, Y., Kim, Y., Rotaru, C., Schultz, J., Stanon, J., Tsudik, G.: Scalable multicast key distribution. IEEE Transactions on Parallel Computers (2004) 468–480
11. Shamir, A.: Identity Based Cryptosystems and Signature Schemes. Advances in Cryptology - Crypto' 84, Lecture Notes in Computer Science 0196, Springer (1984)
12. Boneh, D., Franklin, M.K.: Identity-Based Encryption from the Weil Pairing. Lecture Notes In Computer Science, Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology (2001)
13. Anton, E., Duarte, O.: Group key establishment in wireless ad hoc networks. Workshop on Quality of Service and Mobility (2002)
14. Steiner, M., Tsudik, G., Waidner, M.: Diffie-Hellman key distribution extended to group communication. ACM Conference on Computer and Communications Security (ACM CCS 1996) (1996) 31–37
15. Zou, X., Ramamurthy, B., Magliveras, S.: Chinese remainder theorem based hierarchical access control for secure group communications. Lecture Notes in Computer Science (LNCS), Springer-Verlag (International Conference on Information and Communication Security) **2229** (2001) 381–385
16. Geiselmann, W., Steinwandt, R.: Attacks on a secure group communication scheme with hierarchical access control. Submitted to International Conference on Information Security and Cryptography, Seoul, Korea (2003)

A Privacy Enhanced Role-Based Access Control Model for Enterprises

Cungang Yang[1] and Chang N. Zhang[2]

[1] Department of Electrical and Computer Engineering, Ryerson University,
Toronto, Ontario, M5B 2K3
cungang@ee.ryerson.ca
[2] Department of Computer Science, University of Regina,
Regina, Saskatchewan, S4S 0A2
zhang@cs.uregina.ca

Abstract. The Role-based access control (RBAC) is a super set of mandatory access control (MAC) and discretionary access control (DAC). Since MAC and DAC are useful in information flow control that protects privacy within an application, it is certainly that we can use RBAC for privacy concerns. The key benefits of the fundamental RBAC are simplified systems administration and enhanced systems security and integrity. However, it does not consider privacy protection and support controlling method invocation through argument sensitivity. In this paper, a privacy-enhanced role-based access control (PERBAC) model is proposed. Privacy related components, such as purpose, purpose hierarchy, are added to the new model. Also, an information flow analysis technique and a privacy checking algorithm are introduced to support controlling method invocation through argument sensitivity.

1 Introduction

Privacy protection is essential for an application that manages sensitive data in an enterprise. The privacy protection can be achieved by information flow control models. The first developed model is called Mandatory Access Control (MAC) [1, 2, 3]. The principles of MAC is that the security levels of objects and users are classified according to the "no read up" and "no write down" rules. However, controlling method invocation through argument sensitivity was not considered in the MAC model. The second developed model is called discretionary access control (DAC). DAC is typically implemented through some form of an access control lists (ACL). Samarati [10] uses access control lists (ACLs) of objects to compute ACLs of executions. Interactions among executions are classified into 5 modes and different modes lead to different security policies. Ferrari [5] proposed a more flexible method by allowing exceptions during or after method execution. However, the drawbacks of ACLs are that it cannot be changed according to newly added objects during runtime. Also, controlling method invocation through argument sensitivity is not considered.

Izaki presented a model [6] that uses Role-based Access Control [4, 7, 11, 12] to control information flows. The model classifies object methods and derives a flow

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 1012 – 1021, 2005.

graph from method invocations. From the flow graph, nonsecure information flows can be identified. The disadvantages of the model are that (1) it does not support controlling method invocation through argument sensitivity. (2) It does not include the important component, purpose, for privacy considerations.

In this paper, a Privacy-enhanced Role-based Access Control model is proposed. Privacy related components, such as purpose, purpose hierarchy, are added to the new model. Also, an information flow analysis technique and a privacy checking algorithm are introduced. The significances of the research work are that (1) PERBAC extends the fundamental RBAC model to support privacy protection. (2) By using the information flow analysis technique, the proposed method supports the controlling method invocation through argument sensitivity. (3) Whenever a new object of the PERBAC model (for instance, task role, privilege, purpose, or object) is added, the privacy checking algorithm is invoked to check if the modification violates the privacy principle which will be explained in section 3.

This paper is structured as follows. Section 2 gives an introduction of the PERBAC model. Section 3 illustrates privacy disclosure problem of the model, characterizes the information flow analysis technique on the model and proposes a privacy checking algorithm. Section 4 presents the conclusion.

2 The Privacy Enhanced Role-Based Access Control Model

The class diagram of the PERBAC model represented using UML (United Modeling Language) is shown in Fig. 1. In the proposed model, a *user* is a human being, a *role* is a job function or job title and a *privilege* (Object + Methods) is an object method that can be exercised on an object to carry out a particular task. Roles are divided into two groups: *position roles* and *task roles*. A position role is a collection of tasks performed by a certain job position in an enterprise, such as sales manager, sales clerk, vice president of the sales department, etc. Acording to its responsibility and authority, a position role such as sales manager may carry out tasks (task roles) like "approve order" or "grant loan extension". Position roles are organized as role hierarchy in a partial order $\geq$, so that if $x \geq y$ then position role x inherits all task roles of position role y. In PERBAC model, *position role constraint* represents the user-position role authorization. An example of position role constraint could be a position role may have a limited number of users.

A new component introduced in the PERBAC model is *purpose*. Purpose tells the customer how the collected data from the customer will be used. For example, the privacy statement "we use customer contact information (from the registration form) to send you information about our company and/or to give you updates on products and/or services" [8] defines two purposes for which customer contact information will be used: (1) send company information, and (2) give updates on products and/or services.

There is a hierarchy structure for purposes in many applications. Purpose hierarchy is used to map higher-level purpose to lower-level purposes. If a task is allowed for a higher-level purpose, it is also allowed for all its lower-level purposes.

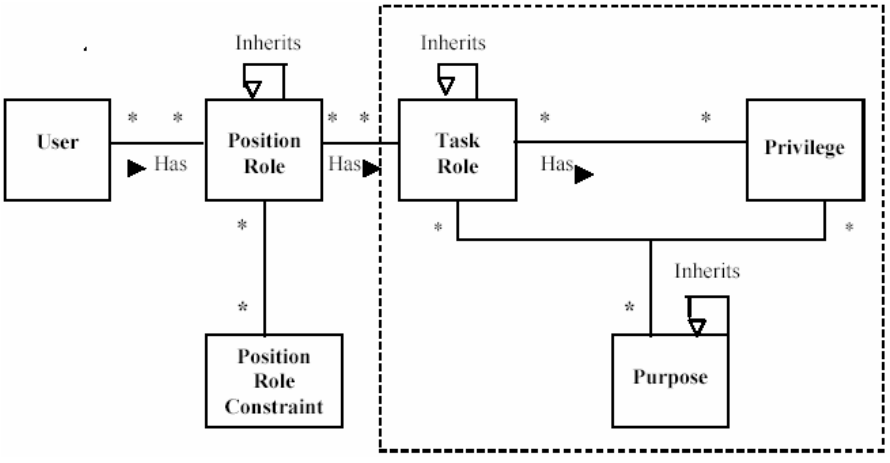


Fig. 1. Class Diagram of the Privacy-Enhanced Role-based Access Control Model

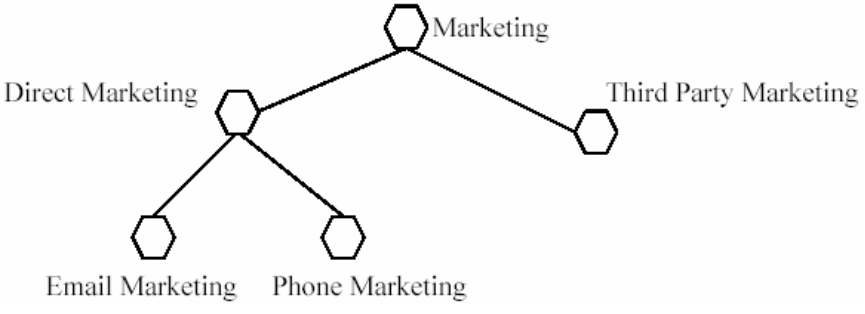


Fig. 2. Hierarchy of Purposes

An example of purpose hierarchy is shown in Fig. 2 where the purposes of direct marketing and third-party marketing are specializations of the marketing purpose. A user assigned to purpose direct marketing (or third-party marketing) will inherit privileges assigned to the more general purpose of marketing. For example, email marketing, and phone marketing are lower-level purposes of direct marketing.

In the PERBAC model, *task role* is introduced to serve as an intermediary entity between position roles and privileges. Task roles are organized as a task hierarchy where higher level task roles inherits the privileges of its lower level task roles. A position role invokes one or more task roles in order to perform some tasks. Each task role has one or more purposes and each purpose corresponds to one or multiple task roles as well. Similarly, each privilege can be accessed according to one or multiple purposes and each purpose also correspond with one or multiple privileges. For the proposed model, the privacy-related components (task hierarchy, privileges and purpose hierarchy) have been enclosed by dashed line in Figure 1.

3 Information Flow Analysis on PERBAC Model

Based on the class diagram shown in Figure 1, an example of object diagram of PERBAC model that emphasized on the interrelationships among task hierarchy, purpose hierarchy and privilege is shown in Figure 3. In Figure 3, objects and their object methods are defined as the privileges and object methods are split into two different categories: basic object methods and complex object methods. If an object method is a function of individual object and does not invoke methods of other objects, it is called as *basic object method,* for instance, o_{11} is a basic object method of object O_1 and o_{21} is a basic object method of object O_2; If an object method is not only a function of an object, but also invokes object methods of other objects, it is called as *complex object method,* for instance, o_{12} is a complex object method of object O_1 and o_{22} is a complex object method of object O_2.

In the object diagram, we need to examine read and write behaviors of the basic object methods and the complex object methods to analyze the information flow [9]. Each basic object method or complex object method can be represented by a list of (object, basic privilege) pairs, where the basic privilege is defined as either read (r) or write (w). In Figure 3, the basic privileges for basic object method o_{11} can be represented as $\{(O_1, r)\}$ and the basic privileges for complex object method o_{12} can be represented as $\{(O_1, r), (O_2, r), (O_3, w)\}$. Task roles are carried out by directly invoking basic object methods or complex object methods on a single object or multiple objects. The basic object methods that invoked by task role m_4 are o_{31} and o_{42}, and complex object methods that invoked by task role m_3 is o_{12}. o_{12} directly invokes basic object method o_{11} and complex object method o_{22}, thus indirectly invokes the basic object methods of o_{21} and o_{33}.

In PERBAC model, the purposes are divided into a number of different categories and create a purpose hierarchy. Each category is assigned a label or purpose level $\lambda(p)$ that signifies the level of the indicated purpose p. In addition, each task role has its own purpose, the purpose level of task s, $\lambda(s)$, is defined to represent the purpose of the task s in a task hierarchy. Moreover, according the privacy policy of an enterprise, objects in the PERBAC model could be accessed for certain purposes, the purpose level of object o, λ (o), is also defined to indicate the purpose level of the object.

The assignments of the purpose level for task roles and objects should meet the following privacy principle.

Privacy principle: If task role s has read or write privilege of the object o, then the purpose level of task role s is greater than or equal to the purpose level of object o, $\lambda(s) \geq \lambda(o)$.

Suppose we designed a purpose hierarchy in Figure 3 that is comprised of purposes of P1, P2, P3 and P4. Assume purpose level of P1, P2, P3 and P4 are M1, M2, M3, M4 and M1>M2, M1>M3, M2>M4 and M3>M4. The purpose level assignments in Figure 3 satisfy the privacy principle because $\lambda(m_1)= M_1$, $\lambda(m_3)=M_2$, $\lambda(m_4)=M_4$, $\lambda(O_1)= M_2$, $\lambda(O_3)=\lambda(O_4)= M_4$, and $\lambda(m_1)>= \lambda(O_1)$, $\lambda(m_1)>= \lambda(O_2)$, $\lambda(m_1)>= \lambda(O_3)$, $\lambda(m_3)>= \lambda(O_1)$, $\lambda(m_4)>= \lambda(O_3)$, $\lambda(m_4)>= \lambda(O_4)$.

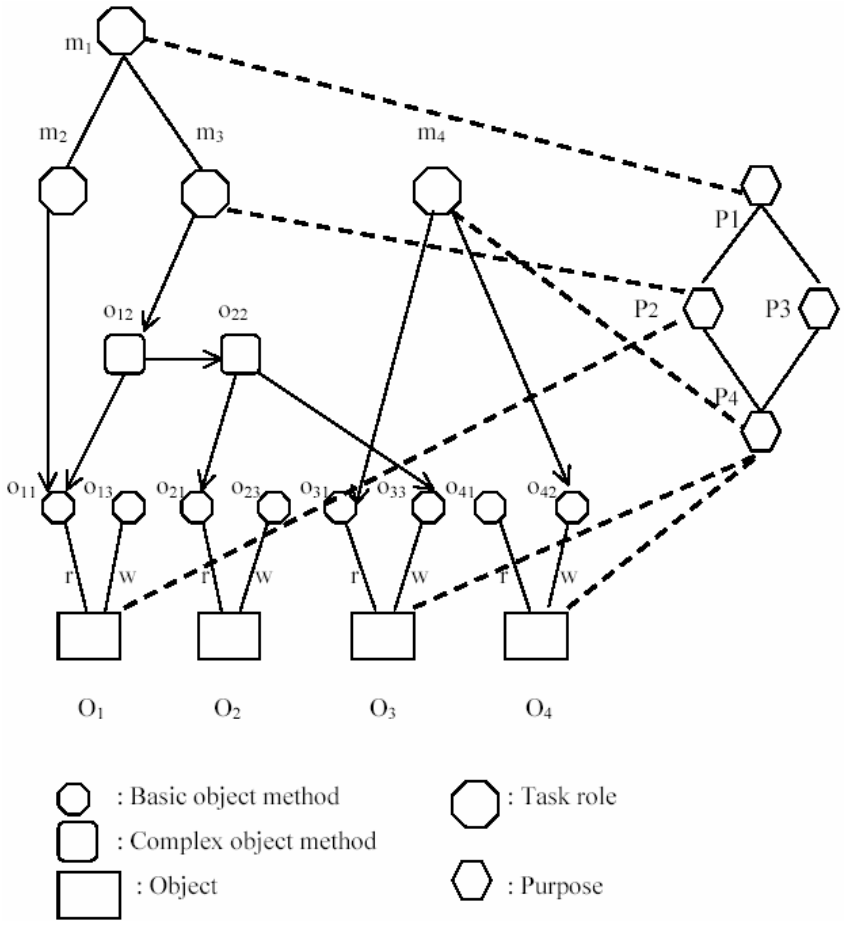


Fig. 3. Task Hierarchy, Purpose Hierarchy and Objects in PERBAC Model

3.1 Privacy Disclosure Problem in PERBAC Model

For a single task role, the privacy concerns could be solved if the privacy principle is satisfied. However, there exist multiple task roles working together in PERBAC systems and privacy disclosure problem may occur. For instance, in Figure 4, suppose the purpose level assignment is the same as the example in Figure 3. Task role m_3 can read on object O_1, O_2 and write on object O_3, task role m_4 can read on object O_3 and write on object O_4. In this case, it is possible that task role m_3 is able to read information from object O_1 and O_2, then write the information to object O_3. After that, task role m_4 can read the information from object O_3 and indirectly obtains information in object O_1 and O_2. Since the purpose level of task role m_4, M4, is less than the purpose level of the object O_1, M1. Thus, the privacy principle is violated. That is, there is the privacy disclose problem in PERBAC systems.

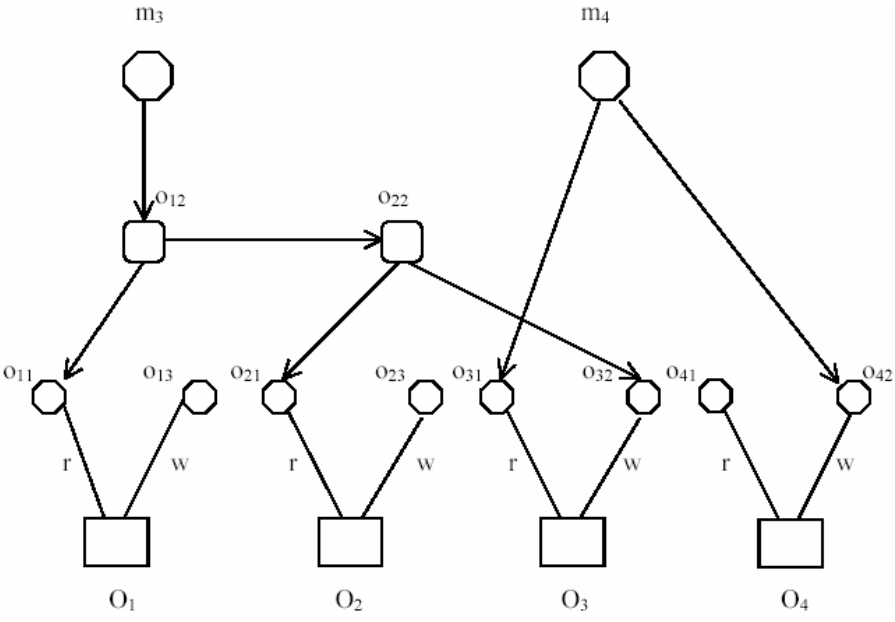


Fig. 4. An Example of Privacy Disclosure Problem on PERBAC Model

In order to solve the privacy disclose problem, in the following sections, an information flow analysis technique is presented. Information flow analysis has been applied to the determination of the security of information in a given system [9, 13]. By analysing the flow of information on PERBAC occasioned by the execution of source operations, it is possible to determine whether such a flow violates the privacy principle. Formally, information flow can be defined as follows: there exists an information flow from object O_i to object O_j ($i \neq j$) if and only if the information is read from object O_i, and written to object O_j. Note that if there is an information flow from object O_i to object O_j, the information written to an object O_j could be the same as the information read from O_i, also, the information written could be different, there still exists information flow from object O_i to object O_j when the information written to object O_j is gotten by executing some computation on the information read from O_i.

3.2 Privacy Checking Algorithm

The information flow analysis is based on task roles. A task role is split into two different groups: (1) A task role that has one or more than one basic object methods on a single object, (2) A task role that has two or more than two basic object methods or complex object methods related with more than one objects. Since basic object method only invokes one object, thus no information flow happens for task roles in group (1). For information flow analysis, we only consider task roles in group (2) as the basic units where information flow could occur. Also, messages are divided into two different categories: primitive message and non-primitive message. The message

sent by object method op_i of object O_i to object method op_j of another object O_j ($i \neq j$) is called non-primitive message. The message sent by object method op_i of object O_i to itself ($i = j$) is called primitive message. Task roles in group (1) only involve primitive messages, whereas task roles in group (2) might have both primitive and non-primitive messages.

If an object method op_i of object O_i sends a message (primitive message) to itself or object method op_j of object O_j (non-primitive message), it is called an *invocation execution* from op_i to op_j and denoted by $op_i \rightarrow op_j$. A set of parameter values, P, are transferred from op_i of object O_i to op_j of object O_j. Each parameter value p_k of P may be the original value directly read from object O_i or a calculated result got from the information on multiple objects, such as O_i, O_l, O_m……O_s, those objects formed the *parameter object set* of p_k, which is written as (O_i) or (O_i, O_l, O_m……O_s). For an invocation from method op_i to op_j, assume the parameter values are P= (p_1, p_2 ……p_m), a set QS ($op_i \rightarrow op_j$) is denoted to represent the parameter values P and their parameter object set which is written as {p_1, (O_1,…, O_n)}, {p_2, (O_1,…, O_m)}…… {p_m, (O_1,…, O_t)}. In the same way, object method op_j of O_j returns values to object method op_i of object O_i, we call it a *reply execution* from op_j to op_i and denote it as $op_i \leftarrow op_j$. After the execution of op_j, a set of return values, R, are transferred from op_j of object O_j to op_i of object O_i. Each return value r_i of R may be the original value of object O_i or a calculated result of the information on multiple objects, such as O_i, O_l, O_m, those objects formed the *reply object set* of r_i which is written as (O_i) or (O_i, O_l, O_m). For instance, the reply value of o_{12} may be the direct value of O_1 or might be a calculated result of O_1 and O_2, thus the reply object set should be (O_1) or (O_1, O_2). A set RS ($op_i \leftarrow op_j$) is denoted to represent the reply values R, r_1, r_2 ……r_m, and their reply object sets which is written as {r_1, (O_1, …, O_n)}, {r_2, (O_1,…, O_m)}…… {r_m, (O_1,…, O_s)}. QS ($op_i \rightarrow op_j$) or RS ($op_i \leftarrow op_j$) will be {NULL} if there are no parameter values for an invocation execution or reply values for a reply execution.

Information flow can be enacted and will be described by the following theorem.

Theorem 1. (Information Flow Condition): *In a task role R if (1) for a primitive message m of O_i, parameter value v of m is written to O_j, and (2) O_i belongs to the parameter object set, S, of v, then there exists information flow from an object O_i to another object O_j.*

Proof

We assume the parameter object set which O_i belongs to is S= (O_i,…… O_h ,O_l) , then according to the definition of parameter object set, v is a calculated result of (O_i,,…… O_h,, O_l.). Since v is written to O_j, so there is information flow from every element of S: (O_i,…… O_h ,O_l) *to* O_j according to the definition of information flow, thus, there is information flow from O_i *to* O_j.

In PERBAC system, suppose that objects and task roles satisfy the privacy principle, the problem to be dealt with is how to check if information flows satisfy the privacy principle. Based on the information flow condition and privacy principles, a privacy checking algorithm is proposed to automatically check whether information flows in each task role violate the privacy principle. The algorithm intercepts every

message exchanged between the objects in each task role and decide how to handle the message.

The privacy checking algorithm dealing with each message works as follows:

For the primitive messages which op_i of O_i sends to itself:

A read message, denoted by h = (READ, QS ($op_i \rightarrow op_i$) , RS ($op_i \leftarrow op_i$)), returns the parameter value from object O_i and its object set { O_i } to RS.

A write message, denoted by h = (WRITE, QS ($op_i \rightarrow op_i$), RS ($op_i \leftarrow op_i$)), writes the parameter values, v_1 v_2…… v_m, and their object sets of QS to object O_i. For each object O_s in the parameter object set of v_1 v_2…… v_m, if the purpose level of object O_s is greater than the purpose levels of all task roles that may access the object O_i, then return a failure message, otherwise success message is returned.

The non-primitive messages op_i of object O_i send message to op_j of object O_j:

A non-primitive message, denoted by h=(R/W, QS ($op_i \rightarrow op_j$), RS ($op_i \leftarrow op_j$)) accepts parameter values from QS, add their reply object set to RS ($op_i \leftarrow op_j$).

```
Privacy Checking Algorithm:
Let h be the message sent.
Let op_i be the execution to be invoked on O_i
Let op_j be the execution to be invoked on O_j
if h is a primitive message {
   case h=(READ, QS (op_i → op_i), RS (op_i ← op_i)){
      read the parameter values from object O_i
      return the parameter values from object O_i and
their object sets { O_i } to RS
   }
   case h=(WRITE, QS (op_i → op_i), RS (op_i ← op_i)){
      if purpose level of all task roles that can ac-
cess O_i>= purpose levels of object O_s in object sets of
QS
        then write parameter values and their object
sets to O_i and return a success message "Safe informa-
tion flow for message h"
      else return a failure message.
   }
else if h is a non-primitive message(op_i of O_i → op_j of
O_j)
   case h=(R/W, QS (op_i → op_j), RS (op_i ← op_j)) {
      accept QS the parameter values and their parame-
ter object sets
      invoke op_j
     reply object set of op_j are added to RS
      return reply values and their reply object sets
to RS
    }
}
```

The privacy disclosure problem can be solved by the following two steps. Step 1: Assign purpose level to each object and task role**.** Each object O_i is classified by a security label $\lambda(O_i)$ and every task *T* in the proposed model is classified and assigned by a purpose level $\lambda(T)$. The assignments of purpose levels for objects and task roles

should follow the privacy principle defined in chapter 3. Step 2: Analyze information flows on each task role. When applying privacy principles on PERBAC and taking information flow analysis into considerations, all the information flows in task roles must satisfy the privacy principle and this can be achieved by applying the privacy checking algorithm on the PERBAC systems. If the privacy principle is violated in a task role, two options could be implemented (1) delete the task role or (2) adjust purpose levels of objects or adjust the relationships between the task role and its objects.

4 Conclusion

In this paper, a privacy-enhanced role-based access control model (PERBAC) is proposed. It is concluded that privacy disclosure problem exists in PERBAC. In order to deal with this problem, based on the information flow analysis and privacy principles, a privacy checking algorithm is presented. The advantages of the algorithm are that (1) Each object is assigned a level of purpose. Thus, controlling method invocation through argument sensitivity is supported. (2) Whenever an object of the PERBAC model is revised, the privacy checking algorithm will be invoked to check if the privacy principle still be satisfied.

RBAC technique has been widely accepted in recent years, we believe that the presented PERBAC model and its information analysis technique can be applied to various applications, especially security and privacy concerns of enterprises.

References

[1] D. E. Bell and L. J. LaPadula, "Secure Computer Systems: Unified Exposition and Multics Interpretation", Technical Report ESDTR-75-306, The Mitre Corporation, Bedford MA, USA, March 1976.

[2] D. E. Denning, "A Lattice Model of Secure Information Flow", *Communication of ACM*, vol. 19, no. 5, pp. 236-243, 1976.

[3] D. E Denning and P. J. Denning, "Certification of Program for Secure Information Flow," *Communication of ACM*, vol. 20, no. 7, pp. 504-513, 1977.

[4] D. Ferraiolo, J. Cugini, and D. R. Kuhn, "Role Based Access Control: Features and motivation." In annual computer security applications conference, *IEEE Computer Society Press*, 1995.

[5] E. Ferrari and E. Bertino, "Providing Flexibility in Information Flow Control for Object-Oriented Systems," *Proc. 13'th IEEE Symp. Security and Privacy*, pp.130-140, 1997.

[6] K. Izaki and K. Tanaka, "Information Flow Control in Role-Based Model for Distributed Objects," *Proc. 8'th International Conf. Parallel and Distributed systems*, pp. 363-370, 2001.

[7] S. Jajodia and B. Kogan "Integrating an object-oriented data model with multilevel security" *Proc. IEEE Symp. on Sucurity and Privacy*, Oakland, CA, pp. 76-85, May 1990.

[8] Gunter Karjoth and Matthias Schunter, "A Private Policy Model for Enterprises", 15th IEEE Computer Security Foundation Workshop, June 24-26, 2002.

[9] F. Potter and S. Conchon, "Information flow in inference for free," *ICFP00, ACM*, pp 46-57, 2000.

[10] P. Samarati and E. Bertino, "Information Flow Control in Object-Oriented Systems," *IEEE Trans. Knowledge Data Eng.*, vol. 9, no. 4, pp.524-538, 1997.
[11] R. Sandhu, E. J. Coyne, H.L. Feinstein, and C.E. Youman "Role based Access Control Models". IEEE Computer, vol. 29, no. 2, pp38-47, 1996.
[12] Ravi Sandhu and Venkata Bhamidipati, "The ARBAC97 Model for Role-Based Administration of Roles: Preliminary Description and outline", *Second ACM workshop on Role-Based-Access-Control* , Fairfax, Virginia, USA, pp 41-54, 1997.
[13] G. Smith, "A new type system for secure information flow", In Proc. 14th IEEE Computer Security Foundations Workshop, Cape Breton, Nova Scotia, pp115—125, 2001.

Text Categorization Using SVMs with Rocchio Ensemble for Internet Information Classification*

Xin Xu[1,2], Bofeng Zhang[1], and Qiuxi Zhong[1]

[1] School of Computer, National University of Defense Technology, Changsha 410073, P.R. China
[2] Institute of Automation, National University of Defense Technology, Changsha 410073, P.R. China
xuxin_mail@263.net

Abstract. In this paper, a novel text categorization method based on multi-class Support Vector Machines (SVMs) with Rocchio ensemble is proposed for Internet information classification and filtering. The multi-class SVM classifier with Rocchio ensemble has a novel cascaded architecture in which a Rocchio linear classifier processes all the data and only selected part of the data is re-processed by the multi-class SVM classifier. The data selection for SVM is based on the validation results of the Rocchio classifier so that only data classes with lower precision is processed by the SVM classifier. The whole cascaded ensemble classifier takes advantages of the multi-class SVM as well as the Rocchio classifier. In one aspect, the small computational cost or fast processing speed of Rocchio is suitable for large-scale web information classification and filtering applications such as spam mail filtering at network gateways. On the other hand, the good generalization ability of multi-class SVMs can be employed to improve Rocchio's precision further. The whole ensemble classifier can be viewed as an efficient approach to compromising processing speed and precision of different classifiers. Experimental results on real web text data illustrate the effectiveness of the proposed method.

1 Introduction

With the wide spread of Internet applications, automated classification and filtering of network information has become an important research topic in recent years since the availability of digital text documents increases dramatically. The applications of Internet information classification and filtering technology range from personal information service agents [1] to spam mail filtering [2]. In these applications, automated text categorization based on machine learning approaches is one of the main

* Supported by the National Natural Science Foundation of China Under Grants 60303012, 90104001, Specialized Research Fund for the Doctoral Program of Higher Education under Grant 20049998027, Chinese Post-Doctor Science Foundation under Grant 200403500202, and A Project Supported by Scientific Research Fund of Hunan Provincial Education Department.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 1022 – 1031, 2005.

techniques that have been studied in the literature [3][4][5]. Automated text categorization is defined as the task of assigning predefined class labels to text documents by learning a set of training samples to construct a classifier or filterer model. The advantages of this approach include an accuracy comparable to that achieved by human experts, and a considerable saving in terms of expert labor power, since no intervention from either knowledge engineers or domain experts is needed for the construction of the classifier or for its porting to a different set of categories.

Slightly different from the concept of text classification, text filtering [6], which means classifying a stream of text documents, is another popular case in Internet information services based on text categorization techniques. A text filtering system observes a stream of text and divides them into two classes, i.e., relevant and irrelevant. Moreover, text classification is often needed to assign detailed categories to the filtered relevant documents. Text filtering is a special case of automated text categorization since it can be viewed as single-label text classification, i.e., classifying incoming documents into two disjoint categories, the relevant and the irrelevant.

Until now, lots of work has been done on applying machine-learning methods to automated text categorization, which include various supervised learning algorithms such as kNNs [3], decision trees [7], Naïve Bayes, Rocchio [4], neural networks [8] and support vector machines (SVMs) [9], etc. However, for classification and filtering of Internet information, the computational efficiency and classification precision of existing methods still have to be improved to meet the requirements of large-volume and complex network background data.

As a relatively new class of machine learning algorithms based on statistical learning theory, SVMs for text classification have obtained several state-of-art results in classification precision [9]. However, the computational cost of SVMs is usually large and becomes a bottleneck for applications to large-scale text documents.

Ensemble learning algorithms [10] train multiple classifiers and then combine their predictions. As studied in [10], the generalization ability of an ensemble classifier can be much better than a single learner so that the algorithms and applications of ensemble learning have been widely studied in recent years. Some of the most popular ensemble learning algorithms include Bagging [11], Boosting [12], etc. In many successful applications, ensemble-learning classifiers usually achieve the best performance in the literature.

In this paper, to overcome the performance problems in real-time Internet information classification and filtering, a novel text categorization method based on SVMs with Rocchio ensemble is proposed. In the proposed method, a multi-class SVM and a Rocchio classifier are cascaded as an ensemble classifier, which can be viewed as a new ensemble architecture different from Bagging and Boosting. The Rocchio classifier is configured to perform rapid coarse filtering of all incoming data since its processing speed is very fast. The multi-class SVM is only used to process selected part of the data output by the Rocchio. The data selection strategy for SVMs is based on the validation results of the Rocchio classifier so that data classes with lower classification precision are re-processed by the multi-class SVM to improve accuracy. The proposed method takes advantages both of the fast speed of the Rocchio and the high precision of the SVM. Experiments on real web text data illustrate the effectiveness of the ensemble classifier.

This paper is organized as follows. Section 2 gives a brief introduction on the techniques involved in text categorization for Internet information classification and filtering. Section 3 presents the principles and algorithms of multi-class SVM with Rocchio ensemble. Experimental results on real web text data are given in Section 4. And some conclusions are drawn in Section 5.

2 Text Categorization for Internet Information Classification

2.1 Internet Information Classification as a Text Categorization Problem

Internet information classification and filtering are used for discriminating various classes of data in web pages and emails. To employ automated text categorization in Internet information classification applications, sampled network data are collected and labeled with their corresponding classes. The original sampled data may have different formats and coding schemes such as Unicode, MIME, etc. So they have to be transformed to a uniform format by extracting ASCII text information from them. The transformed data are then divided into two sets for automated classifier training and performance testing.

After sample data collection, automated text categorization usually involves three steps, namely, document representation, classifier construction, and performance evaluation or validation. Document representation can be viewed as a preprocessing process, which includes stop word elimination, stemming, feature selection and weighting, etc. After preprocessing, a text document is usually represented as a data vector

$$d = [w_{f1}, w_{f2}, ..., w_{fn}] \tag{1}$$

where w_{fi} (i=1,2,…,n) are the weights of document features. n is the number of document features. The feature weights are usually determined by some function of feature frequencies:

$$w_{fi} = g(t_{fi}) \tag{2}$$

where t_{fi} is the occurrences of feature f_i in a document and the selection methods of function $g(.)$ include TF*IDF, log(TF)*IDF, etc [3]. The document representation method is usually called the vector space model. For detailed discussion on the vector space model, please refer to [3].

In the classifier construction step, various machine learning methods can be used to learn a classifier model based on training data. The training data are composed of preprocessed document data vectors from different classes and each data vector is labeled with the corresponding class labels.

The performance evaluation of text classifiers is conducted on a testing sample data set, which is usually different from the training set. In text categorization, there are two main criteria for performance evaluation, i.e., precision and recall. Let N_{ci} denote the number of test samples that are classified correctly to class i, the precision P_i and recall R_i of a text classifier are defined as follows:

$$P_i = \frac{N_{ci}}{N_{ci} + M_i} \qquad R_i = \frac{N_{ci}}{N_{ci} + N_i} \tag{3}$$

where M_i is the number of samples that are misclassified to class i and N_i is the sample number of class i that has not been classified as class i.

2.2 The Rocchio Algorithm for Text Classification

In automated text categorization, linear classifiers usually construct linear profiles of different classes explicitly so that they are easy to be understood. The Rocchio method is one of the most popular linear models in text categorization due to its simplicity and low computational cost. It relies on an adaptation to text categorization of the well-known Rocchio's formula for relevance feedback in the vector space model, i.e.,

$$p_{ik} = \beta \sum_{d_j \in POS_i} \frac{w_{jk}}{N_{pi}} - \gamma \sum_{d_j \in NEG_i} \frac{w_{jk}}{N_{ni}} \tag{4}$$

where w_{ji} are the weight of term i in document d_j, POS_i is the set of documents that belong to class i, NEG_i is the set of documents that are not labeled with class i, N_{pi} and N_{ni} are the document numbers of POS_i and NEG_i, respectively, and β, γ are two control parameters that allow the adjustment of relative importance of positive and negative examples. Then, for each text class i (i=1,2,…,m), by dividing the training examples into a positive subset and a negative subset, a linear profile of the Rocchio classifier is constructed as follows:

$$p_i = [p_{i1}, p_{i2}, ..., p_{in}] \qquad (i=1,2,\ldots,m) \tag{5}$$

After building the linear profile of Rocchio classifier, the class label of a test example is determined by computing the distance between its weight vector and the linear profile, where the following cosine formula is usually used.

$$D(t, p_i) = \frac{\sum_j w_j p_{ij}}{(\sum_j w_j^2)^{1/2} (\sum_j p_{ij}^2)^{1/2}} \tag{6}$$

Then the class label of a test sample t=[w_1, w_2,…, w_n] is assigned by selecting the class whose profile has the minimal distance with the test sample.

The Rocchio algorithm for text categorization has the advantage of simplicity and fast testing speed. However, as all linear classifiers, it separates text documents linearly so that it is hard to obtain better classification precision for large volumes of text documents.

3 The Multi-class SVM with Rocchio Ensemble

In this section, we will present a novel ensemble classifier learning method that combines multi-class SVMs with Rocchio to take advantages both of the nonlinear optimal classifier built by SVMs and of the fast processing speed of Rocchio's linear profiles. First, the structure of the ensemble classifier based on multi-class SVMs and Rocchio is given in the following.

3.1 Structure of the Multi-class SVMs with Rocchio Ensemble

Fig. 1 shows the component structure and running scheme of the proposed ensemble classifier. All the feature vectors of Internet information flow, which are produced by preprocessing and feature extraction of Web pages and e-mails, are processed and classified by the Rocchio classifier. Besides classification, the Rocchio classifier also carries out text filtering by dividing the classes into two sets, i.e., relevant and irrelevant. Usually, there are several classes in the relevant set and the irrelevant set only contains one class. Although the Internet information flow may have large volume of data, the filtering and classification based on Rocchio are time-efficient due to the fast processing speed of linear profiles.

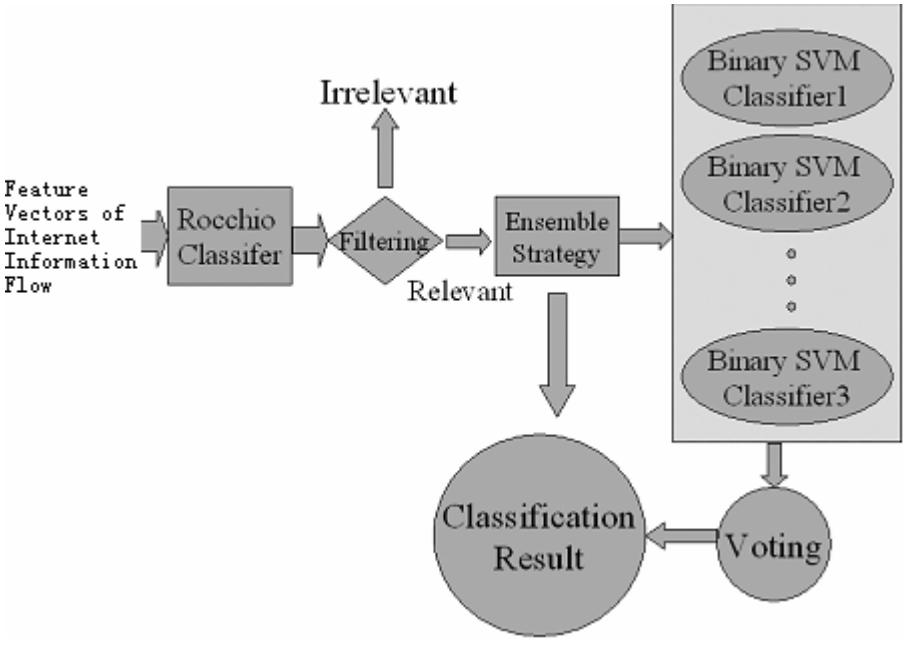


Fig. 1. Structure of the ensemble classifier

After filtering, the relevant data are re-processed by a new ensemble strategy which considers the classification performance of the linear Rocchio classifier for different classes of text documents and then selects the data classes that have lower precision to be processed by the multi-class SVM classifier. Since classifiers based on SVMs have been proved to be very effective in high precision classification, the use of multi-class SVMs to re-classify a part of the data will be beneficial to improve classification precision. Moreover, we can select only a small part of the text data to be re-classified by the SVM so that the computational cost will be lower than processing all the data by

SVMs again. Therefore, the ensemble strategy can be viewed as a compromising mechanism between classification precision and processing speed, which are both crucial to the ultimate performance of text classification methods in real-time Internet information processing applications.

The multi-class SVMs are consisted of multiple binary SVM classifiers and the outputs of the binary SVMs are combined by a voting strategy. Details about the multi-class SVM algorithms will be discussed in the following subsection.

3.2 The Multi-class SVM Algorithm Based on One-vs-All Strategy

Based on the idea of constructing optimal hyper-plane to improve generalization ability, SVMs are originally proposed for binary classification problems. Nevertheless, most real world pattern recognition applications are multi-class classification cases. Thus, multi-class SVM algorithms have received much attention over the last decades and several decomposition-based approaches for multi-class problems are proposed [13]. The idea of decomposition-based methods is to divide a multi-class problem into multiple binary problems, i.e., to construct multiple two-class SVM classifiers and combine their classification results. There are several combining strategies for the implementation of multi-class SVMs using binary SVM algorithms, which include one-vs-all, one-vs-one, and error correcting coding [13], etc. Among the existing decomposition approaches, the one-vs-all strategy has been regarded as a simple method with relatively low precision when compared with other multi-class schemes. However, a very recent research [14] demonstrates that one-vs-all classifiers is extremely powerful and can produce results that are usually at least as accurate as other methods. In the proposed ensemble classifier, we employ the one-vs-all strategy for multi-class SVMs, where a binary SVM classifier is constructed for each partition of one class against all the other classes. For *m* classes of data, there will be *m* binary SVM classifier to be built based on different partitions of the training data. Thus, the multi-class classification problem is decomposed into *m* subtasks of training binary SVM classifiers.

In the training of binary SVM classifiers, a hyperplane is considered to separate two classes of samples. Following is the linear form of a separating hyperplane.

$$(\vec{w} \cdot \vec{x}) + b = 0 \qquad \vec{w} \in R^n, \ b \in R \tag{7}$$

Then the decision function can be given by

$$f(x) = \mathrm{sgn}(\vec{w} \cdot \vec{x} + b) \tag{8}$$

Based on the structural risk minimization (SRM) principle in the statistical learning theory, the optimal linear separating hyperplane can be constructed by the following optimization problem

$$\min_{\vec{w},b} \frac{1}{2} \left\| \vec{w} \right\|^2 \tag{9}$$

subject to

$$y_i(\vec{w}\cdot\vec{x}_i+b)\geq 1,\quad i=1,2,...,N \tag{10}$$

To reduce the effects of noise and outliers in real data, the following soft margin techniques are usually used, which is to solve the primal optimization problem as

$$\min_{\vec{w},b}\frac{1}{2}\|\vec{w}\|^2+C\sum_{i=1}^{N}\xi_i \tag{11}$$

subject to

$$y_i(\vec{w}\cdot\vec{x}_i+b)\geq 1-\xi_i,\quad \xi_i\geq 0,\ \ i=1,2,...,N \tag{12}$$

The Lagrangian dual of soft-margin support vector learning can be formulated as

$$\max_{\alpha}\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j(\vec{x}_i\cdot\vec{x}_j) \tag{13}$$

subject to

$$0\leq\alpha_i\leq C,\ \ i=1,2,...,N\ \text{ and }\ \sum_{i=1}^{N}\alpha_i y_i=0 \tag{14}$$

Since in most real-world classification problems, nonlinear separating planes have to be constructed, a 'kernel trick' is commonly used to transform the above linear form of support vector learning algorithms to nonlinear ones. In the kernel trick, a nonlinear feature mapping is introduced to build linear hyper-plane in the feature space without explicitly computing the inner products in high-dimensional spaces. Let the nonlinear feature mapping be denoted as

$$\vec{x}\rightarrow\phi(\vec{x}) \tag{15}$$

the dot products $(\vec{x}_i\cdot\vec{x}_j)$ in linear SVM algorithms can be replaced by dot products in nonlinear feature space and a Mercer kernel function can be used to express the dot products in high-dimensional feature space

$$k(\vec{x}_i,\vec{x}_j)=(\vec{x}_i\cdot\vec{x}_j) \tag{16}$$

Then the optimization problem of SVMs for two-class soft margin classifiers is formulated as follows:

$$\max_{\alpha}\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j k(\vec{x}_i\cdot\vec{x}_j) \tag{17}$$

subject to

$$0\leq\alpha_i\leq C,\ \ i=1,2,...,N\ \text{ and }\ \sum_{i=1}^{N}\alpha_i y_i=0 \tag{18}$$

The decision function of each binary SVM is

$$f_k(\vec{x}) = \mathrm{sgn}(\sum_{i=1}^{N} \alpha_{ki} y_{ki} k(\vec{x}_{ki}, \vec{x}) + b_k) \quad k = 1,2,...,m \tag{19}$$

where $f_k(\vec{x})$ is the decision function of classifier k and ($\vec{x}_{ki}$, y_{ki}) (k=1,2,…,m) are the corresponding training samples.

4 Experimental Results

The proposed ensemble classifier based on multi-class SVMs and linear Rocchio is evaluated in text documents extracted from real network data. The text data are manually partitioned to 4 classes for training and testing. The features or number of keywords for the four classes of text data are 134, 204, 61, 79, respectively. Then a feature vector of 479 dimensions is constructed for each text document and the elements of every feature vector are normalized to the interval of [0, 1]. The total number of document samples in our experiments is 1111. Table 1 summarizes the data information about training and testing samples.

Table 1. Sample data for the experiments

Class	Feature dimension	Sample number
Class 1	134	427
Class 2	204	275
Class 3	61	181
Class 4	79	228
Total	478	1111

Table 2. Precision and recall of different classifiers

		Class 1	Class 2	Class 3	Class 4
Precision	SVM	98.3%	100%	99.4%	100%
	Rocchio	96.1%	94.1%	97.5%	97.2%
	Rocchio+SVM Ensemble	**98.6%**	**100%**	97.5%	97.3%
Recall	SVM	100%	98.4%	98.1%	97.2%
	Rocchio	96.1%	94.6%	96.3%	97.2%
	Rocchio+SVM Ensemble	99.2%	97.8%	**98.1%**	**99.1%**

In the experiments, the multi-class SVM classifier with Rocchio ensemble is trained and tested on the sample data. We also evaluated the Rocchio classifier as well as the multi-class SVM classifier separately to make comparisons on the precision and testing speed of different algorithms. In the implementation of multi-class SVMs, RBF (Radius Basis Functions) kernels are selected with a width parameter σ=0.1 for each binary SVM classifier.

Table 2 shows the experimental results of text classification using the above three learning algorithms, i.e., the conventional multi-class SVM classifier, the linear Rocchio classifier, and the proposed ensemble classifier based on SVM and Rocchio. It is shown that the precision and recall of the ensemble classifier are better than the Rocchio classifier alone and are comparable to conventional multi-class SVM.

Fig.2 presents the testing speed comparison of the three types of classifiers, where the processing time of each classifier on the testing data set is computed. It is clear that the multi-class SVM with Rocchio ensemble has faster speed than conventional SVMs so that it is more suitable for real-time Internet applications of data filtering and classification with high volume and fast speed.

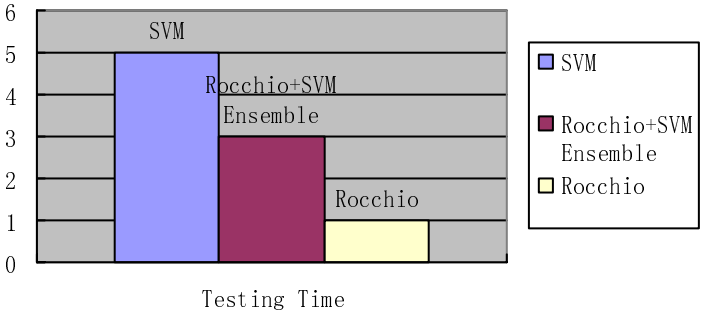


Fig. 2. Testing time comparisons of different classifiers

5 Conclusions

With the wide spread of Internet applications, information filtering and automated classification of Internet data have attracted much attention. To realize high precision information filtering and classification with fast processing speed, this paper presents a novel text classifier ensemble method based on multi-class SVMs and Rocchio. In the ensemble classifier, a Rocchio classifier and a multi-class SVM using one-vs-all strategy are trained separately. Data flows of text documents are firstly processed by the Rocchio classifier, and only a part of the data is reprocessed by the SVM to improve accuracy. Thus, the proposed ensemble method makes advantages of the good generalization ability of SVMs as well as the fast processing speed of linear Rocchio classifiers. Experimental results demonstrate the effectiveness of the proposed method.

References

1. Konstantinos V. C., et al.: Automatic Web Rating: Filtering Obscene Content on the Web. Lecture Notes in Computer Science, vol.1923 (2000)
2. Schneider K.: A Comparison of Event Models for Naïve Bayes Anti-spam E-mail Filtering. In: Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL'03) (2003)

3. Sebastiani F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys (1999)
4. Joachims T.: A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. In: Proceedings of 14th International Conference on Machine Learning. ICML-97 (1997)
5. Kom Y.J., et al.: Automatic Text Categorization by Unsupervised Learning. In: Proceedings of the 17th Conference on Computational Linguistics, Volume 1, (2000)
6. Ittner, D. J., Lewis, D. D., Kim, Y.-H.,et al.: Text Filtering by Boosting Naive Bayes Classifiers. In: Proceedings of 23rd ACM International Conference on Research and Development in Information Retrieval, SIGIR-00, Athens, Greece (2000) 168–175
7. Lewis, D. D., Cartlett, J.: Heterogeneous Uncertainty Sampling for Supervised Learning. In: Proceedings of 11th International Conference on Machine Learning, New Brunswick, NJ, ICML-94 (1994) 148–156
8. Merkl, D.: Text Classification with Self-Organizing Maps: Some Lessons Learned. Neurocomputing, 21, 1/3 (1998) 61–77
9. Joachims, T.: Transductive Inference for Text Classification using Support Vector Machines. In: Proceedings of 16th International Conference on Machine Learning, ICML-99, Bled, Slovenia (1999) 200–209
10. Zhou Z.-H., Wu J., and Tang W.: Ensembling Neural Networks: Many Could Be Better Than All. Artificial Intelligence, 137(1-2) (2002) 239-263
11. Breiman L.: Bagging Predictors. Machine Learning, 24 (2) (1996) 123–140
12. Freund Y.: Boosting a Weak Algorithm by Majority. Information and Computation, 121 (2) (1995) 256-285
13. Dietterich T. G., Bakiri G.: Solving Multiclass Learning Problems via Error-correcting Output Codes. Journal of Artificial Intelligence Research, 2 (1995) 263-286
14. Rifkin R., Klautau A.: In Defense of One-Vs-All Classification. Journal of Machine Learning Research, 5 (2004) 143–151

OpenRouter: A TCP-Based Lightweight Protocol for Control Plane and Forwarding Plane Communication*

Feng Zhao, Jinshu Su, and Xiaomei Cheng

School of Computer, National University of Defense Technology,
Changsha 410073, Hunan, China
fengzhao1978@tom.com

Abstract. The Network Processing Forum (NPF) is delivering specifications for programmable network elements that reduce equipment time-to-market, while increasing time-in-market. ForCES (Forwarding and Control Element Separation) aims to define framework and associated mechanisms for standardizing the exchange of information between the logically separate functionality of the control plane and the forwarding plane. To make good use of the efforts of NPF and TCP reliability, this paper presents a TCP-based lightweight protocol for control plane and forwarding plane communication. This protocol meets many requirements of the ForCES working group charter for a protocol. We provide an evaluation of its applicability for a ForCES protocol. One advantage of this protocol is that it can provide good support for Common Programming Interface (CPI) of NPF. Also it can be easily extended to support new services or new functions. The current version of this protocol has been implemented in our IPv6 core router.

1 Introduction

In traditional network equipment, control plane components are interdependent with the forwarding plane functions implemented by custom ASICs. The traditional model has some problems such as inflexible hardware solutions. To solve these problems, control plane and forwarding plane functions, while still interdependent, should be abstracted from each other in the next-generation building-block model. This means that traditional, monolithic architectures can be broken up into functional blocks connected by specific interfaces. The decoupling of control and forwarding paths has several desirable characteristics. First of all, neither component bottlenecks the other as long as their capacities are sufficient to sustain their input demands. Moreover, because of decoupling, improvements in any one component allow the router to service higher input loads for that component, independent of the other component.

In programmable networks, there are two possible ways to place a controller. One is to put the controller quite close to the controlled Forwarding Elements (FE). The

* This research was supported by the National Grand Fundamental Research 973 Program of China under Grant No. 2003CB314802 and the National Natural Science Foundation of China Grant No.90104001.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 1032 – 1041, 2005.

other is to put the controller remotely away from FEs. ForCES has explicitly defined the controller place in local place, such as in the same room or at a very close proximity. In IP routers, because most of the functions are packet-by-packet-processing based, we do not think that it is possible or necessary for an open programmable IP networks. So we set the architecture to local mode as ForCES does.

Control plane and forwarding plane can communicate over a variety of link, network, and transport media. When developing IP routers such as routers using network processors, cluster-based IP routers [1], distributed multi-processor routers and so on, control plane and forwarding plane can be connected by a high speed system area network. And TCP provides reliability and congestion control. Also the Network Processing Forum (NPF) is delivering specifications for programmable network elements that reduce equipment time-to-market, while increasing time-in-market. ForCES (Forwarding and Control Element Separation) aims to define framework and associated mechanisms for standardizing the exchange of information between the logically separate functionality of the control plane and the forwarding plane. To make good use of the efforts of NPF, so this paper presents a TCP-Based lightweight protocol as communication mechanism between control and forwarding called OpenRouter, supporting control plane extensibility. In this protocol, resources in the forwarding plane that need to be controlled or managed by the control plane are abstracted as objects. Because of this characteristic, this protocol provides good support for Common Programming Interface (CPI) of NPF. Also it is easily to be extended to support new services or new functions. This protocol is simple and easy to be implemented. It has been applied to our IPv6 core routers.

2 The Router Configuration and OpenRouter Protocol

2.1 The Router configuration

To apply OpenRouter protocol, all the CEs and FEs are interconnected with TCP/IP network in a router configuration. These separated CEs and FEs are one hop or multiple hops away from each other. The CEs and FEs communicate to each other by running OpenRouter, and the collection of these CEs and FEs together become one routing unit to the external world. But we neither care how the CEs communicate, nor do we care how the FEs do. FEs may be interconnected with some kind of high speed LAN connection or a switch fabric, etc. Fig.1 shows such a router configuration example.

2.2 FE Resources Abstraction

Dominic Herity [2] points out that an object oriented API in the control plane gives the richness and flexibility you need to abstract a complex entity like a network processor. IEEE P1520 reference model [3] for open programmable computer networks has four horizontal layers and interfaces for IP routers. Each layer defines what is termed as a *level*. Each level comprises a number of *entities* in the form of

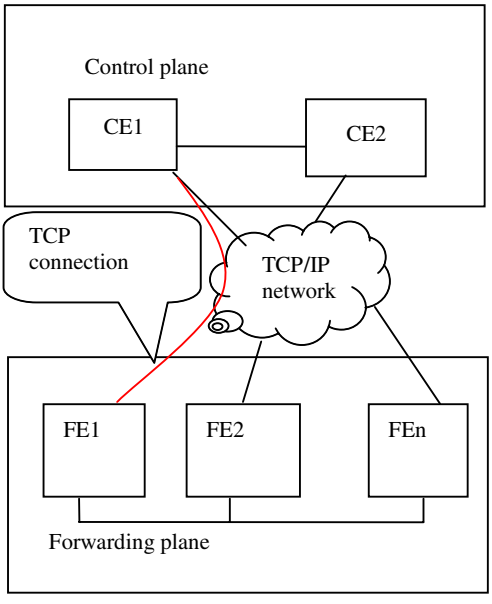


Fig. 1. A router configuration example

object class name: interface	
attributes:	**methods:**
information about the interface type of interface max size of datagram current bandwidth in bits/sec desired state for the interface current operational status …	enable interface disable interface clear statistical counters get MTU set MTU send packets // event notification inform CEs of the status change (link up/down) redirect packets

Fig. 2. An interface object class

algorithms or objects representing logical or physical resources depending on the level's scope and functionality. The *virtual network device level* (L-interface) has as its sole purpose to logically represent resources in the form of objects (entities) thereby isolating the upper layers from hardware dependencies or any other proprietary interfaces. P1520 views core router abstractions in a hierarchical manner. For providing an object oriented API, we think it is useful to abstract FE resources in a

hierarchical manner. So we view FE resources as objects equally. Any resource or element function [4] in the forwarding plane that needs to be controlled or managed by the control plane is abstracted as an object. Fig.2 show an example of an interface object class abstracted from port resources.

2.3 Protocol Overview

We establish two TCP connections for OpenRouter messages: one for control messages, the other for the slow router data path. The TCP connections carry object oriented messages used to configure or control objects in the FEs by the CEs and object oriented messages encapsulating the packets in the slow router data path, as illustrated in Fig.3.

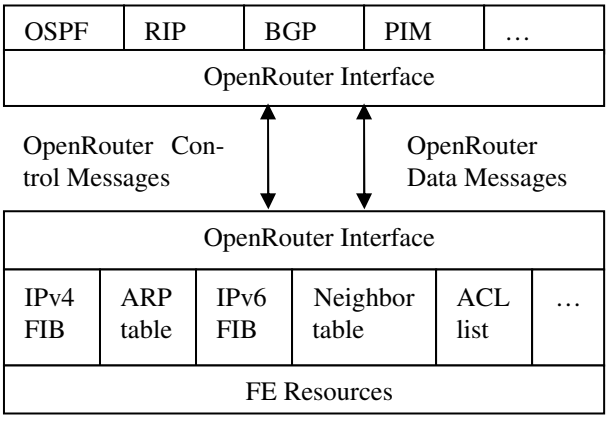


Fig. 3. Object oriented Open Router messages

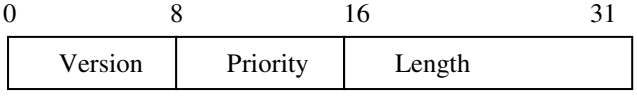


Fig. 4. OpenRouter message header format

OpenRouter protocol consists of OpenRouter protocol messages. The messages can be sent from a CE to a FE as configuration messages, or from a FE to a CE as response or event report messages.

In data format, an OpenRouter protocol message is composed of an OpenRouter protocol header and a message body.

The header format is illustrated in Fig.4.

Version:
Version number, this version of OpenRouter is set to 0x01.

Priority:
The priority is used for receiver of the message to know if the message should be processed ahead of other lower priority messages.
Length:
The message body length, not including the four bytes message header.
The message body format is illustrated in Fig.5.

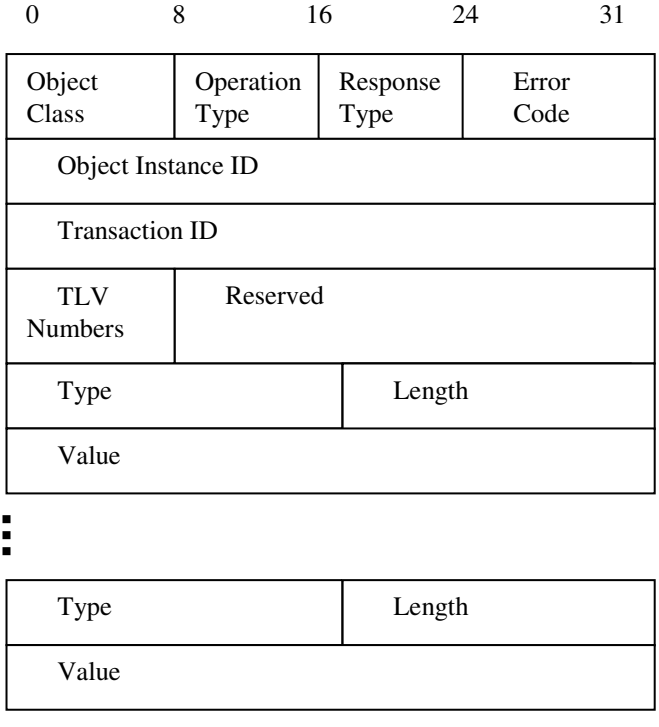


Fig. 5. OpenRouter message body format

Object Class:
Resources that can be managed or controlled by CEs are abstracted as objects, such as ports, IPv4 forwarding table, ARP table, IPv6 forwarding table, neighbor table, access list, and so on.
Operation Type:
This corresponds to an object method. For example, it may be an operation adding a route entry to IPv6 forwarding table.
Response Type:
A CE may want to know whether an operation is completed successfully or not, so we should provide a mechanism to allow the CE to control the nature required from the FE. If the response type is ACK, then the FE that received a message generated by this request will send a response to the CE indicting the operation results. NOACK

indicates that no responses should be generated as a result of this request. NEGACK indictates that only unsuccessful operation results should be reported.

Error Code:

Asynchronous response messages depend on which method of an object that generates the response. Error Code is typically included with each response.

Object Instance ID:

Object Instance ID is needed to identify an object, together with object class.

Transaction ID:

Used for the system to uniquely distinguish individual received messages. It may be generated by message senders as a random number. For request messages, the sender may select any transaction identifier; while for response messages, the transaction identifier is set to the same value as that of the message it responses to.

TLV Numbers:

It indicates how many parameters are required by the object method. These parameters are encapsulated in a type, length, value (TLV) format. Each TLV must be aligned on a word (4-bytes) boundary. But TLVs can be placed in any order.

3 OpenRouter Protocol Evaluation

Though OpenRouter protocol is a proprietary protocol by now, it meets many requirements of the ForCES working group charter for a protocol. This section provides an evaluation of its applicability for a ForCES protocol.

3.1 Architectural Requirements Compliance Evaluation

OpenRouter protocol is designed based on the ForCES architecture requirements [5]. We review its compliance to the individual requirement items as below:

1) For architecture requirement #1
   OpenRouter packets are transported via TCP/IP mediums, against any suitable medium, such as Ethernet, ATM fabrics, and bus backplanes.
2) For architecture requirement #2
   ForCES requires that FEs MUST support a minimal set of capabilities necessary for establishing network connectivity (e.g., interface discovery, port up/down functions). OpenRouter protocol has no restriction on this functionality.
3) For architecture requirement #3
   By properly configuring FEs with their LFBs in a NE via OpenRouter protocol, packets can arrive at one FE and depart at the other FE or FEs.
4) For architecture requirement #4
   By properly configuring LFBs in FEs in a NE via OpenRouter protocol, the NE can appear as a single functional device in a network.
5) For architecture requirement #5
   OpenRouter protocol can be extended to provide a way to prevent unauthorized ForCES protocol elements from joining a NE.

6) For architecture requirement #6
   A FE is able to asynchronously inform the CE of a failure or increase/decrease in available resources or capabilities on the FE via OpenRouter event notification message.
7) For architecture requirement #7
   A FE can establish TCP connections with any CE. So CE redundancy or CE failover can be supported.
8) For architecture requirement #8
   FEs is able to redirect control packets (such as routing messages) addressed to their interfaces to the CE via interface object methods.
9) For architecture requirement #9
   OpenRouter supports RFC1812 compliant router functions by means of following mechanisms in OpenRouter:
   – Fully supporting ForCES FE model
   – Packet redirection messages
   – Datapath management messages
   – Managed Object(MO) management messages
10) For architecture requirement #10
    OpenRouter does not meet this requirement.
11) For architecture requirement #11
    In OpenRouter, a FE is identified by an IP address. So The NE architecture is capable of supporting hundreds of FEs. And a port is identified by a 32 bits object instance identifier and an object class. It is capable of supporting tens of thousands of ports.
12) For architecture requirement #12
    FEs AND CEs can join and leave NEs dynamically by establishing the TCP connections or break them.
13) For architecture requirement #13
    OpenRouter supports multiple FEs working together in a NE by using FE identifiers. OpenRouter supports multiple CEs working together in a NE by supporting CE redundancy or failover functionality.
14) For architecture requirement #14
    CEs can use object oriented messages to get the SNMP MIBs.

3.2 Model Requirements Compliance Evaluation

The OpenRouter protocol message is separated into generic message header and an extensible message body payload which can be used to carry the FE, Logical Functional Block (LFB) specific data which is defined by the FE Model. Thus the OpenRouter protocol is cleanly separated from the data model that it carries. The FE Model draft [4] defines the data model for the Forwarding Element and meets all the Model requirements.

3.3 Protocol Requirements Compliance Evaluation

We don not detailed the compliance levels for OpenRouter Protocol in this paper. A summary of the compliance levels is given in table 1.

Where:

T = Total compliance. Meets the requirement fully.
P+ = Partial compliance. Fundamentally meets the requirement through the use of extensions (e.g. packages, additional parameters, etc.)

Table 1. A summary of the compliance levels

	Protocol Requirements	Compliance levels
1	Configuration of Modeled Elements	T
2	Support for Secure Communication	P+
3	Scalability	T
4	Multihop	T
5	Message Priority	T
6	Reliability	T
7	Interconnect Independence	p
8	CE Redundancy or CE Failover	T
9	Packet Redirection/Mirroring	T
10	Topology Exchange	P+
11	Dynamic Association	T
12	Command Bundling	p
13	Asynchronous Event Notification	T
14	Query Statistics	T
15	Protection Against Denial of Service Attacks	T

4 OpenRouter Protocol Implementation

We implemented our core IPv6 router based on OpenRouter protocol. As shown in Fig.6, OpenRouter Master runs on a CE which uses Linux as its operation system, and OpenRouter Agent runs on a FE using VxWorks as its operation system. When a routing protocol learns some route, routing table management adds this route to Linux kernel and encapsulates this route as an OpenRouter message which is got and sent to the OpenRouter Agent by the OpenRouter Master. When the CE wants to send data from some interface, the OpenRouter Master will get the message from the Linux kernel. The OpenRouter Agent decodes the received messages, takes some action such as writing a forwarding table entry to network processors, decides whether it should report the result to the CE according to the response type in the message. When a packet needs to be redirected to CE or some events happen, the OpenRouter Agent sends the encapsulated message to OpenRouter Master.

The OpenRouter Agent listens on OpenRouter socket ports, waiting for the CE's connections. After the TCP connections are established, the CE and FE exchange the security information, decides whether their association can be established or not. The Heart Beats message is used to decide whether the association is keep alive. Once the association is established, the CE and FE exchange message to control or manage FE resources. Fig.7 shows an example of messages exchange between a CE and a FE.

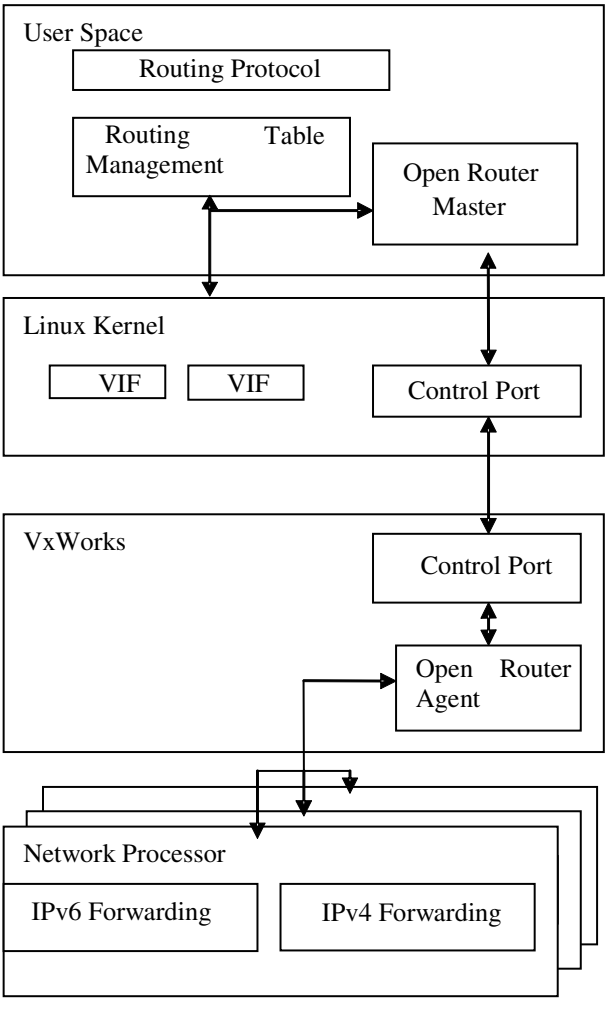


Fig. 6. Router structure based on OpenRouter protocol

5 Conclusion and Future Work

Because the Network Processing Forum (NPF) is delivering specifications for programmable network elements that reduce equipment time-to-market, while increasing time-in-market, we should make good use of the efforts of NPF when we define a protocol for control plane and forwarding plane communication. On the other hand, TCP provides reliability and congestion control. So this paper presents an extensible TCP-based protocol called OpenRouter for control plane and forwarding plane communication. Compared with other protocols [6, 7, 8], this protocol provides good support for Common Programming Interface (CPI) of NPF. Also it can be easily extended to support new services or new functions. This protocol has been adopted and implemented by our IPv6 core router project. Though it is a proprietary protocol by

now, it meets many requirements of the ForCES working group charter for a protocol. We are working to standardize this protocol and intend to draft it.

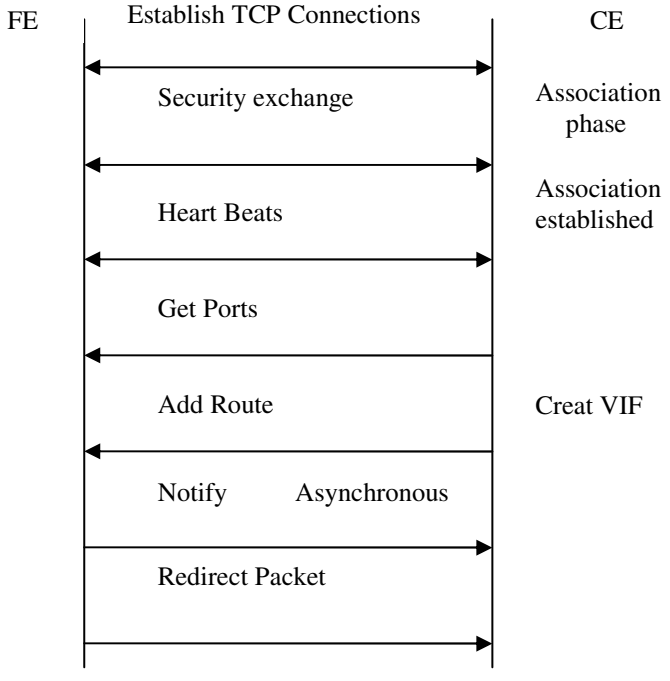


Fig. 7. Example of messages exchange between CE and FE

References

1. Prashant Pradhan Tzi-cker Chiueh, “Implementation and Evaluation of A QoS-Capable Cluster-Based IP Router”, Proceedings of SC 2002
2. Dominic Herity, “Network Processor Software: Some Lessons Learned” http://www.s3group.com/pdf/N_1-2001-NP-SoftwareLessonslearned.pdf, May 2001
3. Spyros Denazis, Kazuho Miki, John Vicente, Andrew Campbell, “Interfaces for Open Programmable Routers”, Proceedings of IWAN, July 1999
4. L. Yang, et. al, ForCES Forwarding Element Functional Model, draft-yang-forces-model-02.txt ,Feb. 2004
5. H. Khosravi, T. Anderson, Requirements for Separation of IP Control and Forwarding, rfc3654, Nov. 2003
6. Alex Audu, Ram Gopal, et. al, “ForwArding and Control ElemenT protocol (FACT)”, draft-gopal-forces-fact-06.txt , Nov. 2003
7. W. Wang, General Router Management Protocol (GRMP) Version 1, draft-wang-forces-grmp-01.txt, Nov. 2003
8. W. Wang, "A Control Scheme and Management Protocol for Open Programmable QoS IP Routers", Proceedings of SCI 2003, July 2003

Efficient Approach to Merge and Segment IP Packets*

Wenjie Li, Lei Shi, Yang Xu, and Bin Liu

Department of Computer Science and Technology, Tsinghua University,
Beijing 100084, P.R. China
{lwjie00, shijim, xy01}@mails.tsinghua.edu.cn,
liub@mail.tsinghua.edu.cn

Abstract. Variable-size IP packets are generally segmented into fixed-size cells for switching and scheduling in scalable input queueing switches. While switch bandwidth loss occurs when packets' sizes are not integral times of the cell size, and the speedup of at least two is required to achieve full line rate. This paper proposes a framing approach called Bit Map Packet Framing (BMPF) to merge and segment IP packets efficiently. In BMPF, the partially filled cell can carry some bytes from the following packet. Thus switch bandwidth loss is avoided and the required speedup is greatly lowered to 1.175. BMPF is superior to other conventional framing methods, such as PPP, HDLC and COBS. Furthermore, BMPF can be also deployed to merge IP packets in optical packet switches.

1 Introduction

Input queueing switches are employed widely in state-of-the-art core routers due to low complexity and high scalability [1][2][3]. In these switches, variable-size IP packets are segmented into fixed-size cells for switching and scheduling in each input port, and reassembled in each output port [4]. Although packet-mode scheduling is proposed to simplify reassembly [5], cell is still the basic switching unit and packet-to-cell segmentation is necessary. In the traditional segmentation method, packets are segmented independently, and padded to integral times of the chosen cell size. E.g., if the cell size is 64 bytes, a 65-byte packet is padded and segmented into two cells, in which the second cell only contains 1-byte valid data. The switch bandwidth is wasted for padding the last cell of an IP packet with useless bytes. In the worst case, the speedup of at least two is required to achieve full line rate.

Furthermore, optical switch fabric is considered as an essential upgrade for terabit core routers due to its "unlimited" bandwidth [6][7]. In optical switches, the switch reconfiguration time is significant because of the mechanical setting and other factors. E.g., in typical micro-electro-mechanical system (MEMS) optical switches, the reconfiguration time is about 10 μs [8]. To reduce the number of reconfiguration,

* This work was supported by NSFC (No. 60173009 and No. 60373007), China 863 High-tech Plan (No. 2002AA103011-1 and No. 2003AA115110), China/Ireland Science and Technology Collaboration Research Fund (CI-2003-02) and the Specialized Research Fund for the Doctoral Program of Higher Education of China (No. 20040003048).

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 1042 – 1051, 2005.

multiple IP packets are merged into a large fixed-size frame and switched together [9][10]. When the size sum of IP packets is larger than the capacity of one frame, another one is needed. Under the extreme condition, only half of the switch bandwidth is utilized. E.g., suppose the frame size is 1000 bytes, each frame only carries one IP packet when packets of 500 and 501 bytes arrive alternately. As far as we know, few works have been done to overcome this problem. Most related works simply adopt the speedup, or make the assumption that one frame exactly contains multiple IP packets. Obviously, the assumption does not hold in real IP switches and routers. Actually, frames in optical switching and cells in electronic switching have no essential difference, and fixed-size frames can be considered as ultra-large cells, so for convenience we use "cell" to refer both of them in the following sections.

In this paper, we propose a practical framing approach called Bit Map Packet Framing (BMPF). BMPF achieves full line rate with a rather low speedup and utilizes the switch bandwidth efficiently. The behavior of BMPF is characterized as three steps. First, IP packets are framed into a byte stream before entering the switch fabric. Second, the byte stream is segmented and switched without considering its boundary, where the concept of IP packets does not exist any more. Finally, in each output port IP packets are extracted from the byte stream. In BMPF, a partially filled cell of the current packet can carry some bytes from the following packet. This makes the switch bandwidth fully utilized. The same idea is also mentioned in [11], but the detailed framing method is not studied.

The overhead of BMPF is about 14.29% in the worst case, and is only 0.1116% for uniformly distributed random data. By comparing with previous framing methods, we obtain: (i) BMPF is more efficient than the point-to-point (PPP) protocol [12]; (ii) BMPF is more efficient and practical than high-level data link control (HDLC), which is developed by ISO; (iii) BMPF achieves less encoding delay than consistent overhead byte stuffing (COBS) [13].

The rest of the paper is organized as follows. Section 2 describes the behavior of BMPF. Section 3 analyzes BMPF's performance and compares the overhead of BMPF with that of PPP, HDLC and COBS. Section 4 presents the hardware implementation architecture of encoding and decoding in BMPF. Section 5 concludes this paper.

2 Framing Approach to Merge and Segment IP Packets: BMPF

Fig. 1 shows the BMPF scheme to merge and segment IP packets. Each framed packet is composed of three parts: flag, stuffing bit map (SBM) and the converted packet. The byte 0x7E is used to mark a packet's boundary. Packets are checked byte by byte. When 0x7E is found, 0x5E is outputted and one bit in SBM is set to '1'. When 0x5E is met, it remains unchanged and one bit in SBM is set to '0'. Otherwise, original bytes are outputted directly and no bits are set. The most significant bit (bit 7) of each SBM byte is set to '1'. Thus in SBM 0x7E never appears and only seven bits of one byte are utilized. The marking bits for 0x7E and 0x5E are placed one by one from bit 6 to bit 0 in SBM. When these bits are not up to seven bits, '0's are padded. SBM of the current packet is placed at the beginning of next packet. If there is no next packet to be transferred, a packet only containing SBM is inserted.

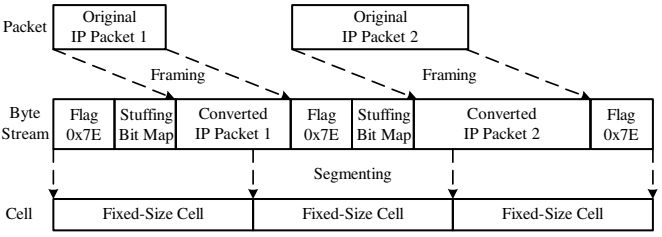


Fig. 1. The BMPF framing approach

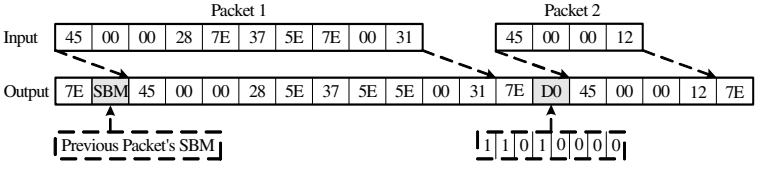


Fig. 2. An example of BMPF

After variable-size IP packets are framed into a byte stream, the byte stream is segmented into fixed-size cells for switching and scheduling. In each output port, the flag byte 0x7E is used to determine the boundary of each packet. The number of 0x5E in the previous packet is counted with a counter, which denotes the number of SBM bytes in the current packet. Since SBM is followed by next IP packet, the current IP packet can be decoded when we get its SBM in next packet. If there is 0x5E in the converted packet and the corresponding bit in SBM is '1', 0x7E is restored. Otherwise, the converted packet is outputted directly.

Fig. 2 shows an example of BMPF. There are two 0x7Es and one 0x5E in IP packet 1, so this packet's SBM is "11010000" (0xD0) and it is transferred at the beginning of the second framed packet. IP packet 2 has no 0x5E or 0x7E, so there is no SBM padded at the beginning of the third framed packet. When there are no packets to be sent, idle cells only including the flag 0x7E are inserted.

In BMPF, consecutive IP packets are merged into a stream and then segmented into cells together. A cell may carry bytes of different packets, and the process of padding useless bytes is avoided when IP packets arrive continuously under a heavy load.

3 Properties of BMPF

Property 1. In the worst case, the overhead of BMPF is about 14.29%, and for uniformly distributed random data the overhead is only 0.1116%.

Note that 0x7E and 0x5E need one marking bit in SBM and only seven bits of each byte in SBM are utilized. Let n be the packet size, and y_W be the overhead of BMPF in the worst case. We obtain

$$y_W = \left\lceil \frac{n}{7} \right\rceil, \tag{1}$$

where $\left\lceil \frac{n}{7} \right\rceil$ denotes the minimum integer that is not less than $\frac{n}{7}$.

Let y_R denote the mean overhead of BMPF for uniformly distributed random data. In an n-byte packet, the probability that any byte equals one of 0x7E and 0x5E is $\frac{2}{256}$, and the probability that there are i bytes to be stuffed is $\binom{n}{i}\left(\frac{2}{256}\right)^i\left(1-\frac{2}{256}\right)^{n-i}$, where $0 \le i \le n$. Then we get

$$y_R = \sum_{i=1}^{n}\binom{n}{i}\left(\frac{2}{256}\right)^i\left(1-\frac{2}{256}\right)^{n-i}\left\lceil \frac{i}{7} \right\rceil. \tag{2}$$

Obviously,

$$\frac{i}{7} \le \left\lceil \frac{i}{7} \right\rceil < \frac{i}{7}+1. \tag{3}$$

From (2) and (3), we obtain

$$\frac{2\times n}{256\times 7} \le y_R < \frac{2\times n}{256\times 7}+1-\left(1-\frac{2}{256}\right)^n. \tag{4}$$

From (1) and (4), we can get Property 1.

Property 2. BMPF is more efficient than PPP.

IP over PPP over SONET/SDH is a typical architecture in backbone networks [14], which makes the PPP protocol popular. PPP uses a byte-stuffing scheme, in which the byte 0x7E is the flag to demarcate the boundary between two consecutive packets. 0x7E in the data is encoded as two bytes 0x7D5E, and 0x7D is encoded as 0x7D5D. Whenever 0x7D appears in the receiver, the receiver discards 0x7D and XOR's the following byte with 0x20 to recreate the original byte.

In the worst case, an n-byte overhead is required for an n-byte packet to be framed into a PPP packet. I.e., the worst-case overhead is 100%. For uniformly distributed random data, the probability that any byte is one of 0x7E and 0x7D is $\frac{2}{256}$. The expected overhead for any n-byte uniformly distributed random data is

$$\frac{2}{256}\times n = 0.0078125n. \tag{5}$$

Fig. 3 compares the overhead of BMPF with that of PPP for uniformly distributed random data. The packet sizes are varied from 0 to 4000 bytes. The two curves are obtained from (4) and (5). We can see that only 5-byte overhead is required in BMPF when the packet size is up to 4000 bytes, and 31-byte overhead is needed in PPP. Obviously, BMPF is more efficient than PPP. The advantage of PPP is that packets can be immediately decoded at the receiver side. However, the fast decoding cannot speedup the reassembly process. In switches and routers, only when the last cell of an IP packet arrives at an output port, can a complete packet be reassembled.

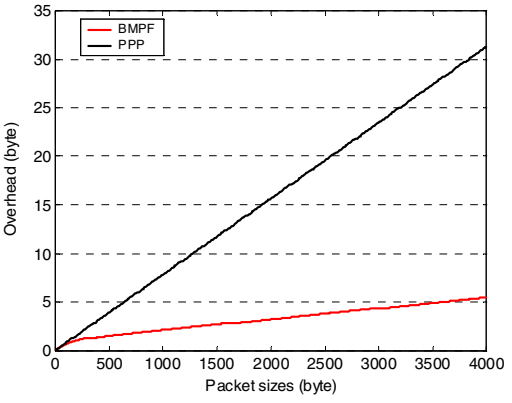


Fig. 3. Overhead comparison between PPP and BMPF

Property 3. BMPF is more efficient and practical than HDLC.
HDLC uses the bit-stuffing technology to eliminate the appearance of the flag byte in the encoded data. The byte 0x7E is also adopted as the boundary flag. Whenever five consecutive '1's are checked in a row, a bit '0' is inserted automatically. At the receiver side, the reverse process is performed: a bit '0' is automatically deleted when five consecutive '1's are met.

In HDLC, the overhead in the worst case is $\left\lfloor \frac{8n}{5} \right\rfloor$ bits for any n-byte data, where $\left\lfloor \frac{8n}{5} \right\rfloor$ denotes the maximum integer that is not greater than $\frac{8n}{5}$. I.e., the worst-case overhead is approximate 20% in HDLC.

For an n-byte uniformly distributed random data, the average stuffing overhead is $\frac{n}{62}$ bytes for a large n. The result is obtained at follows. In HDLC, the framing process can be characterized into six states: $\mathrm{S1}^i \left(0 \le i \le 5\right)$, where $\mathrm{S1}^i$ represents i consecutive '1's have been detected. Fig. 4 shows the state translation in the HDLC encoding process. The initial state is $\mathrm{S1}^0$, and the state $\mathrm{S1}^5$ is reached if and only if five consequent '1' are met in a row.

For uniformly distributed random data, both bit '0' and bit '1' occur with the same probability of 0.5. Therefore, these six states form a finite Markov chain, and the probability translation matrix is

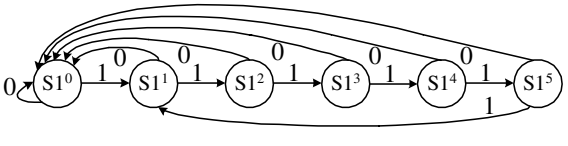


Fig. 4. State translation in HDLC

$$\begin{array}{c} \\ Sl^0 \\ Sl^1 \\ Sl^2 \\ Sl^3 \\ Sl^4 \\ Sl^5 \end{array} \begin{bmatrix} 0.5 & 0.5 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0.5 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0.5 & 0 & 0 \\ 0.5 & 0 & 0 & 0 & 0.5 & 0 \\ 0.5 & 0 & 0 & 0 & 0 & 0.5 \\ 0.5 & 0.5 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Let $P\left(\mathrm{Sl}^i\right)\ \left(0 \le i \le 5\right)$ denote the steady probability of the state Sl^i. From above translation matrix we can easily get

$$P\left(\mathrm{Sl}^i\right) = \begin{cases} 0.5 & i = 0 \\ \dfrac{2^{5-i}}{62} & 1 \le i \le 5 \end{cases}. \tag{6}$$

Only in the state Sl^5, one bit '0' is inserted, so the mean overhead is $\frac{n}{62}$ bytes for an n-byte packet. That is, the overhead is about 1.613% of the original packet size.

From Property 1 of BMPF, we know in BMPF the worst-case overhead is about 14.29% and the mean overhead for uniformly distributed random data is about 0.1116%, so we get BMPF is more efficient than HDLC.

HDLC is easy to implement in bit-oriented hardware. HDLC needs checking each bit of a packet and performing bit shifting. However, the basic processing unit in switches and routers is at least a byte. BMPF just needs to compare each byte with 0x7E and 0x5E, and perform byte shifting. Therefore, BMPF is more practical than HDLC in hardware implementation for switches and routers.

Property 4. BMPF achieves lower mean overhead for uniformly distributed random data and less encoding delay than COBS.

COBS is a counter-based high-efficiency framing method. COBS adds not more than 1 byte in each 254 bytes in the worst case, and the mean overhead is about 0.23% for uniformly distributed random data [13]. Table 1 summarizes the comparison among those referred framing methods. We obtain that COBS achieves better worst-case overhead than BMPF, but the mean overhead for uniformly distributed random data is two times of BMPF. Table 2 shows the actual overhead for packets in three cases that we randomly captured in Internet. From it we obtain that BMPF is similar to COBS, and both of them are better than PPP and HDLC. HDLC is the worst one among the four framing methods.

Table 1. Overhead comparison among BMPF, PPP, HDLC and COBS

Framing methods	Worst-case overhead	Overhead for uniformly distributed random data
BMPF	14.29%	0.1116%
PPP	100%	0.78125%
HDLC	20%	1.613%
COBS	0.4%	0.23%

Table 2. Overhead comparison among different methods with captured packets

Framing methods	Case 1	Case 2	Case 3
BMPF	0.1438%	0.1581%	0.1314%
PPP	0.8629%	0.9398%	0.7499%
HDLC	1.9873%	2.1110%	1.7146%
COBS	0.1320%	0.1775%	0.1615%

COBS needs scanning the content of a packet before encoding, which adds the encoding delay. COBS requires three operations of the input buffer. First, a packet is read from the input buffer and checked byte by byte. Then, the packet is written back into the buffer after encoding. Finally, the packet is read from the input buffer again for switching and scheduling. In BMPF, SBM is padded at the beginning of next packet, so a packet can be encoded immediately and just one reading of the input buffer is needed. Moreover, BMPF introduces no extra decoding delay because the decoding is performed at the same time of reassembling a packet.

Property 5. The minimum speedup to achieve full line rate in BMPF is much less than that in the traditional segmentation method.

When packets arrive consecutively, let ρ denote the switch bandwidth utilization in the worst case. In the traditional segmentation method, such as the segmentation technology deployed in the adaptation layer of ATM networks, we can obtain

$$\rho = \frac{x}{CL \times \left\lceil \frac{x}{CL} \right\rceil}, \tag{7}$$

where x is the packet size, and CL is the chosen cell size.

In BMPF, we obtain

$$\rho = \frac{x}{\left\lceil \frac{x}{7} \right\rceil + x + 1}. \tag{8}$$

From (8) we know that the switch bandwidth utilization ρ in BMPF is independent of the chosen cell size, and it is only related to the packet size x. This feature makes the switches with BMPF much scalable. To support higher line rate, we just need to increase the cell size to achieve enough scheduling time without affecting the switch bandwidth utilization.

Fig. 5 shows the switch bandwidth utilization in BMPF and the traditional segmentation method when CL is set to 64, 128 and 256 bytes. We can see three typical broken curves in the traditional segmentation method. Padding the last partially filled cell causes the switch bandwidth utilization varied greatly under different packet sizes. To achieve full line rate for packets larger than 40 bytes, the minimum speedup is at least two when CL is 64 bytes (such as for 65-byte packets). However, the curve in BMPF is much smooth, which means BMPF is less sensitive to variation of packet sizes. In BMPF the switch bandwidth utilization is improved to 85.1%. In other words, in BMPF the speedup of 1.175 (1/0.851) is enough to guarantee full line rate for all packets larger than 40 bytes in size.

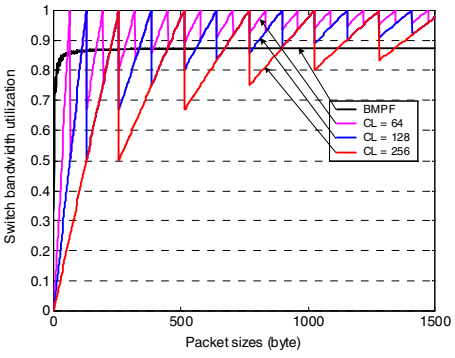


Fig. 5. Switch bandwidth utilization under different packet sizes

Property 6. In BMPF, fewer overheads in the lower link layer are required than that in the traditional segmentation method.

In the traditional segmentation method, IP packets will be padded to integral times of the chosen cell size. Overhead in the lower link layer is still required to identify the actual packet size and the boundary of each packet. In optical packet switches, more overheads in the lower layer are needed to demarcate boundaries of packets in one ultra-large cell. To deal with worst-case conditions, such as lots of short packets encapsulated in the same large cell, we must allocate enough extra overheads. In BMPF, however, there is a unique flag between any two consecutive packets. Therefore, it is not necessary for the lower link layer to mark the boundary of each packet further, and this reduces the overhead required in the lower link layer.

4 Hardware Implementation of BMPF

We develop hardware implementation architecture of BMPF in this section. Fig. 6(a) shows the encoding architecture of BMPF in each input port. When packets arrive at the encoder, the comparer module compares each byte of original packets with 0x7E and 0x5E. If the byte is 0x7E, the corresponding marking bit in the 7-bit register is set to '1', and if the byte is 0x5E, '0' is set. When all the seven bits of the register are set or the packet is encoded completely, the value of the register is written into the SBM FIFO, which stores one packet's SBM when encoding.

The multiplexer module reads data from either the incoming packets, or the SBM FIFO, or the fixed value 0x7E or 0x5E. The behavior of multiplexer module is controlled by the encoding state machine shown in Fig. 6(b). At the beginning, the multiplexer module is in the IDLE state. When a packet arrives, the state machine transfers to the FLAG state. The FLAG state transfers to the SBM state when the SBM FIFO is not empty, to the DATA state when the SBM FIFO is empty and some packets are waiting to be encoded at that time, and to the IDLE state otherwise.The SBM state transfers to the DATA state when the SBM FIFO is empty and to the FLAG state when there are no more packets to be encoded. The DATA state transfers to the FLAG state when one packet is transferred completely.

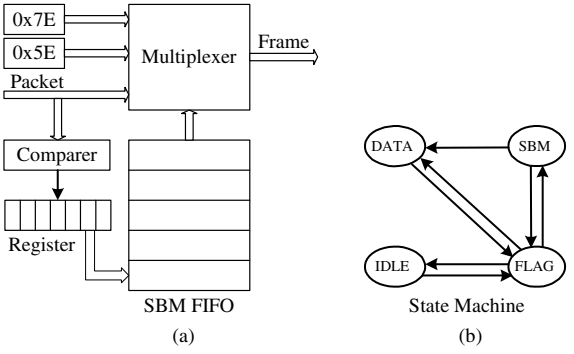


Fig. 6. BMPF encoding architecture

In the FLAG state, 0x7E is outputted. In the SBM state, the data in the SBM FIFO are outputted continuously until the FIFO becomes empty. In the DATA state, the incoming packet is outputted directly and all the 0x7Es in the original packet are converted to 0x5E.

Fig. 7 shows the decoding architecture of BMPF in an output port. When a packet arrives at the output port, a counter is used to record the number of 0x5E in the packet and to extract the SBM bytes by the demultiplexer module. The counter is reset to zero when meeting the flag 0x7E. The data field of a framed packet is buffered in the reassembly buffer. When both a packet and its SBM arrive, the reassembly control module reads the packet in the reassembly buffer and outputs it to the external output link. The 7-bit register is used to read marking bits from the SBM FIFO. When 0x5E is met and its marking bit in SBM is '1', 0x7E is outputted; otherwise, original data are outputted directly.

When unexpected bit errors occur due to random reasons, the decoding process of BMPF can recover to a correct state fast. This is because the decoding procedure is determined by two factors: the flag 0x7E and the counter value. When there are no more bit errors, the flag 0x7E can be recognized again. The counter value only impacts the number of the SBM in next packet and may cause one packet not to be extracted correctly. In a word, when bit errors occur in one packet, next packet may be broken too, but the decoding state machine will enter a right state and all the packets following next packet will be correctly decoded.

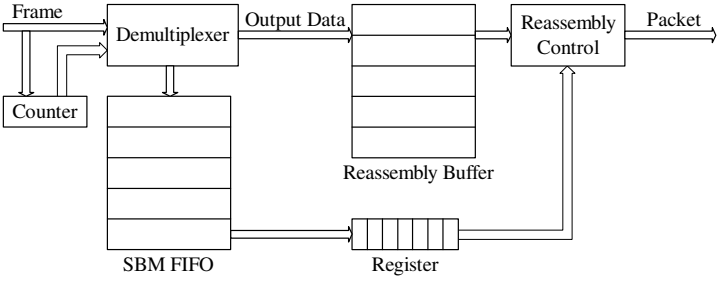


Fig. 7. BMPF decoding architecture

5 Conclusions

In this paper, we propose an efficient framing approach called BMPF to merge and segment variable-size IP packets for both input queueing switches and optical packet switches. BMPF overcomes the framing problem to merge IP packets from the viewpoint of switch bandwidth utilization. In BMPF, the speedup of 1.175 is enough to achieve full line rate, instead of at least two in the traditional segmentation method. BMPF is more efficient than the conventional framing method PPP, both in the worst case and for uniformly distributed random data. BMPF is more efficient and practical than HDLC. BMPF achieves less encoding delay than counter based framing methods, such as COBS. Finally, BMPF can be implemented in hardware at a very low cost.

References

1. Karol M.J., Hluchyj M.G., Morgan S.: Input versus Output Queueing on a Space-Division Packet Switch. IEEE Trans. Commun., Vol. 35, 12 (1987) 1347–1356
2. Anderson T.E., Owicki S.S., Saxe J.B., Thacker C.P.: High-Speed Switch Scheduling for Local Area Networks. ACM Trans. Computer Systems, Vol. 11, 4 (1993) 319–352
3. McKeown N., Izzard M., Mekkittikul A., Ellersick W., Horowitz M.: Tiny Tera: a Packet Switch Core. IEEE Micro, Vol. 17, 1 (1997) 26–33
4. Li W.J., Gong Y.P., Xu Y., Zheng K., Liu B.: Design and Implementation of a 320 Gb/s Switch Fabric. IEEE ICCNMC 2003, (2003) 143–148
5. Marsan M.A., Bianco A., Giaccone P., Leonardi E., Neri F.: Packet-Mode Scheduling in Input-Queued Cell-Based Switches. IEEE/ACM Trans. Networking, Vol. 10, 5 (2002) 666–678
6. Gripp J., et al.: Optical Switch Fabrics for Ultra-High-Capacity IP Routers. J. of Lightwave Technology, Vol. 21, 11 (2003) 2839–2850
7. Keslassy I., et al.: Scaling Internet Routers Using Optics. ACM SIGCOMM 2003, (2003) 189–200
8. Towles B., Dally W.J.: Guaranteed Scheduling for Switches with Configuration Overhead. IEEE INFOCOM 2002, Vol. 1, (2002) 342–351
9. Kar K., Stiliadis D., Lakshman T.V., Tassiulas L.: Scheduling Algorithms for Optical Packet Fabrics. IEEE. J. Sel. Areas Commun., Vol. 21, 7 (2003) 1143–1155
10. Bianco A., Giaccone P., Leonardi E., Neri F.: A Framework for Differential Frame-Based Matching Algorithms in Input-Queued Switches. IEEE INFOCOM 2004, Vol. 2, (2004) 1147–1157
11. Christensen, K., Yoshigoe K., Roginsky A., Gunther N.: Performance of Packet-to-Cell Segmentation Schemes in Input Buffered Packet Switches. IEEE ICC 2004, Vol. 2, (2004) 1097–1102
12. Simpson W.: PPP in HDLC-like Framing. IETF RFC 1662, (1994)
13. Cheshire S., Baker M.: Consistent Overhead Byte Stuffing. IEEE/ACM Trans. Networking, Vol. 7, 2 (1999) 159–172
14. Cisco Corp.: Cisco 12000 Series Routers Application Notes: Packet over SONET/SDH. http://www.cisco.com/

Measuring Internet Bottlenecks: Location, Capacity, and Available Bandwidth

Hui Zhou[1,3], Yongji Wang[1,2], and Qing Wang[1]

[1] Laboratory for Internet Software Technologies,
Institute of Software, Chinese Academy of Sciences, Beijing 100080, China
[2] Key Laboratory for Computer Science,
Institute of Software, Chinese Academy of Sciences, Beijing 100080, China
[3] Graduate School of the Chinese Academy of Sciences, Beijing 100039, China
{hzhou, ywang, wq}@itechs.iscas.ac.cn

Abstract. The ability to measure the location, capacity and available bandwidth of bottleneck in end-to-end network path is of major importance in congestion control, streaming applications, quality-of-service, overlay network and traffic engineering. Existing algorithms either fail to measure all the three bottleneck properties, or generate a large amount of probing packets. In addition, they often require deployment in both end hosts. A novel technique, called BNeck, is presented in this paper. It allows end users to efficiently and accurately measure the three bottleneck properties. The key idea of BNeck is that the per-link dispersion of probing packet train can be applied to measure the properties of congested links. The accuracy and efficiency of BNeck have been verified with elaborately designed simulation. The simulation result indicates that various applications can adopt BNeck to probe for the three bottleneck properties without loss of performance.

1 Introduction

Characterizing the bottleneck in end-to-end network path is a problem that has received considerable attention throughout the history of packet networks, in both research and practice [1], [2]. An end-to-end network path is composed of a sequence of store-and-forward links that transfer packets from source R_0 to destination R_n through routers R_1, $R_2 \ldots R_{n-1}$. Link $L_i=(R_i,R_{i+1})$ is the data connection between R_i and R_{i+1}. The three major properties of link are location, capacity and available bandwidth. The location of L_i is i, i.e. the hop count of L_i along the path. The capacity (C_i) of L_i refers to the maximum data-rate that L_i can achieve. The available bandwidth (A_i) of L_i is the residual bandwidth that isn't utilized by cross traffic. The *bottleneck*, also called bottleneck link, is the link with the smallest available bandwidth among all the links in the path.

The ability to measure the three bottleneck properties is of great assistance to both network operators and Internet Service Providers (ISPs) because these properties are crucial parameters in congestion control, streaming applications, quality-of-service, overlay network and traffic engineering. However, it is a very challenging task for

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 1052 – 1062, 2005.

end users to capture the bottleneck properties because the design of the Internet can't provide explicit support for them to obtain information about the network internals. Furthermore, what make the measurement difficult is that it generally demands knowledge of the properties of all links, and that the bottleneck varies with time due to unbalanced link capacities and changing load conditions.

A novel probing technique, called BNeck, is presented in this paper. It allows end users to accurately and efficiently measure the three bottleneck properties. The key idea of BNeck is to measure the three properties of congested links, instead of all links, using the per-link dispersion of a novel probing packet train. The train consists of many adjacent packets that travel from source to destination. Empirically, when the train traverses a link where the available bandwidth is less than the transmission rate of the train, the dispersion of the train, i.e. the time interval between the head and tail packets in the train, will increase. Formally, congested links are just the links that enlarge the dispersion. BNeck focuses on measuring the congested links since bottleneck is also a congested link that maintains the minimum available bandwidth.

The accuracy and efficiency of BNeck have been verified using simulation that is elaborately designed with bottleneck-determined factors (e.g. link capacity and traffic load). The simulation result indicates that various applications can adopt BNeck to obtain the three bottleneck properties without loss of performance.

This paper is organized as follows. Section 2 summarizes the related work about the measurement of bottleneck properties. Section 3 presents the probing packet train and Per-Packet Dispersion. Section 4 describes the design of BNeck in detail. Section 5 sets up simulation to verify BNeck. Finally, Section 6 concludes the paper.

2 Related Work

Recently, Hu addressed the problem of bottleneck location and presented a tool – Pathneck – to infer the location [3]. Pathneck relies on the fact that cross traffic interleave with probing trains along the path, thus changing the length of the packet train. BNeck takes his idea to smooth dispersion sequence, while our probing packet train is different from Hu's recursive packet train in structure and usage.

Jacobson proposed the single packet method for packet delay to measure link capacity [4]. It estimates the capacity by measuring the time difference of the round-trip time (RTT) to both ends of targeted link, and it is called the single packet method because it assumes that every probing packet travels independently from the source to the destination. Similarly, BNeck measures the capacity of targeted link by sending series of packets. However, the single packet method estimates per-link capacity, while BNeck only captures the capacity of targeted link.

The first tool that attempted to measure the bottleneck available bandwidth was Cprobe [1]. Its assumption is that the dispersion of probing packet train at the destination is inversely proportional to the available bandwidth. However, [5] showed that this is not the case. What the dispersion measures is not the bottleneck available bandwidth, but a different throughput metric that is referred to as the asymptotic dispersion rate (ADR). Recently, Jain and Dovrolis provided Pathload [6], a tool that measured the available bandwidth with periodic packet stream of different rates, but users couldn't benefit from Pathload since it require deployment in both end-points.

3 Probing Packet Train and Per-packet Dispersion

In order to measure the dispersion of packet stream in each link, a novel probing packet train is designed.

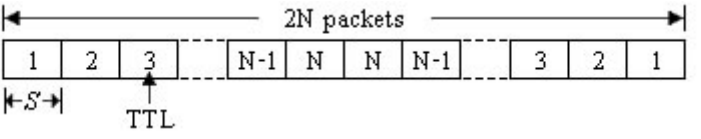


Fig. 1. Structure of the probing packet train

The train consists of 2N packets ($1 \le N \le 255$) that are all of the same size (S). As showed in Fig. 1, every box is a UDP packet and the number in the box is its Time-To-Live (TTL) value, which increments linearly from both ends to the center. This enables the train to measure network paths with up to N routers. Although the number of packets determines the overhead that the train will bring to the network, it also indicates the amount of cross traffic that the train should necessarily interact with [5]. Thus, it pays off to use a fairly long train.

The source sends the train in a back-to-back fashion, i.e. as close as possible, to the destination. When the train arrives at the first router, its head and tail packets expire, since their TTL values are 1. As a result, these two packets are dropped and the router sends two ICMP packets back to the source [7]. The other packets of the train are forwarded to the next router after their TTL values are decremented. Each subsequent router along the path repeats the above process. The source measures the time gap between the two ICMP packets from each router to estimate the dispersion of the train in the incoming link of that router. The dispersion in the last link can't be used because it is distorted by both the accumulated generation time for ICMP packets and the ICMP rate limiting mechanism of the destination.

Dispersion of the train in each link is proportional to the number of packets that the train maintains in that link. To represent the average distance between every two neighboring packets in the train, we introduce a metric: Per-Packet Dispersion (PPD). If the measured dispersion in L_i is Δ_i , then the PPD in L_i is p_i and

$$p_i = \frac{\Delta_i}{2 \cdot (N-i) - 1} \quad (0 \le i < n-1) \tag{1}$$

Additionally, let p_{-1} be the PPD of packets when they are sent out by the source. Since congested links enlarge the dispersion, they expand the PPD too. Moreover, PPD allows us to explore the relation between the train and the links. In an experiment, the train was applied to probe a 7-hop Internet path repeatedly. The path was under our supervision and connected hosts in ISCAS and GSCAS[1]. Packet traces of routers revealed that PPD was expanded proportionally if A_i was lower than the

[1] ISCAS is the Institute of Software, Chinese Academy of Sciences; GSCAS is the Graduate School of the Chinese Academy of Sciences.

rate of the train in L_{i-1} (r_p^{i-1}), while PPD remained almost unchanged if $A_i \geq r_p^{i-1}$ (Fig. 2). The trains were configured as those in Section 5.

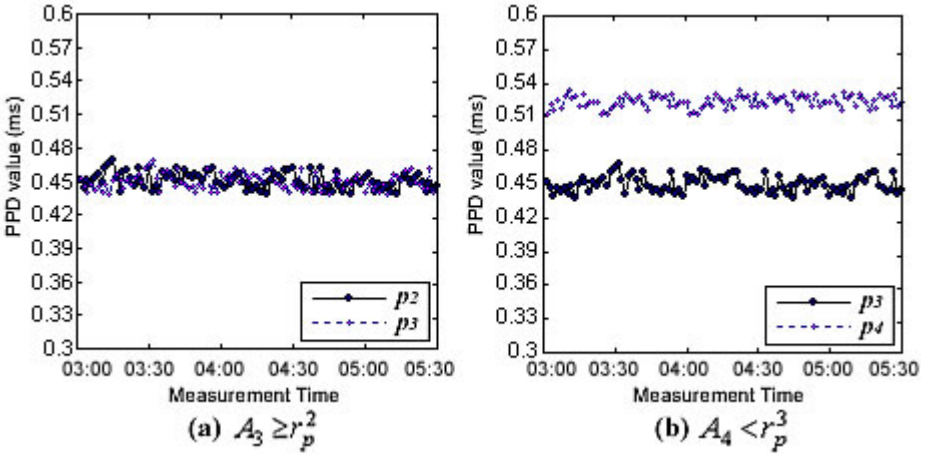


Fig. 2. PPD of probing packet trains in two cases

4 Design of BNeck

Based on the PPD, BNeck measures the bottleneck properties in four steps. First, it captures per-link PPD to locate the congested links. Second, it estimates the capacities of the congested links, while the available bandwidth of them are calculated in the third step. Finally, it identifies the bottleneck and outputs the three properties.

4.1 Locating the Congested Links

BNeck probes the network path once using the train, and collects the returned ICMP packets to calculate PPD in all links except for the last link. All PPD from a single probing, i.e. p_{-1}, p_0, $p_1 \ldots p_{n-2}$ ($n \geq 1$), make up of a PPD sequence.

Ideally, PPD increases if a link is not sufficient to sustain the train; or PPD stays the same if the link has enough space for the train. Therefore, PPD should never decrease. However, in reality, the cross traffic brings noise to the PPD sequence. BNeck has to process the sequence before locating the congested links.

As proposed by [3], BNeck first smoothes the PPD sequence by fixing the hill points and the valley points. A hill point is defined as p_2 in a three-point group (p_1, p_2, p_3) with PPD satisfying $p_1 < p_2 > p_3$. A valley point is defined in a similar way with $p_1 > p_2 < p_3$. The hill points and the valley points are replaced with the closest PPD (p_1 or p_3) since the short-term variance is probably caused by noise.

Then, BNeck locates the congested links by ranking the PPD sequence to identify PPD expansion. All PPD of the sequence are classified into ranks. Every rank contains at least one PPD. If two PPD belong to a rank, their intermediate PPD must

be of that rank. At the same time, PPD in any rank should satisfy $| p_1 - p_i | < threshold$, here p_1 is the first PPD in a rank; p_i is any other PPD in that rank.

As a result, the corresponding hops of the first PPD in all ranks are location of the congested links. Now, BNeck turns to measure the congested link.

4.2 Measuring Capacity of the Congested Links

Let D_i be the transmission delay needed for a packet to travel from R_0 to R_i. As illustrated in Fig. 3(a), C_i is the capacity of L_i; d_i is the fixed delay of L_i; S is the packet size; Q_i is the queuing delay of the packet in L_i. Then D_i is

$$D_i = \sum_{k=0}^{i-1} \frac{S}{C_k} + Q_k + d_k \tag{2}$$

In order to measure D_i, the source (R_0) generates a packet and transmits it to R_i after setting its TTL value to i. The packet will be dropped when it arrives at R_i. Meanwhile, R_i returns an ICMP packet to the source. In this way, the source captures the round trip time between R_0 and R_i to approximate $2D_i$, and then works out D_i.

A common observation with Kapoor [8] is held that queuing delay caused by cross traffic can only increase the transmission delay. Thus, among packets that are sent to R_i, the one that takes the minimum D_i may experience the least queuing delay. In this way, Q_k can be statistically eliminated from formula (2) using the minimum of many observed D_i of a particular packet size. In addition, adjacent probing packets are spaced by a time interval I to avoid interleaving. Generally, $I = \max \{p_i, -1 \le i < n-1\}$.

Furthermore, BNeck probes both ends of L_i, i.e. R_i and R_{i+1}, with packets of size S to get both the minimum D_i and D_{i+1}. The transmission delay for the packets in L_i without being queued ($DL_i(S)$) is then the function of S, and

$$DL_i(S) = \min(D_{i+1}) - \min(D_i) = \frac{S}{C_i} + d_i \tag{3}$$

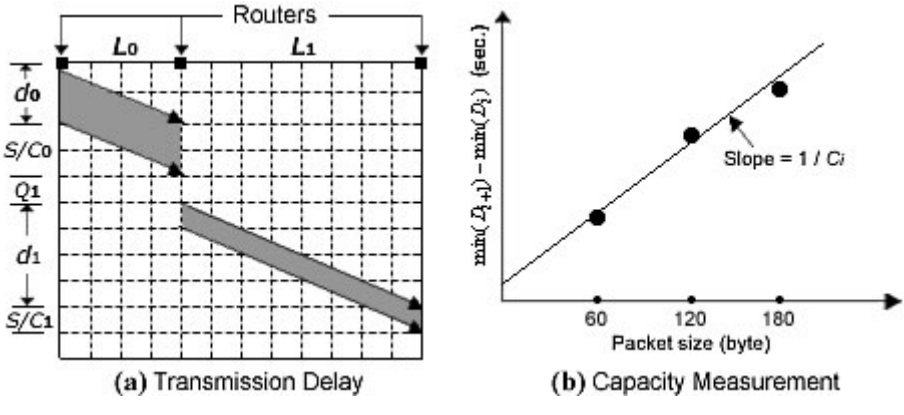


Fig. 3. Illustration of transmission delay and capacity measurement

Based on the above analysis, in order to obtain C_i, the source sends groups of packets for each of different packet sizes to both ends of the targeted link (L_i), plots $\min(D_{i+1}) - \min(D_i)$ of each group versus its size to draw the line of formula (3), and finally inverses the slope of the line to gain C_i. Fig. 3(b) illustrates a typical process.

4.3 Measuring Available Bandwidth of the Congested Links

The rate of cross traffic that travels in L_i (from R_i to R_{i+1}) is denoted as r_c^i. By definition, $r_c^i = C_i - A_i$. As supposed in [9], the cross traffic is evenly spread between every two neighboring probe packets. The amount of cross traffic that arrives at L_i in a p_{i-1} period is $X_i = r_c^i \cdot p_{i-1}$. Meanwhile, only one probing packet arrives at L_i, i.e. $S = r_p^{i-1} \cdot p_{i-1}$. The total amount of packets that L_i accepts during p_{i-1} period is $X_i + S$.

L_i is able to carry $C_i \cdot p_{i-1}$ amount of traffic in the p_{i-1} period. When $r_p^{i-1} \le A_i$,

$$X_i + S = \left(r_p^{i-1} + r_c^i\right) \cdot p_{i-1} \le \left(A_i + r_c^i\right) \cdot p_{i-1} = C_i \cdot p_{i-1} \tag{4}$$

It means that all the incoming packets can be transmitted by L_i without being queued, so PPD remains unchanged, i.e. $p_i = p_{i-1}$. When $r_p^{i-1} > A_i$, it indicates that L_i has to take more time to transmit all the incoming packets to R_{i+1}. In this case, L_i is a congested link and PPD is expanded, i.e. $p_i > p_{i-1}$.

$$p_i = \frac{X_i + S}{C_i} \tag{5}$$

A_i is inferred by substituting formula (5) with $X_i = r_c^i \cdot p_{i-1}$ and $r_c^i = C_i - A_i$,

$$A_i = \frac{S + C_i \cdot (p_{i-1} - p_i)}{p_{i-1}} \tag{6}$$

Since parameters in formula (6) are either configured (S) or have been measured in the preceding steps (p_{i-1}, p_i and C_i), A_i is calculated directly. Note that A_{n-1} is out of scope because the source can't capture p_{n-1} due to the reason stated in Section 3.

4.4 Identifying the Bottleneck

Finally, BNeck identifies the bottleneck that maintains the minimum available bandwidth among all the congested links, and then outputs its three properties.

5 Simulation Verification

The following simulation experiments verify the efficiency and accuracy of BNeck in Network Simulator (NS) [10], which is a reproducible and controlled network simulation environment. Since BNeck is a path-oriented probing technique, a linear topology is used (Fig. 4). Nodes 0 and 7 are the probing source and destination, while

nodes 1-6 are intermediate routers with agents attached to generate cross traffic. The link delays are roughly set based on a traceroute measurement from an ISCAS host to *www.gscas.ac.cn*. Link capacities are configured except that the capacities of L_2 and L_4 (X and Y) depend on scenarios. All links are duplex, and they are sufficiently buffered to avoid packet losses.

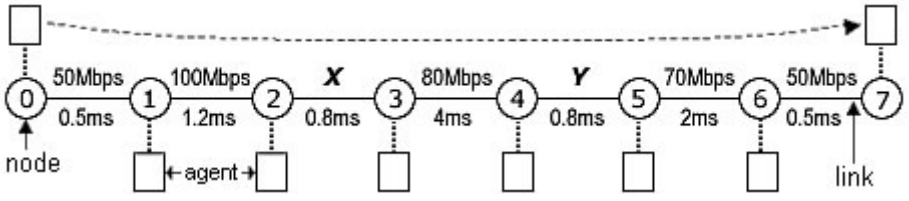


Fig. 4. Simulation topology

The data presented in the experiments is collected using a number of BNeck measurements where each measurement is identically configured as follows. The probing packet train consists of 50 packets that are all of 800 bytes. After locating the congested links with $threshold = 100\mu s$, two 300-packet groups are sent to both ends of each of these links to measure their capacities. Both groups consist of a hundred 60-byte packets, a hundred 120-byte packets and a hundred 180-byte packets. We define the ratio of available bandwidth that is matched as N_m/M. In an experiment, M is the number of all executed BNeck measurements, and N_m is the number of measurements where available bandwidth satisfy

$$\left|1 - \frac{m\_bottleneck_A}{r\_bottleneck_A}\right| \le \beta \tag{7}$$

Here, $m\_bottleneck_A$ is the bottleneck available bandwidth outputted by BNeck; $r\_bottleneck_A$ is the available bandwidth calculated from raw packet traces; $\beta = 5\%$. The ratio of capacity is defined as matched in the same way, while the capacity outputted by BNeck is denoted as $m\_bottleneck_C$.

5.1 Experiment 1: Capacity-Determined Bottleneck

A large fraction of bottlenecks fall into the capacity-determined category where the bottleneck is determined by the capacities, and the traffic load isn't heavy enough to affect the bottleneck properties. In order to generalize the result, this experiment is separated into two cases. In the first case, the Long Range Dependent (LRD) cross traffic is applied. While in the second case, the One Link Persistent (OLP) cross traffic predominates. In both cases, capacity is the key factor that determines the bottleneck. The traffic load, as an additional factor, may exhibit two extreme types, i.e. LRD and OLP [8]. Thus, the two cases are enough for evaluating the ability of BNeck to measure bottlenecks of this category.

In both cases, X is 50Mbps and Y is 30Mbps. In the first case, 20Mbps LRD cross traffic is produced by ten agents that generate Pareto traffic from node 1 to node 7 with $\alpha = 1.9$. The aggregation of many Pareto sources with $\alpha < 2$ has been proved to produce LRD traffic [11]. In the second case, 20Mbps CBR traffic is sent from nodes 1-5 to their next nodes. Thus, the cross traffic from nodes 1-5 will exit the path after traveling only one link. In this way, the OLP cross traffic is produced. In either case, L_4 is the bottleneck, C_4 = 30Mbps and A_4 = 10Mbps.

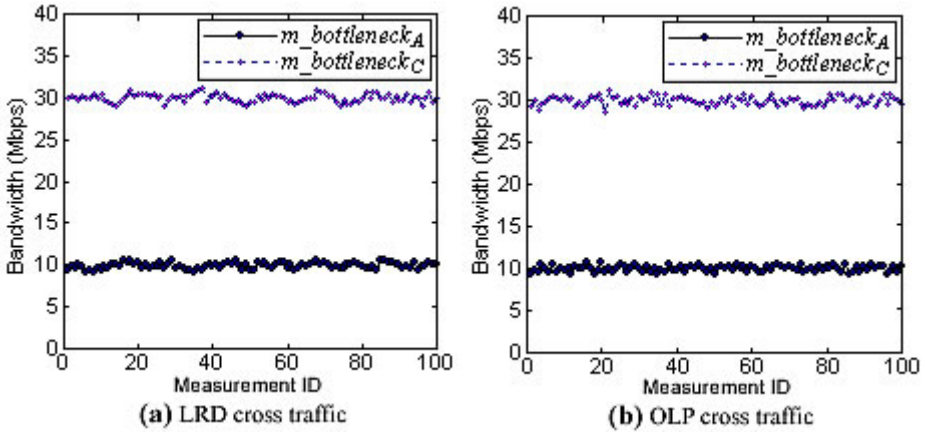


Fig. 5. Capacity and available bandwidth measurements in two cases

Both cases perform 100 BNeck measurements. In the first case, BNeck always outputs L_4 as bottleneck. Obviously, the result is 100% accurate. The 100 BNeck measurements in the second case output the same result, though they identify L_3 as congested link five times. Raw packet traces reveal that the ranking *threshold* is so conservative that L_3 is identified as congested link due to very small PPD variance. BNeck measurements in both cases output similar result of capacity and available bandwidth (Fig. 5). The capacity and available bandwidth are both 99% matched in the first case, while they are 96% and 94% matched in the second case, respectively.

The following experiments adopt the same setting (ten Pareto sources with $\alpha = 1.9$) when they produce LRD cross traffic.

5.2 Experiment 2: Load-Determined Bottleneck

Besides capacity, another factor that determines the bottleneck is the traffic load.

This experiment is configured as follows. X is 45Mbps and Y is 50Mbps. Node 1 sends 20Mbps LRD traffic to node 7. In addition, 15Mbps CBR traffic travels from node 4 to node 5, making the load of L_4 be heavier than that of other links. As a result, L_4 is the bottleneck, C_4 = 50Mbps and A_4 = 15Mbps.

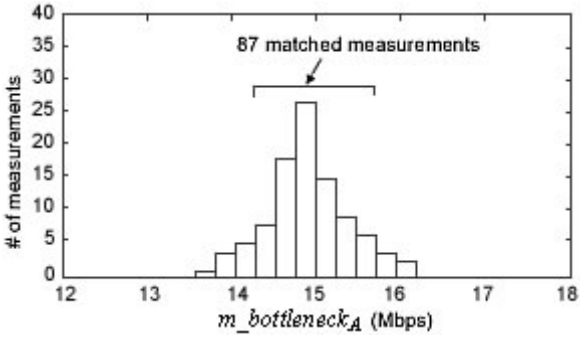


Fig. 6. Available bandwidth measurements (bin width = 0.25Mbps)

This experiment applies BNeck 100 times. L_4 is constantly identified as bottleneck, but L_1 and L_3 are wrongly identified as congested links twice each. Although the capacity of L_4 is 98% matched, the available bandwidth is only 87% matched (Fig. 6).

Compared with experiment 1, this experiment reports lower accuracy of available bandwidth (87%). Analysis of raw packet traces shows that the probing packet train competes un-fairly with the cross traffic. In L_4 that is 70% utilized, a few probings miss the cross traffic that should be captured, while some other probings capture the cross traffic that should be spread elsewhere, leading to slight fluctuation of the result.

5.3 Experiment 3: Reverse Path Traffic

In addition to the forward path cross traffic, the reverse path cross traffic also affects the accuracy of BNeck since it may cause queuing of the returned ICMP packets.

Let X be 30Mbps and Y be 20Mbps. Cross traffic in the reverse path is produced by sending 8Mbps LRD traffic from node 7 to node 1. In this case, L_4 is the bottleneck, C_4 = 20 Mbps, and A_4 = 20 Mbps because all links are duplex.

The experiment performs 100 BNeck measurements. The capacity is 98% matched and the available bandwidth is 92% matched. Meanwhile, as reported in Table 1, all measurements correctly output L_4 as bottleneck, but every link is identified as congested link at least once. This indicates that traffic in reverse path does affect the accuracy of BNeck, similar to the findings in [3]. The last link is not included because the source (node 0) can't measure A_6 due to the reason stated in Section 4.3.

Table 1. The number of times of each link being a congested link or bottleneck

Link	L_0	L_1	L_2	L_3	L_4	L_5
Congested Link	1	2	100	3	100	2
Bottleneck					100	

5.4 Measurement Time

Measurement time is the main indicator of the efficiency of BNeck. Table 2 lists the time that BNeck consumes in the above three experiments.

Table 2. Measurement time taken by BNeck in the three experiments

Experiment #		Min time (Sec.)	Average time (Sec.)	Max time (Sec.)
1	LRD	2.641	2.830	3.024
	OLP	2.686	2.893	3.272
2		2.649	2.874	3.116
3		2.633	2.817	2.972

In each experiment, a BNeck measurement averagely takes 2.81-2.89 seconds. In addition, the minimum time that it takes is 2.63-2.69 seconds; and the maximum time that it takes is 2.97-3.27 seconds. This is acceptable for end users to view the three bottleneck properties of a path. Analysis of raw packet traces reveals that the number of congested links mainly determines the measurement time. Thus, BNeck can be efficient enough for applications, e.g. multimedia streaming, peer-to-peer file sharing and network monitoring, to accurately probe for the bottleneck properties because the number of congested links in a single Internet path tends to be small [3].

6 Conclusion and Future Work

A novel technique, called BNeck, is presented in this paper. It allows end users to efficiently and accurately measure the three bottleneck properties. The simulation result indicates that BNeck can not only be used by end users, but also be applied to various applications without loss of performance.

This paper analyzes some features of the Internet bottleneck and many issues require further study, including the dynamic nature of the bottleneck, the impact of network topology and routing changes on the bottleneck. We also hope to improve BNeck by studying how configuration parameters such as the packet number, packet size, ranking threshold and transmission rate of the probing packet train affect the measurement accuracy and efficiency.

Acknowledgment

This work is supported by the National Natural Science Foundation of China under Grant No.60373053; One-Hundred-People Program of Chinese Academy of Sciences under Grant No.BCK35873; the Chinese Academy of Sciences and the Royal Society of the United Kingdom under Grant Nos.20030389, 20032006.

References

1. R. L. Carter and M. E. Crovella: Measuring bottleneck link speed in packet-switched networks. Perform. Eval., Vol. 27, No. 28, 1996, pp. 297-318.
2. V. Paxson: End-to-end Internet packet dynamics. IEEE/ACM Trans. Networking, Vol. 7, Jun. 1999, pp. 277-292.

3. N. Hu and L. Li: Locating Internet Bottlenecks: Algorithms, Measurements, and Implications. ACM SIGCOMM, Aug. 2004, pp. 41-54.
4. V. Jacobson: pathchar - a tool to infer characteristics of Internet paths. Presented in Apr. 97 MSRI talk, 1997.
5. C. Dovrolis, P. Ramanathan and D. Moore: What do packet dispersion techniques measure? In Proc. IEEE INFOCOM, Apr. 2001, pp. 905-914.
6. M. Jain and C. Dovrolis: End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation With TCP Throughput. IEEE/ACM Trans. Networking, Vol. 11, No. 4, Aug. 2003, pp. 537-549.
7. RFC 792. Internet control message protocol. Sept. 1981.
8. R. Kapoor: CapProbe: A Simple and Accuracy Capacity Estimation Technique. ACM SIGCOMM, Aug. 2004, pp. 67-78.
9. Y. Zhang and N. Duffield: On the constancy of Internet path properties. In Proc. ACM SIGCOMM Internet Measurement Workshop, Nov. 2001, pp. 197-211.
10. NS-2. http://www.isi.edu/nsnam/ns/
11. M. S. Taqqu and W.Willinger: Proof of a Fundamental Result in Self-Similar Traffic Modeling. ACM Computer Communications Review, Apr. 1997, pp. 5-23.

Experiment and Analysis of Active Measurement for Packet Delay Dynamics

Kai Wang, Zhong-Cheng Li, Feng Yang, Qi Wu, and Jing-Ping Bi

Institute of Computing Technology, Chinese Academy of Sciences,
6# KeXueYuan South Road, Zhongguancun, Haidian, Beijing 100080, P.R. China
Telephone: +861062565533 ext 9228
Fax: +861062567724
{wangkai, zcli, yf, abewu, jpingbi}@ict.ac.cn

Abstract. Active measurements have formed the basis for much of our empirical efforts to understand Internet packet delay dynamics. Packet-average performance of user flow is getting more and more important for users, and especially for network service providers. But in network active measurement area, there are not empirical efforts to investigate the performance of active measurement for packet performance from user's standpoint. We quantitatively assess and compare the one-way delay statistics experienced by user flow and active probe flow based on simulation experiments, and find that: (1) Active measurement systematically underestimates statistics of packet delay experienced by user flow and the estimation error is far severe than can be ignored. (2) Increasing sampling frequency is almost helpless for the reducing of the estimation error. (3)The estimation error degrees of active measurement decrease as the increasing of the queue utilization. The above conclusions are based on active measurements using Poisson sampling and Periodic sampling. As they are mainly used sampling methods in active measurement area, so our conclusions indicate that current active method for measuring Internet packet delay suffer from system errors from user's standpoint.

1 Introduction

Network measurement plays a more and more important role in the Internet society, for example, network monitoring and management, traffic engineering, validation of Service Level Agreement (SLA). Active measurement is one of the most important methods in network measurement area. It gathers network performance information by sending probing packets into the network actively. The benefit of active measurement is that it can be run almost anywhere in the network and it can easily estimate end-to-end network performance. Poisson sampling and Periodic sampling are the two most widely used sampling techniques for active measurement. Although there are some disadvantages which would affect the accuracy of measurement results, such as the synchronization with periodic events in the network [1], Periodic sampling is still widely used as it is simple and easy to use. Poisson sampling is a kind of random additive sampling, that is, the intervals between consecutive sampling instances are independent random variables which have the same exponential

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 1063 – 1072, 2005.

distribution. There are three advantages which make Poisson sampling to be a natural candidate for network measurement. The first is that Poisson sampling samples all instantaneous signals of measured process with the same probability [2]. The second is the characteristic of *Poisson Arrivals See Time Averages* (PASTA),which means the proportion of a measured state is asymptotically equal to the proportion of time the network spends on this state, in other words, Poisson sampling is also unbiased for the states it sampled [3]. The third is that it can avoid the potential network synchronization. As a result, the IP Performance Metrics (IPPM) working group of IETF recommends Poisson sampling in official documents [4] and many Internet measurement infrastructures perform measurements continuously by a Poisson-like style [2][5][6][7].

As Quality of Service (QoS) experienced by user flow is getting more and more attentions, and active measurement is often used to obtain this kind of performance in routinely measurements, it is important to know whether the active measurement results are unbiased with the packet performance of user flow. But there have been no published results that systematically study the performance of active measurement for packet performance from user's standpoint. Masaki Aida et al. indicate active measurement will underestimate average performances experienced by user flow with Poisson sampling [8], but they do not give detailed analysis and quantitative conclusions. As one-way packet delay is one of the most important metrics to characterize the packet performance of user flow, we investigate whether the one-way packet delay performance experienced by user flow and active probe flow are consistent in this paper as the first step. And our study is based on simulation experiments and use actual Internet traffic datasets as user flow. We find that both Periodic sampling and Poisson sampling underestimate all the concerned delay statistics systematically, including average and median delay, and the estimation errors of the two sampling methods are similar. We also find that increasing sampling frequency is almost helpless for reducing the estimation error; however the error degrees decrease as the increasing of the queue utilization for both methods. We reveal the reason of underestimation effect of Periodic and Poisson sampling is twofold. First, both sampling methods are uniformly distributed in time, however user traffic is bursty. Second, large arrival rate is likely companied with longer queue length and larger packet delay. That is, if Periodic and Poisson sampling are used to measure the packet delay performance of user flow with bursty characteristic, they will systematically underestimate the delay statistics of user flow.

Our work has implications for different areas in network engineering. First, our simulation experiments lead us to believe that new active sampling methods should be devised to get more accurate estimations of packet delay dynamics from user's standpoint. Next, network service providers should be aware of the underestimate effect of active measurement for one-way packet delay from user's standpoint and maybe need to reevaluate the active measurement results.

The rest of this paper is structured as follows: Section 2 describes the simulation topology and the Internet traffic datasets. Section 3 presents the simulation experiment results. Section 4 we present a qualitative analysis on the underestimate effect of both sampling methods. In section 5 we summarize our findings and discuss the future work.

2 Simulation Experiments

Internet packet delay mainly consists of propagation delay, transmission delay and queueing delay. With the quickly increasing of bandwidth and processor ability, the variation of transmission delay is usually small enough and can be ignored. Propagation delay is equal for all packets on the same path and can be also ignored for evaluating measurement error. As there often exist performance bottlenecks on the Internet paths, and the bottlenecks have a great impact on the end-to-end performance of packets. Cross traffic dynamics on the bottleneck is the dominant factor that influences packet queuing delay dynamics, and it is clear that unbiased packet delay measurements depends on unbiased queue delay measurements. Therefore, we focus on the queueing delay measurement error of active measurement in this section.

We use the LBNL Network Simulator, ns2 [9], as the simulation tool. A single-bottleneck network topology is adopted for the simulation, as showed in Fig.1. We consider the queueing system in router r1 with the following characteristics: single server, fixed service rate, infinite buffer. Poisson-style or Periodic-style probe flow is originated from node n1 and end up on node n3. User flow is originated from node n2 and end up on node n4. Probe flow and user flow share the common bottleneck, namely router r1. And we investigate whether the average delay and median delay of probe flow and user flow are consistent, and we also want to identify which factors will impact the performance of active measurements. As sampling frequency is important parameter and influences the measurement results in many cases, we also experiment and analyze the measurement results with different sampling frequency. Utilization is a very important parameter of network provision and it does influence the performance experienced by user flow. It is heuristic that utilization may also impact active measurement. Then we further study the relationship between the queue utilization and the performance of active measurement.

In order to make our simulation results more reliable, we use Internet traffic datasets as user flows in simulations, and the description of the datasets is showed in Table.1 [10][11][12]. In our experiments, the utilization of the queue in the bottleneck is calculated as $\rho = (\lambda_p + \lambda_u)/\mu$, where λ_p and λ_u is the average arrival rate of active probe flow and user flow respectively, μ is the link service rate. For a given traffic dataset, we adjust the bottleneck link service rate μ to obtain different utilizations of the queue in the bottleneck. And it is must be noted that μ shouldn't be larger than the bandwidth of the link where the traffic dataset was captured, otherwise there is no bottleneck in our simulation topology. As a result, the queue utilization is set within the range between 0.2 and 0.9. Probe flows are generated as follows. Poisson sampling flow is created through Exponential On-Off traffic model as suggested by ns2 Manual. And Periodic sampling flow is created by CBR traffic generator of ns2. As different sampling frequency can be represented by different rate of probe flow, we can obtain different sampling frequency by adjusting the proportion of the rate of probe flow to the rate of a given user flow. And the proportion ranges from 0.1%~1% in this paper. The size of probing packet is set to be 80 bytes, in order to limit the influence of measurement traffic on user flow.

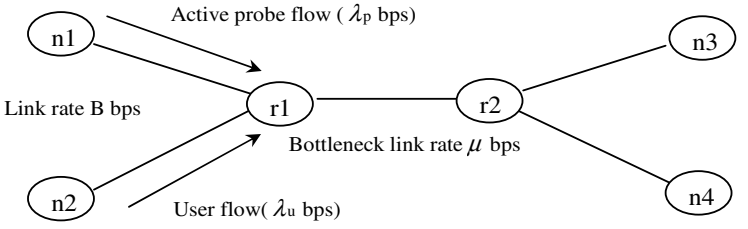


Fig. 1. Simulation Topology

Table 1. Internet traffic datasets used as user flow in the simulation

Traces	Source	Measurement time	Duration (second)	Packet number	Average rate (Mbps)	Link bandwidth (Mbps)
I	NLANR	2001-5-10 16:00	3600	2644442	1.732	10
II	NLANR	2001-6-9 12:00	21600	8928722	1.276	100
III	WIDE	2003-6-16 14:00	900	3408602	18.527	100
IV	WIDE	2003-6-17 14:00	900	3701086	19.34	100
V	LBNL	1989-8-29 11:25	3142.82	999999	1.105	10

3 Simulation Experiment Results

In this section, we describe the simulation results and compare delays experienced by probe flow with those of the user flow. We find no matter Poisson sampling or Periodic sampling is used to estimate the delay statistics, which include average and median delay, active measurement will systematically underestimate the true values, and the estimation errors of Poisson sampling are as approximately same as that of Periodic sampling . The simulation results are showed in Table 2.

The estimation error is defined as

$$\mathrm{err}_d = (d_m - d_u)/d_u , \tag{1}$$

where d_m is the delay statistics for active measurements and correspondingly d_u is for experiences of user flows. Table 2 shows that, for Poisson sampling, the estimation error is always negative within the range from 25% to about 35% for average delay and from 98% to 100% for median delay. For Periodic sampling, the estimation error is always negative within the range from 25% to about 32% for average delay and from 97% to 100% for median delay. Moreover, the estimation errors are almost equal with different sampling frequency for both sampling methods. In other words, changing sampling frequency is helpless for reducing the estimation errors for both sampling methods.

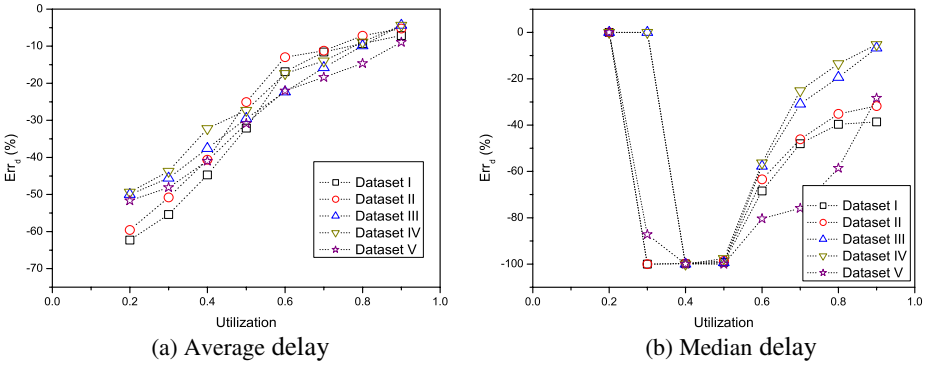
We further find that the underestimation effects of active measurement are weakened with the increasing of the queue utilization. Fig.2 and Fig.3 shows the estimation error of Poisson sampling and Periodic sampling with different utilizations respectively. The x-axis represents the utilization, and y-axis represents the estimation error of average or median queueing delay. From Fig.2, we can see the maximum estimation error of median queueing delay can even be up to 100% for Poisson

Table 2. Estimation error of active measurement to queueing delay statistics of user flow (utilization is 0.5, probe rate is of the proportion of 0.1%, 0.5%, 1% to user flow rate)

Traffic Traces			I	II	III	IV	V
Poisson sampling	Error for average delay (in percentage)	1%	-32.084	-25.071	-29.503	-27.31	-30.858
		0.5%	-32.167	-25.112	-30.275	-27.543	-31.09
		0.1%	-32.212	-25.263	-29.539	-27.371	-31.129
	Error for median delay (in percentage)	1%	-99.216	-98.621	-99.387	-97.653	-99.995
		0.5%	-99.312	-99.001	-99.403	-97.991	-99.83
		0.1%	-99.376	-99.021	-99.587	-98.027	-99.991
Periodic sampling	Error for average delay (in percentage)	1%	-31.916	-24.981	-30.124	-27.63	-31.382
		0.5%	-32.023	-25.172	-29.846	-27.991	-31.230
		0.1%	-29.184	-25.034	-29.052	-27.527	-31.279
	Error for median delay (in percentage)	1%	-100	-98.331	-99.387	-97.141	-99.995
		0.5%	-99.989	-98.983	-99.394	-96.794	-99.990
		0.1%	-98.725	-99.107	-99.401	-97.411	-99.377

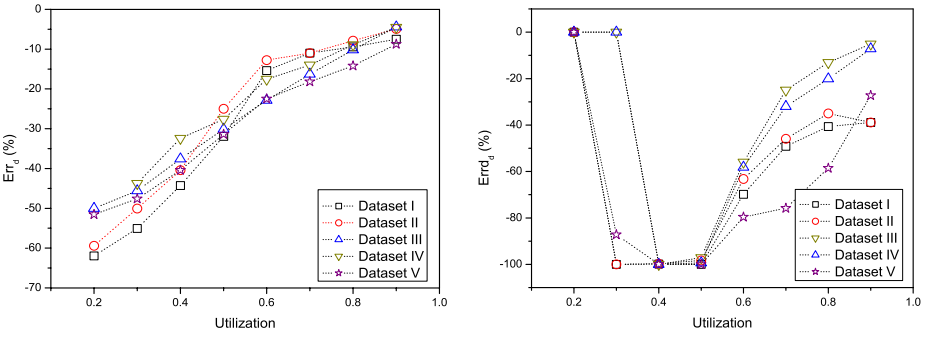
sampling, and there is a very interesting threshold phenomenon. If the queue utilization is no more than the threshold value, the estimation error is zero, and if the queue utilization is larger than the threshold value, the estimation error will achieve a maximum value in the beginning and then decrease as the queue utilization increasing. However, the threshold value differs with different traffic datasets, e.g. the threshold value is 0.4 for dataset III and is 0.3 for dataset I. This phenomenon can be explained as follows. When the queue utilization is small enough, most of the packets do not experience queueing delay. Even though active measurement does underestimate the queueing delay of user flow, the underestimation can't be revealed by median queueing delay. For example, suppose the proportion of packets with zero queueing delay in the user flow and active probe flow are 60% and 80% respectively, that is active measurement underestimates the delay experienced by user packets for about 33% relatively, but the median queueing delays of both user flow and active probe flow are zero yet. When the queue utilization is large enough, the underestimation effect of active measurements can then be revealed by median delay.

Fig.3 shows the similar phenomenon with that of Fig.2 and the estimation errors of Periodic sampling are approximate with that of Poisson sampling.



(a) Average delay (b) Median delay

Fig. 2. The relationship between Poisson sampling estimation error and the queue utilization



(a) Average delay (b) Median delay

Fig. 3. The relationship between Periodic sampling estimation error and the queue utilization

4 Why Does Active Measurement Fail

Since the first empirical evidence of self-similar or long-range dependence characteristics in network traffic was presented more than ten years ago [13], similar phenomenon were observed in wide aspects of network behavior [14][15]. Internet traffic datasets used in this paper also exhibit obvious bursty characteristics and even long-range dependence, for example, Fig.4 shows the autocorrelation functions of packet inter-arrival times for traffic dataset III can be characterized by typical power-law decay.

It is well known that long-range dependence in network traffic will induce scale-invariant burstiness. That is packets arrive in a bursty manner for wide range of time scale. Then it tend to be much more packets arrival when traffic burst than when traffic non burst, and these packets tend to experience larger queueing delays. From the standpoint of unbiased measurement of packet delay characteristics, more samples should be obtained when traffic burst than when traffic non burst, if the sampling period is equal. We analyze the underestimation effect of active measurement as follows.

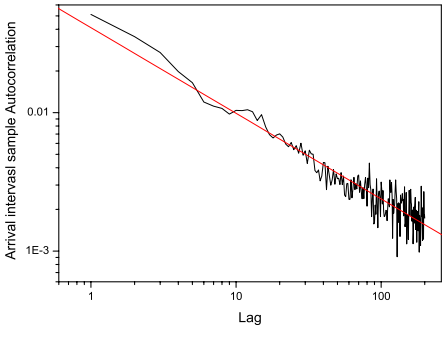


Fig. 4. Autocorrelation function of packet inter-arrival times for traffic dataset III

Let's use Poisson sampling for illustration. Suppose the measurement period is T. The queue utilization is a metric reflect load density and can be expressed as

$$\rho = \lim_{T\to\infty} (\int_0^T I(s)ds) / T \tag{2}$$

The indicators function $I(s) = 1$ or 0 when the server, i.e. the bottleneck output link in our simulations, is in busy state or idle state for instance s. According to (2), If T is large enough, and then ρ is the proportion of busy period in T. If a packet arrives when the server is busy, it must be buffered and will experience queueing delay. Define indicator function $I_p(n)$, $I_u(n)$ as follows, $I_p(n) = 1$ (or $I_u(n) = 1$) if the nth packet of Poisson sampling flow (or user flow) experiences queueing delay, otherwise $I_p(n) = 0$ (or $I_u(n) = 0$). And let

$$\rho_p = N_{pB} / N_p \tag{3}$$

$$\rho_u = N_{uB} / N_u \tag{4}$$

where N_p , N_u represent the total packet number of Poisson sampling flow and user flow in period T respectively, and $N_{pB} = \sum_{i=1}^{N_p} I_p(i)$, $N_{uB} = \sum_{i=1}^{N_u} I_u(i)$ represent the number of packet which experience queueing delay accordingly. It is clear that $\rho, \rho_p, \rho_u \le 1$. Poisson sampling packets arrive uniformly in time in statistical sense. And according to PASTA theorem [3], there is

$$\lim_{N_p\to\infty} \rho_p = \rho \tag{5}$$

Then ρ_p can be regarded to be equal to ρ if the number of sampling packet is large enough, namely the measurement period T is larger enough, and our simulating experiments in section 4 satisfy this qualification, as showed in Fig.5. The uniformity of Poisson sampling means that the number of sampling packet is equal in statistical sense given the length of period, however for user flow, the number of packets arrive when traffic burst will be much larger than that when traffic non burst. In the meanwhile, scale-invariant bustiness of traffic will amplify queue length. And then the proportion of packets which experience queueing delay will be larger than the proportion of busy time of the server, namely $\rho_u > \rho = \rho_p$. And this is why Poisson sampling underestimates packet delay characteristics. And when the number of sample is large enough, (5) is not influenced by increasing sampling frequency, which is why sampling frequency is helpless for reducing estimation errors in our experiments.

According to the above analysis, scale-invariant burstiness exist in user flow will increase queueing delay and also cause the proportion of packets which experience queueing delay larger than the proportion of busy time of the link. And then it causes the underestimation effect for Poisson sampling when delay statistics of user packet

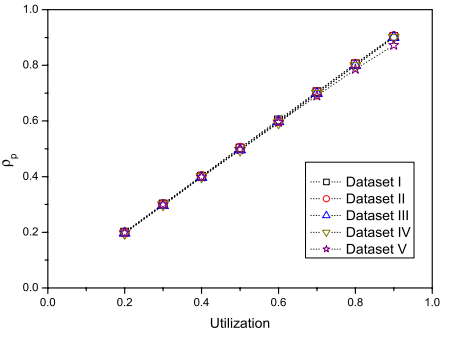


Fig. 5. Comparison of real utilization and ρ_p, which is calculated from Poisson sampling

are concerned. The estimation error of the two proportions decides the degree of underestimation of Poisson sampling. On the other hand, different utilization will influence the queueing behavior, and then the above proportions.

It is heuristic that. $\rho_u \to \rho, \rho_p$ with $\rho \to 1$. And more it is clear that the proportion of busy time of the queue will increase with the increase of the queue utilization. Considering simulations with utilization ρ_1 and ρ_2, $\rho_1 < \rho_2$ .Let $\rho_1 = t_B^1 / T$, $\rho_2 = (t_B^1 + \Delta t) / T$ where T represent the measurement period, t_B^1 represents the total busy time when the queue utilization is ρ_1 in period T, and Δt represent the incremental busy time relative to t_B^1 when the queue utilization is increased from ρ_1 to ρ_2. We mark ρ_p as ρ_p^1 (or ρ_p^2) and ρ_u as ρ_u^1 (or ρ_u^2) when the queue utilization is ρ_1 (or ρ_2). Due to (5), suppose that T is larger enough, and $\rho_p^1 = \rho_1$ and $\rho_p^2 = \rho_2$. We define the relative difference of ρ_p to ρ_u as

$$\mathrm{err}_\rho = (\rho_p - \rho_u) / \rho_u \tag{6}$$

And then

$$\triangle \mathrm{err}_\rho = \mathrm{err}_\rho^1 - \mathrm{err}_\rho^2 = (\rho_p^1 \rho_u^2 - \rho_p^2 \rho_u^1) / \rho_u^1 \rho_u^2 \tag{7}$$

According to (3) and (4), let $\rho_u^1 = N_{uB}^1 / N_u$, $\rho_u^2 = (N_{uB}^1 + \triangle N) / N_u$, where $\triangle N$ represents the incremental number of packets which experience queueing delay when the queue utilization is increased from ρ_1 to ρ_2. Then

$$\triangle \mathrm{err}_\rho = (t_B^1 \triangle N - \triangle t N_{uB}^1) / A \tag{8}$$

where $0 < A = \rho_u^1 \rho_u^2 N_u T$. Let λ_{uB}^1 represents the average arrival rate of packets which experience queueing delay when the queue utilization is ρ_1. From above

description, we can learn that there are a proportion of packets which do not experience queueing delay when the queue utilization is ρ_1, however, when the queue utilization is increased to ρ_2, they do experience queueing delay. The number of these packets is $\triangle N$. Let $\lambda_{\triangle N}$ represents the average arrival rate of these packets. It can be induced that $\lambda_{\triangle N} < \lambda_{uB}^1$, and then

$$\triangle err_\rho = (t_B^1 \triangle t \lambda_{\triangle N} - t_B^1 \triangle t \lambda_{uB}^1) / A < 0 \tag{9}$$

Combined with the fact that err_ρ is negative, Equation (9) means that ρ_p will approaches to ρ_u when the queue utilization ρ approaches to 1. According to the relationship between ρ_p and ρ_u and the underestimation effect of Poisson sampling, in other words, increase of the queue utilization will weaken the underestimation effect.

Although Periodic sampling has deterministic sampling interval, if the start time is randomized, then the predictability can be avoided [16]. And if periodic sampling does not synchronize with measured process, it will also have the same property of (5). Then why Periodic sampling underestimates delay statistics of user flow is similar with that of Poisson sampling, Periodic sampling flow also obtains fewer samples which experience queueing delay in proportion comparing with that of user flow. And the reasons why increasing sampling frequency is almost helpless for reducing the estimation error and why estimation error degrees decrease as the increasing of the queue utilization are similar with the explanations for Poisson sampling.

5 Summary and Future Work

In this paper we first quantitatively analyze how active measurement performs when it is used to evaluate user packet delays by simulation experiments. We find active measurement systematically underestimates the delay statistics of user flow, no matter whether Poisson sampling or Periodic sampling is used. And the estimation errors of the two sampling methods are similar; it can exceed 30% for average delay and even up to 100% for median delay. We also find that increasing sampling frequency is almost helpless for the reducing of the estimation error, and the estimation error decreases as the increasing of the queue utilization. We reveal the reason of underestimation effect of active measurement is twofold. First, Poisson sampling and Periodic sampling are evenly distributed in time, however user traffic is bursty. Second, large arrival rate is likely companied with longer queue length and larger packet delay. That is, if active measurement is used to measure the packet delay performance of user flow with bursty characteristic, it will systematically underestimate the delay statistics of user flow with Poisson sampling or Periodic sampling. Our work has implications for different areas in network engineering, such as packet delay dynamics measurement from user's standpoint, network operations and management.

Our future works will focus on the following aspects.

1) Our simulation experiments show that the degree of estimation error is different for various traffic datasets, even when the queue utilization is equal. That means the traffic characteristics maybe another factor to influence the degree of estimation error of active measurement, such as degree of bustiness. We believe it needs to be further investigated.

2) As the performance measurement from user's standpoint is more and more important, and Poisson sampling and Periodic sampling both are biased, we should further propose a better method for this kind of measurement based on this work.

3) More performance metrics will be considered in the future, such as packet loss, delay jitter etc.

Acknowledgements

We would like to thank YingHua Min, Jing Huang and XiaoLu Huang for their valuable comments and constructive criticisms.

References

[1] S. Floyd and V. Jacobson, "The Synchronization of Periodic Routing Messages," IEEE/ACM Transactions on Networking, 2(2), pp. 122-136, April 1994.
[2] V. Paxson, "Measurements and Analysis of End-to-End Internet Dynamics," Ph.D. dissertation, U.C. Berkeley, 1997.
[3] Wolff, R.W., "Poisson Arrivals See Time Averages," Operations Research, vol 30, 223-231, 1982.
[4] V. Paxson., G. Almes, J. Mahdavi and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, May 1998.
[5] Surveyor Project, http://www.advanced.org/surveyor/
[6] RIPE NCC Test Traffic Measurements, http://www.ripe.net/ripencc/mem-services/ttm/
[7] Active Measurement Project (AMP), http://amp.nlanr.net/
[8] Masaki Aida, Naoto Miyoshi and Keisuke Ishibashi, "A scalable and lightweight QoS monitoring technique combining passive and active approaches," In Proceedings of IEEE Infocom , San Francisco, CA, April 2003.
[9] The Network Simulator ns2. http://www.isi.edu/nsnam/ns/.
[10] Auckland-VI trace archive, http://pma.nlanr.net/Traces/long/auck6.html .
[11] MAWI Working Group Traffic Archive. http://tracer.csl.sony.co.jp/mawi.
[12] The Internet Traffic Archive. http://ita.ee.lbl.gov/.
[13] W.E.Leland, M.S.Taqqu, W.Willinger, and D.V.Wilson. "On the self-similar nature of Ethernet traffic (extended version)". IEEE/ACM Transactions on Networking, 2:1-15,1994.
[14] Vern Paxson and Sally Floyd, "Wide-area traffic: The failure of Poisson modeling" IEEE/ACM Trans. Networking, vol. 3, pp. 226-244, 1995.
[15] M. E. Crovella and A. Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes," IEEE/ACM Trans. Networking, vol. 5, no. 6, pp. 835-846, 1997.
[16] V. Raisanen, G. Grotefeld, A. Morton, "Network performance measurement with periodic streams". RFC 3432, November 2002.

A New Steganalytic Algorithm for Detecting Jsteg

Mingqiao Wu[1,2], Zhongliang Zhu[2], and Shiyao Jin[1]

[1] School of Computer, National University of Defense Technology,
410073 Changsha, China
wumingqiao1021@sohu.com
[2] Key Lab, Southwest Institute of Electron & Telecom Techniques,
610040 Chengdu, China

Abstract. Jsteg is an open steganography software on Internet. It uses the LSB of DCT coefficients to hide secret information. This paper presents a new, fast steganalytic method for detecting Jsteg hiding which is more convenient than the Chi-square attack. The AC coefficients of image are divided into two parts and the distribution of the statistic of the two parts is fitted by Laplacian. The mean of Laplacian is 0 that is proved by Hypothesis testing. The Pearson χ^2 test is done to test goodness-of fit. Basing on this statistic mode, our algorithm can detect Jsteg hiding with high accuracy. The amount of embedding message can be estimate using linear regression.

1 Introduction

Steganography is the art of invisible communication. Its purpose is to hide the very presence of communication by embedding messages into innocuous-looking cover objects such as digital documents, image, video, and audio files. Why the multimedia files are often used as cover objects relies on two factors. First, there is redundancy in them; second, the human perceptual system such as visual system, audio system, has some mask character. The watermarking technology using character of human perceptual system has been studied by many researchers [1]. The steganographic algorithm is evaluated by its imperceptiveness, robustness and high capacity. Today there is hundreds of steganorgraphic software available on Internet. Most of them use digital images as the cover objects. Steganographic algorithms based on image can be divided into two categories: algorithms in spatial domain and algorithms in transform domain.

The art and science of detecting the existence of steganographic data (and possibly reveling the secret message) is called steganalysis. Steganography and steganalysis of digital image is a cat-and-mouse game. In recent years, many steganalysic mehtods have been published. We classify them into four categories. The first category methods are visual attacks that use the human ability to distinguish the stego-image [2]. The second category is methods based on signatures [3]. These methods detect the existence of hidden message by looking for obvious and repetitive patterns which may point to identification or signature of a steganography tool. Johnson [3] points some steganography tool for pallette image will leave some signatures such as Hide&Seek, S-tools, MandelSteg, StegDos. These methods are based on the deep analysis of the specific hiding algorithms, so their ability of detection and extensibil-

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 1073 – 1082, 2005.

ity are limited. The third category is algorithms based on statistical features. These algorithms detect one or one class steganography by using some statistic of image. Westfeld [2] proposes the Chi-square attack to detect the sequential LSB embedding in pixels or DCT coefficients of image. The color pairs method proposed by Fridrich [4] is effective to detecting the embedding in high-color-depth image. When the number of unique colors is more than 1/2 of the number of pixels, the results of the method may become unreliable. Fridrich [5] proposes the RS method to detect the LSB embedding in spatial domain of image. She also proposes a higher-order statistical steganalytic method called Pairs Analysis for palette images [6]. In paper [7], she introduce a steganalysis based on JPEG compatibility detecting steganography in image that are originally stored in the JPEG format. Niimi [8] studies the security of BPCS-Steganography by analyzing the histograms of color components and luminance. Harmsen[9] and Fridrich [10] each proposes a method for detection of F5 steganograhy in JPEG images. In paper [11], Trivedi proposes method that can be used to detect location and length of messages embedding using spread spectrum steganography. Fridrich [12] proposes method to attack OutGuess. The fourth category is the universal blind detection algorithms. Fraid[13] [14] proposes such a detection algorithm based on high-order statistics of image that can be appled to any steganographic scheme after proper training on databases of original and steg-images. Avcibas [15] proposes a blind steganalysic algorithm using image quality metrics. The classifier between cover and stego-images is build using multivariate regression on the selected quality metrics. It is reported that the steganalytic methods to one or one class steganography are more powerful and exact than the universal blind algorithms.

Recently, the JPEG format attracted the attention of researchers as the main steganographic format due to the following reasons: It is the most common format for storing images, JPEG images are very abundant on the Internet bulletin boards and public Internet sites, and they are almost solely used for storing nature images. Upham [16] proposes the Jsteg algorithm as a high capacity JPEG stegnography. Pfitzmann and Westfeld introduce a detection method of Jsteg based on statistical analysis of Pairs of Values (PoVs). It uses the Chi-square test to determine whether the test image is stego-image. Their method is effective to the sequential Jsteg. When the message-carrying DCT coefficients of the image are selected randomly rather than sequentially, this test becomes less effective. Furthermore, every detection need multitimes of Chi-square tests, it is time consuming. In this paper, we propose a steganalytic algorithm for sequential Jsteg and random Jsteg based on the statistic of DCT coefficients of images. We study on the gray images, the developing to color images is the next work. The algorithm can estimate the amount of hidden message in image.

In the next Section, we introduce the Jsteg algorithm. In Section3, we describe the statistical mode of image DCT coefficients that is the base of our algorithm. In Section 4, we describe our algorithm. In Section 5,we present the experimental results. We make conclusion in Section 6.

2 Jsteg Algorithm

Jsteg proposed by Derek Upham [16] replaces the least significant bits (LSB) of the DCT coefficients by the secret message after quantization. The embedding mechanism skips all coefficients with the values 0 or 1. We describe it as fellow.

Let $C=\{C_0,C_1,\ldots,C_{n-1}\}$ represent the set of all the DCT coefficients. Chose a subset of C, $S=\{C_{l_0},C_{l_1},\ldots,C_{l_{(m-1)}}\}$, $(m\le n, l_{(m-1)}<n)$. To all the elements in S which is neither 0 nor 1, do replacement $LSB(C_{l_i})=M_i$, where $LSB(C_{l_i})$ represents the LSB of C_{l_i}, M_i represents the message. The sequential Jsteg proposed by Upham choses S from C as fellow: $\forall i, 0\le i\le m-1$, there is $l_i=i.$

The sequential Jsteg is simple and achievable but is not safe. The statistic of the modified part of the image is different from that of the unmodified part. Pfitzmann and Westfeld have proposed detection method based on this. Another Jsteg algorithm is random Jsteg. A Pseudo-random sequence $k_0,k_1,k_2,\ldots,k_{m-1}$ is generated. Let $l_0=k_0$, then $l_i=l_{i-1}+k_i, 1\le i\le m-1$. Adjust the Pseudo-random sequence to let the elements of S randomly scatter in C. The receiver has the seed to generate the same Pseudo-random sequence to retrieve the embedded message.

Considering the integral set $G=\{G_0,G_1,\ldots,G_{n-1}\}$, we denote $\|G\|$ the number of elements in G, $h_i(G)$ the number of elements whose value is equal to i. The max capacity of two Jsteg algorithm is $\|C\|-h_0(C)-h_1(C)$.

3 Statistical Model of Image DCT Coefficients

JPEG is the international standard of digital compression and coding of continuous-tone still images. In the algorithm, the image is first divided into nonoverlapping blocks of size 8×8 pixels, where each block is then subjected to the discrete cosine transform (DCT) before quantization and entropy coding. The distribution of the DCT coefficients has been studied by many researchers. Figure 1 shows a standard image "bridge" and it's plot of the histograms of the DCT coefficients.

Reininger and Gibson [17] use Kolmogrov-Smirnov tests to show that most DCT coefficients are reasonably well modeled as Laplacian. Muller [18] shows that modeling the DCT coefficients with the generalized Gaussian distribution results in a significantly smaller test statistic χ^2 compared with Laplacian. Lam [19] offers a mathematical analysis of the DCT coefficient distributions and demonstrates how a Laplacian distributin of the coefficients can be derived by using a doubly stochastic model. Barni [20] demonstrates that full frame DCT coefficients can be effectively modeled by a Laplacian density function.

Westfeld has found that the Jsteg algorithm will change the statistic of DCT coefficients. We divided the image into blocks, then calculate the histogram of all the ac coefficients over the blocks. Fig 2 is the plot of the histogram of ac coefficients. Fig 2(a) is the standard image "boat", Fig 2(b) is the original histogram of all AC coefficients in blocks. Dueing the symmetrical distribution of every AC coefficient, the distribution of all AC coefficients based on blocks is also symmetrical. Jsteg algorithm will change the symmetry of AC coefficients. Fig 2(c) is the histogram of AC

coefficients after embedding 3.2K bytes by Jsteg. The symmetry of distribution has been changed. After embedding more bytes, Fig2 (d) shows that distribution become more dissymmetric.

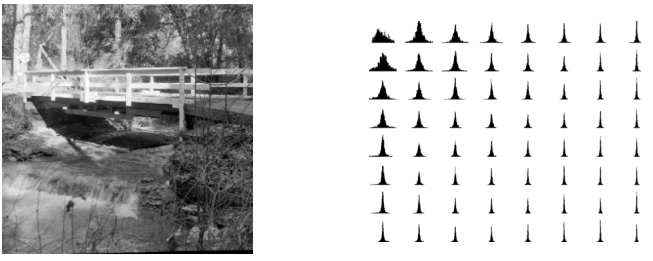


Fig. 1. (a) Standard image "bridge" (b) Histogram of DCT coefficients of "bridge"

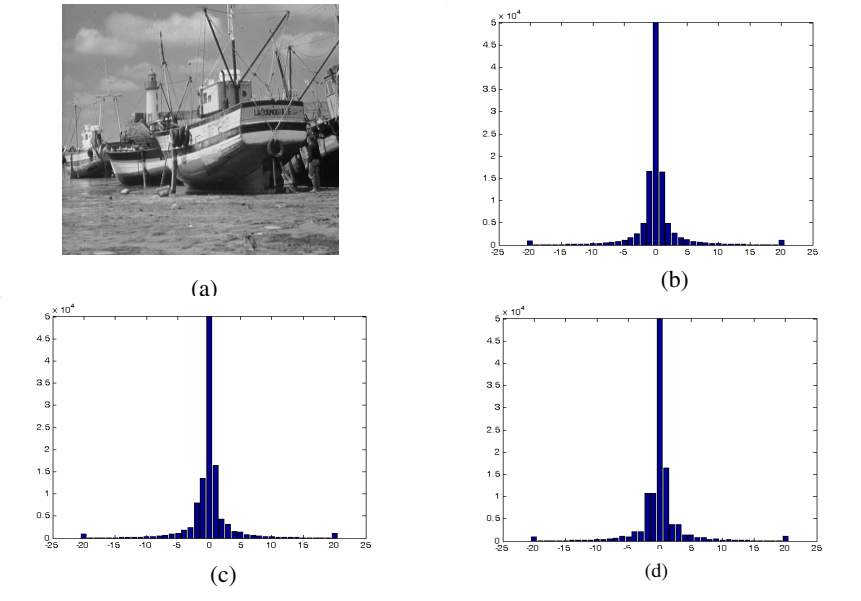


Fig. 2. (a) Standard image "boat". (b) Original histogram of AC coefficients (after quantization) of image "boat"; (c) Histogram of AC coefficients (after quantization) of image "boat" with Jsteg embedding 3.2K bytes; (d) Histogram of AC coefficients (after quantization) of image "boat" with Jsteg embedding 6.3K bytes

Based on the distribution of AC coefficients, we propose a statistical model. We denote $C = \{C_0, C_1, \ldots C_{n-1}\}, n = M \times N - (M \times N / 64)$ as the set of AC coefficients of image with size $M \times N$. Let $f_0(C)$ be the number of elements in

C whose value is either positive even or negative odd. Let $f_1(C)$ be the number of elements in C whose value is either positive odd or negative even. That is:

$$f_0(C) = \sum_{i>0, i \bmod 2 \equiv 0} h_i(C) + \sum_{i<0, i \bmod 2 \equiv 1} h_i(C) \tag{1}$$

$$f_1(C) = \sum_{i>0, i \bmod 2 \equiv 1} h_i(C) + \sum_{i<0, i \bmod 2 \equiv 0} h_i(C) \tag{2}$$

Neither $f_0(C)$ nor $f_1(C)$ includes the 0 elements of C. We construct a statistic

$$x = 2(f_1(C) - f_0(C))/(f_1(C) + f_0(C)) \tag{3}$$

to discriminate the stego-images from the clear images. Because of the symmetrical distribution of the AC coefficients, we have $f_0(C) \approx f_1(C)$. So x should be around 0 for clear images. We suppose that statistic x may satisfy some statistical distribution with 0 mean. We calculate the value of x for thousands images. The images with different sizes are downloaded from UCI image database, KODAK company or pictured by ourselves using digital cameral and changed to gray. Fig 3 is the plot of the density distribution of x.

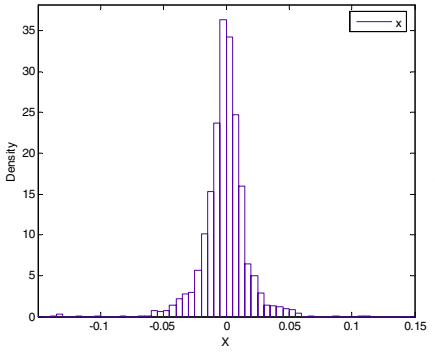


Fig. 3. The density distribution of x

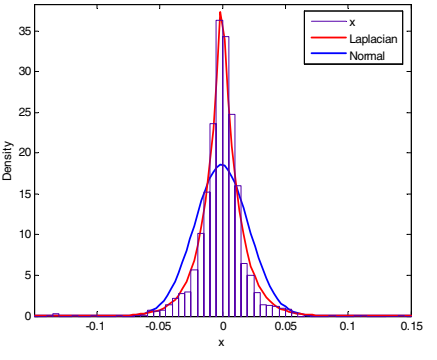


Fig. 4. Using Laplacian distribution and Normal distribution to fit the probability density distribution of x

We use the Laplacian distribution and Normal distribution to fit the distribution of x respectively. The probability density function of a Laplacian distribution can be written as

$$p(x) = \frac{1}{2\sigma} \exp\left\{ -\left| \frac{x-\mu}{\sigma} \right| \right\}. \tag{4}$$

The probability density function of a Normal distribution can be written as

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-(x-\mu)^2 / 2\sigma^2\right\} \tag{5}$$

Fig 4 is the distribution fits for x. From it we can see that the Laplacian distribution fits more well than the Normal distribution.

Pearson χ^2 Goodness of Fit Test

We use the Pearson Chi-square test [21] to test the goodness of fit of Laplacian distribution and Normal distribution. For a sample set $X = (x_1, x_2, \ldots x_n)$, it is used to test the hypothesis

H_0 : the distribution function of X is $F_0(x)$;

H_1 : the distribution function of X is not $F_0(x)$.

Here $F_0(x)$ is the Laplacian distribution and Normal distribution. The statistic of Pearson Chi-square test is χ^2

$$\chi^2 = \sum_{i=1}^{k} \frac{(n_i - np_i)^2}{np_i} \tag{6}$$

where k is the number of groups that the samples are divided into, n_i is the number of sample fall into the group i; p_i is the theory frequency of group i; n is the total number of samples. The results of the tests are exposed in Table 1.

Table 1. χ^2 test of Laplacian distribution and Normal distribution (significance level $\alpha = 0.05$)

distibution	$\hat{\mu}$	$\hat{\sigma}$	groups	$\chi^2 = \sum_{i=1}^{k} \frac{(n_i - np_i)^2}{np_i}$	$\chi^2_{1-\alpha}$	accept H_0
Laplacian	-0.000257	0.012781	63	76.2	78.8	yes
Normal	-0.000217	0.0251611	63	657.7	78.8	no

From Table 1 we can see that the distribution of statistic x is Laplacian but not Normal.

Test for Zero Mean

The statistic x is symmetric. We use the mean test [21] of nonnormal distribution to test the hypothesis:

$$H_o : E(X) = 0; \qquad H_1 : E(X) \neq 0.$$

The statistic of the mean test is

$$\lambda = \frac{\overline{X}}{S / \sqrt{n-1}} \tag{7}$$

where S is the standard deviation of X, n is the number of samples. When n is large enough, λ is $N(o,1)$ distribution. For statistic x, we have (the significance level $\alpha = 0.05$)

$$-1.96 = u_{\alpha/2} < \lambda = -0.38 < u_{1-\alpha/2} = 1.96$$

$u_{\alpha/2}$ is the point at which the value of cumulative distribution function of $N(o,1)$ is $\alpha/2$. λ falls into the confidence interval so the H_0 hypothesis is accepted.

4 Steganalytic Algorithm

The distribution of statistic x is Laplacian with 0 mean. For a significant level of 5% ($\alpha = 0.05$), the confidence interval is $\left[l_{\alpha/2}\ l_{1-\alpha/2}\right]$, where the $l_{\alpha/2}$ represents the point at which the value of cumulative distribution function of Laplacian is $\alpha/2$. For our statistical mode of x, we have

$$l_{\alpha/2} = -0.038, \quad l_{1-\alpha/2} = 0.038.$$

4.1 Detection of Jsteg

We propose a fast steganlytic algorithm for detecting both sequential Jsteg and random Jsteg. The steps of the algorithm are described as fellow:
Input: A set of JPEG images for detecting;
Output: y=1 (embedding), y=0 (no embedding)

1) Preparing the image: using the JPEG coding technique to get the quantizated DCT coefficients of image;
2) Calculate the value of statistic x as described in Section 3 ;
3) Determination: for a significant level of α, if the value x fall into the confidence interval $\left[l_{\alpha/2}\ l_{1-\alpha/2}\right]$, then output $y = 0$, else output $y = 1$.

4.2 Calculating the Amount of Embedding

Let β be the length of the embedded message in bits divided by $\|C\| - h_0(C) - h_1(C)$. We have

$$z = \frac{[f_1(C) - f_0(C)]}{h_1(C)} \tag{8}$$

There is some relations between β and z. When $\beta = 0$, the value of z is around 0; when $\beta = 1$, the value of z is far from 0. Lineal model describes it as.

$$\beta = a_1 z + a_2 \tag{9}$$

We use linear regression to get coefficients a_1, a_2. For 50 training images, we embed message into them to let $\beta = 0,0.2,0.4,0.6,0.8,1.0$ respectively. The least squares solution is

$$\hat{a} = (Z^T Z)^{-1} Z^T \beta \tag{10}$$

For the sequential Jsteg, we get

$$\beta = 1.02z - 0.0059$$

For the random Jsteg, we get

$$\beta = 1.0z - 0.0041$$

We can see that for furthermore simplification, we have $\beta \approx z$.

5 Experimental Results

We use our steganalytic algorithm to test on 680 images download from websites. We embed data in these images with amount of 0%,20%,40%,60%,80%,100% of the max capacity using sequential Jsteg and random Jsteg respectively. We use the detection algorithm proposed in Section 4.1 to detect the stego-images, and use the algorithm proposed in Section 4.2 to estimate the amount of embedding. Table 2 is the results of

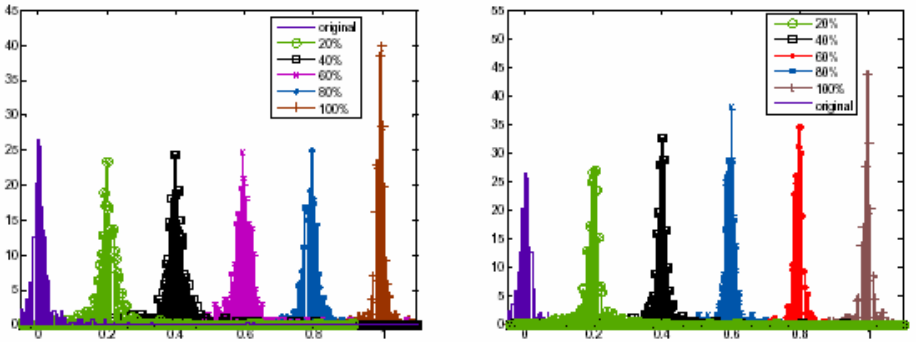


Fig. 5. Estimation of the amount of embedding of Jsteg. (a) Estimation of the sequential Jsteg.(b) Estimation of the random Jsteg

Table 2. Results of the detection of test images

Embedding amount	0	20%	40%	60%	80%	100%
Accurate detection for sequential Jsteg	96.76%	91.18%	95.29%	96.62%	97.65%	97.79%
Accurate detection for random Jsteg	96.76%	91.62%	95.25%	96.76%	97.20%	97.94%

detection accuracy for test images. It can be see that our algorithm is effective to detect either sequential Jsteg or random Jsteg. Fig 5 is results of the estimation of the amount of embedding. We can see that the estimations are very closed to the exact values.

6 Conclusions

In this paper, we propose a fast steganalytic algorithm for Jsteg based on the statistical model of images. We use the Laplacian distribution to fit the distribution of statistic. We perform the Pearson Chi-square test for the goodness of fit. We also prove that the mean of the Laplacian is zero. Whether the image is stego-image or not is determined by the statistic of its AC coefficients. We use linear model to estimate the amount of embedding and use regression to get the coefficients of linear model. The experiment results show that our detection algorithm is effective to both sequential Jsteg and random Jsteg. The estimation of the embedding amount is closed to the accurate value. The further work is to develop this algorithm to color image.

References

1. Christone, I, Podilchuk, and Wenjun Zeng : Image-Adaptive Watermarking Using Visual Models. *IEEE Journal Select. Areas Commun*, vol .16,pp.525~539,1998.
2. A. Westfield, A. Pfitzmann: Attack on steganographic systems. In: Proc. Information Hiding Workshop. Volume 1768 of Springer LNCS. (1999) 61-76.
3. N.F. Johnson, S. Jajodia: Steganalysis of Images Created Using Current Steganography Software. In: Proc. Information Hiding Workshop. Volume 1525 of Springer LNCS. (1998) 32-47.
4. J. Fridrich, R. Du, M. Long: Steganalysis of LSB Encoding in Color Images. Proc.of ICME 2000,NYC,july 31-Aug. 2,USA. 355.
5. J. Fridrich, M. Goljan, R. Du: Reliable Dectection of LSB Steganography in Color and Grayscale Images. Pro. of ACM Workshop on Multimedia and Security, Ottawa,Oct.5, (2001) 27-30.
6. J. Fridrich, M. Goljan, D. Soukal: High-order Statistical Steganalysis of Palette Images. Proc. Security and Watermarking of Multimedia Contents V. Vol. 5020 of SPIE. (2003) 178-190
7. J. Fridrich, M. Goljan, R. Du: Steganalysis Based on JPEG Compatibility. Proc. Multimedia Systems and Applications IV. Vol. 4518 of SPIE. (2002) 275-280

8. M. Niimi, R.O. Eason, H. Noda, E. Kwaguchi: Intensity histogram steganalysis in BPCS steganography. Proc. Security and Watermarking of Multimedia Contents. Volume 4314 of SPIE. (2001) 555-564
9. J.J. Harmsen, W.A. Pearlman: Kernal Fisher Disriminant for steganlysis of JPEG Hiding. Proc. Security, Steganography and Watermarking of Multimedia Contents VI. Vol. 5306 of SPIE. (2004)
10. J. Fridrich, M. Goljan, D. Hogea: Steganalysis of JPEG Images: Breaking the F5. In: Proc. Information Hiding Workshop. Volume 2578 of Springer LNCS. (2003)
11. S.Trivedi, R. Chandramouli: Active Steganalysis of Sequential Steganography. Proc. Security and Watermarking of Multimedia Contents V. Vol. 5020 of SPIE. (2003) 123-130
12. J. Fridrich, M. Goljan, D. Hogea: Attacking the OutGuess. Pro. of ACM Workshop on Multimedia and Security. France, December 6,2002.
13. H. Fraid: Detecting Steganographic Message in Digital Images. Technical Report, TR2001-412, Dartmouth College, (2001)
14. H. Fraid: Detecting Hidden Message using higher-order statistics and Support vector machines. Pro. of IEEE Int. on Image Processing , Rochester,NY, (2002)
15. I. Avcibas: Steganalysis Using Image Quality Metrics. In: IEEE Transactions on Image Processing. Vol. 12. No. 2. (2003) 231-239
16. JPEG-Jsteg-V4. http://www.funet.fi/pub/crypt/steganography/jpeg-jsteg-v4.diff.gz
17. R.C. Reininger, J.D. Gibson: Distribution of the Two-Dimensional DCT Coefficients for Images. In: IEEE Transaction On Communicatins. Vol. Com-31,No.6.June 1983
18. F. Müller: Distribution Shape of Two-Dimensional DCT Coefficients of Nature Images. In: Electron. Letter. Vol. 29, Oct, (1993) 1935-1936
19. E.Y. Lam, J.W. Goodman: A Mathematical Analysis of the DCT Coefficient Distributions for Images. In: IEEE Transactions on Image Processing. Vol. 9, No. 10. (2000)
20. M. Barni, F, bartolini, A. Piva, F. Rigacci: Statistical Modelling of Full Framee DCT Coefficients.
21. Wu Yi: Applied Statistics. Publishing Company of National University of Defence Technology, Chang Sha, 1995

Packet Classification Algorithm Using Multiple Subspace Intersecting*

Mingfeng Tan, Zexin Lu, and Lei Gao

School of Computer Science, National University of Defense Technology,
Changsha 410073, China
{mftan, lzx}@nudt.edu.cn, wuyufeng315@yahoo.com.cn

Abstract. Packet classification on multi-fields is difficult and has poor worst-case performance due to its character of multiple dimensions. Thus his paper proposes a efficient hardware algorithm MSI (multiple subspace intersecting) to solve it. MSI cuts each dimension of the classifier into several subspaces and then utilizes the parallelism of hardware to do the classification on these subspaces. For a classifier with n rules of width W, MSI needs only less than $n[4W+\log_2(n)]$ bits. MSI is able to classify 100 M packets/s with pipelined hardware, supports fast incremental update, and has good flexibility in specification of rule and. By simulation, we find MSI has better performance comparing to some existing algorithms.

1 Introduction

Internet needs to provide more different qualities of services in the future, and this requires the switching units to do the packet classification with higher performance. For this purpose, this paper proposes a hardware scheme named MSI (Multiple Subspace Intersection). We find the new and effective techniques: divide each dimension of the classifier into several smaller subspaces which are processed in parallel by hardware during search. By this mean, MSC is able to get the classification result per clock cycle with small memory and multi-pipeline architecture. For n rules of width W, MSI needs less than $n[4W+\log_2(n)]$ bits to store the data structures. MSI can do fast incremental update, and supports general rules, including prefixes, ranges, operators (less than, equal to, etc), and wildcards.

Chapter 2 defines the classification problem formally to specify MSI accurately and clearly. The related works and some important conclusions are briefly introduced in chapter 3. Chapter 4 describes MSI in detail, then proves its correctness and analyzes its performance. Moreover, examples are illustrated through this chapter to explain how MSI works. Chapter 5 shows the simulating results, and then the conclusions are drawn in chapter 6.

* This work is supported by the 973 Program of China (Grant 2003CB314802), the NSF of China (Grant 90104001).

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 1083 – 1093, 2005.

2 Description of Problem

When a packet arrives, the router compares the packet header with the rules in the classifier, and then and then the action related to the best match is executed. A packet is likely to match several rules, but only the least cost one is the best match. So there are three main elements in classification problem: 1. packet headers to be classified; 2. a set of rules, that is, the classifier; 3. classification algorithm to find the best rule for the packet header. Then we define these key concepts as follows:

Packet Header: H=<h_1,…,h_d> is a packet header and W is its width. Here h_i is the *ith* field of H, and the width of h_i is w_i.

Rule: A d dimension rule of W bits is R=<r_1,…, r_d, *action*>. Here r_i is the *ith* filed of the rule, also called as the *ith* dimension of R. The width of r_i is w_i. and r_i can be prefix, operator, ranges, or wildcard. Here *span*(r) is an operator, which is defined as: *span*(r) ={x|x is a binary string representation of r , here r is a filed of the rule }, and *span*(F) ={x|x ∈ *span*(r), r ∈ F, here F is a set of fields }. For example, span (1*0)={100,110},span({1*0,11*})={100,110,111}.

Classifier: Classifier=$\{R_j \mid 1 \le j \le n, \text{here } R_j \text{ is a rule}\}$, each rule has different cost; the cost of R_j is cost(R_j). Here n is the size of the classifier.

Packet Classification Algorithm: Packet classification algorithm chooses the least cost rule that matches the incoming packet, and it often includes the process of building the matching information data structures. A rule R_{best}=<r_1,…, r_d, *action*> is the best match of H=<h_1,…, h_d,> iff the following conditions are satisfied: 1. R_{best} is a match of H, that is, for i from 1 to d, $h_i \in span(r_i)$. And all the matched rules of header H can be defined as a set: *Match*(*Classifier*,H); 2. If rule $R' \in Classifier$ is a match of H and $R \neq R'$, then there must be $\cos t(R) < \cos t(R')$.

An operator on packet header H and the *ith* dimension of the classifier is defined as $Match_{\dim}(Classifier, H, i) = \{R \mid R \in Classifier \text{ and } h_i \in span(r_i)\}$, then *Match* $(Classifier,H) = \bigcap_{i=1}^{d} Match_{\dim}(Classifier,H,i)$.

3 Related Works

Classification on multi-fields is difficult due to its character of multiple dimensions, and many researchers try to solve it. We briefly discuss some efficient works here.

The Bitmap-Intersection[1] scheme associates a bitmap to each value of different dimensions. When a header H arrives, it does the 1-dimensional match on each dimension according to h_i, hence gets d bitmaps and intersects them, then it gets *Match*(*Classifier*,H). So its searching speed is limited by the 1-dimensional searching.

RFC[2] (Recursive Flow Classification) is a multiple phases scheme. It cuts or combines the fields to segments with same length, and then recursively pre-calculates

the match information. When searching, RFC maps the packet header to a shorter binary string in each phase and then gets the class ID at the last stage. RFC can search fast but needs to rebuild the whole data structures for each update, and its memory usage increases fast when the size of classifier increases.

HiCuts[3] algorithm recursively chooses and cuts one dimension into smaller spaces, and then calculates the rules intersecting with each smaller space to build a decision tree With which HiCuts guides the classifying process. It works well for small classier, but its performance decrease fast when the size of classifier increasing.

HyperCut[4] algorithm extends the HiCuts: it cuts multiple dimensions each step, and get the hyper cubes to intersect with the rules. Thus this "hyper cut" searches faster and requires smaller memory.

The EGT-PC[5] paper draws an important conclusion based on the investigation on the real classifiers: "for 99.9% of the source-destination cross products, the number of matching rules was 5 or less. Even in a worst case sense, no cross product (and hence packet) for any database matches more than 20 rules when considering only source destination field matches". And then it extends Grid-of-Tries[6] based on this observation and uses path-compressing to get better performance.

We will compare the performance between our MSI and these algorithms in chapter 5 by simulating, and will find MSI has better performance.

4 MSI Algorithm

4.1 Intuitionistic Explanation of MSI

A d-dimensional classification problem is equivalent to the point location problem in a d-dimensional space, if we can pre-compute all the best matching rule for each point, then each classifying operation can be done in only 1 step. But the number of points in the space is enormous. For a typical 5 dimension classification there are $2^{32+32+8+16+16}=2^{104}$ points. Hence this can't be practical. Instead, MSI tries to indirectly calculate the matching rules for *H*.

We use an example to introduce the idea of MSI. Suppose a classifier has 4 2-d rules: {R1=<(8,10),(4,9)> R2=<(9,13),(9,11)> R3=<(4,9),(7,13)> R4=<(1,4),(3,8)>}, Both dimensions are 4 bits. As left part of figure2 shows, these 4 rules can be thought as 4 rectangles in the 2D space. And MSI cuts each dimension into two 2-bits sub-dimensions hence get 16 sub-spaces that is highlighted by grey shadow as right part of figure 2 shows:

During updating, MSI pre-calculates the rules intersecting with each subspace, and then stores the intersecting information in bitmaps. When searching, it uses the sub-fields of the packet header as indexes to get proper subspaces. For example, the searching process for packet header (1001, 0111) can be explained as figure 3. The solid black point at the left of the equation is the point (1001, 0111), notice that the intersection of these subspaces is exact the point (1001, 0111). Using the sub-fields '10' on $subDim_{1,1}$, the sub-fields '01' on $subDim_{1,2,}$ the sub-fields '01' on $subDim_{2,1}$, and the sub-fields '11' on $subDim_{2,2}$, MSI gets four subspaces: $Subspace_{1,1,2}$, $Subspace_{1,2,1}$, $Subspace_{2,1,1}$and $Subspace_{2,2,3}$. As figure 1 shows, they are the shadows in hyper cubes at the right of equation. The intersection of these four subspaces is just

the point (1001, 0111). MSI calculates the rules intersecting with each of these subspace, then it get $\{R_1,R_2,R_3\}$ for $Subspace_{1,1,2}$, $\{R_1,R_2,R_3,R_4\}$ for $Subspace_{1,2,1}$, $\{R_1,R_3,R_4\}$ for $Subspace_{2,1,1}$, and $\{R_1,R_2,R_3,R_4\}$ for $Subspace_{2,2,3}$. Then:

$$\begin{aligned} Match_{candidate}(Classifier,H)&=\{R_1R_2R_3\}\cap\{R_1R_2R_3R_4\}\cap\{R_1R_3R_4\}\cap\{R_1R_2R_3R_4\} \\ &=\{R_1R_3\} \end{aligned} \quad (1)$$

These two rules are the exact matches of the header (1001, 0111).Section 4.3 will prove $Match(classifier,H) \subseteq Match_{candidate}(classifier,H)$. So we call $Match_{candidate}(classifier,H)$ "the candidate matching rules set". After get it, MSI chooses the best rules in efficient way as section 4.2 describes.

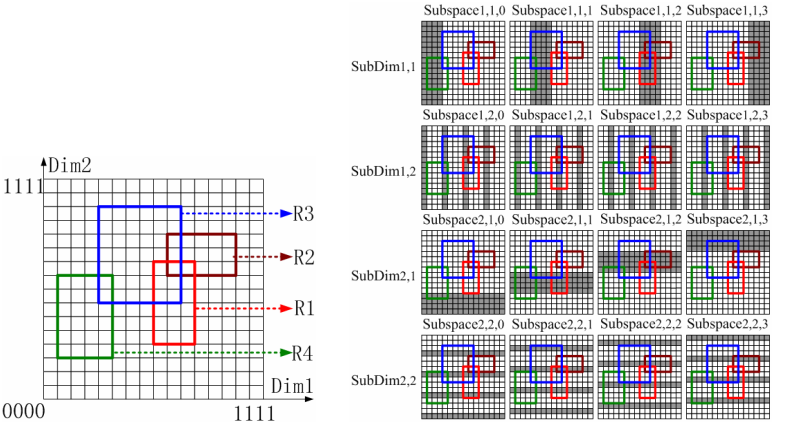


Fig. 1. Left: the Rules' images in space. Right: Subspaces after cutting

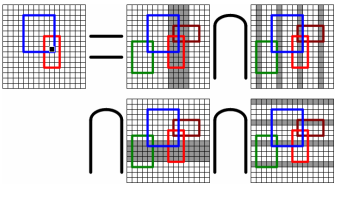


Fig. 2. Intuitionistic explanation of MSI

4.2 Description of MSI

In this section, the *jth* rule of the classifier is denoted as R_j=<$r_{j,1}$,…, $r_{j,d}$, *action*>. The *ith* dimension is divided into sub-dimensions with width of $s_i\,(1\le i\le d)$, namely $subDim_{i,k}$. So the number of sub-dimension in *ith* dimension is $c_i=w_i/s_i$. Here k $(1\le k\le c_i)$ is the index of the sub-dimension. Suppose a rule, R, is to be added, then the process of is shown as follows. And figure 1 shows the hardware structure of MSI.

- Adds a rule

1. Store R and its cost: store R and its cost to a high speed memory entry, and taking the entry's number as R's number. Suppose the newly added rule is R_j.
2. Convert range to prefixes: for each field $r_{j,i}$ of R_j, convert $r_{j,i}$ to a set of prefixes, namely $\text{Prefix}_{j,i}= span(r_{j,i})$.
   r_{11}=(8,10)={100*,1010} r_{12}=(4,9)={01**,100*}
   r_{21}=(9,13)={1001,101*,110*} r_{22}=(9,11)={1001,101*} r_{31}=(4,9)={01**,100*}
   r_{32}=(7,13)={0111,10**,110*} r_{41}=(1,4)={0001,001*,0100} r_{42}=(3,8)={0011,01**,1000}
3. Cut sub-dimension: Cut each $p_{j,i} \in \text{Prefix}_{j,i}$ into c_i sub-prefixes with length s_i, namely $p_{j,i,k}$. So for each sub-dimension $\text{subDim}_{i,k}$ of all rules, MSI builds $j \times c_i$ sets of sub-prefixes, each set is denoted as $Psub_{j,i,k}$.
   $\text{Psub}_{1,1,1}$={10} $\text{Psub}_{1,1,2}$={0*,10} $\text{Psub}_{1,2,1}$={01,10} $\text{Psub}_{1,2,2}$={**,0*}
   $\text{Psub}_{2,1,1}$={10,11} $\text{Psub}_{2,1,2}$={01,1*,0*} $\text{Psub}_{2,2,1}$={10} $\text{Psub}_{2,2,2}$={0, 1*}
   $\text{Psub}_{3,1,1}$={01,10} $\text{Psub}_{3,1,2}$={**,0*} $\text{Psub}_{3,2,1}$={01,10,11} $\text{Psub}_{3,2,2}$={11,**,0*}
   $\text{Psub}_{4,1,1}$={00,01} $\text{Psub}_{4,1,2}$={01,1*,00} $\text{Psub}_{4,2,1}$={00,01,10} $\text{Psub}_{4,2,2}$={11,**,00}
4. Get subspace index in sub-dimension: The MSI calculates a set of indexes: $Index_{j,i,k}=span(Psub_{j,i,k})$, and each $idx \in Index_{j,i,k}$ is a subspace index on $\text{subDim}_{i,k}$.
   $\text{Index}_{1,1,1}$={2} $\text{Index}_{1,1,2}$={0,1,2} $\text{Index}_{1,2,1}$={1,2} $\text{Index}_{1,2,2}$={0,1,2,3}
   $\text{Index}_{2,1,1}$={2,3} $\text{Index}_{2,1,2}$={0,1,2,3} $\text{Index}_{2,2,1}$={2} $\text{Index}_{2,2,2}$={1,2,3}
   $\text{Index}_{3,1,1}$={1,2} $\text{Index}_{3,1,2}$={0,1,2,3} $\text{Index}_{3,2,1}$={1,2,3} $\text{Index}_{3,2,2}$={0,1,2,3}
   $\text{Index}_{4,1,1}$={0,1} $\text{Index}_{4,1,2}$={0,1,2,3} $\text{Index}_{4,2,1}$={0,1,2} $\text{Index}_{4,2,2}$={0,1,2,3}
5. calculate bitmaps: Each possible subspace's index x(x=0,…, 2^{s_i} -1) has an associated bitmap, namely $\text{Bitmap}_{i,k,x}$. And each bitmap has n bits: Value '1' of the jth bit means R_j intersects with this sub-space. So for each $idx \in Index_{j,i,k}$, set the jth bit of $\text{Bitmap}_{i,k,idx}$ to '1'. In this way, a set of rules intersecting with the sub-space represented by x is called $\text{SubDimRuleSet}_{i,k,x}$ and is associated with $\text{Bitmap}_{i,k,x}$.
   $\text{Bitmap}_{1,1,0}$=0001 $\text{Bitmap}_{2,1,0}$=0001 $\text{Bitmap}_{1,1,1}$=0011 $\text{Bitmap}_{2,1,1}$=1011
   $\text{Bitmap}_{1,1,2}$=1110 $\text{Bitmap}_{2,1,2}$=1111 $\text{Bitmap}_{1,1,3}$=0100 $\text{Bitmap}_{2,1,3}$=0010
   $\text{Bitmap}_{1,2,0}$=1111 $\text{Bitmap}_{2,2,0}$=1011 $\text{Bitmap}_{1,2,1}$=1111 $\text{Bitmap}_{2,2,1}$=1111
   $\text{Bitmap}_{1,2,2}$=1111 $\text{Bitmap}_{2,2,2}$=1111 $\text{Bitmap}_{1,2,3}$=0111 $\text{Bitmap}_{2,2,3}$=1111

- Deletes a rule

1. Removes jth entry of the high speed memory for deleting R_j and its cost.
2. Clears jth bits of all bitmaps. Or instead, calculates the subspace indexes of R_j for each sub-dimension, then find the corresponding bitmaps and set their jth bit to '0'.

- Classification

Here we show the classification examples in table 1. When a packet $H =< h_1, \cdots, h_d >$ arrives, MSI gets the best match as follows:

1. Cut h_i into c_i sub-fields, hence MSI gets $h_{i,k}$.
2. Get bitmaps: Uses $h_{i,k}$ as the subspace index on $\text{subDim}_{i,k}$ and gets $Bitmap_{i,k,h_{i,k}}$.
3. Calculate $\bigcap_{i=1}^{d}\bigcap_{k=1}^{c_i} Bitmap_{i,k,h_{i,k}}$: This is equivalent to $Match_{candidate}(classifier,H)=\bigcap_{i=1}^{d}\bigcap_{k=1}^{c_i} SubDimRuleSet_{i,k,h_{i,k}}$.
4. Get the final match: MSI chooses the exact match from $Match_{candidate}(classifier,H)$ and takes the least cost one.

Table 1. Examples of classification

Header	Match Rules	Bitmap					Candidate rules
		1,1,x	1,2,x	2,1,x	2,2,x	result	
1001,1001	$R_1R_2R_3$	1110	1111	1111	1111	1110	R_1R_2
1001,0111	R_1R_3	1110	1111	1011	1111	1010	R_1R_3
1010,0101	R_1	1110	1111	1011	1111	1010	R_1R_3
0010,1011	none	0001	1111	1011	1111	0001	R_4

Obviously the searching of MSI can be pipelined in 5 stages: 1. Get bitmaps on all sub-dimensions according to the packet header H; 2. Calculate the result bitmap; 3. Gets $Match_{candidate}(classifier,H)$; 4. Use the comparing units to compare H and the rules in $Match_{candidate}(classifier,H)$, and then find out the exact matches; 5. Choose the least cost one among the exact matches by the arbitrator.

Being pipelined, MSI can get 1 classification result per clock cycle in most cases. If $|Match_{candidate}(classifier,H)|>k$, where k is the number of comparing units, then stage 4 and stage 5 must be executed again until all the candidate rules are processed. But this happens scarcely in real situation.

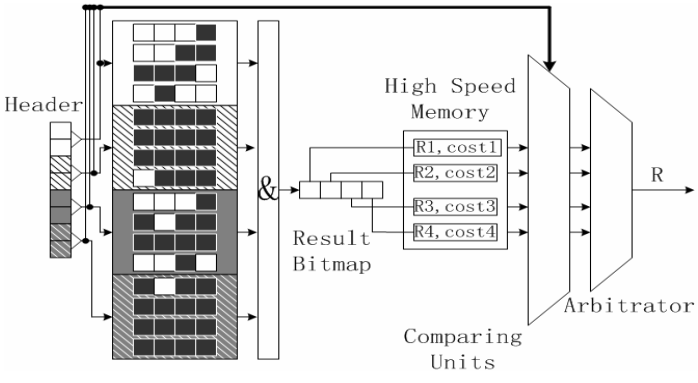


Fig. 3. Example and structure of MSI

4.3 Theorem Proving of MSI

This section proves MSI can get the correct result and $|Match_{candidate}(classifier,H)|$ is small. For this purpose theorem 1 and 2 are put forward at first:

Theorem 1: Suppose prefix P matches packet header H. If extracting the same bits from them to build a new prefix P' and a new header H', then P' still matches H'.

Theorem 2: Suppose prefixes P_i matches headers H_i $(1 \le i \le d)$ and P_i has same length with H_i, then the combination $P=<P_1,\ldots, P_d>$ matches $H=<H_1,\ldots, H_d>$.

Obviously these two theorems are true. And theorem 3 is brought forward to show the correctness of MSI.

Theorem 3: $Match(classifier,H) \subseteq Match_{candidate}(classifier,H)$

Proof: For $\forall R_j \in Match(Classifier,H)$, suppose $R_j=<r_{j,1},\ldots, r_{j,d}, action>$, then there must be $R_{j,i} \in Match_{dim}(Classifier,H_i)$, then $h_i \in span(r_{j,i})(1 \le i \le d)$. If convert $r_{j,i}$ to a prefix set A, then there must be a prefix $p \in A$ satisfying $h_i \in span(p)$. Suppose h_i's sub-filed on $subDim_{i,k}$ is $h_{i,k}$, and p's sub-prefix on $subDim_{i,k}$ is $p_{i,k}$. According to theorem 1, $p_{i,k}$ is a match of $h_{i,k}$, then $h_{i,k} \in span(p_{i,k})$. From MSI's updating process we can know the *jth* bit of $Bitmap_{i,k,h_{i,k}}$ is '1'. So we get the following relationships:

$$R_j \in SubDimRuleSet_{i,k,h_{i,k}} \Rightarrow Match(classifier,H) \subseteq SubDimRuleSet_{i,k,h_{i,k}} \Rightarrow$$

$$Match(classifier,H) \subseteq \bigcap_{i=1}^{d} \bigcap_{k=1}^{c_i} SubDimRuleSet_{i,k,h_{i,k}} = Match_{candidate}(classifier,H)$$

Theorem 3 tells us that all the matches of H must be in $Match_{candidate}(classifier,H)$, so MSI can get the right rule. And then theorem 4 is brought forward to help us find the size of the $Match_{candidate}(classifier,H)$, hence shows the efficiency of MSI.

Theorem 4: If all fields of the classifier are represented as prefixes, then there must be $Match(classifier,H) \supseteq Match_{candidate}(classifier,H)$.

Proof: $\forall R_j \in Match_{candidate}(classifier,H)$, then $R_j \in \bigcap_{i=1}^{d} \bigcap_{k=1}^{c_i} SubDimRuleSet_{i,k,h_{i,k}}$, and then $R_j \in SubDimRuleSet_{i,k,h_{i,k}}$ $(1 \le i \le d,\ 1 \le k \le c_i)$. According to the caculating process of $SubDimRuleSet_{i,k,h_{i,k}}$, we have $h_{i,k} \in Index_{j,i,k}=span(Psub_{j,i,k})$, so for all $1 \le i \le d$ and $1 \le k \le c_i$ there exists $p_{j,i,k} \in Psub_{j,i,k}$ makes $h_{j,i,k} \in span(p_{j,i,k})$. So $p_{j,i,k}$ is a match of $h_{j,i,k}$. Since R_j is in prefix representation, so there is only one sub-prefix in $Psub_{j,i,k}$, that is $p_{j,i,k}$. And we know $R_j =< p_{j,1,1},\cdots, p_{j,i,k},\cdots, p_{j,d,c_d} >$, so according to theorem 2, R_j is a match of H, then we have :

$R_j \in Match(Classifier,H) \Rightarrow Match(classifier,H) \supseteq Match_{candidate}(classifier,H)$.

From theorem 4 we can infer that if some fields of the classifier are represented as prefixes, then $Match_{candidate}(classifier,H)$ is the subset of S, which is the matching rules set on these fields . So there must be $|Match_{candidate}(classifier,H)| \le |S|$.

And with theorem 3 and 4 we can get following inference:

Theorem 5: If all fields of the classifier are represented as prefixes, then: $Match(Classifier, H) = Match_{candidate}(classifier,H)$.

According to the conclusion of EGT-PC paper[5] : **"for 99.9% of the source-destination cross products, the number of matching rules was 5 or less. Even in a worst case sense, no cross product (and hence packet) for any database matches more than 20 rules when considering only source destination field matches",** and according the above theorems, we know $|Match_{candidate}(classifier,H)|$ must be smaller than 5, and will never exceeds 20. In fact, $|Match_{candidate}(classifier,H)|$ would be smaller for multi-filed classification. And MSI needs only several comparing units to select the exact matches from $Match_{candidate}(classifier,H)$, each of these comparing units should be able to compare all the dimensions in parallel. By this means and being pipelined, MSI can find out the final rule in 1 clock cycle.

Besides, for the classifiers whose fields are all in prefixes, we can store the rules in the sequence of their priority, then only 3 pipeline stages is needed: 1. getting bitmaps on all sub-dimensions according to the packet header H; 2. calculating the result bitmap; 3. get the final result. In this situation, MSI needs only $2nW$ bits to store the data structure as section 4.5 describes, and can get 1 result in 1 pipeline loop.

4.4 Performance Analysis

- Searching time, Updating time, and the choice of s_i

The searching time is related to the size of $Match_{candidate}(classifier,H)$ and the number of comparing units. But from the proof of section 4.4 and chapter 5 we can know that $|Match_{candidate}(classifier,H)|$ is small, and so MSI has good searching performance. And our experiments shows that when s_i =2, MSI is able to update faster with less memory. So s_i =2 is the best choice.

MSI needs writing bits to the bitmaps. The largest number of this writing operation is $O\left(\sum_{i=1}^{d}\left[2(w_i-1)\times c_i\times 2^{s_i}\right]\right)$. When s_i = 2 it is $O\left(4\sum_{i=1}^{d}\left[(w_i-1)\times w_i\right]\right)$. For a typical 5D classification, this is $O(10^4)$. But as our experiments shows, this number never exceeds 350. So each update time is less than 350 memory accesses.

- Storage requirements

MSI needs $n(2W+\log_2 n)$ bits to keep the original rules and their cost, but if some fields, with total length of L bits, are in prefix representation, then MSI needs not store these fields in the high speed memory, because theorem 5 tell us they will be exactly matched by the AND gates , and need not compare them again at the last stage of the pipeline. So Each rule takes $(2W - 2L)$bits, and each cost takes $\log_2 n$ bits. Besides, MSI needs $n\sum_{i=1}^{d}\left[c_i\times 2^{s_i}\right]$ bits to store the bitmaps. When s_i=2, it's only $2nW$ bits. Then the total storage is $n(4W-2L+\log_2 n)$ bits. For the classifiers for the classifiers whose fields are all in prefixes and all the rules are stored in the sequence of their cost, then MSI needs only $2nW$ bits.

- Requirement on logical units.

To get best performance, MSI requires n AND gates. As we know that AND gate is the most basic logic circuit, and therefore is very cheap. Furthermore, less than 20 comparing units are needed to select the exact matches.

5 Simulation and Result

Due to privacy and commercial secrets, we have no sufficient real classifiers. So we use the real route tables (http://bgpview.6test.edu.cn) to generate the classifiers according to the characters of real classifiers[2][5].Both the source and destination IP address fields are prefixes and other fields are ranges representation. A real IP trace of 18,523,793 headers (http://pma.nlanr.net) is tested for each simulation.

Table 2 shows the average result and the worst result among all the tests on the 20 classifiers with the same size. We can see $|Match_{candidate}(classifier,H)|$ is near to

$|Match\ (classifier,H)|$, this shows the the efficiency of MSI. As for the 2D cases, $|Match\ (classifier,H)| = |Match_{candidate}\ (classifier,H)|$, which can be explained by theorem 4. When d=5, $|Match\ (classifier,H)| \leq |Match_{candidate}\ (classifier,H)|$, and their size is very close, this can be explained by theorem 3. And we can find that both the worst and average size of $Match_{candidate}(classifier,H)$ are often small values, so only a small number of comparing unit is needed in the last pipeline stage. Therefore, MSI can achieve good performance with only one comparing unit.

Table 3 shows the update performance of MSI. When s_i=2 MSI get its best update performance, and the size of the classifier has nothing to do with the update time. As described before, MSI consumes the least memory while s_i=2. So we use s_i=2 as the best parameter and make the further simulation.

Table 2. Simulation results for searching. (Notice that the default rule is not included in the test rule set. so the average size of the rule set seems very small. In the real situation, it proberbly needs to add 1 for the default rule. Table4 is in the same situation)

d	Rule	$\|Match(Classifier,H)\|$		$Match_{candidate}(classifier,H)$ (s_i=1,2,4,8)							
		worst	average	worst				average			
2	128	1	9.235e-5	1				9.235e-5			
	256	1	1.324e-4	1				1.324e-4			
	512	2	1.851e-4	2				1.851e-4			
	1024	2	5.560e-3	2				5.560e-3			
	2048	2	4.547e-3	2				4.547e-3			
	4096	3	8.929e-3	3				8.929e-3			
	8192	4	1.714e-1	4				1.714e-1			
5	128	1	2.55e-6	1	1	1	1	2.6e-6	2.6e-6	2.6e-6	2.55e-6
	256	1	1.25e-6	1	1	1	1	1.25e-6	1.25e-6	1.25e-6	1.25e-6
	512	1	3.6e-6	1	1	1	1	3.8e-6	3.8e-6	3.8e-6	3.6e-6
	1024	1	5.765e-5	1	1	1	1	6.2e-5	6.2e-5	6.195e-5	5.765e-5
	2048	2	1.876e-4	2	2	2	2	2.011e-4	2.008e-4	1.997e-4	1.876e-4
	4096	2	3.878e-3	2	2	2	2	4.130e-3	4.130e-3	4.130e-3	3.878e-3
	8192	2	1.672e-3	2	2	2	2	1.791e-3	1.787e-3	1.786e-3	1.672e-3

Table 3. Simulation results for searching

d	**Number of Bitmaps to be write of each update s_i=(1,2,4,8)**							
	worst				**average**			
2	120	116	226	1793	90.2715	70.9592	109.757	720.076
5	198	188	348	2825	166.623	141.01	241.555	1734.26

Table 4 shows the simulation result for the large classifiers while s_i=2. The result shows that $|Match_{candidate}\ (classifier,H)|$ is also almost equal to $|Match\ (classifier,H)|$. For the 2D situation, the worst $|Match_{candidate}\ (classifier,H)|$ is under 20, and for 5D situation it is under 6. Moreover, we find the average $|Match_{candidate}\ (classifier,H)|$ is very small. So, one comparing unit in the last pipeline stage is enough to get good average performance for the large classifier.

As described before, MSI needs less than $n(4W+\log_2 n)$ bits to store the whole data structure. For a large classifier, its memory usage is only about 2 times of linear

searching scheme. So we can use the smaller but fast memory to make MSI achieve better performance.

Table 4. Simulation results for large classifier, s_i=2

Rules	d	$\|Match\|$		$\|Match_{candidate}\|$		d	$\|Match\|$		$\|Match_{candidate}\|$	
		worst	average	worst	average		worst	average	worst	average
16384	2	6	0.030523	6	0.030523	5	2	0.008504	2	0.009035
32768		6	0.05938	6	0.05938		2	0.008971	2	0.009551
65536		13	2.11409	13	2.11409		4	0.007302	4	0.007807
131072		19	2.2133	19	2.2133		5	0.025719	19	0.027473

Figure 4 shows the simulation results of MSI and several existing high performance classification schemes. From these simulation results we can find that the MSI has the best searching performance with smallest memory usage.

The link speed of OC768 is 40 GB/s. It can receive 104.2M packet/s at most with the least packet size of 48 bytes. But this is only for the 2D situation. As for the simple multiple classification, we consider the UDP packet. Then the OC768 can receive 89.2M packet/s at most. According to the structure of MIS in figure 1, if each pipeline stage of MSI can finish its work within 10 ns, which is not difficult to achieve by now, then MSI can process 100M packet/s and then it can fully catch the line speed forwarding and classification of OC768.

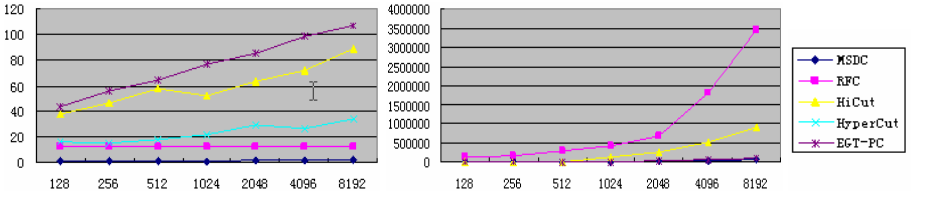


Fig. 4. Comparision of the classification algorithms according to table 5

6 Conclusion

Internet intends to provide more services and to be more reliable, secure, scalable and controllable. The classification algorithms with high performance are expected to meet some of these requirements. So we proposed MSI, a high performance classification algorithm. And we will try to find the efficient way to utilize the main idea of MSI, the sub-space cutting and intersection techniques to develop the new software classification algorithms. And another main point of our research will be on the forwarding architectures of the routers. For example, the routers often queue the packets after classification. If we queue the packet with some transcendental knowledge before the classification, and do the classification in the queue, then the classification time can be hidden.

References

1. T.V.Lakshman and D.Stiliadis. “High-Speed policy-based packet forwarding using efficient multi-dimensional range Matching”, Proc. Sigcomm’98, Sep. 1998.
2. Pankaj Gupta and Nick McKeown, “Packet Classification on Multiple Fields”, Proc. Sigcomm’99, Sep 1999.
3. Pankaj Gupta and Nick McKeown, “Packet classification using hierarchical intelligent cuttings,” IEEE Micro, 20(1): 34-41, Jan./ Feb. 2000.
4. Sumeet Singh and al. “Packet classification using multidimensional cutting,” Proc. Sigcomm’03, Aug. 2003.
5. Florin Baboescu and al, “Packet Classification for Core Routers: Is there an alternative to CAMs?” Proc. of IEEE Infocom’03, Mar./Apr. 2003.
6. V.Srinivasan and al, “Fast and scalable layer 4 switching,” Proc of Sigcomm’98, Sep. 1998.

RSA Extended Modulus Attacks and Their Solutions in a Kind of Fair Exchange Protocols

Ping Li[1,2], Lalin Jiang[1], Jiayin Wu[1], and Jing Zhang[2]

[1] School of Computer and Telecommunication Engineering,
Changsha University of Science and Technology, Hunan, China 410076
[2] Institute of Computer and Telecommunications, Hunan University Hunan, China 410008
{lping9188, jiayin528, mail_zhangjin}@163.com,
jiangln@csust.edu.cn

Abstract. By applying the concept of VRES (Verifiable and Recoverable Encrypted Signature), a kind of RSA-based fair exchange protocols try to provide security mechanism on information delivery assurance. However, due to incomplete verification on VRES, RSA extended modulus attacks addressed in this paper can be launched on those protocols successfully, causing failure on fairness of exchange. The RSA-DEMCA algorithm is proposed, achieving complete verifications on VRES without sensible information leakage. Security analysis on RSA-DEMCA is also addressed.

Keywords: Extended-modulus computation, fair exchange, VRES, RSA.

1 Introduction

Fair data exchange focus on achieving real-time fairness in exchange activities, based on the assumption that the two exchanging party do not trust each other or misbehave in exchange processes.

A kind of fair exchange protocols[1,2] utilizes the concept of VRES (Verifiable and Recoverable Encrypted Signature) to achieve fairness on signature exchange. Several protocols[3,4] based on RSA signature scheme, construct VRES expressed as $y = x^e \bmod N$, where x is a sensible element and N is the multiplicative of two bases of RSA cryptosystems. We call such a kind of computation as RSA extended modulus computation, as it is a little different from RSA standard encryption algorithm.

That kind of computation does provide theoretic support on data recovery for a third party. However, the premise condition of such an achievement is that x is required within an expected area. Otherwise the third party would fail to recover a sensible element to an applicant. The authors of [3] present RSA-CEGD protocol to achieve fairness on certified e-goods delivery and make an unsuitable assumption that a VRES creator (one of the two exchanging parties) should choose a random number in a specific area as a sensible element. It is certainly contradicted with the assumption that misbehaviors by the two exchanging parties may occur, such as choosing random

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 1094 – 1104, 2005.

numbers without any restrictions. In practical applications, a VRES creator may act as an active adversary, forge a verifiable but unrecoverable VRES, and launch RSA extended modulus attacks successfully addressed in this paper.

Similarly, the authors of [4] present an optimistic fair signature exchange protocol (we call it EPRSS for simplicity), constructing VRES by utilization of zero-knowledge proofs. However, due to lack of sufficient verifications on those presented proofs, RSA extended modulus attacks also cause the protocol failure to reach its expected achievements.

In this paper, we propose the RSA-DEMCA algorithm, achieving complete verification on VRES summarized as follows: (1). Introducing a delayed RSA modulus in exchange processes, requiring a VRES creator provides two VRES orderly. (2). Introducing an item called VRES-ER which is used to identify the recoverability of VRES, without influencing fairness addressed in RSA-CEGD protocol. (3). Transforming the problem of judging the value area of a sensible element into that of VRES-ER.

The rest of the paper is organized as follows. Section2 describes briefly RSA-CEGD and EPRSS protocols, combined with the properties of RSA extended modulus computations. RSA extended modulus attacks on those two protocols are presented in Section3. Section4 is arranged to propose RSA-DEMCA algorithm, aiming at erasing security flaws of RSA-CEGD. In Section5 we provide security analysis on our algorithm, before drawing the conclusion of this paper in Section6.

2 VRES Constructed in RSA Extended Modulus Computation

2.1 VRES Concept in RSA-CEGD

The protocol assumes that Party P_a has a valuable e-goods or e-payment, expressed as document D_a, and a symmetric key k_a for the encryption and decryption of D_a. Party P_a wishes to send D_a to party P_b in exchange for P_b's receipt for D_a.

The VRES represents a signature (i.e. receipt) encrypted in such a way that a receiver of the VRES is assured that it indeed contains the correct signature without obtaining any information about the signature itself (verifiability). He is also assured that a designated trusted party can help to recover the original signature from the VRES, in case the original signature sender refuses to release his receipt after receiving the e-goods.

2.2 VRES Implementation

Every party P_i ($\in \{P_a, P_b, P_t\}$) has a pair of public and private RSA keys, expressed as $pk_i = (e_i, n_i)$ and $sk_i = (d_i, n_i)$. Party P_b has obtained a RSA-based public-key certificate $C_{bt} = (pk_{bt}, w_{bt}, s_{bt})$ issued by P_t prior to the exchange. The public key pk_{bt} and its corresponding private key sk_{bt} are denoted as $pk_{bt} = (e_{bt}, n_{bt})$, $sk_{bt} = (d_{bt}, n_{bt})$, respectively. Notes that $e_b = e_{bt}$, $w_{bt} = (h(sk_t, pk_{bt})^{-1} \times d_{bt}) \bmod n_{bt}$ and $s_{bt} = E_{sk_t}(h(pk_{bt}, w_{bt}))$. P_b's receipt for document D_a, denoted as $receipt_b = (h(D_a))^{d_b} \bmod n_b$. Assume that P_b generates the VRES of his signature.

- VRES construction

The protocol constructs (y_b, x_b, xx_b) as VRES with the following expressions:

$x_b = (r_b \times (h_a)^{d_b}) \bmod n_b = (r_b \times receipt_b) \bmod n_b$ where r_b is a random number chosen by P_b. $y_b = r_b^{e_b} \bmod (n_b \times n_{b,t})$, encryption of r_b with P_b's public key pk_b. $xx_b = (r_b \times E_{sk_{bt}}(h(y_b))) \bmod n_{bt}$.

- VRES verification

(1). $x_b^{e_b} \bmod n_b = (y_b \times h_a) \bmod n_b$. (2). $xx_b^{e_b} \bmod n_{bt} = y_b \times h(y_b) \bmod n_{bt}$.

With the assumption that $e_b = e_{bt}$, P_a is sure that y_b can be decrypted using either private key sk_b or sk_{bt} to recover r_b.

- VRES recovery

In case that disputes occur, P_t uses key sk_{bt} from C_{bt} to decrypt y_b to recover r_b.

2.3 VRES in EPRSS

The EPRSS[4] protocol has applications in contract signing (exchange of two signatures over the same document), and in exchange of two signatures over different documents. The protocol assumes that two parties (A and B) share document M to be signed, and wish to fairly exchange their signatures over document M. For example, party A's RSA signature over M can be expressed as:

$s = Sign_A(M, sk_A) = H(M)^{d_A} \bmod n_A$ where d_A is the secret exponent of party A's private RSA key, expressed as $sk_A = (d_A, n_A)$. The two parties have agreed to employ an off-line TTP to help them with the exchange if they cannot reach a fair completion of the exchange themselves.

Assume that A computes $C_A = s^{e_{TTP}} \bmod N_{TTP,A}$ where $N_{TTP,A}$ is the product of the two bases of RSA keys of TTP and party A (n_{TTP}, n_A), denoted as $N_{TTP,A} = n_{TTP} \cdot n_A$.

In addition, corresponding proofs with zero-knowledge properties are required: $\omega = H(C_A \| e_{TTP} \| r^{e_{TTP}} \bmod N_{TTP,A})$, $z = r \cdot s^{\omega}$ where r is a random number chosen by the party A.

As a VRES verifier, after having received C_A, V_A ($V_A = (\omega, z)$), B performs the following VRES verifications:

1). $C_A^{e_A} = (H(M))^{e_{TTP}} \bmod n_A$; 2). $\omega = H(C_A \| e_{TTP} \| z^{e_{TTP}} C_A^{-\omega} \bmod N_{TTP,A})$;

3). $|z| < (\omega + 1)k$

If verifications are positive, B is sure that TTP has abilities to recover s for him once misbehaviors by A occur.

2.4 The Properties of RSA Extended Modulus Computations

The properties of RSA extended modulus computation are a little different from those of RSA standard cryptographic algorithms. In order to clearly describe extended modulus attacks, several corollaries are presented in this subsection.

Theorem 1. Let *p*,*q* be two different primes, *n*=*pq*. For any integer y （$0 \le y < n$）, then $y^{k\varphi(n)+1} \equiv y \bmod n$ where $k \geqq 0$.

Corollary 1. For integer $y\,(0 \le y < N)$, where $N = n_1 \cdot n_2$ (n_1 and n_2 are the two bases of RSA cryptosystems), and *e* is a common exponent, then $y = x^e \bmod N$ where $0 \le x < N$. Correspondingly, $y \bmod n_1 = x_1^e \bmod n_1, \quad (0 \le x_1 < n_1)$, $y \bmod n_2 = x_2^e \bmod n_2, \quad (0 \le x_2 < n_2)$.

Proof. According to Theorem1, $y^{k\varphi(N)+1} \equiv y \bmod n_1$. Similarly, $y^{k\varphi(N)+1} \equiv y \bmod n_2$. As gcd($n_1$,$n_2$)=1, $y = y^{k\varphi(N)+1} \bmod N$. For a given *e*, there exists corresponding d with $e \cdot d \equiv 1 \bmod(\varphi(N))$. Let $x = y^d \bmod N$, Thus $y = x^e \bmod N$ where $0 \le x < N$ .Suppose $x_1 = x \bmod n_1$, $x_2 = x \bmod n_2$, then $y \bmod n_1 = x_1^e \bmod n_1$, $y \bmod n_2 = x_2^e \bmod n_2$.

Corollary 2. For any two integers *x* and *y*, which satisfy the requirements as described in Corollary1, the following conclusions are held:

1). If $0 \le x < \min(n_1, n_2)$, then *x* can be obtained by RSA decryption on $y \bmod n_1$ or $y \bmod n_2$ with corresponding secret exponents d_1 or d_2.

2). If $\min(n_1, n_2) \le x < \max(n_1, n_2)$, then *x* can only be obtained by decrypting $y \bmod(\max(n_1, n_2))$ with corresponding $d_i (i = \max(n_1, n_2))$.

3). If $\max(n_1, n_2) \le x < N$, then only congruence solutions moduli n_1 and n_2 can be obtained by decrypting $y \bmod n_1$ and $y \bmod n_2$ with corresponding secret exponents d_1 and d_2.

Proof. Assume that $n_1 < n_2$ might as well, then $y \bmod n_1 = x_1^e \bmod n_1$, $y \bmod n_2 = x_2^e \bmod n_2$ according to Corolary 1.

1). If $0 \le x < n_1$, then $y^{d1} \bmod n_1 = x^{e \cdot d1} \bmod n_1 = x \bmod n_1 = x$. Similarly, the conclusion is also held for $y \bmod n_2$.

2). If $n_1 \le x < n_2$, then $y^{d1} \bmod n_1 = x^{e \cdot d1} \bmod n_1 = x \bmod n_1 < x$.

3). If $n_2 \le x < N$, then $y^{d2} \bmod n_2 = x^{e \cdot d2} \bmod n_2 = x \bmod n_2 < x$. Furthermore, $x_1 \ne x_2$, and they are not congruence modulo n_1 or n_2.

3 RSA Extended Modulus Attack

In this Section we present two instances of RSA extended modulus attacks on RSA-CEDG and EPRSS.

3.1 An Attack Instance in RSA-CEGD

In RSA-CEGED protocol, assume that P_b chooses a random number r_b, within the area of $\max(n_b, n_{bt}) < r_b < n_b \times n_{bt}$. Then he constructs VRES as addressed in Section 2.1.Notes that P_b is assumed impliedly to choose $r_b < n_b$ in RSA-CEGD protocol, so that there are no such verification available as to check if r_b is within the expected area. Thus the presented VRES verifications are certainly held.

That means if P_b is not honest, RSA extended modulus attacks can be launched successfully. Due to constraints of RSA encryption scheme, P_t fails to recover r_b for P_a to obtain $receipt_b$. P_t decrypts $y_b \bmod n_{bt}$ by the private key sk_{bt} and could only obtain r_b' expressed as $r_b' = r_b \bmod n_{bt}$. Certainly, $r_b' \neq r_b \bmod n_b$ according to Corollary 2.

3.2 An Attack Instance in EPRSS

In EPRSS protocol, party A computes $C_A' = S^{e_{TTP}} \bmod N_{TTP,A}$. It is worth noticing that $s = S \bmod n_A$, where s is the party A's correct signature on M, and $N_{TTP} < S < N_{TTP,A,r}$. It is fairly easy for the party A to find such an item S as satisfies the above requirements.

A then chooses a random number and constructs proofs addressed in Section2.4. Obviously, the forged VRES can also be verified "correct":

1). $C_A'^{e_A} \bmod n_A = (S^{e_{TTP}})^{e_A} \bmod n_A$
$= (s^{e_{TTP}})^{e_A} \bmod n_A = (H(M))^{e_{TTP}} \bmod n_A$

2). As we have

$z'^{e_{TTP}} \bmod N_{TTP,A} = r^{e_{TTP}} \cdot (S^{e_{TTP}})^{\omega} \bmod N_{TTP,A,r} \bmod N_{TTP,A}$

$= r^{e_{TTP}} \cdot C_A'^{\omega} \bmod N_{TTP,A}$

$\omega' = H(C_A' \| e_{TTP} \| z'^{e_{TTP}} \cdot C_A'^{-\omega} \bmod N_{TTP,A})$ is also held.

3). Since $|N_{TTP,A,R}| \leq (\omega+1)k$, then $|z'| \leq |N_{TTP,A,R}| \leq (\omega+1)k$

Unfortunately, TTP would fail to recover a correct s for the party B once misbehaviors occur. Notes that S is within the area $n_{TTP} < S < N_{TTP,A}, r$, TTP could only achieve s_1 denoted as $s_1 = S \bmod n_{TTP}$ by decrypting $C_A{}'$. According to Corollary2, there exist infinite solutions of S satisfying $s_1 \neq s$.

4 RSA-DEMCA Algorithm

As illustrated in Section3, a malicious VRES creator would forge a verifiable but unrecoverable VRES by choosing a random number in a larger area as a sensible element. Thus, a complete verification on VRES is required to consist of value judgement on the encrypted data as well as formal verifications on VRES.

Here we expand the meaning of recoverability of VRES. Even though P_t only obtains a congruence solution of r_b modulo n_{bt}, we also say that VRES is recoverable if and only if P_a can achieve $receipt_b$ with help of P_t. In order to check if the presented VRES is recoverable or not, we introduce an effective item, denoted as VRES-ER, to identify the recoverability of VRES. Thus, the problem of value restriction of VRES is transformed into that of VRES-ER.

In this section we propose RSA-DEMCA algorithm totally based on the assumptions addressed in RSA-CEGD protocol.

4.1 Assumptions

1. We assume that A generates VRES, and sends it to B, which is responsible for VRES verification. A and B have agreed to employ an off-line STTP called T. A and B want to exchange their valuable items named *S* and *D* respectively, where *S* is a RSA signature on *D* signed by A. T is involved in data recovery of the item r.

2. Each party $i(i \in \{A,B,T\})$ has a pair of public and private RSA keys, expressed as $pk_i = (e_i, n_i)$, and $sk_i = d_i$, notes that pk_i is certified by CA and is known to the public.

3. A and T share a pair of RSA keys for purpose of data exchange, expressed as $pk_m = (e_m, n_m), sk_m = d_m$. Notes that the pair of RSA keys is not used for the entity's identification.

4. A owns a pair of assistant RSA keys after a time interval. We denote the keys as $pk_s = (e_s, n_s)$, $sk_s = d_s$. Notes that the keys are stored in a kind of a certificate C_s, denoted as $C_s = (pk_s, c_1, c_2)$, where $c_1 = E_{pkA}(d_s) = d_s^{e_A} \bmod n_A$ $c_2 = h(pk_s \| c_1)^{d_T} \bmod n_T$. Clearly, C_s can be verified by the public and only A can obtain the private key by decrypting c_1. At the initial stage, T sends C_s to B secretly.

5. Assume that $e = e_A = e_m = e_s$ and $n_A < n_s < n_m$.

4.2 Preliminaries

We summarize notations and definitions of the items used in RSA-DEMCA algorithm in the following table.

Table 1. Definitions of the items

A, B, T: a VRES creator, a verifier VRES, and STTP employed by A and B.
S: Party A's RSA digital signature on document D, denoted as $S=h(D)^{dA} \bmod n_A$
r: A random number generated by A
C_s: A kind of certificate signed by T, containing public key information of the assistant RSA pair keys. $C_s=(pk_s,c_1,c_2)$, $c_1=E_{pkA}(d_s)$, $c_2=E_{skT}(h(pk_s\|\|c_1))$
N,N_1,N_2: RSA extended modulus, where $N_1=n_A \times n_m$, $N_2=n_A \times n_s$, and $N=n_A \times n_s \times n_m$
$VRES_{T1}$: Presented VRES in the time interval T1.$y_1=r^e \bmod N_1$, $y2=S^e \bmod N_1$, $x_1=(r\times s) \bmod n_A$, $x_2=(r\times s) \bmod n_m$, $xx_1=(r\times(h(y_1)^{dm}) \bmod n_m$, $xx_2=(s\times(h(y_2)^{dm}) \bmod n_m$
$VRES_{T2}$: Presented VRES in the time interval T2.$y_3=r^e \bmod N_2$, $y_4=S^e \bmod N_2$, $x_3=(r\times S) \bmod n_s$, $xx_3=(r\times(h(y_3)^{ds}) \bmod n_s$, $xx_4=(S\times(h(y_4)^{ds}) \bmod n_s$
X,X_{T1},X_{T2}: the residue of *Val* moduli N,N_1,N_2.
$FV_1(y_1,x_1,xx_1)_{T1}$, $FV_2(y_2,x_2,xx_2)_{T1}$: Formal verification on $VRES_{T1}$.
$FV_1(y_3,x_1,xx_3)_{T2}$, $FV_2(y_4,x_3,xx_4)_{T2}$: Formal verification on $VRES_{T2}$.
$DV(X_{T1},X_{T2})$: Value judgement on VRES-ER
Val： VRES-ER, $Val=r\times s$

4.3 Formal Verification on VRES

1. $VRES_{T1}$ Formal Verification

The purpose of this verification is to make sure that $VRES_{T1}$ provided by A have expected expressions. The formal verification on $VRES_{T1}$ consists of two parts: $FV_1(y_1,x_1,xx_1)_{T1}$ and $FV_2(y_2,x_2,xx_2)_{T1}$. $FV_1(y_1,x_1,xx_1)_{T1}$ is used to check that y_1 and x_1 have the expected expressions, and we omit the process since it is similar with that of the verifications addressed in RSA-CEGED.

$FV_2(y_2,x_2,xx_2)_{T1}$: (*a*). $y_2 \bmod n_A = S^e \bmod n_A = h(D)$

(b). $xx_2^e \bmod n_m = (y_2 \times h(y_2)) \bmod n_m$

(c). $x_2^e \bmod n_m = (y_1 \times y_2) \bmod n_m$

The purpose of $FV_2(y_2,x_2,xx_2)_{T1}$ is to verify $y_2 = S^e \bmod N_1$. (a) makes sure that y_2 has contained correct *S*. (b) together with (c) confirms that the modulus operation in y_2 is based on $n_A \times n_m$, and x_2 has an expected expression.

2. $VRES_{T2}$ Formal Verification

Similar with those of $VRES_{TI}$, $VRES_{T2}$ Formal Verification consists of $FV_1(y_3, x_1, xx_3)_{T2}$ and $FV_2(y_4, x_3, xx_4)_{T2}$, and confirms that

$y_3 = r^e \bmod N_2$, $y_4 = S^e \bmod N_2$, and $x_3 = (r \times S) \bmod n_s$.

4.4 Value Judgement on VRES-ER

Let $Val = r \times S$ be VRES-ER in VRES verifications.

1. X_{T1} Computation

$$X_{T1} = (x_1 \times n_m \times (n_m^{-1} \bmod n_A) + x_2 \times n_A \times (n_A^{-1} \bmod n_m)) \bmod N_1 < n_A^2$$

2. X_{T2} Computation

$$X_{T2} = (x_1 \times n_s \times (n_s^{-1} \bmod n_A) + x_3 \times n_A \times (n_A^{-1} \bmod n_s)) \bmod N_2 < n_A^2$$

3. $DV(X_{T1}, X_{T2})$:

(a) $X_{T1} = X_{T2} < n_A^2$;(b) $X_{T1} \bmod n_s = x_3$; (c) $X_{T2} \bmod n_m = x_2$

(d). Computes and verifies

$$X = (x_1 \times n_m \times n_s \times t_1 + x_2 \times n_A \times n_s \times t_2 + x_3 \times n_m \times n_A \times t_3) \bmod N = X_{T1}$$

where $t_1 = (n_m \times n_s)^{-1} \bmod n_A$, $t_2 = (n_A \times n_s)^{-1} \bmod n_m$,

$t_3 = (n_m \times n_A)^{-1} \bmod n_s$.

If $DV(X_{T1}, X_{T2})$ =1, then $Val = r \times S < n_A^2$.

For expression convenience, we denote *Val* in $VRES_{TI}$ as $Val_{T1} = (r \times S)_{T1}$, and *Val* in $VRES_{T2}$ as $Val_{T2} = (r \times S)_{T2}$ respectively. Val_{T1} and Val_{T2} are congruence moduli *N* if (a)-(c) are positive. Thus, $Val_{T1} = kN + a$ where $k \geq 0$ and $a < n_A^2$. As the assumptions of the algorithm have described, A does not know the value of n_s when constructing $VRES_{TI}$. If $Val_{T1} > N$, A has no means to create such an integer, of which the residue modulo *N* (i.e. a) is known while *N* is dubious for A at that time. Thus, we have

$Val_{T1} = r \times S < n_A^2$ if $DV(X_{T1}, X_{T2})$ =1.

4.5 VRES Recovery

Two situations are needed to deal with since we have verified that $r \times S < n_A^2$.

1). If $r < n_A$, it is just the case that RSA-CEGD has processed. Thus T can obtain *r* by decrypting y_1, and B computes: $S = (x_1 \times r^{-1}) \bmod n_A$

2). If $S < n_A$, obviously we have $S < n_m$ since $n_A < n_s < n_m$.

As we have $x_2 = (r \times S) \bmod n_m$, B computes $S = x_2 \times r'^{-1} \bmod n_m$.

4.6 The Process Flow

1. $A \rightarrow B : x_1, x_2, xx_1, xx_2, y_1, y_2$
2. $B \rightarrow A : C_s$
3. $A \rightarrow B : x_3, xx_3, xx_4, y_3, y_4$
4. B transfers y_1 to T, in a case that A terminates process after receiving the sensible element sent by B.

5 Security Analysis

In this section, we analyze the security of RSA-DEMCA algorithm with three aspects: data confidentiality, recovery assurance and misbehaviors prevention.

1. Data confidentiality

In our algorithm, similar with RSA-CEGD protocol, we apply RSA extended modulus computations on VRES construction. As illustrated in RSA-CEGD, $y_1 = r^e \bmod N_1$ is a minor variation of RSA encryption, so it's hard for any other party to decrypt it to obtain *r*. We believe it is also hard for an adversary to factor $x_1 = (r \times S) \bmod n_A$ to obtain *r*. The conclusion is also held with respect to such items as x_2, xx_1, xx_2, y_2, $x_3, xx_3, xx_4, y_3, y_4$. Thus we believe VRES-ER$_{TI}$ and VRES-ER$_{T2}$ are secure without sensible information leakage.

In addition, VRES-ER in our algorithm expressed as $Val = r \times S$ is also hard to be factored. Based on the difficulty of the integer factorization problem (IF Problem[8]), we believe another party excluding A has no means to factor *Val* and thus achieves *S*.

2. Misbehaviors prevention

Our algorithm provides a security mechanism that can prevent successfully misbehaviors by A during VRES construction. In normal situations, VRES-ER$_{TI}$ =VRES-ER$_{T2}$, and they are irrespective to N_2, the base of the secondary RSA extended modulus computations. Once A is a malicious one, he has no means to launch RSA extended modulus attacks by forging VRES$_{TI}$ in a larger area. Firstly, VRES$_{TI}$ and VRES$_{T2}$ have been verified formally to make sure that presented VRES have expected expressions. Secondly, by performing consistency check, B is then sure that VRES-ER$_{TI}$ and VRES-ER$_{T2}$ are consistent and irrespective to N_2, which is still invisible to A when he constructs VRES$_{TI}$. Thus A has to provide a correct VRES for purpose of verification requirements.

Thus we provide an intrusion-aware security mechanism to detect three kinds of possible misbehaviors by a VRES creator: (1). Presented items with unexpected expressions. (2). Inconsistent elements e used to construct VRES. (3). Constructed VRES-ER beyond the expected area.

On the other hand, as a verifier of VRES, B has no chance to act as an adversary. In our algorithm, B only transmits an appointed certificate *Cs* to A.

3. Recovery assurance

Due to constraints of RSA encryption scheme, a trusted third party would fail to recover a complete sensible element for the VRES verifier. However, that does not influence information assurance on VRES recovery, since value judgement on VRES-ER is presented. Furthermore, even we lack an effective scheme to check if a sensible element is within an expected area, VRES recovery can also be achieved successfully based on assurance of an expected VRES-ER.

6 Conclusion

It's essential to trade off the contradiction between the trust model and constraints of adapted security mechanisms on algorithm design of fair exchange protocols. On one hand, RSA encryption and signature schemes are performed in modulus computations, thus there exist infinite solutions satisfying verification requirements. On the other hand, the exchanging parties are assumed to be active adversaries, having abilities to forge sensible information without any restrictions.

According to the characteristics of RSA extended modulus attacks, only does the VRES creator know the bases of RSA cryptosystems, then he can launch such a kind of attacks. Motivated by the idea of "hiding" effective information (i.e. n_s) from the VRES creator after a time interval, we actually restrict the forging abilities of the VRES creator successfully in RSA-DEMCA algorithm.

References

[1] Ateniese, G.: 'Efficient Verifiable Encryption (and Fair Exchange) of Digital Signatures,' Proc. ACM Conference on Computer and Communications Security, Singapore, November 1999, pp. 138-146.

[2] Chen, L.: 'Efficient Fair Exchange with Verifiable Confirmation of Signatures,' Proc. Advances in Cryptology - ASIACRYPT '98, Springer-Verlag, Berlin, Germany, 1998, pp. 286-99.

[3] Nenadic, A., Zhang, N., Barton, S., A Security Protocol for Certified E-Goods Delivery, Proceedings of IEEE International Conference on Information Technology, Coding and Computing (ITCC 2004) - Information Assurance and Security Track, Las Vegas, Nevada, USA, IEEE Computer Society, 2004, pp. 22-28.

[4] ZHOU YB, ZHANG ZF, QING SH, JI QG. A Fair Exchange Protocol Based on RSA Signature Scheme. Journal of Software, 2004,15(07):1049-1055.

[5] Franklin, M., Reiter, M.: 'Fair Exchange with a Semi-Trusted Third Party,' Proc. ACM Conference on Computer and Communications Security, Zurich, Switzerland, April 1997, pp. 1-5.

[6] Bao, F., Deng, R.: 'An Efficient Fair Exchange Protocol with an Off-Line Semi-Trusted Third Party,' Proc. International Workshop on Cryptographic Techniques and E-Commerce, 1999, pp. 37-47.
[7] Schneier, B.: 'Applied Cryptography,' John Wiley & Sons, 1996.
[8] Wenbo Mao. Modern Cryptography: Theory and Practice[M]. Publishing House of Electronics Industry, 2004.

Using Ambient in Computational Reflection Semantics Description*

Jianghua Lv[1,2], Shilong Ma[1], Aili Wang[3], and Jing Pan[4]

[1] National Lab of Software Development Environment,
Beijing University of Aeronautics and Astronautics, Beijing, China 100083
{jhlv, slma}@nlsde.buaa.edu.cn
[2] College of Computer Science and Technology, Jilin University,
Changchun, China 130012
lvjh@jlu.edu.cn
[3] 307 Military Hospital, Beijing, China 100039
ailimoon@sina.com
[4] School of Management, University of Science and Technology Beijing,
Beijing, China 100083
panjing@manage.ustb.edu.cn

Abstract. With the development of wide-area network, distribution and mobility have become the main character of computation. Similarly, reflection systems are mostly like to implement under this environments, as a result how to describe reflection semantics in distributed and mobile environment is necessary indeed to understand and automatically generate reflection mechanism. We give a new semantics description for distributed computational reflection system in ambient calculus and also we give the proof to verify our approach.

1 Motivation

Currently, numerous new technologies have been used in either computer software or hardware in order to enhance computer system's expansibility or performance further. Most current systems may be dynamically reconstructed or extended to perfect themselves as possible. Among these, computational reflection [1] that the possibility of a software system to inspect and modify itself at runtime became a feasible mechanism to meet the needs of strong adaptability conditions for heterogeneous environment and computing system. As a result, it is gaining interest in more and more practical applications. Especially with the development of wide-area network, computation scale up to widely distributed, intermittently connected and well administered computational environments which bring new moment for computational reflection.

As a new programming technology that allows a user to extend and modify the inner information of systems, computational reflection has become an important issue in the

* Supported by the National 973 Project under the grant number G1999032701 and China Postdoctoral Science Foundation.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 1105 – 1114, 2005.

field of computing architecture, many researchers investigated its powerful capability and emphasized its usefulness [4], [5], Concretely, the JAVA programming environment [6] relies heavily on the use of reflection for the implementation of the JAVABEANS and RPC component models. Furthermore, adaptability is a prime requirement of middleware systems and several groups are therefore doing research on reflective middleware [7], [8]. However, since reflective systems offer different reflection programming interfaces, the design of reflection mechanism is subject to a number of constraints relating to expressive power, efficiency, security properties and so on. So in different applications, these constraints are different from one application to another. In order to unify these constraints and describe generic character of computational reflection, we should give a common semantic framework for reflection, which on the one hand makes object systems be nested or extended reflection easily and on the other hand makes computational reflection be understood and generated easily.

Nevertheless it is difficult to describe reflection in conventional formalization methods since it involves control actions which transform between base level and meta-level. It is also a factor that discussions on reflection semantics are not as so many as on reflection implement. Malmkjaer [9] presents an un-strict framework for reflective tower based on meta-level access. Hook and Sheard [10] introduce a formal semantic of compile-time reflective ML, which is a reflective language with compile-time type checking and run-time type safety. It supports compile-time reflection. Malenfant [11] studies a new model for behavioral reflection based on meta-objects in a prototype-based programming language, and then gives a formal semantics of this protocol using the theory of priority rewrite system. Douence [12] presents present a reification mechanism for object-oriented interpreters based on program transformation techniques. Based on extensible denotational semantics, we give an extensible semantics framework for reflection in [13].

In this paper, we present formalism for reflection based on mobile ambient [2] and safe ambient [3]. And a type system will be given in order to express and verify sound reflection mechanism. The paper is structured as following: in section 2, we present how to describe computation reflection according to mobile ambient idea –CRA and following with an example to show how to describe it in CRA. To verify CRA, in section 3 typing system of CRA is given to ensure sound static semantic for CRA. At last we conclude our work in the paper in section 4.

2 Computational Reflection Ambient – CRA

2.1 CRA

In a conventional system, computation is performed on data which represent entities, and it is external to the computation system. In contrast, a reflective computation system must contain some data that represent the structural and computational aspect of the system itself, and such data must be manipulated within the system itself, and more importantly, changes made to such data must be causally connected to the actual computation being performed. As a result, to realize reflection in a system, the causally connection between the system and its self-representation must exist within the system.

Algebraically speaking, a self-representation and the computation represented by the self-representation are surjective and satisfy Glois connection [14]. Since reflective computation depends on the way in which self-representation is described, the expression form of self-representation is the primary concern when a reflective system is being built.

Based on above description, a reflective program is a particular architecture in which all programs are not executed by the primitive and inaccessible interpreter, but rather be performed by the explicit running of a program that represents that interpreter, which is usually written in the same languages as the user program for efficiency. The reflection structure can be shown in Fig. 1.

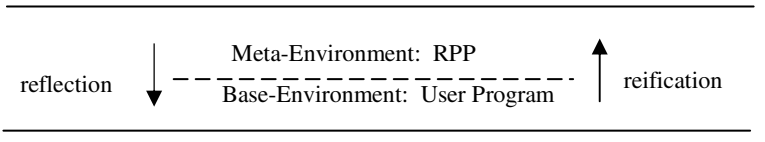


Fig. 1. Reflection Structure-- the explicit interpreter is called as Reflective Processor Program (RPP) executed in meta-environment. To implement explicit access to the inner information, two meta-level processes are needed: reification and reflection. Reification makes the current inner information of the entity available and conversely, reflection returns the entity that is represented by current information

In order to ensure safe access to meta-environment or base environment, for every action, we define synchronic mechanism between meta-environment and base environment, namely, whether an operation is executed depends on the synchronization of the both levels. For example, if in current level *m* a process is needed to deliver to another level *n*, the transfer is executed only when there is a synchronism between *m* and *n* that current capability are respectively request for entering n in *m* and acceptance for entering from *m* in *n*. So we define following co-capabilties as prefix to processes:

in↗*: entering meta environment;*
in ↘ *: entering base environment;*
m↗*in: accepting m that comes from base environment;*
m↓*in: accepting m that comes from meta environment*
update m : updating m ambient;
update{m} : using m to update ambient
open m: opening m's boundary;
open{m}: accepting meta ambient m to open ambient

And reduction rules are:

$m[in\nearrow \bullet P] \mid n[m\nearrow in \bullet R] \rightarrow n[m[P] \mid R]$
$m[n\downarrow in \bullet P] \mid n[in\searrow \bullet R] \rightarrow m[P \mid n[R]]$
$n[m[open\{n\} \bullet P] \mid open\ m \bullet R] \rightarrow n[P \mid R]$
$m[update\{n\} \bullet P] \mid n[(update\ m \bullet R)]] \rightarrow m[P] \mid m \bowtie n[R]$

$\bowtie$ *denotes* $\nwarrow$ *or* $\swarrow$

Function $\ltimes$: Env$\times$MEnv $\rightarrow$Env, is defined to denote using new meta environment to update base environment. Giving definition above, reflection and reification processes can be defined as:

$$\varphi(m[P]) \triangleq m[(in\nearrow \bullet\ open\{\overline{m}\}\bullet P)] \mid \overline{m}\,[m\nearrow in\bullet\ open\ m\bullet 0]$$

$$\psi(\overline{m}\,[R]) \triangleq m[\overline{m}\downarrow in\ \bullet open\,\overline{m}\,\bullet 0] \mid \overline{m}\ \ [(in\searrow \bullet\ open\{m\}\bullet\ R)]$$

However, $\overline{m}$ is interpreter representation of m which means that any computation should result in same answer whenever it is executed in m or $\overline{m}$. Therefore any changes taken on base environment m should affect m's meta-environment $\overline{m}$, and the reverse is the same. Function $\ltimes$ updating base environment using meta-environment guarantees that changes on meta-environment casually affect base environment. Then how to guarantee the reverse?

It is known that in reflection system, computations are not always taken in meta-level but in base level for execution efficiency, except that meta-actions are needed to access to meta-level. This means some time the meta-environment is not casually connected with base environment. In order to ensure the accuracy of meta-computation, we define function $\rtimes$: Env $\times$ MEnv $\rightarrow$ MEnv to update meta-environment using current base environment, when reification occurs, $\rtimes$ is applied first. Hence though base environment and meta-environment are not casually connected in microcosmic view, but for any computation taken into the two environments, it results in the same answer, furthermore if $\varphi \circ \psi$ and $\psi \circ \varphi$ are identity functions, then we can say they are casually connected microscopically, where "$\circ$" denotes the compound of functions. Based on the definitions above, we prove that our definitions satisfy Gloris connection [in appendix]. Consequently, Reflection and reification processes can be defined as:

$$\varphi(m[P]) \triangleq m[(update\,\overline{m}\bullet in\nearrow \bullet\ open\{\overline{m}\}\bullet P)] \mid \overline{m}\,[update\{m\}\bullet m\nearrow in\bullet\ open\ m\bullet 0]$$

$$\psi\ (\overline{m}\,[P]) \triangleq m[\overline{m}\downarrow in\bullet\ update\{\overline{m}\}\bullet open\ \overline{m}\,\bullet 0] \mid \overline{m}\,[(in\searrow \bullet\ update\ m\bullet open\ \{m\}\bullet P)]$$

In Fig. 2, we define CRA's syntax and semantics. Structural congruence of Computational reflection ambient can be easily derived from the structural congruence in [2], so here for space reason we ignored. Reduction rules and reflection function φ and ψ are given in Fig. 3.

2.2 A Java Example in CRA

In fact our CRA is a generic framework for computational reflection, the reflection operations are not defined in detail, and they depend on users. For example an ambient may be thought as a location where computations are carried, it concludes static environment and dynamic state information that are needed when these computations are executed, and ambients are divided into two kinds: base-ambients and meta-ambients. Conventional computations are executed in base-ambient, while computations involving internal information are submitted to be executed in meta-ambients.

As an example, in Java, a class's compile information is stored in class file with extensible name".class", Java API is used for users to define compile-time reflection in order to access or modify compile-time information. The sample program referenced from [15] modifies the width field of a Rectangle object by invoking the set method in reflective API. Since the width is a primitive type, an *int*, the value passed by set is an Integer, which is an object wrapper.

Capability M ::= in↗ entering meta environment;
| in↘ entering base environment;
| m↗ in accepting m that comes from base environment;
| m↘ in accepting m that comes from meta environment
| update m updating m ambient;
| update{m} using m to update ambient
| open m opening m's boundary;
| open{m} accepting meta ambient m to open ambient
| M•M path connection

Process P ::= 0 null process
| M•P capability
| m[P] ambient
| P1 | P2 parallel composition
| (νn : A)P restriction
| rec X•P recursion

Fig. 2. Syntax for Computational Reflection Ambient— a system is composed of processes, X is process variable. As for reflective expressions, it depends on users in detail definition and in concrete system

---

$m[in\nearrow \bullet P] \mid \overline{m}\,[m\nearrow in \bullet R] \rightarrow \overline{m}\,[\,m[P] \mid R\,]$

$m[n\searrow in \bullet P] \mid \overline{m}\,[\,in\searrow \bullet R] \rightarrow m[\,P \mid \overline{m}\,[Q]\,]$

$n\,[m[open\{n\}\bullet P] \mid open\ m \bullet R] \rightarrow n\,[\,P \mid R\,]$

$m\,[update\{\overline{m}\}\bullet P] \mid \overline{m}\,[update\ m \bullet R] \rightarrow (m \ltimes \overline{m}\,)\,[P] \mid \overline{m}\,[R]$

$m\,[update\,\overline{m} \bullet P] \mid \overline{m}\,[update\ \{m\} \bullet R] \rightarrow m[P] \mid (m \rtimes \overline{m}\,)[R]$

$\varphi\,(m[P]) \triangleq m[(update\,\overline{m} \bullet in\nearrow \bullet open\{\overline{m}\}\bullet P)] \mid \overline{m}\,[update\{m\}\bullet m\nearrow in \bullet open\ m \bullet 0]$

$\psi\,(\overline{m}\,[P]) \triangleq m[\,\overline{m}\searrow in \bullet update\{\overline{m}\}\bullet open\,\overline{m}\,\bullet 0] \mid \overline{m}\,[(in\searrow \bullet update\ m \bullet open\{m\}\bullet P)]$

Fig. 3. Reduction rules and definitions of reflective functions in Computational Reflection Ambient are shown

```
import java.lang.reflect.*;
import java.awt.*;
class SampleSet {
  public static void main(String[] args) {
      Rectangle r = new Rectangle(100, 20);
```

```
      System.out.println("original: " + r.toString());
      modifyWidth(r, new Integer(300));
      System.out.println("modified: " + r.toString());
  }
  static void modifyWidth(Rectangle r, Integer widthParam){
        Field widthField;
        Integer widthValue;
        Class c = r.getClass();
        try {
           widthField = c.getField("width");
           widthField.set(r, widthParam);
        } catch (NoSuchFieldException e) {
            System.out.println(e);
          }
    }
  }
```

The output of the sample program verifies that the width changed from 100 to 300:

original: java.awt.Rectangle[x=0, y=0, width=100,height=20]

modified: java.awt.Rectangle[x=0, y=0, width=300,height=20]

SampleSet. Class is the meta-level of SampleSet, which concludes internal compile-time information of class SampleSet. Because procedure modifyWidth involves meta-accessing operations, it will be reified into meta-ambient SampleSet.Class. ModifyWidth is also an ambient that is needed to confine and prevent from the contents of modification interfering with outer processes. Hence, we can describe above programs into computational reflection ambient briefly:

$$\psi \circ \varphi(SampleSet\ [ModifyWidth])$$

$$\equiv \psi(SampleSet[(update\ SampleSet.Class \bullet in\nearrow \bullet open\ \{SampleSet.Class\} \bullet\ ModifyWidth)]\ |\ SampleSet.Class[update\{SampleSet\} \bullet SampleSet\nearrow in \bullet\ open\ SampleSet \bullet 0])$$

Where the detail definition of ModifyWidth is just as conventional programs done as described in [2]. We focus on reflection mechanism here, so the conventional part are neglected to talk about.

3 Tying System

Tying system grammar is given in Fig.4 including ambient type, process type and exchange type. In order to distinguish different level processes, for example P is in meta-ambients, we define its type as $\overline{\text{Prc}}\,[T]$ rather than define a new structure to denotes meta-ambient type. Namely, we have $\overline{\text{Amb}[T]} \equiv \text{Amb}[\,\overline{T}\,]$.

The rules defining the typing judgments are given on Fig. 5. Γ represents type environment which is a list of the form E: T, denoting T is the type of the specified entity E.

A	::= Amb[T]	ambient type
U	::= Prc[T]	process type
	$\overline{\text{Prc}}$ [T]	meta-process type
S, T	::= Shh	no exchange
	U , A	
	$\overline{\text{T}}$	

Fig. 4. Types grammar for computational reflection ambient

$$\Gamma\text{-NIL}\ \ \Gamma \vdash 0 : U \qquad \Gamma\text{-VAR}\ \frac{\Gamma(X) = U}{\Gamma \vdash X : U} \qquad \Gamma\text{-NAME}\ \frac{\Gamma(m) = A}{\Gamma \vdash m : A}$$

$$\frac{\Gamma \vdash m\text{: Amb[T]}, P\text{: Prc[T]}}{\Gamma \vdash m[P]\text{: Prc[T]}} \qquad \Gamma\text{-RES}\ \frac{\Gamma, m\text{: A} \vdash P\text{: U}}{\Gamma \vdash (\nu n\text{: A})P : U}$$

$$\Gamma\text{-PAR}\ \frac{\Gamma \vdash P_1\text{: U}, P_2\text{: U}}{\Gamma \vdash P_1 | P_2\text{: U}} \qquad \Gamma\text{-REC}\ \frac{\Gamma, X\text{: U} \vdash P\text{: U}}{\Gamma \vdash \text{rec X P : U}}$$

$$\frac{\Gamma \vdash P\text{: U}}{\Gamma \vdash \text{in}\nearrow \bullet P\text{: } \overline{U}} \qquad \Gamma\text{-}^{\text{IN}}\ \frac{}{}$$

$$\Gamma\text{-COIN}\ \frac{\Gamma \vdash P\text{: } \overline{U}}{\Gamma \vdash \text{in}\searrow \bullet P\text{: U}} \qquad \Gamma\text{-}_{\text{IN}}\ \frac{\Gamma \vdash P\text{: U}, m\text{: A}}{\Gamma \vdash m\diamondsuit \text{in}\bullet\ P : U}$$

$$\Gamma\text{-COUPD}\ \frac{\Gamma \vdash P\text{: U}, m\text{: A}}{\Gamma \vdash \text{update } \{m\}\bullet\ P : U} \qquad \Gamma\text{-UPD}\ \frac{\Gamma \vdash P\text{: U}}{\Gamma \vdash \text{update n}\bullet P\text{: U}}$$

$$\Gamma\text{-COPEN}\ \frac{\Gamma \vdash P\text{: Prc[S]}, m\text{: Amb[T]}}{\Gamma \vdash \text{open } \{m\}\bullet\ P\text{: Prc[T]}} \qquad \Gamma\text{-OPEN}\ \frac{\Gamma \vdash P\text{: U}}{\Gamma \vdash \text{open n}\bullet\ P : U}$$

Fig. 5. Typing Rules

As we know that reflective functions transform between base environment and meta-environment, the type of φ should be: Amb[S] → Amb[$\overline{\text{S}}$] and ψ should have type: Amb[$\overline{\text{S}}$] → Amb[S], where S denotes the type of process. Now we use type system defined above to verify our reflective functions φ andψ. In order to illustrate the typing process pithily, we give a delaminated structure:

$$\frac{\frac{a_1}{a_2}\ \ldots\ l\ level}{a_3}\ \ldots\ s\ level \qquad \ldots\ m\ level$$

a_1, a_2, a_3 are assertions, in the delaminated structure an assertion in a level l is the condition of the assertion in s level if exists level s down l and at the same time it is also the conclusion of the assertion in l's up level m if exists up level m. For example, under

the condition of a_1, a_2 holds and under the condition of a_2, we can gain a conclusion a_3, and we use a longer line to denote down direction reduction. Hence, based on typing rule defined above, we have:

$$\varphi(m[P]) \triangleq m[(update\ \overline{m} \bullet in\nearrow \bullet\ open\{\overline{m}\}\bullet P)] \mid \overline{m}\,[update\{m\}\bullet m\nearrow in\bullet\ open\ m\bullet 0]$$

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash P: Prc[S],\ \overline{\mathrm{m}} : Amb[\overline{\mathrm{S}}\,]}{\Gamma \vdash open\ \{\overline{\mathrm{m}}\,\}\bullet\ P : Prc[\overline{\mathrm{S}}\,]}}{\Gamma \vdash in\nearrow \bullet\ open\{\overline{m}\,\}\bullet P:\ \overline{\mathrm{Prc}}\,[\overline{\mathrm{S}}\,]}}{\Gamma \vdash update\ \overline{m} \bullet in\nearrow \bullet\ open\{\overline{m}\,\}\bullet P :\ \overline{\mathrm{Prc}}\,[\overline{\mathrm{S}}\,]}$$

$$\psi(\overline{m}\,[P]) \triangleq m[\overline{m} \searrow in\bullet\ update\{\overline{m}\,\}\bullet open\ \overline{m}\ \bullet 0] \mid \overline{m}\,[(in\searrow \bullet update\ m\bullet open\{m\}\bullet P)]$$

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash P:\ \overline{\mathrm{Prc}}\,[\overline{\mathrm{S}}\,],\ m: Amb[S]}{\Gamma \vdash open\{m\}\bullet\ P :\ \overline{\mathrm{Prc}}\,[S]}}{\Gamma \vdash update\ m\bullet\ open\ \{m\}\bullet P:\ \overline{\mathrm{Prc}}\,[S]}}{\Gamma \vdash in\searrow \bullet\ update\ m\bullet\ open\ \{m\}\bullet P:\ Prc[S]}$$

We can see that these reflective functions transform a process of type Prc[S] (or $\overline{\mathrm{Prc}}\,[\overline{\mathrm{S}}\,]$) in base (or meta) environment to a process of type $\overline{\mathrm{Prc}}\,[\overline{\mathrm{S}}\,]$ (or Prc[S]) in meta (or base) environment.

4 Conclusion

In the paper based on the principle of mobile ambient we give computational reflection ambient – CRA to describe reflection computation. Compared to other similar works in reflection description, we based on ambient calculus to formalize reflection mechanism, and at the same time in order to enable security of accessing to meta-level information, in CRA co-capabilities are defined. At last typing rules are given verify the CRA.

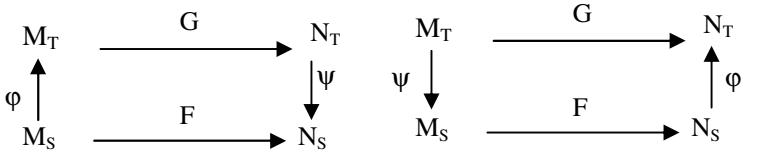
In CRA we imply that the representation of source programs is shown in the form of source programs, it is intuitive to comprehend reflection without additional interpreter to execute this representation program.

References

1. Brian Cantwell Smith. Reflection and Semantics in Lisp. Technical Report CSLI84 -8, Stanford University Center for the Study of Language and Information, December 1984.
2. Luca Cardelli, Andrew D. Gordon. Mobile Ambient. In: Proceedings of FOSSACS'98. Lecture notes in computer science, Vol. 1378. Springer, Berlin Heidelberg New York, pp 140–155.
3. Levi F, Sangiorgi D (2000) Controlling Interference in Ambients. In: Proceedings of the symposium on principles of programming languages. ACM Press, New York, pp 352-364.

4. D.P.Friedman, M.Wand. Reification: Reflection without Metaphysics. In: ACM Conference Proceedings of Lisp and Functional Programming, pp. 348--355, 1987.
5. Maes.Pattie. Concepts and Experiments in Computational Reflection. In: OOPSLA'87 Conference Proceedings, pp 147-155, 1987.
6. JAVA home page. Sun Microsystems, Inc. http://java.sun.com.
7. G. Blair, R. Campbell (chairs). Workshop on Reflective Middleware, 2000. http://www. comp. lancs. ac.uk/computing/RM2000/
8. P. Cointe (ed.). Proceedings of Reflection'99, LNCS 1616, Springer Verlag, 1999.
9. Karoline Malmkjaer. On Some Semantic Issues in the Reflective Tower. In Mathematical Foundations of Programming Semantics. (Lecture Notes in Computer Science, vol. 442) (1989), M. Main, A. Melton, M. Mislove, and D. Schmidt, Eds., pp. 229--246.
10. James Hook, Tim Sheard. A Semantics of Compile-time Reflection. Technical Report 93-019, Department of Computer Science and Engineering, Oregon Graduate Institute, November 1993.
11. Jacques Malenfant. A Semantics of Introspection in a Reflective Prototype-Based Language. Lisp and symbolic computation, 9(2/3):153--180, 1996.
12. Rémi Douence, Mario Südholt. A Generic Reification Technique for Object-Oriented Reflective Languages.
13. Jianghua Lv, Shilong Ma. Towards an Extensible Semantics for Reflection. Accepted by International Conference on Information Technology: Coding and Computing 2005. onsored by IEEE Computer Society. To appear.
14. Amr Sabry, Philip Wadler. A Reflection on Call-by-Value. ACM Transactions on Programming Languages and Systems, 19(6): 916-941, Nov. 1997.
15. The JavaTM Tutorial. http://java.sun.com/docs/books/tutorial/reflect/
16. Luca Cardelli, Andrew D. Gordon. Types for Mobile Ambients. In Proc. 26th POPL, pages 79-92. ACM Press, 1999.

Appendix



Definition 1. Maps φ and ψ form a Galois connection from source domain S to target domain T whenever

$$M_S \rightarrow^*_S \psi(N_T) \quad \text{if and only if} \quad \psi(M_S) \rightarrow^*_T N_T$$

Assume $\rightarrow_S$ for reduction relation of S and $\rightarrow_T$ for T.

Proposition 1. Map φ and ψ form a Galois connection from S to T if and only if the following four conditions hold

(1) $M_S \rightarrow^*_S \psi \circ \varphi\, M_S$ (2) $\varphi \circ \psi\, N_T \rightarrow^*_T N_T$

(3) $M_S \rightarrow^*_S N_S$ implies $\varphi\, M_S \rightarrow^*_T \varphi\, N_S$

(4) $M_T \rightarrow^*_T N_T$ implies $\psi\, M_T \rightarrow^*_S \psi\, N_T$

Definition 2. Maps φ and ψ form a reflection in S of T if they form a Galois connection and $M \equiv \psi \circ \varphi M$.

Proposition 2. Let φ and ψ form a reflection in S of T, then for F: $M_S \rightarrow_S N_S$, and G: $M_T \rightarrow_T N_T$, satisfy:

(1) $\psi \circ G \circ \varphi = F$ (2) $\varphi \circ F \circ \psi = G$

Proposition 3. In CRA, and form a reflection in base ambient of meta-ambient.

Proof: if G or F function affects its own environment respectively, the update operation ⋊ or ⋉ will be applied to ensure base environment to be consisted with meta-environment, so we only need to prove that ψ and φ satisfy proposition 2 when G and F have no effect on its own environment. $\overline{P} \equiv \overline{m}[P]$, Since F and G are functions in base and meta-level environments respectively, so given $P \in m$ and $\overline{Q} \in \overline{m}$ we have $F(m[P]) \equiv F(P)$ and $G(\overline{m}[Q]) \equiv G(\overline{Q})$.

(1) $\psi \circ G \circ \varphi = F$

Assume that on meta-level there exists a function G: $\overline{P} \rightarrow \overline{Q}$, $\varphi(P) = \overline{P}$ and $\psi(\overline{Q}) = Q$, our target is to prove that on base level function F: $P \rightarrow Q$ holds. Based on the definition of φ and ψ above, we have:

$\psi \circ G \circ \varphi(m[P])$

$\equiv \psi \circ G$ $(m[(update\ \overline{m} \bullet in\nearrow \bullet open\{\overline{m}\} \bullet P)] \mid \overline{m}[update\{m\} \bullet m\not\in in \bullet open\ m \bullet 0])$

$\rightarrow^* \psi \circ G$ $(\overline{m}[P]) \equiv \psi \circ G(\overline{P})$

$\rightarrow \psi(\overline{Q}) \equiv \psi(\overline{m}[Q])$

$\equiv m[\overline{m} \not\searrow in \bullet update\{\overline{m}\} \bullet 0] \mid \overline{m}[(in\searrow \bullet update\ m \bullet Q)]$

$\rightarrow^* m[Q]$

(2) $\varphi \circ F \circ \psi = G$

Assume that on base level there exists a function F: $P \rightarrow Q$, $\varphi(Q) = \overline{Q}$ and $\psi(\overline{P}) = P$, our target is to prove function G: $\overline{P} \rightarrow \overline{Q}$ holds on meta-level. Based on the definition of φ and ψ above, we find:

$\varphi \circ F \circ \psi(\overline{m}[P])$

$\equiv \varphi \circ F(m[\overline{m} \not\searrow in \bullet update\{\overline{m}\} \bullet 0] \mid \overline{m}[(in\searrow \bullet update\ m \bullet P)])$

$\rightarrow^* \varphi \circ F(m[P]) \rightarrow \varphi(m[Q])$

$\equiv m[(update\ \overline{m} \bullet in\nearrow \bullet open\{\overline{m}\} \bullet Q)] \mid (m \rtimes \overline{m})[update\{m\} \bullet m\not\in in \bullet open\ m \bullet 0]$

$\rightarrow^* \overline{m}[Q]$ □

Here we only prove the part involving reflective mechanism, others proof can be easily referred from [16].

Energy Aware Routing Based on Adaptive Clustering Mechanism for Wireless Sensor Networks*

Sangho Yi, Geunyoung Park, Junyoung Heo[1], Jiman Hong[2], Gwangil Jeon[3], and Yookun Cho[1]

[1] Seoul National University
{shyi,gypark,jyheo,cho}@ssrnet.snu.ac.kr
[2] KwangWoon University
gman@daisy.kw.ac.kr
[3] Korea Polytechnic University
gijeon@kpu.ac.kr

Abstract. The main goal of research concerning energy aware routing algorithm for wireless sensor network is to increase the lifetime and long-term connectivity of the wireless sensor networks. However, most of energy aware routing algorithms do not take into account the clustering mechanism efficiently. In this paper, we present an efficient energy aware routing algorithm for the wireless sensor networks. In our algorithm, the data aggregation technique and adaptive clustering mechanism are considered for reducing and compacting the cumulative size of packets on the wireless sensor networks. Simulation results show that the energy usage of EAR-ACM is significantly reduced compared with the previous clustering based routing algorithm for the sensor networks.

1 Introduction

Wireless sensor networks typically consist of hundreds or thousands of sensor nodes deployed in a geographical region to sense events. Wireless sensor networks provide a high-level description of the event being sensed. They are used in many applications such as environmental control, offices, robot control, and automatic manufacturing environments, and can be used even in harsh environments[1,2]. Developing wireless sensor networks entails significant technical challenges due to the many environmental constraints.

Recent advances in sensor technology, low power electronics, and low-power RF design have enabled the development of relatively inexpensive and low-power wireless sensor network[2]. However, power is a scarce yet critical resource in battery-powered sensor networks and efficient and utilization of battery power becomes an important issue because most wireless sensor nodes today are powered by batteries, It is desirable to make such nodes as energy efficient as possible and rely on their large numbers to obtain high quality results.

* The present research was conducted by the Research Grant of KwangWoon University in 2005, and was supported in part by the Brain Korea 21 project.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 1115–1124, 2005.

Routing protocols for the wireless sensor networks are also guilty of expending energy power needlessly. Most existing routing protocol used in the sensor networks do not take power into consideration, and could therefore result in either parts of, or the whole sensor network being temporarily unavailable. In most of those protocols the paths are computed based on minimizing the hop count or delay. Thus, some nodes, become responsible for routing packets from many source-destination pairs. Over time, the energy reserves of these nodes will become deleted, resulting in node failure.

Most previous researches related to routing[3,4,5] have been focused on the algorithm design and performance evaluation in terms of the packet overhead and loss rate. On the other hand, many routing algorithms have been proposed in [1,2,6,7,8,9] in order to improve the scalability of routing algorithms for large sensor networks. However, previous works did not take into account the residual energy of each sensor and data aggregation simultaneously while sending a packet.

Selecting a route that consumes the least amount of energy as possible requires that the information on all of the candidate routes be known as accurately as possible. This in turns entails high cost for transmitting a single packet. Therefore, the probabilistic approach where a route is selected based on certain criteria is more widely used. Therefore, the parameters and the constraints affect performance very much. In addition, amount of transmission of a packet in the wireless sensor networks is closely related to the lifetime of sensor network because the energy usage of the transmitting 1 bit of data is much greater than processing it[10]. Therefore, energy aware routing with data aggregation that is highly correlated with energy consumption, is a critical factor determining the performance of the wireless sensor networks.

In this paper, we present an efficient energy aware routing algorithm for the wireless sensor networks. In our algorithm, adaptive clustering mechanism is considered for reducing and compacting the cumulative size of packets on the wireless sensor networks. The proposed algorithm can provide much longer connectivity and lower energy usage of the wireless sensor networks through efficient use of energy among the nodes in the wireless sensor networks.

The rest of the paper is organized as follows. In section 2, we present related works. Section 3 describes the energy aware routing algorithm proposed by Shah and Rabaey in [1] in detail and a new energy aware routing algorithm based on adaptive clustering mechanism. Section 4 presents and evaluates the performance of the proposed energy aware routing algorithm against previous clustering based routing algorithm. Finally, some conclusions are given in Section 5.

2 Related Works

In this section, we present a brief overview of the proposed energy-aware routing algorithms. Considerable research efforts[1,2,3,4,11,12,13] have been made to reduce the energy consumption of the wireless sensor networks.

In [3], Ganesan et al. proposed the use of braided multipaths instead of completely disjoint multipaths in order to keep the cost of maintaining the multipaths low. The costs of such alternate paths are also comparable to the primary path because they tend to be much closer to the primary path.

In [4], Chang and Tassiulas proposed an algorithm for maximizing the lifetime of a network by selecting a path whose nodes have the largest residual energy. In this way, the nodes in the primary path retain their energy resources, and thus avoid having to continuously rely on the same route. This contributes to ensuring longer life of a network.

LEACH(Low Energy Adaptive Clustering Hierarchy)[2] has been introduced for the sensor networks where an end user wants to monitor an environment remotely. In such a situation, a packet from an individual node must be sent to a central base station, often located far from a sensor network, through which the end-user can access the packet. LEACH includes distributed cluster formation, local processing to reduce global communication, and randomized rotation of the cluster-heads. Together, these features allow LEACH to achieve the desired properties. LEACH-C[11], an improved scheme of LEACH, was proposed. In LEACH-C, cluster formation is made by centralized algorithm at the base station.

In [12], Lindsey et al. proposed PEGASIS which improved the LEACH more than LEACH-C. In PEGASIS, each node communicates only with close neighbors and only one designated node sends the combined data to the base station. In [13], Bandyopadhyay and Coyle proposed randomized clustering algorithm to organize the sensors into clusters in wireless sensor network. They showed computation of the optimal probability of becoming a cluster head.

3 Energy Aware Routing with Adaptive Clustering Mechanism

In this section, we describe Energy Aware Routing with Adaptive Clustering Mechanism(EAR-ACM) algorithm in detail.

3.1 Energy Aware Routing

Energy Aware Routing(EAR)[1] was proposed by Shah and Rabaey. EAR is an energy efficient routing for the wireless sensor networks. The primary goal of EAR is to improve the survivability of the networks. For this, EAR occasionally uses suboptimal paths rather than always using the optimal path. These paths are selected with the energy-based probabilities as explained above. This is intended to slow down the depletion of the energy of the nodes across networks. As a consequence, the entire networks will have a longer lifetime than that using the algorithms such as Directed Diffusion[14].

EAR finds multiple routes, if any, from source to destination nodes. Each route is assigned a probability of being selected to transmit a packet, based on residual energy and the energy for communications at the nodes along the route.

Then, based on these probabilities, one of the candidate routes is chosen in order to transmit a packet. The probability is proportional to the energy level at each node, so the route with high energy is more likely to be selected than the route with low energy level.

The operation of EAR consists of three phases. In *S*etup phase, a destination node initiates a route request and a routing table is built up by finding all the paths from a source to the destination and their energy cost. In *D*ata Communication phase, data packets are sent from the source to the destination. Each of the intermediate nodes forwards the packet to a neighboring node that is chosen randomly based on the probability computed in the setup phase. In *R*oute Maintenance phase, local flooding is performed to keep all the paths alive[1].

3.2 Overhearing

Concept of Overhearing. In the wireless sensor networks, radio wave is a physical media of communications. Radio wave propagates in every direction. Hence, when two nodes communicate with each other, the neighboring nodes of a sender can hear the packet being transmitted. Figure 1 shows this situation.

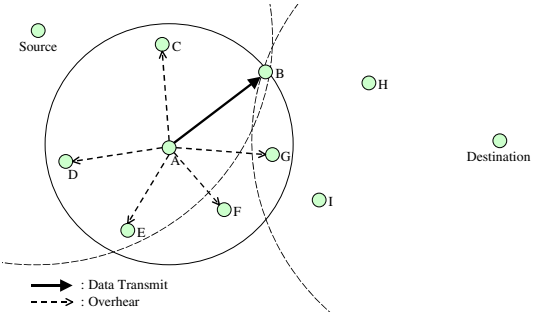


Fig. 1. Data communication in the wireless Sensor networks

In Fig. 1, node A sends a packet to node B, and then all nodes in the circle, the center of which is the location of A and the radius of which is the distance between A and B (C,D,E,F and G in this figure) can hear the data packet. In general cases, such data heard unintentionally is ignored. In Fig. 1, for example, nodes $C \sim G$ can hear A transmitting to B, which means $C \sim G$ can know that B is received packet and will forward the packet to the destination.

Such a characteristic, which is called *overhearing*, can be used to make dynamic clusters. By using overhearing information, each neighbor node can act as a participant of cluster of B, which is the head node of a cluster and forward data packets to the node B. Then the head node, B can aggregate the entire information to reduce the cumulative amount of data packets.

Set of Clustered Nodes. For the formal description of the proposed algorithm, we define the sets of nodes as follows.

Definition 1. *$NodeSet(N_i, N_j)$: Set of all nodes in a circle (or sphere in a 3-dimensional space). The center of the circle is N_i. The radius of the circle is the distance between node N_i and node N_j.*

Definition 2. $ClusterSet(N_i, N_j) =$
$NodeSet(N_i, N_j) \cap NodeSet(\text{Source}, N_j) \cap \neg NodeSet(N_j, \text{Destination})$

Definition 3. $PartySet(N_i, N_j) = ClusterSet(N_i, N_j) - \{N_j\}$

For example, in Fig. 1: $NodeSet(A, B)$, $ClusterSet(A, B)$ and $PartySet(A, B)$ are $\{A, B, C, D, E, F, G\}$, $\{A, B, C, D, E\}$ and $\{A, C, D, E\}$ respectively, and the cluster head of $ClusterSet(A, B)$ is node B.

3.3 Energy Aware Routing with Adaptive Clustering Mechanism(EAR-ACM)

Similar to EAR, the operation of EAR-ACM consists of three phases: *Setup phase*, *Data Communication phase* and *Route Maintenance phase*.

Setup Phase. Setup phase of EAR-ACM is the same as that of EAR. A destination node initiates a route request and all of the intermediate nodes relay the request in the direction to the source node.

Data Communication Phase. After the setup phase is completed, each sensor, i.e. the source node, sends data that it collected to a destination node.

1. A source node sends a data packet to any of its neighbors in the forwarding table, with the probability of the neighbor being selected set to the probability in the forwarding table[1].
2. Each intermediate node chooses the next hop among its neighbors. The probability of the neighbors being selected is set to that in the forwarding table. Each node delays packet forwarding within a given period, T_{delay}, to the next hop to aggregate the multiple packets.
3. Each node, which is belong to $PartySet(N_i, N_j)$ and has the forwarding table entry which contains the probability of choosing N_j as its next hop, sets the overheard flag of N_j field in the forwarding table which they overheard. At the same time, the cluster set $ClusterSet(N_i, N_j)$ is dynamically created. All overheard nodes are joined to $ClusterSet(N_i, N_j)$ as participants and the N_j becomes the head of the $ClusterSet(N_i, N_j)$ during T_{delay}.
4. Similarly to EAR, this process continues until the data packet reaches the destination node[1].

Route Maintenance Phase. Like in the setup phase, the route maintenance phase of EAR-ACM is the same as that of EAR. Flooding is occasionally performed in order to keep all the paths alive and to compute new metrics[1]. However, computing new metrics can be performed in the data communication phase of EAR-ACM, and thus route maintenance phase is not necessary as in EAR.

Figure 2 shows the example scenario of adaptive clustering hierarchy of EAR-ACM. There is no overhead of selecting a cluster head because the clustered regions are dynamically created using overheard information.

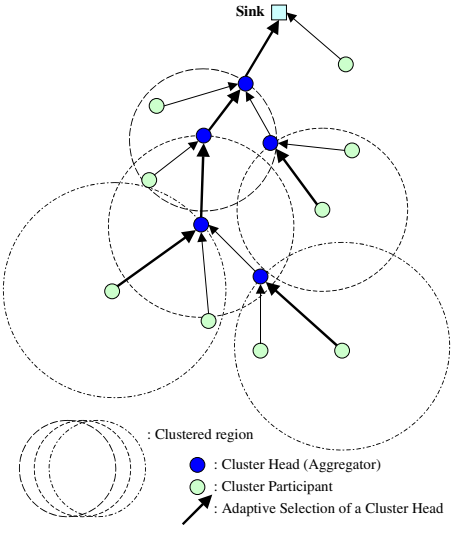


Fig. 2. Adaptive Clustering Hierarchy on EAR-ACM

4 Performance Evaluation

4.1 Simulation Environment

A simulator was developed by our research group. This program can simulate both EAR-ACM and LEACH[2] and provide some statistical information such as the residual energy of each node, the energy metrics between any two nodes and the communication cost of each node.

The area of the sensor networks was assumed to be $100m \times 100m$. The number of the nodes in networks was assumed to be 100 nodes – one node is controller while the others are sensors. The controller was located at the center of the field and the sensor nodes are placed randomly in the field as shown in Fig. 3(a). All of the sensors sent data to the controller at fixed time. The length of interval at each node was all-identical. This interval can be considered as a virtual time

unit. That is, in a single time unit, all of the sensor nodes send their data to the controller altogether.

The parameters in numeric formulas are as shown in [1]. Every node was given an identical amount, $0.05J$, of initial energy. The energy for transmission was assumed $20nJ/bit + 1pJ/bit/m^3$. The energy for reception was assumed $30nJ/bit$. The packet length was assumed 32 bytes. The energy metric function with α= 1 and β= 50 was used.

In this simulation, the following assumptions were made:

- Every node knows its position and the distance between itself and the other nodes.
- Every node has an identical maximum radio range.

4.2 Simulation Results and Evaluation

Figure 3(b) shows the residual energy of every sensor node after 400 time units by comparing EAR-ACM with LEACH. The x-axis indicates the distance between the sensor and controller or destination. Figure 3(b) shows that the energy of the nodes using EAR-ACM are more uniform. Table 1 shows the statistics of

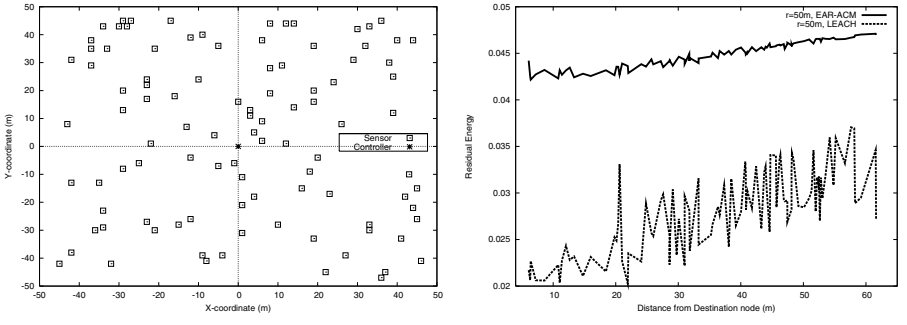


Fig. 3. (a) Layout of nodes in the sensor networks (b) Residual energy of each node

the residual energy of the nodes in networks for two algorithms. After 200 time units, the average residual energy of each node for EAR-ACM is ∼1.17 times that for LEACH. After 600 time units, the ratio exceeds over 3. It should be noted that the standard deviation of energy for EAR-ACM is less than that for LEACH, and that the minimum energy for EAR-ACM is much greater than that for LEACH. This shows that the nodes using EAR-ACM will survive much longer than those using LEACH.

Figure 4(b) shows the energy distribution at a certain time point of the sensor networks using EAR-ACM. It can be seen that the difference in energy among nodes is smaller than LEACH as shown in Fig. 4(a), respectively. In addition, the energy of the entire networks using EAR-ACM is still higher than that of LEACH. The detailed graph is shown below.

Table 1. Residual Energy Statistics

Time	Protocol	Energy (J) Average	Std.Dev.	Max	Min
200	EAR-ACM	0.04726	0.001	0.048	0.046
	LEACH	0.04026	0.003	0.045	0.034
400	EAR-ACM	0.04495	0.001	0.047	0.042
	LEACH	0.02871	0.004	0.037	0.020
600	EAR-ACM	0.04262	0.002	0.046	0.039
	LEACH	0.01335	0.008	0.033	0

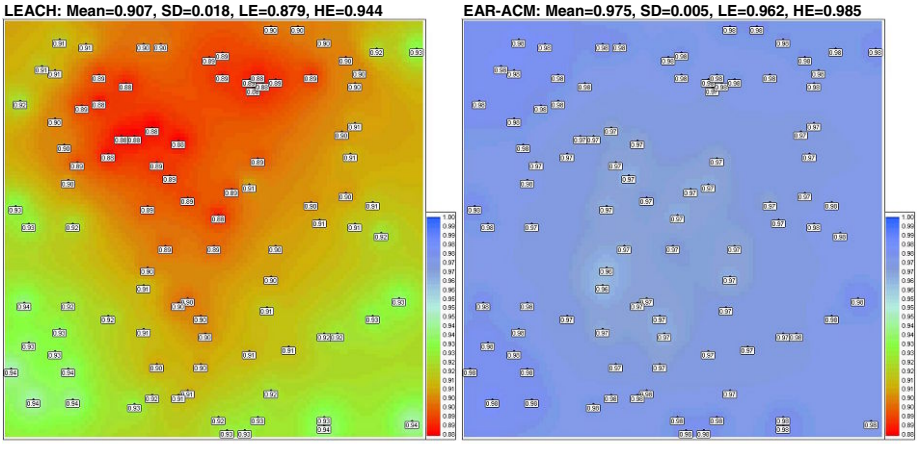


Fig. 4. Energy consumption map (a) LEACH (b) EAR-ACM

Figure 5(a) shows the number of active nodes in networks versus time. In the case of networks using EAR-ACM with the maximum radio range $r = 20m$, the first event where a node runs out of energy occurs after 387 time units, whereas in the case of LEACH, it occurs after 55 time units. Then, in the case of networks using EAR-ACM with $r = 50m$, it occurs after 266 time units and in the case of networks using LEACH, after 64 time units.

In the sensor networks, it is not fatal if a small number of nodes run out of energy. However, it becomes a critical problem when all of the surrounding neighboring nodes die. Therefore, in this paper, we measured the time elapsed until the energy of all the nodes which are neighbors of a destination run out. That is, the controller is no longer able to receive any more data packet and the entire networks become useless. According to the simulation results, when the maximum radio range was assumed $20m$, it occurred after 506 time units for EAR-ACM, while it occurred after 48 time units for LEACH. When r is $50m$, it took 590 and 217 time units, respectively.

Figure 5(b) shows the average of residual energy of each node versus time. Then the total remaining energy of the networks is computed by multiplying the

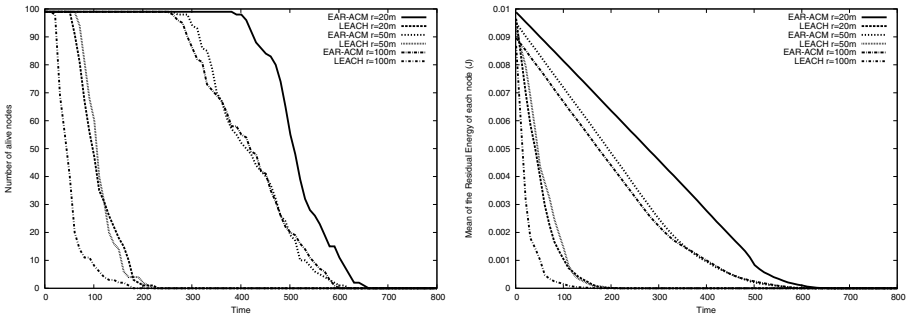


Fig. 5. (a) The number of active nodes during communication phase (b) Mean of residual energy of each node

average with the number of nodes. As we expect, the total energy of EAR-ACM is greater than that of LEACH.

5 Conclusion

Most existing routing protocol used in the sensor networks do not take power into consideration, and could therefore result in either parts of, or the whole sensor network being temporarily unavailable. In this paper, we presented an efficient energy aware routing algorithm for high survivability of the wireless sensor networks. Our algorithm is based on EAR[1] and enhanced the survivability of networks using adaptive clustering mechanism. The proposed algorithm can reduce the cumulative amount of data packets and ensure efficient use of energy among the nodes in the sensor networks. Simulation results show that the energy usage of EAR-ACM is significantly reduced compared with the previous clustering based routing algorithm for the sensor networks.

References

1. Shah, R., Rabaey, J.: Energy aware routing for low energy ad hoc sensor networks. In: Proc. IEEE Wireless Communications and Networking Conference(WCNC). (2002)
2. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: Hawaii International Conference on System Sciences (HICSS). (2000)
3. Ganesan, D., Govindan, R., Shenker, S., Estrin, D.: Highly-resilient, energy-efficient multipath routing in wireless sensor networks. Mobile Computing and Communications Review **5** (2001) 11–25
4. Chang, J.H., Tassiulas, L.: Maximum lifetime routing in wireless sensor networks. IEEE/ACM Transactions on Networking **12** (2004) 609–619
5. Li, Q., Aslam, J., Rus, D.: Hierarchical power-aware routing in sensor networks. In: DIMACS Workshop on Pervasive Networking. (2001)

6. Singh, S., Woo, M., Raghavendra, C.S.: Power-aware routing in mobile ad hoc networks. In: Mobile Computing and Networking. (1998) 181–190
7. Schurgers, C., Srivastava, M.B.: Energy efficient routing in wireless sensor networks. In: IEEE Military Communications Conference (MILCOM'01). (2001) 357–361
8. Chang, J., Tassiulas, L.: Energy conserving routing in wireless ad-hoc networks. In: IEEE Infocom. (2000) 22–31
9. Braginsky, D., Estrin, D.: Rumor routing algorithm for sensor networks. In: 1st ACM international workshop on Wireless sensor networks and applications WSNA02, pages 22.31, Atlanta, Georgia, USA, 2002. ACM Press. (2002) 22–31
10. Pottie, G.J., Kaiser, W.J.: Embedding the internet: wireless integrated network sensors. Communications of the ACM **43** (2000) 51–58
11. Heinzelman, W.: Application-specific protocol architectures for wireless networks. Ph.D. thesis, Massachusetts Institute of Technology (2000)
12. Lindsey, S., Raghavendra, C., Sivalingam, K.M.: Data gathering algorithms in sensor networks using energy metrics. IEEE Trans. Parallel Distrib. Syst. **13** (2002) 924–935
13. Bandyopadhyay, S., Coyle, E.J.: An energy-efficient hierarchical clustering algorithm for wireless sensor networks. In: IEEE INFOCOM. Volume 3. (2003) 1713–1723
14. Intanagonwiwat, C., Govindan, R., Estrin, D.: Directed diffusion: a scalable and robust communication paradigm for sensor networks. In: Mobile Computing and Networking. (2000) 56–67

Curve-Based Greedy Routing Algorithm for Sensor Networks

Jin Zhang[1], Ya-ping Lin[1], Mu Lin[2], Ping Li[1], and Si-wang Zhou[1]

[1] College of Computer and Communication, Hunan University, Changsha 410082, China
[2] College of Mathematics and Econometrics, Hunan University, Changsha 410082, China
{mail_zhangjin, yplin888, kevin9908, liping9188, myswzhou}@hotmail.com

Abstract. Routing packets along a specified curve is a new approach to forwarding packets in large-scale dense sensor networks. Forwarding packets along trajectories can be very effective in implementing many networking functions when standard bootstrapping or configuration services are not available, as will be the case in sensor networks where nodes are thrown or dropped to form a one-time use network. In this paper, investigating Trajectory-Based Forwarding (TBF), we propose a novel curve-based greedy routing algorithm (CBGR) for sensor networks. In CBGR, by using the location information of a source node and the sink, the source constructs a B-spline curve as forwarding trajectory and encodes the curve into the packets. Upon receiving each packet, the intermediate nodes decode it and construct a simple dynamic forwarding table (DFT) by different greedy forwarding strategies. Then, the packets are forwarded along the selected curve based on DFT. Several greedy forwarding strategies are discussed. CBGR is a distributed routing strategy and easy to implementation. By selecting multiple forwarding curves, CBGR can balance the energy consumption of the nodes effectively. The analysis and simulation also show that CBGR has better performance.

1 Introduction

Integrated collecting data, managing data and communicating, wireless integrated network sensors can be used in many applications. The uncertainty of environment usually requires disposing hundreds of sensors to cooperate, and the research to sensor network is regarded as a challenging area especially when considering its characteristics of high density of nodes and limited resource of nodes. Three criteria drive the design of large-scale sensor networks: scalability, energy-efficiency and robustness. For these criteria, the traditional routing protocols are not suitable and new routing algorithms must be researched for sensor networks.

As a stateless routing algorithm, greedy forwarding [1] is one selection to sensor networks. SBR [2] is desirable on traffic engineering needing multi-path. Niculescu and Nath [3] proposed the idea of TBF (Trajectory-Based Forwarding) which combines

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 1125 – 1133, 2005.

SBR with greedy forwarding. Yuksel [4] developed the idea of TBF by Bezier parametric curve. However, the algorithms in [3,4] suffer a high computation overhead because each node has to calculate next-hop to be forwarded for each packet even though these packets are from the same sources and to the same destinations. In this paper, a curve-based greedy routing algorithm (CBGR) is proposed. In CBGR, a source node selects a curve as trajectory of forwarded packets and encodes the curve into the packets. The intermediate nodes construct dynamic forwarding tables (DFT) based on adaptive greedy strategies. By checking DFT, packets along the same route can be forwarded without extra computation. CBGR is a distributed routing strategy and easy to implementation. By selecting multiple forwarding curves, CBGR can balance the energy consumption of the nodes efficiently. The rest of the paper is organized as follows. In the next section, the curve-based greedy routing model is proposed. Section 3 describes how to construct CBGR using B-spline parametric curve. In section 4, we evaluate the performance of CBGR. We conclude in Section 5.

2 Curve-Based Greedy Routing Model

CBGR combines TBF with Dynamic Source Routing (DSR)[5]. The basic idea can be described as follows: (1) a source node selects a suitable curve and encodes the curve into each packet; (2) upon receiving the packets, intermediate nodes decode the curve and use greedy strategies to decide next-hop to be forwarded and construct dynamic forwarding tables (DFT); (3) the sequent packets can be forwarded according to the constructed DFT; (4) after sending a number of packets, sources select another curve and forward packets along this curve.

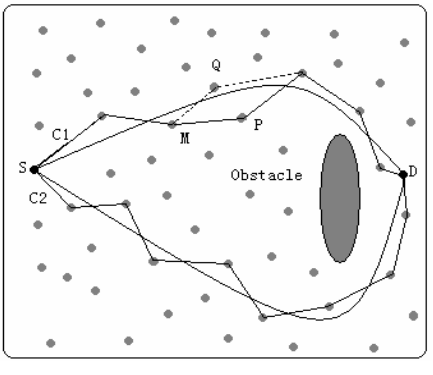


Fig. 1. CBGR model

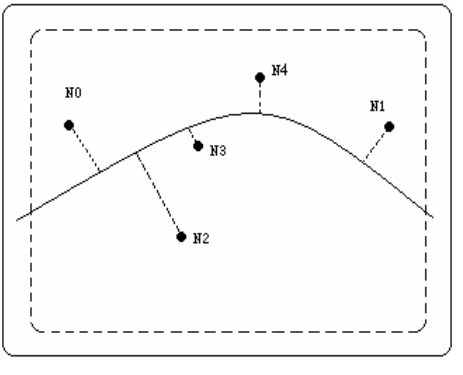


Fig. 2. Neighbors of one node

The model of CBGR can be described in Figs.1 and 2. In Fig.1, the source (denoted by S) selects a suitable curve and encodes it into the packets. Then intermediate nodes select one node as next-hop in its neighbors according to greedy strategies and then construct DFT. For example, if an intermediate node's neighbors are represented as Fig. 2, then N0 can be selected as the next-hop. After that, the sequent packets can be

forwarded by lookup DFT. The intermediate nodes, such as P in Fig.1, will record the number of forwarded packets and tell its previous hop node (PH), such as M in Fig.1, to select another node, such as Q in Fig.1, to replace it. If M can't find a suitable node according to the greedy strategy, it will send a special packet to inform S which will select another curve to forward packets. After sending a number of packets along curve C1, the source node S selects another curve, such as C2 in Fig.1, to forward packets. These curves can be selected evenly in exploring area to make more sensors work. By this, CBGR can balance the energy consumption of nodes effectively.

How to select the curve is also an important issue. Some simple curves, such as Beeline and Cosine, can be used in general applications. However in some special applications such as military application, the shape of curves must be controlled strictly in order to avoid dangerous or doubtful area during battle. Compared with Bezier curve, B-spline curve is more suitable because it has the following good properties: (1) the computation cost is lower because the degree of the curve is independent of its control points; (2) a B-spline curve is naturally connected, so its shape can be controlled easily.

However, there is a problem with B-spline curves. In order to make things clear, we give the following formulas:

$$P(t) = [t^2 \quad t \quad 1]\frac{1}{2}\begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix}\begin{bmatrix} V_0 \\ V_1 \\ V_2 \end{bmatrix} \quad 0 \le t \le 1 \tag{1}$$

$$\left.\begin{aligned} t{=}0 \quad & P_{0,2}(0) = (V_0 + V_1)/2 \\ t{=}1 \quad & P_{0,2}(1) = (V_1 + V_2)/2 \end{aligned}\right\} \Rightarrow \begin{cases} V_0 = 2P_{0,2}(0) - V_1 \\ V_2 = 2P_{0,2}(1) - V_1 \end{cases} \tag{2}$$

From formula 1, each point in B-spline curves can be calculated when parameter t changes from 0 to 1 if the control points V_i (i=1,2,3) are known. We use S and D to denote the source and destination nodes respectively. However, S and D are not the start point and end point of B-spline curves respectively when we assign S and D corresponding to control points V_0 and V_2. This is a shortcoming for B-spline curves compared with Bezier curves. However, we can do some changes in formula 1 to make the B-spline curve start from S and end to D. Let t=0 and t=1 respectively, we get formula 2. By setting $P_{0,2}(0)$=S, $P_{0,2}(1)$=D and selecting one control point V_1 , we can calculate the other two control points. The three control points can ensure that S and D are the start and end points of the B-spline curves respectively. Because S and D are fixed, we need only to choose V_1 to distribute different curves in the detecting area evenly.

How to select next-hop is another important issue in CBGR. Basically, the next-hop node is selected from the set of neighbors. However, different greedy strategies will cause different results for selecting next-hop. Depending on the different applications, we give several strategies and corresponding examples described in Fig.2:

(1) Maximal Advancement along Curve (MAC): in Fig. 2, N1 is selected as the next-hop by N0;

(2) Least Advancement on Curve (LAC): in Fig. 2, N2 is selected as the next-hop by N0;
(3) Closest to Curve (CTC): in the Fig. 2, N3 is selected as the next-hop by N0.

3 Curve-Based Greedy Routing Algorithms

In this section, we develop CBGR based on B-spline curve. The process can be divided into four stages: local topology built stage (LTBS), interest distributed stage (IDS), data forwarded stage (DFS) and path adjusting stage (PAS). Each stage is described in the following.

3.1 Local Topology Built Stage (LTBS)

In this paper, nodes are assumed to know its position by GPS or LPS [6]. After nodes are disseminated, each node sends Topology Building Packets (TBP) three times during a short time interval to assure that each neighbor can receive at least one TBP. It is unnecessary that nodes send acknowledgment to respond TBP. According to the TBP, each node constructs its Neighbors Table (NT) including ID, Position (POS), and Energy (E) of neighbors. By setting TBP.TTL (Time to Live)=1,TBP can only be received by its neighbors. When a neighbor receives a TBP, it checks NT first to assure there isn't an item before it adds a new item. After LTBS, each node knows the information of its neighbors.

3.2 Interest Distributed Stage (IDS)

The aim in this stage is for each node to get the position of the sink and its interest (required data). In IDS, the distribution of interest is based on TBF and rumor routing [7]. The basic idea can be described as follows: (1) each node has been configured with a scope of interest before disseminated and it collects data after LTBS; (2) the nodes collecting data send data agent packets (DAP) to notice TD (type of data) and the sink sends interest agent packets (IAP) to notice its position and required TD. DAP and IAP is forwarded along simple curve such as straight line. (3) On receiving DAP or IAP, the nodes record $DAP.ID_S$, DAP.TD, DAP.PH or $IAP.POS_D$, $IAP.ID_D$, IAP.TD. (4) If DAP.DT is equal to IAP.DT, nodes reverse to send Announcing Packets (AP) to tell sources some information about destinations. Because nodes have recorded the information when forwarding DAP, it is very simple to forward AP.

According to [7], we can assure that the probability of intersecting of the forwarding curve of DAP and IAP is fairly high (99.7%) by selecting randomly and separately five curves. But the influence of network size should be considered when distributing interest.

3.3 Data Forwarded Stage (DFS)

In this stage, DFT will be constructed according to Data Forwarding Packets (DFP). And then packets can be forwarded easily from S to D**.** The constructed process can be described as follows: (1). S selects a control point of two degree B-spline curve and calculates the two other control points according to formula 1; (2). S selects next-hop

from the set of neighbors based on the suitable greedy strategy and constructs DFP. (3). S broadcasts DFP to all of its neighbors, but only the neighbor node whose ID is equal to DFP.NH deals with it. (4). the right node calculates the value of *t* and selects the NH (next-hop node). Then the node replaces DFP.NH and DFP.PH with new NH and the ID of node. At last, the node broadcasts DFP. (5). the process doesn't terminate until D receives DFP.

3.4 Path Adjusting Stage (PAS)

In order to balance routing load among the nodes, the basic idea is to replace nodes that have forwarded a number of packets with other nodes. There is a constant variable, DFT.E, for each node in CBGR. When one node's energy consumption reaches the value, it sends energy announcing packets (EAP) to its PH. Then the PH will select another node to replace it. The PH has to find another NH and broadcast path adjusting packets (PAP). The new NH will deal with PAP to adjust its DFT. And the previous NH has to delete the corresponding item when it receives the PAP. The PH node adjusts its DFT by replacing DFT.UH with the new UH. If PH can't find one node to replace, the EAP will be forwarded to S upwardly and S will select the different control point to construct another curve. The process is like the description in 3.3. The active adjustment can balance the energy consumption effectively.

4 Evaluation

In order to analysis the performance of CBGR, we define some metrics and compare it with other algorithms.

4.1 Comparison Metrics

(1) Time Complexity (TC): time spent on selecting the nodes which forward packets.
(2) Message Complexity (MC): number of packets sent on selecting the next-hop nodes.
(3) Message Scope (MS): scope of local message needed on selecting the next-hop nodes.
(4) Average Deviation (AD): the average distance from each node on the path to its projection in the curve.
(5) Average Path Length (APL): the average hops of path from S to D.
(6) Node Raito (NR): the proportion of the number of nodes on the path to the total of nodes.

4.2 Theoretic Analysis

CBGR are compared with Directed Diffusion (DD) [9] and LEACH [10] in the metrics of TC, MC and MF. The results are listed in Table 1. DD uses flooding once to build the gradients and its TC and MC are $O(n)$. n is the number of nodes in the sensor networks. LEACH needs only to select the leader once in each cycle, so its TC is also $O(n)$. In [8], it has been proved that any distributed algorithm for leader election and spanning tree

need send at least *O(nlogn)* messages. Sensor networks can be considered as a special kind of ad hoc networks. So TC of LEACH is *O(nlogn)*. To CBGR, its TC and MC are not more than *O(n)* according to the described process. From the Table 1, we know CBRG has better performance.

Table 1. Performance Comparison

Metric	DD	LEACH	CBGR
TC	*O(n)*	*O(n)*	<= *O(n)*
MC	*O(n)*	*O(nlogn)*	<= *O(n)*
MS	*1 hop*	*1 hop*	*1 hop*

4.3 Simulation

We simulate CBGR on randomly generated a connected network in an area of $100*100m^2$ with *N* nodes on different positions. *N* and the communication radius *R* of nodes are respectively varied in the simulation. Firstly, the performance of different greedy strategies is compared. In the figures, s1, s2 and s3 denote MAC, LAC and CTC described in section 2 respectively. Figure 3 and 4 show the performance on AD and it is obvious that CTC has the best performance. Figure 5 and 6 show the performance on APL and it is easy to see MAC has the best performance.

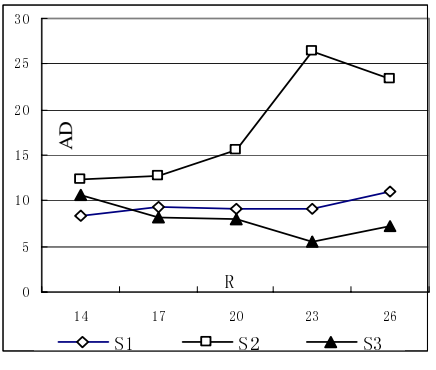


Fig. 3. AD changes with different R

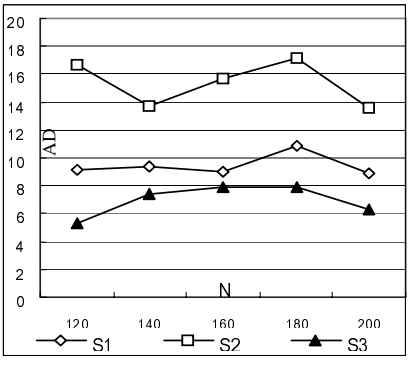


Fig. 4. AD changes with different N

Then we select CTC as greedy strategy of selecting next-hop and use two curves to forward the data. We compare CBGR with DD which uses the number of hop to destination as gradients. Figs.7 and 8 show the number of nodes with different energy expenditure (EE) when two-unit data is sent from source to destination if *R* and *N* change. The results show the balancing energy expenditure of CGBR is better than DD. And the more the number of selecting curves is, the less the energy consumption of each node is.

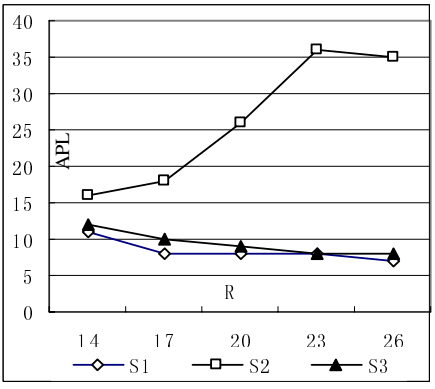


Fig. 5. APL changes with different R

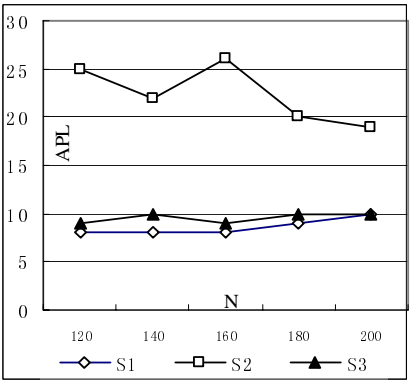


Fig. 6. APL changes with different N

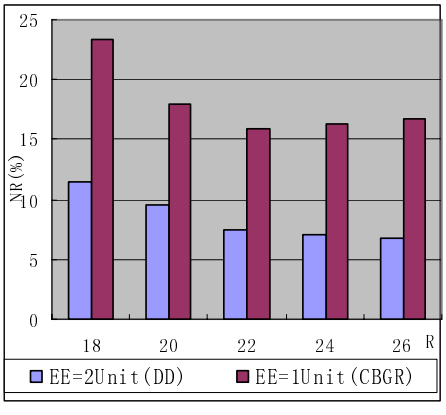


Fig. 7. EE changes with different R

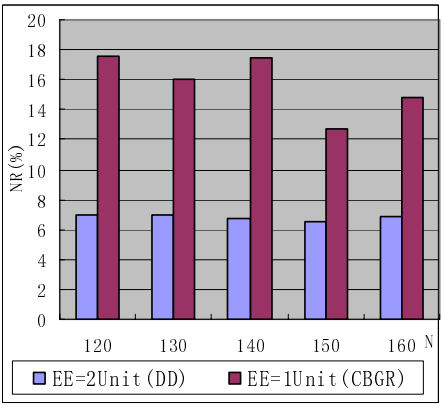


Fig. 8. EE changes with different N

Figs. 9 and 10 show the ratio of node (NR) forwarding packets. For LEACH, the number of cluster is set to 5% generally. In Fig. 9, when *R* increases, NR decreases. The reason is the average hops decrease when the distance from S to D is changeless. In Fig. 10, when *N* increases, NR also decreases but more slowly compared with Fig. 9. It's because the probability of next-hop near D is higher when *N* increases.

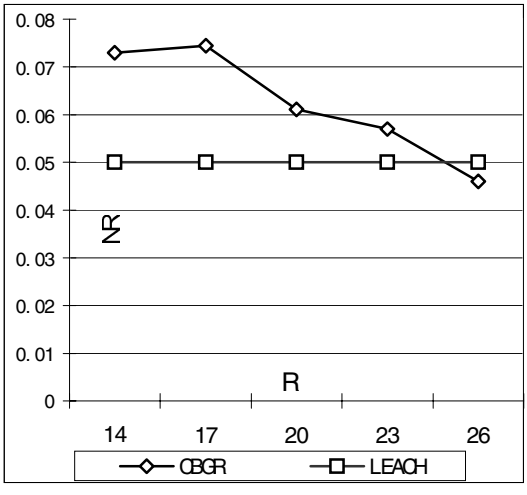


Fig. 9. Changes of NR in different algorithms with different R

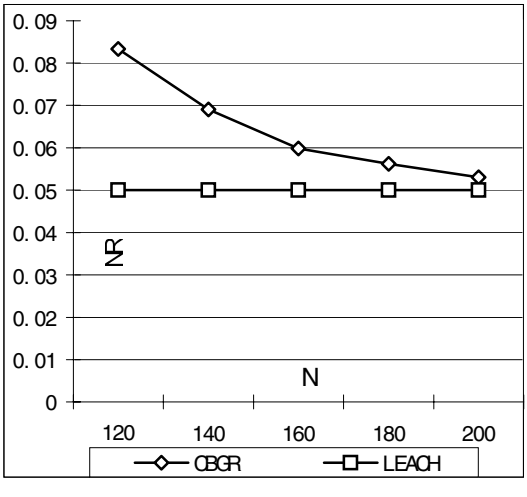


Fig. 10. Changes of NR in different algorithms with different N

5 Conclusions

In this paper, we propose a novel B-spline curve-based greedy routing (CBGR) model. There are several conclusions we can draw from our analysis and simulation. First, CBGR can balance the energy consumption among the nodes. Second, Closest to Curve (CTC) has better performance in different greedy forwarding strategies. Finally, CBGR is an easy to implementation and distributed model. Still, several issues remain to be investigated, such as mobility, the density of nodes, etc. Our future works also include energy-efficient multi-path routing and security routing based on CBGR.

References

1. B. Karp and H. T. Kung, GPSR: greedy perimeter stateless routing for wireless networks, in Proceedings of ACM 2000.
2. E. Crawley, et al., A Framework for QoS-based Routing in the Internet, RFC 2386, August 1998.
3. D. Niculescu and B. Nath. Routing on a curve, in Proceedings of Workshop on Hot Topics in Networks (HOTNETS-I), 2002.
4. M. Yuksel, R. Pradhan and S. Kalyanaraman. Trajectory-Based Forwarding Mechanisms for Ad-Hoc Sensor Networks, in Proc. of the IEEE 2nd Upstate New York Workshop on Sensor Networks, Syracuse, NY, October, 2003.
5. D. Johnson and D. Maltz. Dynamic source routing in ad hoc wireless networks, in Mobile computing, vol. 353, 1996.
6. D. Niculercu ond B. Nath, Position and Orientation in Ad Hoc Networks, Elrevier Ad *Hoc* Nehwrkr, vol. 2, no. 2, Apr. 2004, pp. 133-51.
7. David Braginsky, Deborah Estrin. Rumor routing algorthim for sensor networks, in Proc. of the 1st ACM international workshop on Wireless sensor networks and applications, Sep. 2002.
8. Peng-Jun Wan, Alzoubi K.M., Frieder O. Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks, in Proc. of Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Volume: 3, 23-27 June 2002, pp. 1597- 1604.
9. C. intanagonwiwat, R. Govindan, and D. Estrin, “Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks”, in: Proc ACM MobiCom ’00, Boston, MA, 2000, pp. 56-57.
10. W. Heinzelman, Application-Specific Protocol Architectures for Wireless Networks, Ph.D. thesis, Massachusetts Institute of Technology, 2000.

Traffic Adaptive MAC Protocol for Wireless Sensor Network*

Haigang Gong, Ming Liu, Yinchi Mao, Li-jun Chen, and Li Xie

State Key Laboratory for Novel Software Technology,
Department of Computer Science and Technology, Nanjing University, China
{ghgang,wing_lm,maoyc}@dislab.nju.edu.cn

Abstract. In this paper, we propose TA-MAC, a traffic load adaptive Medium Access Control protocol for wireless sensor network. TA-MAC modified the contention window mechanism of S-MAC. It adjusts the initial contention window according to the current traffic load to reduce the collision probability and employs a fast backoff scheme to reduce the time for idle listening during backoff procedure, which both reduce the energy consumption. Simulation results show that TA-MAC outperforms S-MAC.

1 Introduction

A wireless sensor network is a distributed system comprised of a large number of extremely small, low-cost and battery-powered sensors, which can be used to collect useful information (i.e. temperature, humidity) from a variety of environment. WSN have been envisioned to have a wide range of applications in both military as well as civilian domains [1][2] such as battlefield surveillance, machine failure diagnosis, and chemical detection.

Like in all other shared-medium networks, medium access control (MAC) is also an important technique that ensures the successful operation of WSN. A MAC protocol decides when competing nodes may access the shared medium and tries to ensure that no two nodes are interfering with each other's transmissions. In contrast to nodes in traditional wireless network, the main constraint of sensor nodes in WSN is their low finite battery energy, which limits the lifetime and the quality of the network. MAC protocol running on WSN must consume energy efficiently in order to achieve a longer network lifetime. However, when running a MAC protocol on a wireless sensor network, much energy is wasted due to the following sources of overhead: a) **Idle listening**: Since a node does not know when it will be the receiver of a message from one of its neighbors, it must keep its radio in idle listening mode at all times. b) **Collisions**: If two nodes transmit at the same time and interfere with each other's transmission, packets are corrupted. Hence, the energy used during transmission and reception is wasted. c) **Overhearing**: Since the radio channel is a

* This work is partially supported by the National Natural Science Foundation of China under Grant No.60402027; the National Basic Research Program of China (973) under Grant No.2002CB312002.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 1134 – 1143, 2005.

shared medium, a node may receive packets that are not destined for it; it would have been more efficient to turn off its radio. d) **Protocol overhead**: The MAC headers and control packets used for signaling (ACK/RTS/CTS) do not contain application data and are therefore considered overhead. e) **Traffic fluctuations:** A sudden peak in activity raises the probability of a collision; hence, much time and energy are spent on waiting in the random backoff procedure.

There are several solutions addressing the problem of energy wastage [8]-[16]. In this paper, we propose TA-MAC, a traffic adaptive MAC protocol for wireless sensor networks. TA-MAC tries to achieve energy savings by: 1) reducing the time spent in idle listening during backoff procedure after collision happens; (2) reducing the probability of collision by adjusting the size of contention windows under different traffic load.

The rest of the paper is organized as follows. Section 2 reviews the related work. Section 3 describes our MAC protocol in detail. Section 4 discusses the simulation results. Finally, Section 5 concludes the paper and presents future research directions.

2 Related Work

Current MAC protocol design for wireless sensor network can be broadly divided into schedule-based and contention-based protocol [14].

1) Schedule-based MAC protocols. For example, TDMA protocols [8]-[10] are typical scheduled protocol that are naturally energy preserving, because they have a duty cycle built-in, and do not suffer from collisions [7]. However, using TDMA protocol usually requires the nodes to form communication clusters, like LEACH [11]. Most nodes in cluster are restricted to communicate within the clusters. Managing inter-cluster communication and interference is not an easy task. Moreover, when the number of nodes within a cluster changes, it is complicated to dynamically change its TDMA frame length and time slot assignment. So its scalability is poorer than contention-based protocol.

2) Contention-based MAC protocols. The standardized IEEE 802.11 distributed coordination function (DCF) [12] is an example of the contention-based protocol, and is widely used in ad hoc wireless networks because of its simplicity and robustness to the hidden terminal problem. However, the energy consumption using 802.11 MAC is very high when nodes are in idle mode.

Sensor-MAC (S-MAC) protocol [13] is an effective MAC protocol designed by Ye et al. for wireless sensor network. The basic idea of S-MAC is that time is divided into large frames. Every frame starts off with a small synchronization phase, followed by a fixed active part and a sleep part. During synchronization phase, nodes receive or send SYNC packet contained the schedule information (i.e. when to sleep). During the sleep part, a node turns off its radio to preserve energy. During the active part, it can communicate with its neighbors and send any messages queued during the sleep part, as shown in Fig. 1. Since all messages are packed into the active part, instead of the whole frame, therefore the energy wasted on idle listening is reduced. Besides addressing the idle-listening overhead, S-MAC includes collision avoidance (RTS/CTS handshake) and overhearing avoidance, which further saves energy.

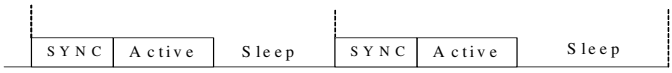


Fig. 1. Duty cycle of S-MAC

Timeout-MAC (T-MAC) protocol [15] introduces an adaptive duty cycle too. In T-MAC, a node keeps listening and potentially transmitting as long as it is in an active period. If a node does not detect any activity within the time-out interval, it can safely assume that no neighbor wants to communicate with it and goes to sleep. The activation time events include reception of any data, the sensing of communication on the radio, etc. The downside of T-MAC is that it introduces early sleep problem [15].

However, both S-MAC and T-MAC neglect the possibility that the number of active sensor nodes can change dynamically over time, leading to dynamically changing contention intensity. They both use fixed contention window to contend for the shared medium. When traffic is high, too small contention window may lead to excessive collisions and retransmission, which introduces unnecessary energy consumption. When traffic is low, large contention window may lead to much idle airtime during which no station attempts to transmit. So an adaptive contention window mechanism should be employed to reduce collisions under different traffic load.

3 TA-MAC Protocol Design

TA-MAC protocol is extended from S-MAC, modifying the contention window scheme of S-MAC and employing a fast backoff scheme to reduce the time of idle listening during backoff procedure. We describe the contention window mechanism of IEEE 802.11 MAC protocol first.

3.1 IEEE 802.11 Protocol

The basic access method in the IEEE 802.11 MAC protocol [12] is the distributed coordination function (DCF). A node wanting to transmit a packet must first test the radio channel to check if it is free for a specified time called the Distributed Inter Frame Space (DIFS). If so, a DATA packet is transmitted, and the receiver acknowledges the reception of the data by sending an ACK packet. If the sender does not receive the acknowledgement, it assumes that the data was lost due to a collision at the receiver and enters a *binary exponential backoff (BEB)* procedure. In BEB, backoff time is decided using the following expressions:

$$CW = CW_{min} \tag{1}$$

$$CW = CW \times 2 + 1 \tag{2}$$

$$CW = \min (CW, CW_{max}) \tag{3}$$

$$BT = \text{rand}(0, CW\text{-}1) * SlotTime , \tag{4}$$

where CW is contention windows; CW_{min} is the initial contention window at the first transmission of a packet; BT is backoff time select from contention window;

rand(0,CW-1) is an integer randomly chosen from a uniform distribution over the interval (0, CW-1). At each retransmission attempt, CW is doubled. Since contending nodes randomly select a time from their CW, the probability of a subsequent collision is reduced by half. To bound access latency somewhat, CW is not doubled once a certain maximum (CW_{max}) has been reached. After a successful transmission, CW will be reset to CW_{min} for the next packet.

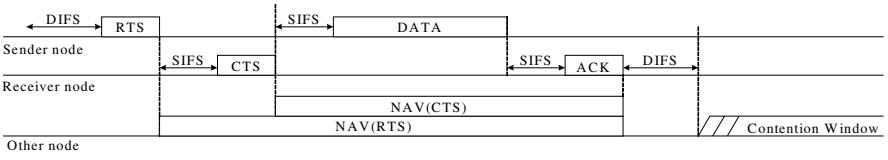


Fig. 2. IEEE 802.11 access control

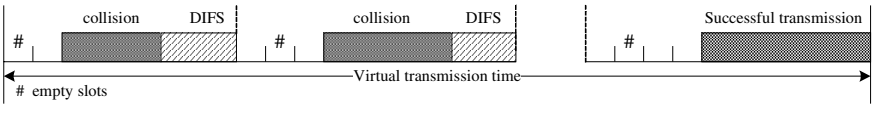


Fig. 3. One successful data transmission

To account for the hidden terminal problem in ad-hoc networks, the 802.11 standard defines a virtual carrier sense mechanism. The RTS/CTS control packets include a time field in their header, which specifies the duration of the upcoming DATA/ACK sequence. This allows neighboring nodes overhearing the control packets to set their Network Allocation Vector (NAV) and defer transmission until it expires (see Fig. 2.). To save energy, the radio can be switched off for the duration of the NAV.

3.2 Contention Window Adjustment Algorithm

Resetting CW to a fixed initial contention window as (1) after a successful transmission has some disadvantages. When many nodes contend for medium, collisions will happen frequently. After a successful transmission, if the next transmission uses the same initial CW as before, the collision probability remains high. The initial CW should be enlarged to reduce the number of collision. On the other hand, if the initial CW is high, there may be considerable backoff delay. S-MAC uses a constant size of contention window. Whenever collision happens, backoff time is selected from the same size of CW, which is inflexible. To solve the problem, we employ a dynamic contention window adjustment algorithm in which the initial CW size is adaptive to the current load of network.

As in [17], the time interval between two successful transmissions is referred to as *virtual transmission time*. A virtual transmission time includes a successful transmission and may include several collision intervals (see Fig. 3).

Define channel load as the average number of collisions $n_{coll\text{-}avg}$.

$$n_{coll-avg} = w \cdot n_{coll-avg} + (1-w) \cdot n_{coll} \ , \tag{5}$$

where n_{coll} is the number of collisions during current successful transmission, w is weight. According to [18], w is set to 0.2. When $n_{coll\text{-}avg}$ is greater than a preset threshold $l_{threshold}$, the channel is busy and traffic load is high. In [18], authors advise the threshold value is set as 4. By monitoring the channel load, we adjust the initial contention window to avoid collision after a successful transmission. The contention window adjustment algorithm is shown in Fig. 4.

```
if n_coll-avg > l_threshold
    CW = min ( CW_max, (CW + 1) *2 –1)
else
    if CW > CWmin
            counter ++
            CW = max (CW_min, (CW+1) / 2 –1)
            if counter > MAX_Counter
                    CW = CW_min
                    counter = 0
```

Fig. 4. CW adjustment algorithm

When traffic load is light, the contention window adjustment follows BEB scheme as (1)-(4). When traffic load ($n_{coll\text{-}avg}$) is higher than the threshold ($l_{threshold}$), which means the current initial CW is too small, more nodes contend for medium access in the small CW, which introduce more collision and energy wastage. According to our algorithm, the initial contention window doubles itself after this successful transmission rather than resetting to CW_{min} as in 802.11. The large initial CW would decrease the chance of accessing medium at the same time so that it reduces the number of collision. When $n_{coll\text{-}avg}$ is less than the threshold, which means traffic load goes lighter; CW would decrease itself by half. If traffic load is less than the threshold during MAX_Counter times consecutive successful data transmission and CW is greater than CW_{min}, it means traffic load turns from heavy to light, and CW is reset to CW_{min} directly in order to reduce the backoff delay at the next transmission. The parameter MAX_Counter is discussed in section 4.2.

3.3 Fast Backoff Scheme

In 802.11 protocols, if a node has a packet to transmit, it will check the medium status by using the carrier sensing mechanism. If the medium is idle, the transmission may proceed. If the medium is determined to be busy, the station will defer until the medium is determined to be idle for DIFS and the backoff procedure will be invoked. The station will set its backoff timer to a random backoff time based on (4).

However, a node can be considered as in idle listening during the backoff time. Small backoff time will decrease the idle listening time during backoff procedure, but large backoff time will decrease the collision probability at subsequent contention periods. We borrow the fast collision resolution (FCR) idea proposed in [19]. When a

node detects the medium is idle for a fixed number of slots during backoff, it would conclude that no other nodes are transmitting. The backoff procedure is then divided into two phases: linearly decrement and exponentially decrement. When the node performs the backoff procedure, it performs linearly decrement first and decreases its backoff timer by a slot time as in 802.11 after detecting the medium idle for a slot. If a number of consecutive idle slots are detected and the remaining backoff timer value is less than or equal to the backoff threshold value BT_{thre}, it will decrease exponentially the backoff timer as (6):

$$BT = BT / 2 . \tag{6}$$

When the backoff timer reaches zero, the node starts to transmit.

Define the backoff threshold BT_{thre} as (7):

$$BT_{thre} = \frac{CW_{\max} - CW}{CW_{\max} - CW_{\min}} \cdot BT . \tag{7}$$

The function adapts to the channel load by including the contention window CW. The logic behind this adaptation is illustrated by these two conditions: a) when traffic load increases and CW doubles itself, it must reduce the exponential decrease stage by decreasing its BT_{thre}, in order to avoid a new collision. b) When traffic load decreases with the decreasing CW, it must extend the exponential decrease stage by increasing its BT_{thre}, in order to reduce the idle listening time.

4 Performance Evaluation

We implemented our protocol in the ns-2 network simulator with the wireless extension. For comparison, The IEEE 802.11(CSMA/CA) MAC protocol and S-MAC will serve as the baseline.

4.1 Simulation Setup and Parameters

We have built a realistic model of the Rene Motes, developed at UCB [20]. The transceiver on the mote is the model TR1000 from RF Monolithics Inc. [21]. Energy consumption in the model is based on the amount of energy the TR1000 uses: 5 μ A while sleeping, 4.5 mA while receiving and 12 mA while transmitting. Other simulation parameter are listed in table I. The contention window in S-MAC is fixed at 63.

Table 1. Simulation parameters

Parameters	Value
Packet Length	100 bytes
CW_{min}	31
CW_{max}	127
SlotTime	200 us
$l_{threshold}$	4

Our simulation uses a 10×10 grid topology with 100 nodes. The sink node is located in the center of the grid and the edge nodes are source nodes to send messages. Concurrent transmissions are modeled to cause collisions if the radio coverage of the senders intersects. We use a randomized shortest path to route the messages to the sink. A node randomly chooses its next hop nodes if they have a shorter path to the sink. Thus routing path is not the same every time. We change the traffic load by varying the inter-arrival period of the messages. If the message inter-arrival period is 1 seconds, a message is generated every 1 seconds by each source node. The active period is set to 100 ms for S-MAC with the duty cycle of 10%.

4.2 The Setting of MAX_Counter

MAX_Counter is a parameter influences the performance of TA_MAC. When MAX_Counter is large, there'll be long delay if traffic load turns from heavy to light quickly. When MAX_Counter is small, there'll be more collisions if traffic load fluctuates drastically, and then more energy consumed. Fig. 5 shows the trade-off between energy consumption and packet delay. As we can see, when MAX_Counter is 3, there is better trade-off than other values. We set MAX_Counter to 3 for the following simulations.

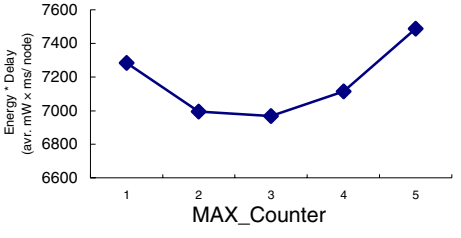


Fig. 5. The influence of MAX_Counter

4.3 Simulation Results

We choose 3 metrics to evaluate the performance of TA-MAC.

1) Energy consumption: the average energy consumption per node to deliver a certain number of packets from sources to sink. This metric shows the energy efficiency of the MAC protocols.
2) Latency: the average end-to-end delay of all nodes.
3) Throughput or delivery ratio: the ratio of the number of packets arrived at the sink to the number of packet sent by sources.

Fig. 6 shows the average packet latency for different message inter-arrival period. Clearly, IEEE 802.11 has the lowest latency for all traffic loads. S-MAC, however, has much higher latency, especially when traffic load is heavy (e.g. at small message inter arrival). TA-MAC has a slightly lower latency than S-MAC by 23%. The reason is that when traffic load is very high, collisions would significantly increase packet

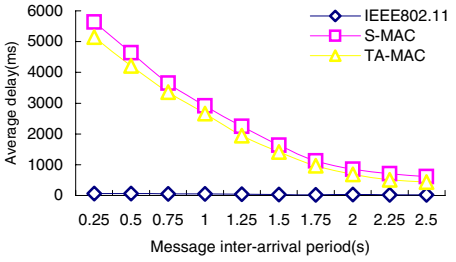


Fig. 6. The average packet latency under different traffic load

latency as a retransmission can only be done after on total schedule interval. But TAMAC enlarged the contention window under heavy traffic load, so that the probability of collision is lower than that of S-MAC with constant contention window. So the number of retransmission of TA-MAC is less than S-MAC. When traffic load is light, the size of contention window is randomly distributed from 0 to CW_{min}, which is less than the contention window of S-MAC. The small contention window and the fast backoff scheme lead to the lower latency than S-MAC when traffic load is low, too.

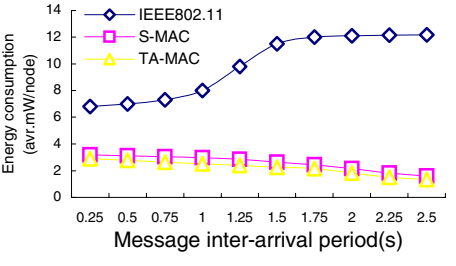


Fig. 7. The average energy consumption per node under different traffic load

Fig. 7 plots the average energy consumption of each MAC protocol when intensifying the traffic load. Energy consumption decreases as traffic load decreases. TA-MAC and S-MAC has better energy property, and far outperforms 802.11 MAC. TA-MAC performs better than S-MAC by about 15%-20%. It achieves energy saving mainly by reducing the number of collision under heavy load and reducing the idle listening time during backoff procedure under light load.

Fig. 8 shows the delivery ratio achieved for different MAC protocols. All MAC have quite good data delivery ratio near 1 when traffic load is very light. Obviously, when traffic load turns high, the throughput of 802.11 performs better than TA-MAC and S-MAC due to its low latency. Also, the throughput of TA-MAC improves by 150% than S-MAC when traffic is very high. This is because the collision probability of TA-MAC is less than that of S-MAC so that the number of retransmission is small.

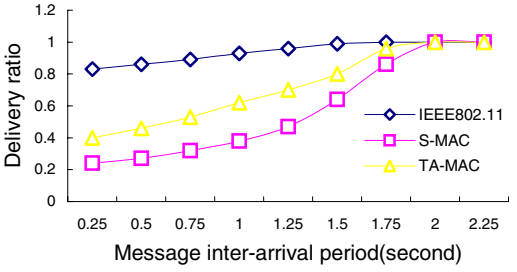


Fig. 8. Data delivery ratio under different traffic load

5 Conclusion and Future Work

This paper has proposed TA-MAC, a traffic load adaptive MAC protocol with fast backoff scheme in wireless sensor network. TA-MAC modified the contention window mechanism of S-MAC. It adjusts the initial contention window according to the current traffic load to reduce the collision probability and employs a fast backoff scheme to reduce the time for idle listening during backoff procedure, which both reduce the energy consumption.

Our simulation results have shown that TA-MAC achieves energy savings and higher throughput when traffic load is heavy. In our future work, we aim to implement this MAC on a Mote-based sensor network platform and evaluate its performance through real experiments.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey", *Computer Networks*, Vol. 38, pp. 393-422, March 2002.

[2] D. Estrin and R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: scalable coordination in sensor networks", in *Proc. of MobiCOM '99*, August 1999.

[3] T. S. Rappaport, *Wireless Communications, Principles and Practice*, Prentice Hall, 1996.

[4] N. Abramson, "Development of the ALOHANET", *IEEE Transactions on Information Theory*, vol. 31, no. 2, pp. 119-123, Mar, 1985.

[5] L. Kleinrock and F. Tobagi, "Packet switching in radio channels: Part I – carrier sense multiple access modes and their throughput delay characteristics," *IEEE Transactions on Communications,* vol. 23, no. 12, pp. 1400-1416, Dec. 1975.

[6] M. Stemm and R. H Katz, "Measuring and reducing energy consumption of network interfaces in hand-held devices," *IEICE Transactions on Communications*, vol. E80-B, no. 8, pp. 1126-1131, Aug. 1997.

[7] P. Havinga and G. Smit, "Energy-efficient TDMA medium access control protocol scheduling.," *In Proc. of Asian International Mobile Computing Conference (AMOC 2000)*, pp. 1-9, Nov. 2000.

[8] V. Rajendran, K. Obraczka, and J. Garicia-Luna-Aceves, "Energy efficient, collision-free medium access control for wireless sensor networks," *in 1st ACM Conf. on Embedded Networked Sensor Systems (SenSys 2003)*, pages 181-192, Los Angeles, CA, Nov. 2003.

[9] S. Kulkarni and M. Arumugam, "TDMA service for sensor networks," *In 24th int. Conf. on Distributed Computing Systems(ICDCS04), ADSN workshop*, pp. 604-609, Tokyo, Japan, March 2004.

[10] L. van Hoesel and P. Havinga, " A lightweight medium access protocol (LMAC) for wireless sensor networks," *In 1st Int. Workshop on Networked Sensing Systems (INSS 2004)*, Tokyo, Japan, June 2004.

[11] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, " Energy-efficient communication protocol for wireless microsensor networks", in *Proc. of 33rd Annual Hawaii International Conference on System Sciences*, Hawaii, January 2000.

[12] IEEE 802.11 standard. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11-1999 edition.

[13] W. Ye, J. Heidemann, and D. Estrin., "An energy-efficient MAC protocol for wireless sensor networks," in *21st Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 3, pages 1567–1576, June 2002.

[14] K. Langendoen and G. Halkes, "Energy-Efficient Medium Access Control", *Embedded Systems Handbook*, CRC press, to appear.

[15] T. van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *1st ACM Conf. on Embedded Networked Sensor Systems (SenSys 2003)*, pp. 171–180, Los Angeles, CA, November 2003.

[16] G. Lu, B. Krishnamachari, and C. Raghavendra, "An adaptive energy-efficient and low-latency MAC for data gathering in sensor networks," in *Int. Workshop on Algorithms for Wireless,Mobile, Ad Hoc and Sensor Networks (WMAN)*, Santa Fe, NM, April 2004.

[17] F. Cali, M. Conti, and E. Gregori, " IEEE802.11 Protocol: Design and Performance Evaluation of an Adaptive Backoff Mechanism," *IEEE Journal on Selected Areas in Communications*, Vol. 18, No. 9, pp. 1774-1786, Sep, 2000.

[18] Banchs A., Perez. X., "Providing throughout guarantees in IEEE 802.11 Wireless LAN," *in Proc. of IEEE WCNC 2002*, pp. 130-138.

[19] Y. Kwon, Y. Fang, and H. Latchman, "A Novel MAC Protocol with Fast Collision Resolution for Wireless LANs," *IEEE Infocom 2003,* 2003.

[20] http:// www.cs.berkeley.edu/~awoo/smartdust/.

[21] RF Monolithics Inc., http:// www. rfm. com/, ASH Transceiver TR1000 Data Sheet.

Semantic Sensor Net: An Extensible Framework

Lionel M. Ni[1], Yanmin Zhu[1], Jian Ma[1], Minglu Li[2], Qiong Luo[1], Yunhao Liu[1], S.C. Cheung[1], and Qiang Yang[1]

[1] Department of Computer Science,
Hong Kong University of Science and Technology, Hong Kong
[2] Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai, P.R. China
[1]{ni,zhuym,majian,luo,liu,scc,qyang}@cs.ust.hk,
[2]li-ml@cs.sjtu.edu.cn

Abstract. Existing approaches for sensor networks suffer from a number of critical drawbacks. First, homogeneous deployments have been commonly assumed, but in practice multiple deployments of sensor nets and heterogeneity of sensor networks are a serious problem. Second, existing approaches are very application-dependent and engineering-oriented. Third, there has been little standard available for WSNs. These drawbacks have significantly limited the further development of sensor networks. To overcome these critical drawbacks, we propose an extensive framework: Semantic Sensor Net (SSN). In brief, a semantic sensor net is a heterogeneous sensor network which enables dynamic tagging of semantic information to sensory data to allow more efficient and systematic monitoring and handling of the environmental dynamics to provide demanded services.

1 Introduction

Recent advances in wireless communications and microelectromechanical systems (MEMS) have led to the wide deployment of large-scale *wireless sensor networks* (WSN), which promises to revolutionize the way we monitor and control environments of interest [1, 2]. WSN was identified as one of the ten emerging technologies that will change the world in MIT Technology Review [3]. A wide variety of attractive applications [4] will come into reality, such as habitat monitoring [5], search and rescue, disaster relief, target tracking, precision agriculture and smart environments.

A sensor node is a low-cost and typically battery-powered device that integrates micro-sensing, onboard processing and wireless communication. A WSN is a self-organizing network composed of a large number of sensor nodes, tightly interacting with the physical world. Such a self-organizing network is able to not only disseminate sensory data across the network, but also provide in-networking real-time processing capability. WSN is a promising technology that effectively bridges the physical world and the digital world, by which we can extract critical information from physical environments, and therefore better monitor and control dynamics of environments.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 1144 – 1153, 2005.

Distinct from traditional computer networks, each sensor node in a sensor network plays a minor role. In addition, we are more interested in sensory data. The data-centric nature of sensor networks provides an innovative approach to solve a greater class of applications. However, current work in wireless sensor networks suffers some major drawbacks.

- Most solutions are based on homogeneous sensor array.
- Each solution is usually for a specific application.
- The solution is usually an engineering approach without a common framework.
- There is no standard to allow communication among different sensors at different levels.

This paper proposes a new concept called "semantic sensor net" to alleviate the above drawbacks. A semantic sensor net (SSN) is a heterogeneous sensor network which enables dynamic tagging of semantic information to sensory data to allow more efficient and systematic monitoring and handling of the environmental dynamics to provide demanded services. SSN has the following advantages

- The tagging of semantic information to sensory data allows efficient handling of large-scale distributed heterogeneous sensory data.
- SSN can provide a sound theoretical foundation to research in wireless sensor networks at different levels.
- SSN can help develop a semantic-based framework to systematically solve various applications.

The rest of the paper is organized as follows. Section 2 discusses characteristics and challenges of WSNs. We review existing approaches and their limitations in Section 3. In Section 4, we give an overview of Semantic Sensor Net. Some of our preliminary results are discussed in Section 5. Finally, Section 6 concludes the paper and introduces the directions of future work.

2 Characteristics and Research Challenges

Compared with traditional computer networks and mobile ad-hoc networks (MANET) sharing more similarities, sensor networks present many unique characteristics. Every sensor node is highly resource-constrained, with limited computational capability and small storage. The wireless communication is unreliable. Each sensor node is typically powered by battery and usually not rechargeable. Sensor nodes may be deployed in unattended environments, exposed to unpredictable damages from environments, and hence any sensor node is prone to failure.

Unlike computers in traditional computer networks, most sensor nodes have no global ID due to low-cost mass production. In addition, global IDs introduce too much overhead which are not affordable for resource-constrained sensor nodes. Because of the lack of global IDs, traditional networking methodologies are not appropriate for WSNs. Since any sensor node may become unavailable at any time and wireless communications are unreliable, the network topology may change constantly over time. Thus, the capability of self-organizing and self-configuring is fundamentally important. In many applications, senor nodes are usually deployed without per-node placements (e.g., dropping sensor nodes from a flying airplane). Given the

dynamic nature of sensor nodes, it is essential that these sensor nodes can cooperate with each other, form a network automatically, and work as a whole without human intervention.

With the rapid development of WSNs and its growing commercialization, it is probable that in an environment of interest different types of sensor nodes are deployed. Therefore, a sensor network could be very heterogeneous. Heterogeneity exists in both individual sensor nodes and sensor networks as a whole. So far there have been few standards available for WSNs. Different manufacturers produce distinct sensor network systems, adopting different hardware and software components. Even if we do not consider the heterogeneity caused by different manufacturers, sensor nodes from the same manufacturer can still be very heterogeneous in terms of function, capability, and so on. Sensors with different functions have been available, such as temperature, pressure, light, motion, and their combinations. Sensor networks as a whole can also be heterogeneous, since different sensor networks may employ different network-organizing strategies, routing algorithms, and aggregation methods. How to integrate heterogeneous sensor nodes to develop various, flexible and extensible applications has been a great challenge!

3 Existing Approaches and Their Limitations

In the past several years, sensor networks have received considerable research efforts, covering different aspects of design. In this section, we give an overview of existing approaches and study their limitations.

Because of resource constraint, the popular layered architectures used in traditional system design are not appropriate for sensor networks. Although a layered architecture can provide better organization and is more extensible, it introduces too much overhead since a packet may be added multiple headers of protocols, dominating the size of a packet. Existing research of sensor networks can still be roughly classified into several categories according to their functions.

- **Hardware and Wireless Communication** include sensor architecture, radio module design, MAC [6] and power control .
- **Infrastructure Establishment** includes deployment, localization [7], time synchronization [8], ID assignment and calibration, and middleware.
- **Network Organization** includes topology control [9], density control and cluster management.
- **Data Dissemination** includes routing [10], aggregation, compression, diffusion and query processing [11].
- **Applications** such as habitat monitoring, target tracking, battlefield surveillance, pollution monitoring [12], industry control, and so on.

Although extensive research has been conducted and some real applications have been in place, existing approaches suffer from several critical drawbacks which have significantly restricted the wide deployment of sensor networks in practical applications.

- **Homogeneous sensors are commonly assumed.** In a small-scale sensor network, it may be reasonable to have homogeneous sensors which are usually identical or similar in terms of function, node architecture and software. With the homogene-

ous assumption, solutions can be greatly simplified. However, in practice sensor networks are inherently very heterogeneous. Existing solutions based on the homogeneous assumption can hardly work in such heterogeneous systems.

- **Existing solutions are very application-dependent.** Applications of sensor networks are very diverse and could have very different requirements and objectives. For example, the battle field surveillance application requires sensor nodes report to the gateway only when some events are detected. In contrast, for a building temperature monitoring application, sensor nodes report to the gateway regularly. Given such two distinct application scenarios, the respective designs have been very different. Such application-specific solutions are not extensible and cannot be reused. Most existing applications of sensor networks adopted tailor-made designs.
- **Existing solutions are engineering-oriented.** In contrast to traditional computer networks, so far there has been few standard, like TCP/IP, available for WSNs. Due to the lack of widely-accepted standards, we have to re-design most building blocks when developing a new sensor network, such as topology control, routing algorithm, and query processing. Such engineering-oriented approaches are particularly inflexible, and pose developers a big burden of re-engineering. In order to avoid unnecessary re-engineering, it is essentially important to develop core standards for sensor networks.

Besides the above, existing solutions are not extensible in the sense that once a sensor network for a specific application has been deployed, it is extremely hard to accommodate application dynamics and new application additions over the same sensor network. It results from two reasons. First, the current hardware limitation does not allow frequent updates of software burned in sensor nodes, as it leads to unreliability and high power overhead. Second, there is no effective mechanism to support such application dynamics and new application additions, which in practice are very desirable to make applications better meet real needs.

Having suffered much from these major drawbacks, we come to realize that a solid foundation and framework for WSNs is highly necessary, through which we can overcome these drawbacks, and hence promote the further development of sensor networks.

4 Semantic Sensor Net: An Overview

To alleviate the drawbacks experienced by existing approaches, we propose an extensible systematic framework: *Semantic Sensor Net* (SSN). In brief, a SSN is a heterogeneous sensor network which enables dynamic tagging of semantic information to sensory data to allow more efficient and systematic monitoring and handling of the environmental dynamics to provide demanded services. The important concept *semantics* is introduced to address various challenges, which exists in different levels of designs of sensor networks, effectively enabling the integration, exchange, and reuse of sensory data across various applications and multiple sensor nets.

As mentioned in the previous section, any single sensor node is of negligible importance, and instead sensory data is what we are concerned the most. It suggests the data-centric principle, which is fundamentally different from the node-centric principle for traditional computer networks. The data-centric principle should be incorpo-

rated throughout designs of sensor networks. However, it is very challenging. Sensory data have unique characteristics and can be utilized in very different ways. High level applications usually require the integration of various sensory data. We believe the semantics-based framework can well address these challenges, and provide a solid foundation for WSN. Although sensory data can be very diverse, semantics inherently associated with sensory data can enable the integration and exchange of various sensory data, and accommodate different requirements of high-level applications.

The concept of semantics has been successfully introduced in semantic web, which is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation [13]. Traditionally, web pages are composed mainly for human's comprehension. Being intelligent, human beings have the ability to understand web information. However, it is difficult for digital computers to understand the meaning behind web pages. The semantic web brings structure to the meaningful content of web pages, so called semantics, enabling computers to carry out sophisticated tasks for users.

In SSN, semantics presents more flexible usage, which not only allows sensory data to be shared and integrated across various applications, but also provides a powerful framework for designs of sensor networks. Basically, *semantics refers to the critical meaning of sensory data, senor nodes and application requirements, which we believe can help better decision making in various designs of sensor networks.* Within SSN, semantics can exist in different levels from bottom to top, as shown in Fig. 1. Semantics in sensory data is the most basic, effectively supporting the realization of semantics in upper levels. Semantics in various applications is the most complex and can be factorized into much simpler forms. Semantics can be converted and form new semantics, and support efficient operations in different levels.

To demonstrate how semantics helps, let's take an example. Suppose in a building, a large number of heterogeneous sensor nodes are deployed for monitoring the environment inside the building. Example sensors include temperature, light and humidity. Now we may be concerned with whether there is a fire emergency inside the building. The 'fire emergency' certainly encompasses much semantics that can only be understood by human, and it resides on the service level. A fire can be roughly interpreted as a combination of strong light detections and high temperature detections in the same area. So the fire emergency is converted to a query with more specific semantics, "a strong light detection (>= 10 candlepower) plus a high temperature detection (>= 80 ℃) in the same region (distance <= 1 m) within 10 seconds". The query is then sent across the sensor network. On receiving the query, a sensor node will be able to interpret and then set up new routing rules. The basic semantics of the new routing rule, which resides on the data dissemination level, may act like the following. This query is prioritized to be forwarded to temperature and light neighbors such that other sensors can avoid being involved in, which is a power-efficient design. And, if a neighbor within ten meters reports a strong light detection and the node itself detects a high temperature event, it will form a fire alarm event and reports it to the gateway.

Besides semantics in the upper levels, semantics in the lower levels are also essential. A sensor node must maintain its own semantics, such as ID, location, sensing type, and sensing accuracy. And when a sensory data is available for transmission, its

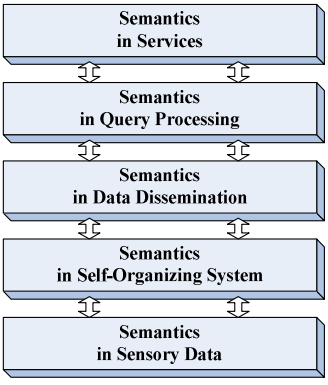


Fig. 1. Semantics exists in different levels of sensor networks

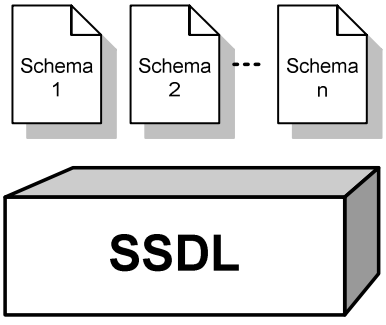


Fig. 2. Sensory-data Semantic Modeling Framework (SSMF)

semantics should be enclosed, enabling other sensors to interpret it; otherwise, other sensors may have no idea what the received data is. The semantic is on the bottom sensory data level. Given the highly heterogeneous sensors in the environment, an integrated network is expected to set up, including all possible sensors. Since different sensors may adopt different wireless technologies, direct communication between some sensors may not be possible. In this case, some nodes should act as bridges for these sensors. Bridges are on the networking level, and may require new semantics to annotate for themselves.

The success of SSN requires the following be well addressed.

- **Semantic Sensory Data Modeling.** Sensory data can be very diverse. To enable the integration and exchange of various sensory data, an expressive data model is highly needed.
- **Semantic Sensor Network System Architecture.** To facilitate the development of sensor network systems, extensible system architecture plays a fundamental role.
- **Semantic-based Data Dissemination.** Data dissemination is at the core of sensor networks. Semantic-based data dissemination protocols are able to enable dissemination of various sensory data over heterogeneous sensor networks.
- **Semantic Query Processing.** With the unique abilities of built-in computation and data storage, sensor networks are very promising in data management. Queries are the effective way to acquire useful information from sensor networks. Due to sensory data distributed across the whole sensor network as well as the heterogeneity of sensory data, query processing is challenging. The semantics-based framework could help more efficient query processing.
- **Services.** We need abstract and form services. Based on which various applications can be built conveniently.

5 Some Preliminary Work of SSN

In this section, we present some of our preliminary work in SSN. More advanced and detail work will be carried out in our future work.

5.1 Semantic Sensory Data Modeling

Each sensor generates some kind of raw data. To make the raw data meaningful, we have to tag it, i.e., we should attach the semantics to it. The semantics of sensory data is the necessary description about data generation environment where the raw data was generated. The description should include:

- **Meta data.** These are the necessary description about the raw data itself. Take raw data produced by a temperature sensor for example. We have to make it clear that the raw data is a temperature measurement, how accuracy it is, and in what conditions it is valid. Meta data usually depends on the capability of sensing devices. Different sensing devices may have different kinds of meta data.
- **Context information.** These are about the context information in which the raw data was generated, which are usually related to the sensor node which the sensing device is attached to [14]. Take the above for example. In general, it should be made clear that where the temperature measurement was made (i.e., location of sensor node), which node took the measurement if applicable (i.e., ID of sensor node), and when it was captured (i.e., timestamp).

Without these necessary descriptions, raw data itself is meaningless. Therefore, raw data should be attached with respective semantics. We have the intuition that the sensory data produced by two identical sensors should have the same presentation pattern of semantics. We term presentation pattern as schema. The same type of sensory data follows the same schema and different types of sensory data follow different schemas. As long as one has the right schemas, he is able to dynamically interpret the sensory data for various usages, despite how heterogeneous these sensory data are. We propose an expressive Sensory-data Semantic Modeling Framework (SSMF), as depicted in Fig. 2. Sensor-data Semantic Description Language (SSDL) is the language for defining schemas. Such a framework is very expressive and extensible. Users can conveniently define customized schemas using SSDL.

5.2 Semantic-Based Data Dissemination

Data dissemination plays a major role in enabling commands to be propagated from gateway nodes to sensor nodes and collecting sensed data of interest from sensor nodes back to the gateway nodes for further study. It is very challenging, however, to design effective data dissemination protocols for sensor networks. It would become even more difficult if we consider the heterogeneity in both sensors and sensor networks as a whole. Heterogeneous sensors in a sensor network produce heterogeneous sensory data. Based on the sensory data, different actions may be required. Such semantic-based routing requirements make the design of data dissemination protocols greatly complicated. In addition, heterogeneous sensor networks may deploy different network protocols and network organization. Some applications may require cooperation among these networks. Such inter-network heterogeneity makes the design of data dissemination protocols even more challenging.

We need to propose **semantics-aware routing**. It is not a specific routing algorithm. Instead, it is rather a framework, which aims to enable efficient data dissemination over large-scale heterogeneous sensor networks. Despite of the heterogeneity, it is certainly desirable that all of the sensor nodes work as a whole network, other than

several separate networks working independently. Working as a whole can provide longer network lifetime, more efficient data dissemination, more complete sensory data and more flexible application integration.

Built over various existing routing algorithms, semantic-based routing should be extensible in the sense that it can address new emerging semantics possibly added to the sensor networks. It should take the advantage of available semantics in both sensory data and sensor nodes. Some possible scenarios are summarized as follows.

- **Semantics in sensor node.** After receiving a packet from its neighbors, the sensor node should take proper actions based on its capability (one kind of semantics). Suppose the destination of the packet is defined by a geographic region, and the node has no knowledge of this physical location. In this case, the sensor node should forward the packet to one of this neighbor which knows its physical location, or simply drop the packet.
- **Semantics in sensory data.** To save energy, data aggregation is a popular technique associated with routing in sensor networks. Rather than routing back every piece of sensory data, some sensory data are aggregated at some nodes, and the resulting data are then routed back to the gateway. Such in-network decisions should be based on semantics in the sensory data.
- **Semantics in query.** In general, queries are about some specific kind of sensory data. If a sensor node is able to interpret semantics in queries, irrelevant sensor nodes can avoid being involved so that more power can be saved. For example, if a query is issued to ask about temperature information, it is desirable that light sensors are not involved. After receiving a query, a sensor node can selectively forward the query to those neighbors which are also temperature sensors. By this means, light sensors will less be involved, and therefore such a query is more power-efficient.

5.3 Semantic Query Processing

The advent of wireless sensor networks provides a unique distributed platform to acquire and query streams of data. Traditional computational models transmit and process the data at each time instant and each individual record at a time. These models can no longer hold true for sensor networks. Furthermore, each sensor node only provides a small piece of the picture. To obtain an overall picture of the area, we must be prepared to answer queries about high-level patterns in place, instead of answering queries that concern low-level information that concern only limited space and time [15]. In other words, we must be able to provide ***semantic-level answers*** to ***pattern-related queries.***

Semantic queries distinguish themselves from the traditional queries with two major characteristics:

- Queries can involve aggregated environmental conditions, are location-context dependent and related to the tracking and monitoring of moving objects
- Queries are answered by taking into consideration of the device semantics on sensor-distribution topology, state and capabilities and answers can depend on power, sensors, and levels of confidence

Equipped with a variety of sensors, a wireless sensor network could make semantic inference about its environment, with queries on such conditions as the overall temperature, humidity, vibration, sound and lighting. All these queries can be answered dependent on location context [15]. A query planning system must be intelligent in order to provide timely and confident answers using a minimum amount of energy.

6 Conclusions and Future Work

For wireless sensor networks, any individual sensor node is by itself unimportant. Instead, sensory data collected from a group of sensors are what we are most concerned about. This suggests a data-centric principle in data processing, which is distinct from the node-centric principle in traditional computer networks. In response to the new challenges posed by the sensor networks, in this paper we have proposed an extensible framework known as the Semantic Sensor Networks. By explicitly exploiting the semantic information, which uncovers the machine-understandable meaning embedded in low-level sensory data, sensor nodes and application requirements, SSN enables the integration, reuse, and exchange of sensory data across various applications.

In this paper, we have just begun to touch the tip of the iceberg in SSN research. Our future work aims to make SSN practical for real developments of sensor networks, especially for large-scale heterogeneous sensor networks. To this end, a wide range of topics should be extensively studied, including an extensible architecture to address the highly heterogeneous nature of WSNs used in practice, the practical semantics-based data dissemination protocols, the semantics-based query processing methods and service methodologies for developing sophisticated applications.

Acknowledgement

We thank HKUST, Hong Kong Research Grant Council, and National Natural Science Foundation of China for providing support to this research. L. Ni, Q. Yang, Q. Luo and S.C. Cheung are supported in part by RGC grants HKUST 6264/04E, HKUST 6187/04E, HKUST 6263/04E, and 6167/04E, respectively. M. Li is supported by National Natural Science Foundation of China (No. 60473092) and Program for New Century Excellent Talents in University (No. NCET-04-0392).

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," Computer Networks, vol. 38, pp. 393--422, 2002.
[2] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," ACM Communications, vol. 43, pp. 51-58, 2000.
[3] MIT Technology Review, Feb. 2003, http://www.techreview.com.
[4] D. Estrin, G. P. L. Girod, and M. Srivastava, "Instrumenting the world with wireless sensor networks," presented at ICASSP, Salt lake City, UT, 2001.

[5] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," presented at the 1st ACM international workshop on Wireless sensor networks and applications, 2002.
[6] E. Shih, S. H. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan, "Physical Layer Driven Protocol and Algorithm Design for Energy-Efficient Wireless Sensor Networks," presented at 7th Annual International Conference on Mobile Computing and Networking (MOBICOM), 2001.
[7] L. Doherty, K. S. J. Pister, and L. E. Ghaoui, "Convex Position Estimation in Wireless Sensor Networks," presented at INFOCOM'01, Anchorage, AK, 2001.
[8] J. Elson and D. Estrin, "Time Synchronization for Wireless Sensor Networks," presented at the 15th International Parallel and Distributed Processing Symposium, 2001.
[9] A. Cerpa and D. Estrin, "ASCENT: Adaptive Self-Configuring sEnsor Networks Topologies," presented at 21 Annual Joint Conference of the IEEE Computer and Communications Societies, 2002.
[10] A. Salhieh, J. Weinmann, a. M. Kochhal, and L. Schwiebert., "Power Efficient Topologies for Wireless Sensor Networks," presented at International Conference on Parallel Processing (ICPP). Valencia, Spain, 2001.
[11] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "The Design of an Acquisitional Query Processor for Sensor Networks," presented at SIGMOD'03, San Diego, CA, 2003.
[12] H. Ngan, Y. Zhu, and L. M. Ni, "SAS: Stimulus-based Adaptive Sleeping for Wireless Sensor Networks," presented at the 34th International Conference on Parallel Processing (ICPP'05), Norway, 2005.
[13] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," Scientific American, 2001.
[14] C. Xu, S. C. Cheung, C. Lo, K. C. Leung, and J. Wei, "Cabot: On the Ontology for the Middleware Support of Context-Aware Pervasive Applications," presented at Building Intelligent Sensor Networks (BISON'04) in conjunction with IFIP International Conference on Network and Parallel Computing, Wuhan, China, 2004.
[15] A. Deshpande, C. Guestrin, S. Madden, J. M. Hellerstein, and W. Hong, "Model-Driven Data Acquisition in Sensor Networks," presented at VLDB'04, Toronto, Canada, 2004.

Loop-Based Topology Maintenance in Wireless Sensor Networks

Yanping Li[1], Xin Wang[2], Florian Baueregger[3], Xiangyang Xue[4], and C.K. Toh[5]

[1] Software School, Fudan University, Shanghai 200433, China
042053011@fudan.edu.cn
[2] Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, Shanghai 200433, China
xinw@fudan.edu.cn
[3] Department of Computer Sciences, Friedrich-Alexander University, Erlangen-Nuremberg, Germany
florian.bauerreger@gmx.net
[4] Department of Computer Sciences, Fudan University, Shanghai 200433, China
xyxue@fudan.edu.cn
[5] Department of Electronic Engineering, Queen Mary University of London, Britain
c.k.toh@elec.qmul.ac.uk

Abstract. Clustering is recognized as an effective topology maintenance framework to provide better scalability for Wireless Sensor Networks. When loop structures are deployed to topology maintenance, it can provide better robustness and convenient route recovery, for the topology information is disseminated within a loop instead of a single node and the nature of the loop that there are two paths between each pair of nodes within a loop provides backup routes. In this paper, we propose a novel topology maintenance approach based on loop structures. The algorithm is a distributed, on-demand and configurable. It prefers a low-mobility network which is typically suitable for wireless sensor networks.

1 Introduction

Large-scale wireless sensor networks are envisioned to be the next generation computing systems for information monitoring and tracking. Although a single low-power sensing device may not reliable or accurate, by networking a number of sensors, autonomous coordination against themselves extends the ability of signal processing. Preferable model of a wireless sensor network is an ad hoc network, which is a self-organizing network of mobile wireless nodes that do not depend on any fixed infrastructure [1] [2]. In such a network, one way to achieve better scalability and robustness is to control the topology by *clustering*. Clustering is a process that divides the network into interconnected substructures, called *clusters*. Each cluster has a *cluster head* (CH) as coordinator within the substructure.

In the research of ad hoc networking, many clustering algorithms have been proposed. These algorithms can be categorized according to the graph formed by all the

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 1154 – 1162, 2005.

CHs [3]. One big family is to cluster by *independent dominating sets*, where every two CHs have no direct connections. The linked cluster algorithm proposed in [4], the lowest-ID algorithm and the highest degree algorithm presented in [5] and the max-min D-hop clustering strategy in [6] are of this family. Clustering with independent dominating set is normally easy to implement but the generated structures have *chain reactions* problem [7]. Another family is based on the connected dominating set, where each CH is connected with at least one another. Though it is ideal to have the *minimum connecting dominating set* (MCDS) as the set of CHs, which ensures the least broadcast redundancy, yet it is proved that the MCDS decision problem is NP-hard in a general graph [8]. Some approximation algorithms for finding MCDS are presented in [9] [10] [11].

In this paper we propose a novel topology control approach which regards a loop as the structure of a cluster. The loop-base clustering is aimed to benefit good features of the loop-based topology and to achieve better robustness and convenience to the wireless sensor network.

The paper is organized as follows. In this section we explain the motivation of our work. In section 2 we describe the models and algorithms of our approach. In section 3 we present the result and analysis of the simulation. And in section 4 we give the conclusion.

2 Models and Algorithms

2.1 Definitions

A loop is a bidirectional path which begins and ends with the same node. Given an undirected graph $G = (V,E)$ representing an ad hoc network, where V representing the set of nodes and E the set of connections. There is at most one connection between every two node, that is, if there exist two edges e_i and e_j connecting v_x and v_y, then e_i and e_j must be identical, so a path from v_n to v_m can be defined as a sequence of only vertices $\{v_n, v_{n+1}, \ldots, v_m\}$. We define a loop as a sequence of vertices $\{v_n, v_{n+1}, \ldots, v_m\}$ where $v_i \neq v_j (i \neq j)$ for any $n \leq i \leq m-1$ and $n \leq j \leq m-1$, and $v_n = v_m$. The length of a loop is the number of hops from v_i to v_j, equal to the numbers of nodes on the loop minus 1. Let l be a loop. When $len(l)$ is smaller than 3, either the node on l is isolated or l is a round trip between two nodes. A loop with only two nodes is regarded as a special loop. Fig. 1 shows an example.

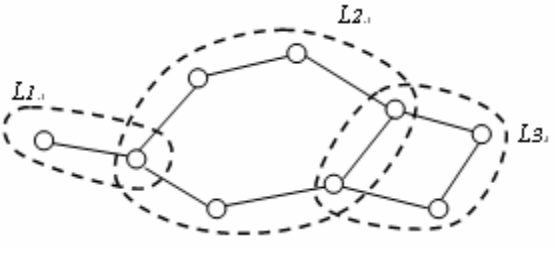


Fig. 1. This figure shows an example for loop-based topology, where L_2 and L_3 are typical loops and L_1 is a two node special loop

2.2 The Loop-Based Topology

Our approach regards a selected loop as a cluster and the entire network is grouped into interconnected loops. Within a loop, nodes can exchange information with each other by forwarding the messages along the loop in either direction. For inter-loop communications, messages are first routed to the nodes that are on more than one loop, called *gateways*, then by forwarding from gateway to gateway, the message reaches the destination loop and then along the destination loop, as is a inner-loop transmission, the message is finally forwarded to the destination.

Loop topologies have many good features. 1) There is no critical clusterheads defined in a loop, so the loop-based topology never suffers from chain reactions [7] caused by the changes of CHs. 2) Within a loop, since every node is necessary to have knowledge of other nodes on the loop, if the information of the local loop reserved in one node is corrupt, by querying the neighbor nodes, the loop knowledge can be recovered, which provides the network with better robustness. 3) One of the natures of a loop that there are two paths between every two nodes on the same loop provides a backup route for connection loss during message transmission. Fig. 2 shows how backup route works.

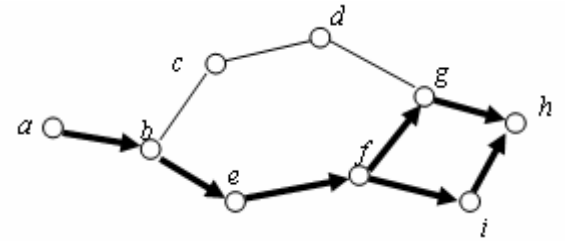


Fig. 2. Suppose data packets are being transferred from node *a* to *h* along the path *{a, b, e, f, g, h}*, and suddenly node *f* detects that the connection to node *g* is lost (perhaps due to power exhaustion). Because *f* knows that there is another path to *h*, that is *{f, i, h}*, for *f* and *h* are in the same loop, it then forward the data packets to node *i*, and relayed by *i*, the data packets will finally reach the destination *h*

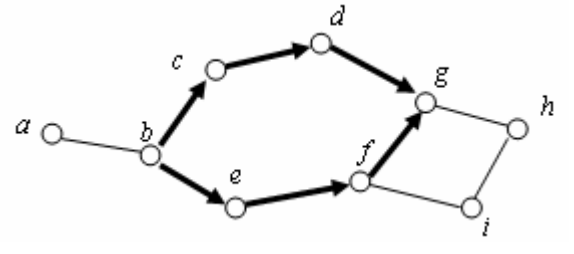


Fig. 3. In the execution of source routing discovery algorithm, node *g* receives a route request with a precursors list *{b, c, d}* and another one with *{b, e, f}*, both of which are forwarded by node *b*, but along different paths. Since there are two different paths from *b* to *g* and all connections are bidirectional, a loop *{b, c, d, g, f, e, b}* is then discovered in between

The loop concept is often found in relation to the concept of loop-freedom in literatures on routing, where algorithms are carefully designed to avoid message loops which cause resource waste. Here, on the contrary, we want to make use of loops.

The idea of the loop-based approach was inspired by the method used in *source routing* [12] especially the one described in [13]. Usually a source routing protocol defines a route request message which includes a special field called *precursors list* containing the identifiers of all nodes that has forwarded the message. We observe that during the flooding of the route request messages, it is common that a node receives more than one messages forwarded by a same node. See Fig. 3.

2.3 The Basic Algorithm

Here, we propose a distributed loop-based clustering algorithm. It executes on an on-demand basis and discovers a configurable loop topology for the network. Because the nodes in sensor networks are likely to be fixed and the transmission radius of each sensor is configurable, the algorithm assumes that 1) it is applied in a fully connected network with only bidirectional links, 2) each node has a unique identifier throughout the network, and 3) during one execution of the algorithm, topology of the network keeps unchanged. The algorithm is executed at each node independently. As the result of the algorithm, each node on any discovered loop knows the other nodes that are on the same loop.

The basic algorithm is described as follows.

When a source node s attempts to transfer data to a destination node d, a *Loop Request (LREQ) lreq* is generated with *lreq.source*$\leftarrow s$ and *lreq.precursors*$\leftarrow \{s\}$. After the LREQ is generated, it is broadcast locally (within the transmission range).

When a node v receives the LREQ *lreq*, it follows the steps below.

1. If v is in *lreq.precursors*, which implies that v has already received and processed this message, then *lreq* is simply discarded and go to step 6.
2. If *v.loop* is empty, then search each path in *v.path list* for common node in *lreq.precursors*. Let $\{p_1, p_2, \ldots, p_i, \ldots, p_m\}$ denote *lreq.precursors*. Let $\{q_1, q_2, \ldots, q_j, \ldots, q_n\}$ denote a path in the *v.path list*. If for some i, j that $p_i = q_j$, and for all $i<x<m$ and $j<y<n$, we have $p_x \neq q_y$, then a loop $\{p_i, p_{i+1}, \ldots, p_m, v, q_n, q_{n-1}, \ldots, q_{j+1}, p_i\}$ denoted as l is discovered. Then
   a) A *Loop Reply (LREP) lrep* is generated with *lrep.loop*$\leftarrow l$ and *lrep.source*$\leftarrow v$ and is sent to either node p_n or q_m (choose one according to some criteria).
   b) *v.loop*$\leftarrow l$.
3. Add *lreq.precursors* to *v.path list*.
4. Let *lreq.precursors* be $\{p_1, p_2, \ldots, p_m, v\}$ and broadcast it locally.
5. End.

When a node v receives the loop reply *lrep*, it follows the steps below.

1. If v=*lrep.source*, which indicates that *lrep* has tripped back to the home node, discard it and go to step 5.
2. Add *lrep.loop* to *v.loop list*.

3. If *v.loop* is empty, then *v.loop*←*l*. This means that node *v* joins the loop discovered by *lrep.source*.
4. Forward *lrep* to the next hop on *lrep.loop*.
5. End.

Fig. 4 shows a typical execution of the algorithm.

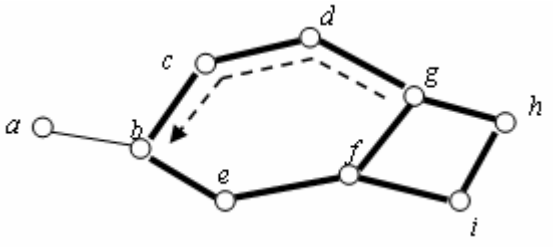


Fig. 4. Suppose node *b* initiates the algorithm by generating a loop request *lreq{source*←*b, precursors*←*b}* and broadcasts it locally. Respectively, node *a*, *c* and *e* will receive this message, add their identifiers at the end of the *lreq.precursors* and broadcast it locally. When node *b* gets the *lreq* forwarded from *a*, *it* will discard the message because *b* is found in *lreq.precursors*. When node *d* and *f* receive *lreq* from *c* and *e* respectively, they add their identifiers at the end of *lreq.precursors* and broadcast it locally. When node *g* receives *lreq* from node *d* and *f* successively, two paths from the same node *b* generate the loop *{b, c, d, g, f, e, b}* and an LREP *lrep{loop*←*loop, source*←*g}* is relayed along the loop, then each other node *d*, *c*, *b*, *e*, *f* joins the loop successively. Similarly, node *g* adds its identifier at the end of *lreq.precursors* and broadcasts it locally. Same process is proceeding at every node. After a while, loop *{f, g, h, i, f}* or/and *{b, c, d, g, h, i, f, e, b}* will be found. Finally, each node *v* (if it is on any loop with the length equal or greater than 3) decides which loop it belongs to, and reserves it in *v.loop* and any other loop containing *v* that is reserved in *v.loop list*

2.4 The Configurable Algorithm

In the basic algorithm, we notice that each node is prone to join the earliest discovered loop and does not take into account whether it is good. So the result of loop-based topology discovered is hard to predict. The simplest criterion for determining good loops is the length of the loop. If the network is clustered with bigger loops, the number of loops is smaller and it is easier for routing. However, since there are more nodes on one loop, the possibility of connection loss increases and then the topology becomes less stable. Contrarily, if the network is clustered with smaller loops, the entire topologies are prone to be more stable since changes of the nodes only affects the local loop. However, because of the big number of the loops, the routing can be very complex. So to choose a good loop length is a trade-off between stability and routing performance which is related to the scale of the network. However, we want to make the result of the algorithm at lease controllable. And thus we define a parameter at each node: *Expected Loop Length (Lexp)*, and make a slight change to the process of LREQ and LREP. That is whether it has reserved a loop or not, it execute loop discovery and before reserving the loop discovered either by matching paths during processing LREQ or from reading LREP message, compare the length of the

loop with Lexp. Only when the length of the loop is closer to Lexp than the reserved one at present if there is any, the new loop is reserved, overwriting the old one. For example, in Fig. 4, suppose node *i* discovers loop *{b, c, d, g, h, i, f, e, b}* with length 8 and loop *{f, g, h, i, f}* with length 4 and its Lexp is 3. Then *i* will choose to join loop *{f, g, h, i, f}* which has better length.

Since LREQs are sending by broadcast, it is likely to cause *broadcast storm*, investigated in [14], which will lead to dramatic power consuming and will be vital for sensor networks. In order to relieve it, we add a *TTL* field to LREQ. TTL determines the maximum number of hops an LREQ can reach, thus to reduce number of messages.

A side effect brought by TTL is the bounded affected area of the algorithm in the network. It is possible that part of the network is affected by the previous execution of the algorithm due to the TTL of LREQ. So in response to a received LREQ, a node may utilize the topology knowledge it has reserved if there is any.

The algorithm does not discover two-node special loops. However, a node can detect whether it is on-loop or not from the *path list* and *loop list*. If the *loop list* is empty but the *path list* is not, considering the bounded affected area of the algorithm, probably the node is on a special loop.

3 Simulation

The effectiveness of the configurable algorithm and some features of it have been evaluated in a static simulation with *n* uniformly distributed nodes in a 1000x1000 square area. For simulation, we assume that all nodes share the same transmission radius *r* and that all nodes within connection range establish a bidirectional link between each other. Values for *n* range from 20 to 60, *r* is increased in steps of 20 at a starting value of 50, TTL is 3 or 4. *Lexp* is 3. All simulation results are the average values of 200 independent executions of the algorithm.

At first we have a look at how many nodes the algorithm can reach during the loop discovery. Thus we examine the coverage, defined as the percentage of nodes on loops after a single execution of the algorithm. Fig. 5 show the coverage for TTL=3 and respectively 4.

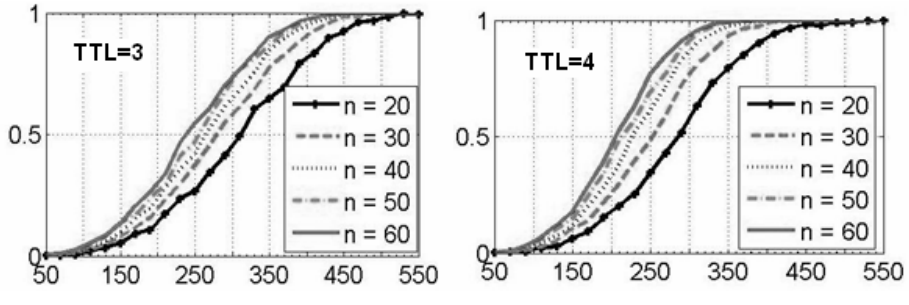


Fig. 5. As can be seen, coverage reaches 90% for *r* between 250 and 350 with TTL=4 while reaching the same value with TTL=3 needs a value for *r* which is approximately 70 to 100 higher in order to cover 90% of the nodes. So increasing the value of TTL helps keep transmission range low and save power consumption

Next to consider are the number of gateways the algorithm discovers (Fig. 6) and their degree (Fig. 7). The degree of a gateway is defined here as the average number of loops a gateway connects. Only loops with 3 or more nodes are counted.

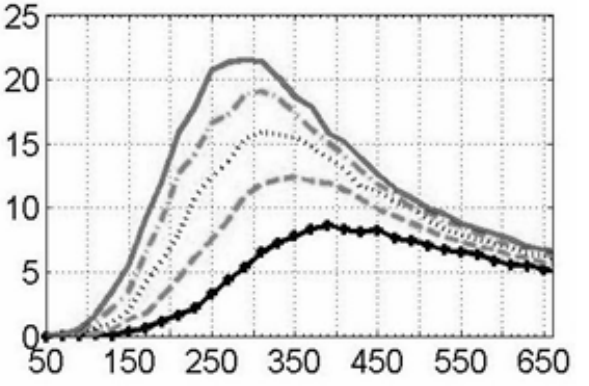


Fig. 6. shows that the number of gateways rapidly increases to a maximum of 30 to 50 percent of all nodes when transmission radius *r* is between 200 and 350. As stated before, in this range a coverage of 90% of the nodes is ensured

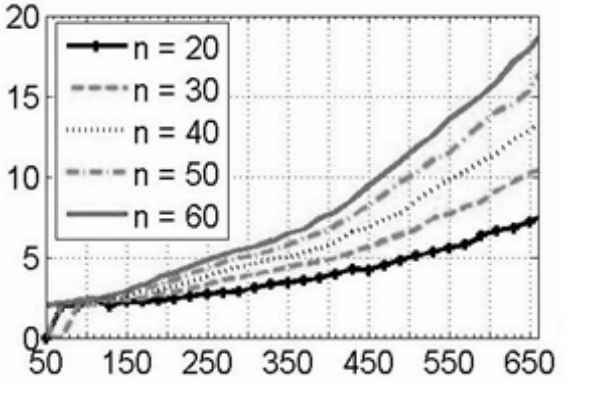


Fig. 7. As can be seen in this figure, every gateway has an average degree of about 5, when transmission radius *r* is between 200 and 350. So typically an average node has to transfer data among 5 adjacent loops

Our special interest is on how TTL can improve loop discovery at a low transmission radius. In Fig. 8, 30 nodes have been tested with increasing values for TTL. Since a TTL of 6 increases the number of packets used during flooding already by a big amount, the algorithm has been run with every set of parameter only 30 times and then averaged. Still, *Lexp*=3.

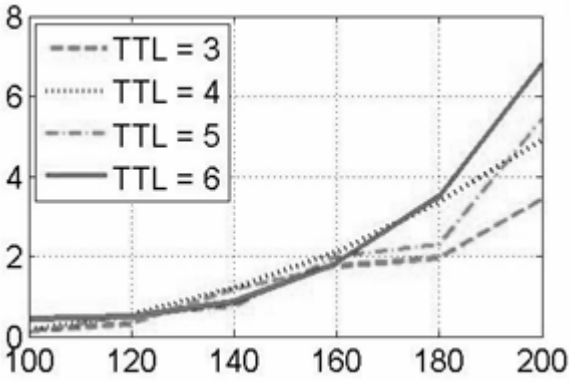


Fig. 8. Although the values are not very representative, due to the low number of repetitions, it is yet noticeable that with TTL=3 only about 3 loops could be discovered at transmission range r=200, the algorithm could discover about 7 loops with TTL=6

3 Conclusion

In this paper, we proposed a loop-based topology maintenance approach. The objective of the loop-based approach is to utilize good features of loop topologies in low-mobility sensor networks. Loop-based clustering algorithm is a distributed, on-demand and configurable algorithm. The simulation result shows that the algorithm can discover loop-based topologies and by configuring the parameters such as *TTL* and *Lexp*, we can control the size of the discovery loops and the affected area of the algorithm.

For the next step, we will evaluate the performance of the proposed algorithm in terms of complexity and the quality of the discovered loops. Further research will be carried out on the resilience technology to the node and link failures based on the loop topology.

Acknowledgement

This work was supported in part by 863-2002AA103011-5, Shanghai Municipal R&D Foundation under contracts 035107008, MoE R&D Foundation and Shanghai Key Laboratory of Intelligent Information Processing (IIPL).

References

1. D. Estrin, R. Govindan, J. Heidemann, S. Kumar: Next century challenges: Scalable coordination in sensor networks. In: Proc. MOBICOM1999, Seattle, 263-270.
2. D. Estrin, L. Girod, G. Pottie, M. Srivastava: Instrumenting the World with Wireless Sensor Networks. In: International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001), Salt Lake City, Utah, May 2001.

3. Yuanzhu P. Chen, Arthur L. Liestman, Jiangchuan Liu: Clustering Algorithms for Ad Hoc Wireless Networks. In: Y. Pan and Y. Xiao (eds): Ad Hoc and Sensor Networks. Nova Science Publishers (2004).
4. D. J. Baker, A. Ephremides: The architectural organization of a mobile radio network via a distributed algorithm. In: IEEE Transactions on Communications, COM-29 (1981), pp. 1694-1701.
5. M. Gerla, J. T.-C. Tsai: Multicluster, mobile, multimedia radio network. In: Wireless Networks, 1 (1995), pp. 255-265.
6. A. D. Amis, R. Prakash, T. H.P. Vuong, D. T. Huynh: Max-min d-cluster formation in wireless ad hoc networks. In: Proc. IEEE INFOCOM, 2000, pp. 32-4.
7. M. Gerla, T. J. Kwon, G. Pei: On demand routing in large ad hoc wireless networks with passive clustering. In: Proc. IEEE WCNC, September 2000.
8. B. N. Clark, C. J. Colbourn, D. S. Johnson: Unit disk graphs. In: Discrete Mathematics, Vol.86, 1990, pp. 165-177.
9. K.M. Alzoubi, P.-J. Wan, O. Frieder: New distributed algorithm for connected dominating set in wireless ad hoc networks. In: Proc. HICSS 2002.
10. U. Feige: A threshold of ln n for approximating set cover. In: Proc. ACM Symposium on Theory of Computing, 1996, pp. 314-318.
11. S. Basagni: Distributed clustering for ad hoc networks. In: Proc. ISPAN'99 Int. Symp. on Parallel Architectures, Algorithms, and Networks, 1999, pp. 310-315.
12. David B. Johnson, David A. Maltz, Yih-Chun Hu: The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR). Internet Draft, draft-ietf-manet-dsr-10.txt, July 2004, Work in progress.
13. Y. Sagawa, T. Asano, H. Higaki: Loop-Based Source Routing Protocol for Mobil. In: Proc. AINA'03, March 27~29, Xi'an, China, 2003, p834-837.
14. Y. Chen S. Ni, Y. Tseng, J. Sheu: The broadcast storm problem in a mobile ad hoc network. In: Proc. MOBICOM. Seattle, Washington, 1999.

Generating Minimal Synchronizable Test Sequence That Detects Output-Shifting Faults

Chuan-dong Huang[1] and Fan Jiang[2]

[1] University of Science and Technology of China, Department of Computer Science, Postalcode 230027, Hefei, People's Republic of China
yellow@mail.ustc.edu.cn
[2] University of Science and Technology of China, Department of Computer Science, Postalcode 230027, Hefei, People's Republic of China
fjiang@ustc.edu.cn

Abstract. During the application of a test sequence in a distributed test architecture, the existence of multiple testers brings out the possibility of synchronization problems among remote testers and the possibility that output-shifting faults go undetected. These problems often require the use of coordination message exchanges among testers. This paper proposes a method that generates synchronizable test sequences that detect output-shifting faults and the use of coordination message is minimized. This method utilizes a set of transformation rules to construct an auxiliary digraph from a given specification and test generation involves finding a rural Chinese postman tour in the digraph to yield the minimal test sequence.

1 Introduction

The objective of testing is to determine whether an *Implementation Under Test* (*IUT*) conforms to its specification. Testing is often realized by generating test sequences from the specification and applying them to the implementation in a test architecture. When testing a distributed system, a distributed test architecture (Fig.1) is needed. In this architecture, the *IUT* contains a number of separate interfaces, called ports and the test system consists of a local tester for each port of the *IUT*. Each local tester communicates with the *IUT* through its corresponding port.

During the application of a test sequence in a distributed test architecture, the existence of multiple testers brings out the possibility of synchronization problems among remote testers. The synchronization problem arises if a tester cannot determine when to apply a particular input to the *IUT*. It is therefore important to construct a synchronizable test sequence that no two consecutive inputs cause a synchronization problem and hence the coordination among testers is achieved indirectly through their interactions with the *IUT* [6]. However, for some specifications, there is no synchronizable test sequence [2]. In this case, it is necessary for testers to exchange coordination message directly through a reliable communication medium which is independent of the *IUT*. Another problem in distributed testing is that output-shifting faults may go undetected. Due to the lack of a global clock, it is difficult to determine the input

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 1163 – 1170, 2005.

which is the cause of a particular output. Even if the behaviors of all the ports are the same as expected, the output-shifting faults may stay in the *IUT* [5]. To ensure the detectability of output-shifting faults, the test sequence needs to be augmented either by additional subsequences selected from the specification of the *IUT* [5] or by coordination message exchanges [3]. Again, for some specifications, there may not exist a test sequence where output-shifting faults can be detected without using coordination message [8].

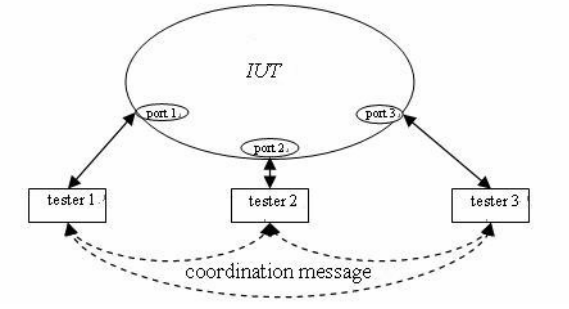


Fig. 1. A distributed test architecture

Both of these problems often require the use of coordination message. However, there is a cost associated with the use of such messages. It is desirable to construct a minimal test sequence where both coordination messages and input/output operation costs are taken into consideration. In this paper, we will introduce a digraph in which every path represents a synchronizable test sequence that detects output-shifting faults. Test generation is then expressed in terms of this digraph.

The rest of the paper is organized as follows: Section 2 gives the preliminaries. Section 3 presents the proposed method. Section 4 gives the conclusions.

2 Preliminaries

2.1 Multi-port Finite State Machine and Its Graphical Representation

A multi-port Finite State Machine with n ports (np-FSM) is a 6-tuple $(S, \Sigma, \Gamma, \delta, \lambda, s_0)$, where

- S is a finite set of states and $s_0 \in S$ is the initial state.
- $\Sigma = (\Sigma_1, \Sigma_2, ...\Sigma_n)$, where Σ_k is the input alphabet of port k, and $\Sigma_i \cap \Sigma_j = \varnothing$, for $i \neq j,\ i,\ j, k \in [1, n]$. Let $I = \Sigma_1 \cup \Sigma_2 \cup ... \cup \Sigma_n$.
- $\Gamma = (\Gamma_1, \Gamma_2, ...\Gamma_n)$, where Γ_k is the output alphabet of port k, and $\Gamma_i \cap \Gamma_j = \varnothing$, for $i \neq j,\ i,\ j, k \in [1, n]$. Let $O = (\Gamma_1 \cup \{\varepsilon\}) \times (\Gamma_2 \cup \{\varepsilon\}) \times ... \times (\Gamma_n \cup \{\varepsilon\})$, where ε stands for the null output.
- δ is the transition function : $D \rightarrow S$, and λ is the output function $D \rightarrow O$, where $D \subseteq S \times I$.

A *transition* of an *np*-FSM is a triple $(s_j, s_k; x/y)$ where $s_j, s_k \in S$, $x \in I$, $y \in O$, such that $\delta(s_j, x) = s_k$ and $\lambda(s_j, x) = y$. An *np*-FSM M can be represented by a directed graph $G = (V, E)$ where a set of vertices V represents the set S of states of M and a set of directed edges E represents all specified transitions of M. Each edge $e_{jk} = (v_j, v_k; x/y)$ represents a state transition from state v_j to state v_k with input x and output y where the *input/output pair x/y* is the *label* of e_{jk}. Two 3*p*-FSMs are represented in Fig.2, with $\Sigma_1 = \{a\}$, $\Sigma_2 = \{b\}$, $\Sigma_3 = \{c\}$, $\Gamma_1 = \{\pi, \zeta\}$, $\Gamma_2 = \{\beta\}$, $\Gamma_3 = \{\gamma\}$. For example, if s_0 is the current state and the input a is received, then state changes to s_1 and the outputs π and γ are sent in ports 1 and 3 respectively.

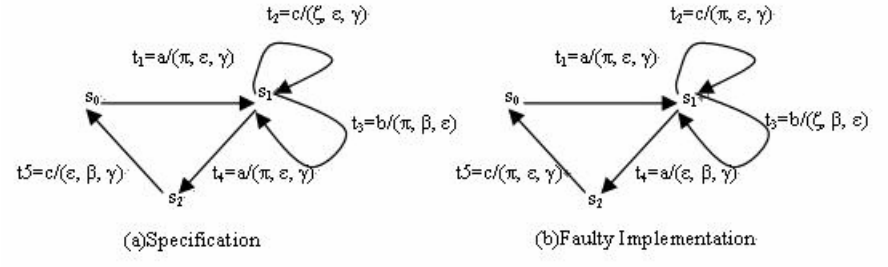


Fig. 2. Two examples of 3*p*-FSMs

An *np*-FSM M is *deterministic* if, for each input, there is at most one transition at each state of M. An *np*-FSM M is *minimal* if none of its states are equivalent. In a digraph $G = (V, E)$, given a vertex $v \in V$, let $d_v^+(G)$ denotes the number of edges that leave v and $d_v^-(G)$ denotes the number of edges that enter v. A *walk* is a finite sequence of adjacent but not necessarily distinct edges. A *tour* is a walk that starts and ends at the same vertex. A *rural postman tour* of G over $E_c \subseteq E$ is a tour that traverses every edge in E_c at lease once. The cost of each edge of G is equal to the number of input/output pairs in its label. The cost of a tour is the sum of the cost of edges included in the tour. A *rural Chinese postman tour* (*RCPT*) over $E_c \subseteq E$ is a minimum-cost rural postman path. Computing an *RCPT* is known to be NP-complete. But, if the subgraph induced by the edges in E_c is weakly connected, then finding an *RCPT* over E_c takes polynomial time [1].

2.2 The Synchronization Problem

During the application of a test sequence in a distributed test architecture, the synchronization problem arises if a tester cannot determine when to apply a particular input to *IUT* because it is not involved in the previous transition, i.e. it does not send the input or receive the output in the previous transition.

Given $x \in I$, let *port*(x) denotes the port associated with input x and the tester at port k be denoted by Tester$_k$. Given $y = (y_1, y_2, \ldots y_n) \in O$, let *ports*($y$) denotes the set of ports associated with values from y that are not null.

Definition 1. The sequence *tt'*, for transition $t = (s_i, s_j; x/y)$ and $t' = (s_j, s_k; x'/y')$, has a *synchronization problem* if $port(x') \notin ports(y) \cup \{port(x)\}$ [4].

Let us consider the *IUT* of Fig.2b, which contains four faults (in transitions t_2, t_3, t_4 and t_5) with regard to the specification of Fig.2a. Let us consider the following sequence which corresponds to sequence of transitions $t_1\ t_3\ t_2\ t_4\ t_5$ of Fig.2a: $<a/(\pi, \varepsilon, \gamma)$, $b/(\pi, \beta, \varepsilon)$, $c/(\zeta, \varepsilon, \gamma)$, $a/(\pi, \varepsilon, \gamma)$, $c/(\varepsilon, \beta, \gamma)>$. The test sequence requires that when s_1 is reached, the *IUT* receives *b* (sent by $Tester_2$) before it receives *c* (sent by $Tester_3$). Since $Tester_3$ does not send the input or receive any output of *IUT* in transition t_3, then $Tester_3$ can not know whether *IUT* has already received *b*. In other terms, $Tester_3$ has no means to determine the order of inputs *b* and *c*. Here is an example which shows the effect of this problem on fault detectability. From state s_1, the three testers observe the same outputs in the following two situations: 1) the correct *IUT* receives *c* before *b* and 2) the faulty *IUT* receives *b* before *c*. Therefore, the test system cannot deduce whether the *IUT* is correct or not because it is not aware of the order of inputs *b* and *c*.

If *tt'* does not have a synchronization problem then it is said to be synchronizable. A test sequence is said to be synchronizable if all subsequences within it are synchronizable. However, for some *np*-FSMs, there may be no synchronizable test sequence [2]. In this case, it is possible to include coordination message among testers. The solution proposed by [3] can be explained as follows, for two consecutive transitions $t = (s_i, s_j; x/y)$ and $t' = (s_j, s_k; x'/y')$. Let $port(x) = h$, $port(x') = k$. If $y \neq \varepsilon$, let $Tester_m$ be defined as follows:

- if $y_k \neq \varepsilon$: $Tester_m = Tester_k$
- if $y_k = \varepsilon$ and $\exists p$ such that $y_p \neq \varepsilon$ and $y'_p = \varepsilon$: $Tester_m = Tester_p$
- otherwise : $Tester_m$ is any tester such that $y_m \neq \varepsilon$

The synchronization problem is then resolved by the use of a message *C* as follows:

- if $y = \varepsilon$ and $h \neq k$: after it sends *x*, $Tester_h$ sends a message *C* to $Tester_k$
- if $y \neq \varepsilon$ and $m \neq k$: after it receives y_m , $Tester_m$ sends a message *C* to $Tester_k$

2.3 The Output-Shifting Faults

Due to the lack of a global clock in distributed testing, it is difficult to determine the input which is the cause of a particular output. Even if the behaviors of all the ports are the same as expected, the output-shifting faults may stay in the *IUT* [5]. In the following, given $y \in O$ and $y_k \in \Gamma_k$, let $y \oplus (k, y_k)$ denote *y* with its *k*th component replaced by y_k.

Definition 2. Suppose two transitions t = (si, sj; x/y) and t' = (sj, sk; x'/y') are sequenced to form tt'. Suppose also that $yk \neq \varepsilon$, $y'k = \varepsilon$, $port(x') \neq k$, and that the actual outputs in the corresponding transitions in the IUT are $y \oplus (k, \varepsilon)$ and $y' \oplus (k, yk)$ respectively. This combination of faults is called a forward output-shifting fault [7].

Definition 3. Suppose two transitions t = (si, sj; x/y) and t' = (sj, sk; x'/y') are sequenced to form tt'. Suppose also that $yk = \varepsilon$, $y' \neq \varepsilon$, $port(x') \neq k$, and that the actual outputs in the corresponding transitions in the IUT are $y \oplus (k, yk)$ and $y' \oplus (k, \varepsilon)$ respectively. This combination of faults is called a backward output-shifting fault [7].

Definition 4. A fault is an output-shifting fault if it is either a forward output-shifting fault or a backward output-shifting fault.

In Fig.2b, let us consider the consecutive transitions t_4 and t_5. If we compare with the specification of Fig.2a, output π has been "shifted" from t_4 to t_5 and β has been "shifted" from t_5 to t_4. With a distributed architecture, these faults are not detected because all the testers observe their expected outputs (Fig.3) although the *IUT* is faulty.

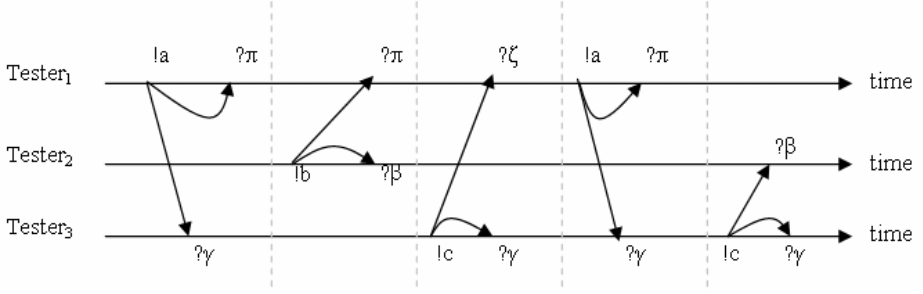


Fig. 3. Expected behaviors of each tester

Remark 1. The sending of an input x is denoted !x. The reception of an output x is denoted ?x.

To ensure the detectability of potential output-shifting faults, the test sequence needs to be augmented either by additional subsequences selected from the specification of the *IUT* [4] or by coordination message exchange among testers [3]. Again, for some specifications, there may not exist a test sequence where output-shifting faults can be detected without using direct coordination message exchange [8]. As it will be used in this paper, the solution proposed in [3] can be explained as follows, for any two consecutive transitions $t = (s_i, s_j; x/y)$ and $t' = (s_j, s_k; x'/y')$. Let $port(x) = h$, $port(x') = k$ and let

- Tester$_p$ be any tester such that $p \neq k$, $y_p \neq \varepsilon$, $y'_p = \varepsilon$ and after it receives y_p, Tester$_p$ does not send a message C to Tester$_k$
- Tester$_q$ be any tester such that $q \neq k$, $y_q = \varepsilon$, $y'_q \neq \varepsilon$

The output-shifting faults can be detected as follows: Before it sends x', Tester$_k$ sends a message O to each Tester$_p$ (detect forward output-shifting faults) and Tester$_q$ (detect backward output-shifting faults).

3 Proposed Method

Formally, the proposed method is a solution of the following problem: consider a minimal and deterministic *np*-FSM M represented by a digraph $G = (V, E)$. Let $\Phi(M)$ be the set of all implementations of M, each of which has only output faults and has

the same sets of states, inputs, outputs and the initial state as M. Taking into consideration both transition and coordination message costs, construct a minimum-cost synchronizable test sequence that detects output-shifting faults and distinguishes M from any faulty implementation of M in $\Phi(M)$. This problem can be resolved by finding an *RCPT* over all the transitions of M in an auxiliary digraph $G' = (V', E')$ obtained from G. Suppose that the cost of each coordination message (message C or message O) is ω and the cost of each transition is 1.

3.1 Description of the Proposed Method

The digraph $G' = (V', E')$ can be constructed from $G = (V, E)$ as follows:

1. for each edge $t = (s_i, s_j; x/y) \in E$, create a pair of vertices $I_i^{port(x),ports(y)}$, $F_j^{port(x),ports(y)}$ and a solid edge $(I_i^{port(x),ports(y)}, F_j^{port(x),ports(y)}; x/y)$
2. Given vertices $F_j^{port(x),ports(y)}$ and $I_j^{port(x'),ports(y')}$, $t = (s_i, s_j; x/y)$, $t' = (s_j, s_k; x'/y')$, create a dashed edge $(F_j^{port(x),ports(y)}, I_j^{port(x'),ports(y')})$ from $F_j^{port(x),ports(y)}$ to $I_j^{port(x'),ports(y')}$
3. for each dashed edge$(F_j^{port(x),ports(y)}, I_j^{port(x'),ports(y')})$, $t = (s_i, s_j; x/y)$, $t' = (s_j, s_k; x'/y')$, if there are no other edges that leave $F_j^{port(x),ports(y)}$ or enter $I_j^{port(x'),ports(y')}$, then change the dashed edge into a solid one.

Every path in digraph G' represents a synchronizable test sequence that detects output-shifting faults. In (1), each solid edge $(I_i^{port(x),ports(y)}, F_j^{port(x),ports(y)}; x/y)$ represents a transition whose cost is 1. In (2), each dashed edge $(F_j^{port(x),ports(y)}, I_j^{port(x'),ports(y')})$ represents the addition of coordination messages. Its cost is determined by the number of coordination messages related to synchronization problems and potential output-shifting faults between $t = (s_i, s_j; x/y)$ and $t' = (s_j, s_k; x'/y')$. This is given by the following two cases:

- if $port(x') \notin (ports(y) \cup \{port(x)\})$ and $ports(y) \setminus ports(y') = \varnothing$ then cost is $\omega(1 + |ports(y') \setminus (ports(y) \cup \{port(x')\})|)$
- otherwise cost is

$$\omega(|ports(y) \setminus (ports(y') \cup \{port(x')\})| + |ports(y') \setminus (ports(y) \cup \{port(x')\})|)$$

In (3), there is only one dashed edge that leaves $F_j^{port(x),ports(y)}$ or enters $I_j^{port(x'),ports(y')}$. Since at lease one solid edge enters $F_j^{port(x),ports(y)}$ or leaves $I_j^{port(x'),ports(y')}$, that dashed edge will be traversed in the minimum-cost tour that traverses every solid edge at lease once. So, that dashed edge is changed into a solid edge to increase the likelihood of obtaining a weakly connected subgraph consisting of solid edges. If the solid edges form a weakly connected spanning subgraph, a polynomial time algorithm [1] can be used to obtain an *RCPT* over the solid edges in the digraph G' which becomes a minimal synchronizable test sequence that detects output-shifting faults. For example, let us consider the 2p-FSM of Fig.4.

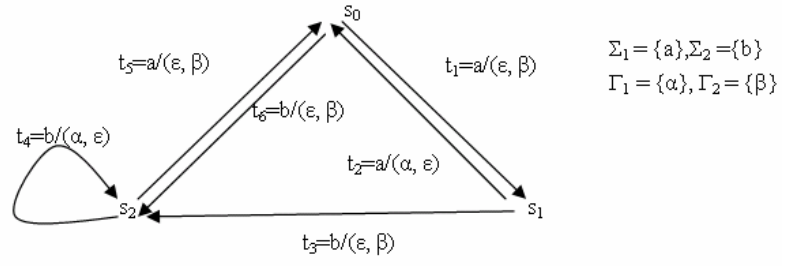


Fig. 4. Digraph of 2*p*-FSM

After step (1), (2) and (3), we will obtain an auxiliary digraph G' (Fig.5). The costs of edges in G' are given below:

1. Cost 0 for the edges: $F_0^{1,\{2\}} \to I_0^{2,\{2\}}$, $F_0^{1,\{2\}} \to I_0^{1,\{2\}}$, $F_1^{1,\{2\}} \to I_1^{2,\{2\}}$
2. Cost ω for the edges: $F_0^{1,\{1\}} \to I_0^{1,\{2\}}$, $F_1^{1,\{2\}} \to I_1^{1,\{1\}}$, $F_0^{1,\{1\}} \to I_0^{2,\{2\}}$, $F_2^{2,\{2\}} \to I_2^{2,\{1\}}$, $F_2^{2,\{2\}} \to I_2^{1,\{2\}}$, $F_2^{2,\{1\}} \to I_2^{1,\{2\}}$
3. cost 1 for other edges(transitions)

Then, an *RCP*T over the solid edges : $I_0^{1,\{2\}} \to F_1^{1,\{2\}} \to I_1^{1,\{1\}} \to F_0^{1,\{1\}} \to I_0^{2,\{2\}} \to F_2^{2,\{2\}} \to I_2^{2,\{1\}} \to F_2^{2,\{1\}} \to I_2^{1,\{2\}} \to F_0^{1,\{2\}} \to I_0^{1,\{2\}} \to F_1^{1,\{2\}} \to I_1^{2,\{2\}} \to F_2^{2,\{2\}} \to I_2^{1,\{2\}} \to F_0^{1,\{2\}} \to I_0^{1,\{2\}}$, gives the minimal synchronizable test sequence that detects output-shifting faults for a total of five coordination message exchanges and eight transitions.

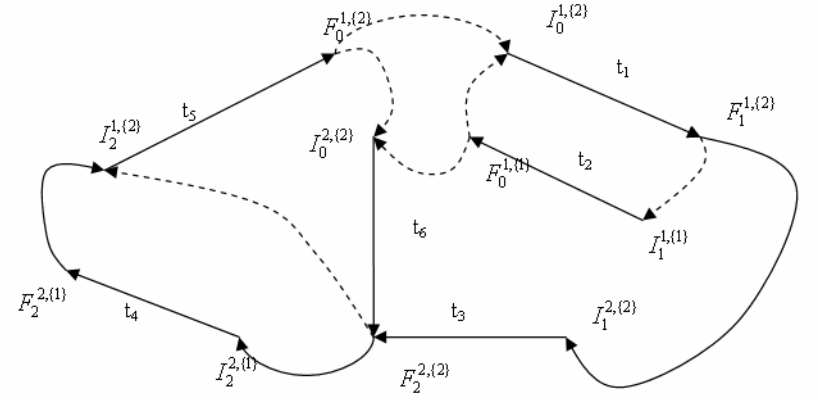


Fig. 5. Auxiliary digraph G'

3.2 Properties of the Proposed Method

Given the digraph $G = (V, E)$ of *np*-FSM $M = (S, \Sigma, \Gamma, \delta, \lambda, s_0)$, let $| V | = | S | = m$, $| E | = r$, $| I | = | \Sigma_1 \cup \Sigma_2 \cup ... \cup \Sigma_n | = p$. As every transition from M leads to at most 2 vertices in $G' = (V', E')$, G' has $O(r)$ vertices. There are $O(r)$ edges repre-

sented transitions of M in G'. Let $G'' = (V'', E'')$ denotes the line graph of G. There are $O(|E''|)$ edges that represent the addition of coordination messages. Since there are at most $|I|$ edges that leave each vertex in G, $d_v^+(G) = |I| = p$, for $i = 1, 2 ..., m$. And,

$$|E''| = \sum_{v \in V} d_v^+(G) d_v^-(G) = O(p \sum_{v \in V} d_v^-(G)) = O(p|E|) = O(pr) . \quad (1)$$

Thus, G' has $O(r) + O(pr) = O(pr)$ edges.

4 Conclusions

The increasing significance of distributed systems has lead to much interest in issues relating to the test of such systems. In distributed testing, the existence of multiple testers complicates testing because remote testers may encounter synchronization problems and output-shifting faults may go undetected during the application of a test sequence. It is important to generate synchronizable test sequence that detects output-shifting faults in distributed testing. This may necessitate the addition of coordination messages among testers.

This paper has proposed a method that utilizes a set of transformation rules to obtain an auxiliary digraph from a given *np*-FSM. Test generation involves finding a rural Chinese postman tour in the digraph to yield a minimal synchronizable test sequence that detects output-shifting faults.

For some *np*-FSMs, the output-shifting faults may be detected by the insertion of test subsequences instead of coordination message exchanges. Producing the necessary and sufficient conditions under which such test subsequences exist and finding a minimum-length subsequence are our future work.

References

1. A. Aho, A. Dahbura, D. Lee, M. Uyar: An optimization technique for protocol conformance test generation based on UIO sequences and rural Chinese postman tours, IEEE Trans. Comm.39(11)(1991) 1604-1615
2. S. Boyd, H.Ural: The synchronization problem in protocol testing and its complexity, Inform. Process. Lett. 49(1991) 131-136
3. L. Cacciari, O.Rafiq: Controllability and observability in distributed testing, Inform. Software Technol. 41(1999) 767-780
4. R.M. Hierons: Testing a distributed system: generating minimal synchronised test sequences that detect output-shifting faults, Inform. Software Technol. 43(9)(2001) 551-560
5. G. Luo, R. Dssouli, G.v. Bochmann, P. Venkataram, A. Ghedamsi: Test generation with respect to distributed interfaces, Comput. Standards Interfaces 16(1994) 119-132
6. B. Sarikaya, G.v. Bochmann, Synchronization and specification issues in protocol testing, IEEE Trans. Comm. 32(1984) 389-395
7. Y.C. Young, K.C.Tai, Observation inaccuracy in conformance testing with multiple testers, IEEE WASET(1998) 80-85
8. J. Chen, R.M. Hierons, H. Ural, Conditions for Resolving Observability Problems in Distributed testing, FORTE 2004, LNCS 3235, 229-242

Considering Network Context for Efficient Simulation of Highly Parallel Network Processors

Hao Yin[1], Zhangxi Tan[1], Chuang Lin[1], Geyong Min[2], and Xiaowen Chu[3]

[1] Department of Computer Science and Technology, Tsinghua University, China
{hyin, xtan, clin}@csnet1.cs.tsinghua.edu.cn
[2] Department of Computing, School of Informatics, University of Bradford, UK
G.Min@Bradford.ac.uk
[3] Department of Computer Science, Hong Kong Baptist University, Hong Kong
chxw@comp.hkbu.edu.hk

Abstract. Researching aspects of parallel architecture and system design concerning network processors requires excellent simulation tools. In this paper, we develop a tool called NPNS on top of a widely used network simulator ns-2, which combines full-system simulators to provide a unified framework for processor and software design within a network context. In this article, we describe the architecture of NPNS and its implementation issues. In addition, an example of simulating a multi-threaded Intel IXP1200 network processor with NPNS is illustrated.

1 Introduction

In response to the continuous growth in network bandwidth and flexibility requirements in packet routers, application specific chips called network processors (NPs) have been developed. To assist researchers in more efficiently evaluating the functionality and performance of NPs, execution-driven full-system simulation tools seem to be a good choice. Processor designers can use these tools to arbitrarily parameterize, control and inspect system components. For system builders, they provide a virtual target platform running unmodified application binaries and help to evaluate general implementation issues. In many popular simulators, such as SimpleScalar [1] and SimOS [2], much emphasis is focused on the exploration of processor architecture, parallel characteristic of system and system components. One important factor that has been consistently simplified or ignored in network processor simulations is the network context, in which NPs are finally working. NPs are specially designed and optimized for network applications, such as protocol processing, packet classification and routing. However, these applications should be tested and simulated within a complete network scenario, which is a description of network topology, protocols, workloads and control parameters. Hence, a small toy external workload is not sufficient any more for evaluating the performance of NPs. We must validate and simulate hardware or software designs within the context of network.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 1171–1180, 2005.

To address this issue, we combine full-system NP simulators with popular network simulators, which provide substantial support for network simulation like TCP, routing and multicast protocols without the expense of building a real network. Hence, the network context is created with a network simulator and full-system NP simulators are embedded as network nodes to simulate the actions of these processors. Based on this idea, we have developed a tool called Network Processor Network Simulator (NPNS) built upon a widely used network simulator - ns-2 [3]. NPNS does not modify the fundamental structure of ns-2, but rather extends it by defining a new scheduler and a set of components. Thus, almost all features inherent in ns-2 are preserved and NPNS can be easily used with basic knowledge about ns-2.

The rest of this article is organized as follows. We briefly review ns-2 and introduce the architecture of NPNS. Then, we discuss implementation issues of NPNS, including modifications and extensions to ns-2 and the interface with full-system NP simulators. After that, we illustrate an example of simulating a packet classification algorithm on Intel IXP1200 network processor to demonstrate the usage of NPNS. Finally, we describe related work and draw some conclusions.

2 Ns-2 Brief and NPNS Architecture

Ns-2 (Network Simulator Version 2) is an object-oriented discrete event-driven simulator that can simulate a variety of IP networks. It implements network protocols such as TCP and UDP, traffic source behavior, router queue management mechanism, routing algorithms and more. The components of ns-2 are separated into two languages spaces: C++ and OTcl (Tcl script language with object-oriented extensions). Components implemented in C++ space run fast that can efficiently interpret packet headers and implement algorithms running over large data sets. Scripts in OTcl space run much slower but are flexible in changing, making them ideal for configurations, topology setup and manipulations of components in C++ space.

NPNS has inherited this split-language architecture for network processor simulations, as shown in Fig. 1. In addition, it includes a Network Processor Application space for simulating applications executed on network processors, which mainly consists of execution-driven full-system network processor simulators, namely NP simulators for the sake of simplicity. NP simulators can perform the functional simulation of network processors. They accept application programs in an appropriate binary format even unmodified codes as input, and simulate the instruction execution. Also, performance statistics of network processors such as memory references, processor utilization rate are exported. For performance concerns, NPNS also allows NP simulators to be implemented as NPNS components in C++ space. Like ns-2, NPNS is an interpreter as well, taking OTcl scripts. Since NPNS components in C++ are modified or implemented following ns-2 definitions, initializing NPNS is quite similar to that of ns-2 except that it uses a new scheduler and configures NP simulators. NPNS inherits most of components in ns-2 but make necessary modifications when combining

with NP simulators. Moreover, components communicate with each other and NP simulators by exchanging packets, which can be traced and outputted using ns-2 output format. Besides, these traces can also be viewed with visualization tools, like Nam (Network Animator) [4].

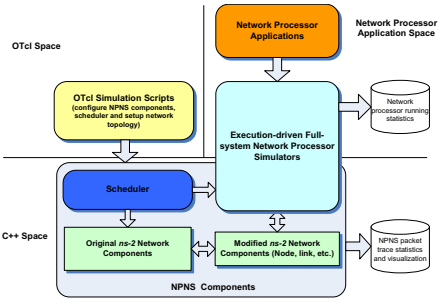


Fig. 1. NPNS architecture

3 NPNS Implementation Issues

Several problems should be solved in NPNS implementation when combining two different types of simulators:

- **Scheduler.** Network simulators are event-driven while NP simulators are execution-driven. Once they work in NPNS simultaneously, the new NPNS scheduler should be able to drive them at the same time.
- **Packet format transformation.** Since network simulators tend to simulate an abstract network, simplified packet descriptors are employed instead of packets with real data. On the contrary, NP simulators consume real packets. Thus, NPNS should perform necessary transformation of packet formats when passing packets between two types of simulators.
- **Modifications on Node and Link objects in ns-2.** Node and link are compound objects in ns-2. However, their capability to specify packet processing or transfer delay is limited (ns-2 specifies this by advancing simulation time through simple calculations). This can be compensated when NP simulators are involved. Hence, these objects should be redesigned in order to cope with the packet processing in NP simulators.

3.1 Scheduler

The scheduler is used to select the next event from the event queue and advances the simulation time. ns-2 has implemented two different types of scheduler (i.e., real-time and non-real-time schedulers). The new scheduler in NPNS is developed on a non-real-time ns-2 scheduler. By using it, we extend the original event engine of ns-2s and make it capable of scheduling execution-driven NP simulators as

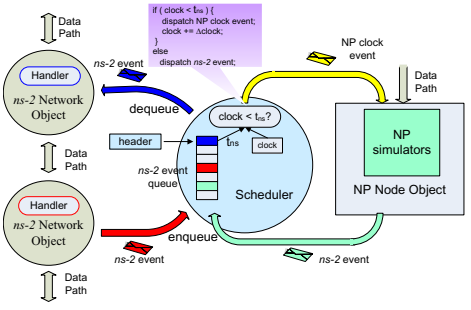


Fig. 2. The new scheduler in NPNS

well, as illustrated in Fig. 2. The left part is to schedule ns-2 network objects, collections of unmodified ns-2 network components. When running, these objects (e.g. delay timers) generate ns-2 compatible events (simply called ns-2 events). They are placed in the ns-2 event queue (red arrow). Also, these events will be sorted in the queue according to the rule of earliest time first. The right part is to schedule NP node objects, modified version of Node objects in ns-2 containing NP simulators for nodal processing simulation. The new scheduler has defined a variable named clock to track the execution of NP simulators (usually clock cycles). Once NP simulators have been notified by the scheduler with NP clock event (yellow arrow), they will run a definite period (clock) and then automatically halt, advancing the clock variable. To synchronize the simulation time, the scheduler inspects the clock variable and compares it with the time in the header of ns-2 event queue and dispatch a proper event (text balloon in purple). Moreover, NP node objects can also generate ns-2 events when packets cross the boundary between NP simulators and the network simulator. These events are also queued in the ns-2 event queue.

3.2 Packet Format Transformation

In most NPNS components inherited from ns-2, packets in ns-2 format are used. These packets are composed of a stack of headers and an optional data space, as presented in Fig. 3. Usually, the header stack includes all usable or registered headers during simulator initialization, such as a common header used by ns-2 objects, TCP/IP header and etc. These headers are reordered and exist whether or not they have been used. In most scenarios, data space is not allocated. On the other hand, this format is incompatible with full-system NP simulators, which take real packets and strictly follow the situation in an actual network. Fortunately, the format transformation is straightforward. First, a data space is allocated to each packet. Then, packet headers are sorted while unused headers are removed. In addition, this procedure can be carried out in the converse direction when converting packets back to the ns-2 format. Fig. 3 demonstrates packet transformation between ns-2 format and that used in NP simulators (typically, TCP/IP packet).

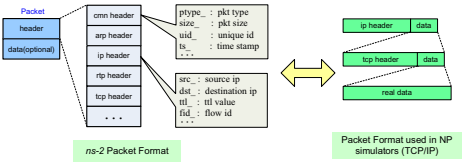


Fig. 3. Packet format transformation in NPNS

3.3 NP Node Object

NP node object is the core object of NPNS derived from an ns-2 unicast node object, which is a compound object of a node entry, an address classifier and a port classifier. Additionally, NP node object includes two packet format transformer and NP simulators for packet processing simulations, shown in Fig. 4. During the simulation, packets flow from left to right illustrated by the arrow line in Fig. 4. First, a packet in ns-2 format enters the node entry from a link object, which is used to connect nodes and complete the topology. It is then transformed to the format available to NP simulators and forwarded to them, as described early. Once the packet has been processed by NP simulators, it is converted back to ns-2 format and passed to an address classifier. In original ns-2 definitions, the address classifier conducts activities of routing and classification. Since these are completed when using NP simulators, the address classifier has been simplified and only need to identify where the packet is going. If the packet should be transferred to other nodes, the address classifier sends it to link objects that connect the target nodes. If the packet destination is the NP node object itself, it will be forwarded to a port classifier to locate a proper agent that represents endpoints where network-layer packets are consumed. Although NP simulators can deal with packets whose destination is the NP node itself, they still generate virtual ns-2 packets and pass them to ns-2 agents. Here, agents are no longer used with protocol processing but simply maintain the application level connections with other unmodified ns-2 objects in NPNS.

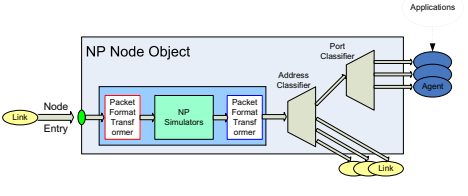


Fig. 4. NP node object

3.4 Link Object

Like node object, link object is another major compound in ns-2. As mentioned early, Link objects connect node objects and complete the topology. In normal

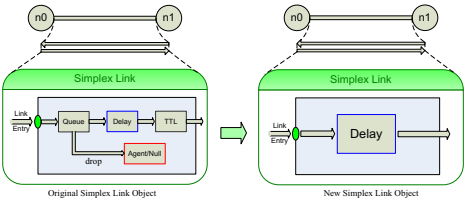


Fig. 5. Modifications on the simplex link object

ns-2 simulation, link objects do more than transferring packets. Some operations which ought to be implemented on network nodes in real applications are also carried out by link objects, such as output queue, delay, TTL calculation and many well known queue management algorithms. In NPNS, they are also completed by NP simulators on NP node objects, thus link objects are only responsible for transfer delay when connected with them. Fig. 5 illustrates the new link object in NPNS, which is modified on the base of the fundamental unidirectional link object in ns-2, known as the simplex link object. Other complex link objects are created with multiple simplex links or derived from it.

3.5 Full-System NP Simulators

When involving full-system NP simulators, NPNS not only tries to utilize numerous academic and commercial processor simulators, but also allows processor designers to write their own system simulators and integrate them with NPNS. NPNS employs NP simulators in a non-intrusive (or black-box) manner and makes minimum controls (e.g. stop and run) and modifications on existing simulators. One advantage of this method is that a system design with network processors can be prototyped using a standalone NP simulator without any internal modifications on NPNS. There are four types of NP simulators that NPNS currently or is going to support:

- Instruction set simulators, for example, the well known processor simulator SimpleScalar [1], the ARMulator [5] for ARM microprocessors and the MIPS Free GNU toolkit [6] for MIPS architectures.
- Complete machine simulators, such as SimOS [2] and Simics [7]. These simulators include all system componets, like processor, memory, I/O devices, etc., and model them in sufficient details to run real programs.
- Cycle-accurate simulators provided by network processor vendors. These simulators can execute codes compiled for network processors and obtain exhaustive hardware/software statistics. In our first implementation of NPNS, we use Workbench 2.01 for Intel IXP1200 network processor [8].
- Full-system simulators implemented as NPNS components. Sometimes full-system simulators especially complete machine simulators and cycle-accurate simulators suffer from low performance because they consider too many hardware details and serve as virtual machines constantly interpreting binary

codes at runtime to take relevant actions. Fortunately, network processors, especially their processing engines (PEs) for data plane applications, can be viewed as a special case of embedded processors. They only have small code storages or firmware. For instance, Intel IXP1200 only has 2k lines of code for each microengine (data plane processor) and 2 Mbytes flashrom for StrongArm (control plane processor) applications. To improve the performance, we no longer need to implement the simulator as an interpreter. The instructions can be translated at compile time, thus exempting the need for runtime decoding.

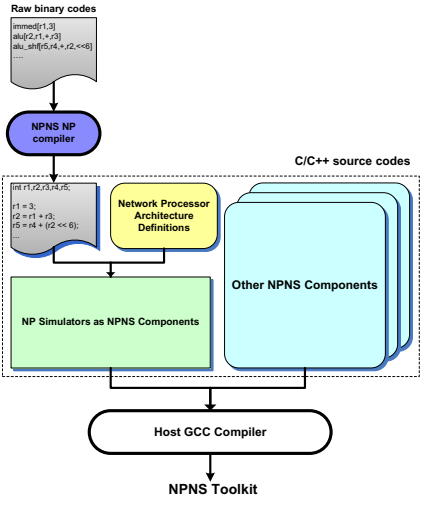


Fig. 6. Building NPNS toolkit with NP simulators implemented as NPNS components

Fig. 6 demonstrates building the NPNS toolkit with NP simulators implemented as NPNS components. In this case, NP simulators still exploit unmodified binaries as input. But, they have been pre-compiled using the NPNS NP compiler, which translates NP instructions to C codes. The outputted C codes are fed directly into network processor architecture definitions, which depict hardware details and are implemented in C++ as NPNS components. Together, they are compiled with GCC to generate the final executable at construction time of NPNS. Once the application running on the NP simulators has been changed, we only recompile it with the NPNS NP compiler and rebuild the toolkit. NPNS components including the definitions of the network processor architecture need not to be revised. In the prototype of NPNS, the NPNS NP compiler supports StrongArm instructions and partial microengine instructions of Intel IXP1200.The StrongArm architecture has also been implemented as NPNS components.

4 A Simulation Example Using NPNS

In this section, we will present a simulation example using the prototype of NPNS. The target system is an edge router based on Intel IXP1200 network processors. Aside from the basic packet forwarding, the major application running on IXP1200 is a packet classifier with 128 rules [9]. NPNS is built on top of ns-2 version 2.1b9a. To simulate IXP1200, we employ two NP simulators in NPNS. To be specific, the StrongArm core of IXP1200 is simulated with a simulator implemented as an NPNS component and IXP1200s microengines are simulated using the aforementioned Workbench 2.01. The simulated edge router supports Fast Ethernet connections and configures IXP1200s core speed at 200 MHz. To illustrate the impact of network context, we create the following topology shown in Fig. 7:

- The edge router connects 2 subnets. One has 20 terminals (with IP address 166.111.*.*) and another has 60 terminals (with IP address 162.105.*.*).
- Terminals run different network services accessing the same sink network. Service name or network protocols are marked on the graph together with the number of terminals on which they run.

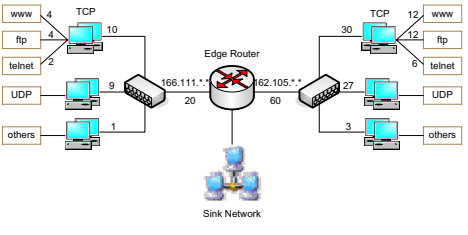


Fig. 7. Sample network topology and services

In our simulation, we investigate the performance of IXP1200 with 20000 packets and different classifier rule hit rate (20%, 60% and 90%). The processing statistics of the classifier on IXP1200 gathered by NPNS are given in Fig. 8, including throughput and SDRAM access count. When we change the hardware configuration of IXP1200, for instance, using four hardware threads instead of one thread in one microengine, per port throughput changes dramatically due to resource contentions (left and center chart in Fig. 8). Moreover, network parameters like packet size also have strong impacts on the overall performance. Some internal metrics (e.g. SDRAM access) which are important for locating processors bottlenecks are still greatly affected by network topology and flow patterns. Hence, our example again demonstrated that simulating network processors considering network context is essential for improving the accuracy of network models and leading to more realistic network simulations.

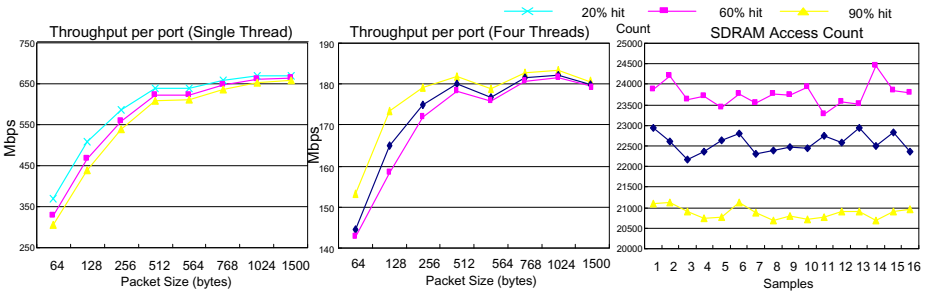


Fig. 8. Statistics of the packet classifier on IXP1200 by NPNS

5 Related Work

Traditional simulation technologies could not offer accurate results for network processor simulations as they put emphasis on either network or processor standalone. Some improvements have been made recently on current network or processor simulators.

On network simulator side, some processing estimators are introduced to enhance the ability to specify processing cost on network nodes. Most recently, NPEST [10] has been developed to estimate application processing cost on network processors. NPEST provides a programming interface and allows users to develop applications using NPEST API in general C code, but currently it can not be integrated into a network simulator like ns-2. Wang et al. [11] have developed a new simulator that combines ns-2 and another event-driven OS/server simulator logsim [12] to simulate network and OS/server bottlenecks. However, it concentrates on the overall behavior not system details.

On network processor simulator side, NP-click [13], can return application specific metrics, but not processor specific metrics. Related work has also been conducted using light weight network processor simulators to improve system level performance given specific hardware platforms and software tasks [14], whereas their workloads are derived from abstract models but not from a complete network environment.

6 Conclusions

There are several interesting issues and open problems that could be further investigated. One is to provide better support on interfacing NP simulators and make communications between the two different types of simulators more efficient. Another line of progression is to further improve the simulation speed by porting NPNS to a parallel environment, especially when mixing cycle-accurate NP simulators with large-scale networks. We believe considering network context will help both processor designers and system builders to improve accuracy of network processor simulations.

References

1. T. Austin, E. Larson, D. Ernst: SimpleScalar: an Infrastructure for Computer System Modeling. IEEE Computer, Vol. 35. no. 2 (2002) 59–67
2. M. Rosenblum et al.: Using the SimOS Machine Simulator to Study Complex Computer Systems. Modeling and Comp. Sim., vol. 7, no. 1 (1997) 78–103
3. LBNL, Xerox PARC, UCB, and USC/ISI, http://www.isi.edu/nsnam/ns/. The Network Simulator - ns-2.
4. Deborah Estrin et al: Network Visualization with Nam, the VINT Network Animator. IEEE Computer, Vol. 33, No. 11, (2000) 63–68
5. ARM Ltd. ARM Development Suite 1.2, 2003.
6. MIPS Technologies Inc. MIPS Free GNU Toolkit, 2003.
7. Peter S. Magnusson et al: Simics: A Full System Simulation Platform. IEEE Computer, Vol. 35, No. 2 (2002) 50–58
8. Intel Corporation. IXP1200 Network Processor Datasheet, 2003.
9. Liqin Tian, Chuang Lin and Zhangxi Tan: A fast packet classification algorithm based on classifier's characteristic applying to multi-fields. Proc. of Internal Conference on Communication Technology, Beijing China, vol. 1 (2003) 255–258
10. R. Ramaswamy, N. Weng, T. Wolf: Improving network simulation: Considering processing cost in network simulations. Proc. of the ACM SIGCOMM workshop on Models, methods and tools for reproducible network research, Karlsruhe, Germany, (2003) 47–56
11. L. Wang, V. Pai, and L. Peterson: The effectiveness of request redirection on CDN robustness. Proc. of the Fifth Symposium on Operating Systems Design and Implementation, Boston, MA, USA (2002)
12. V. S. Pai, M. Aron, G. Banga, M. Svendsen, P. Druschel, W. Zwaenepoel, and E. Nahum: Locality-aware request distribution in cluster-based network servers. Proc. of the Eight International Conference on Architectural Support for Programming Languages and Operating Systems, San jose, CA, USA (1998) 205–216
13. N. Shah, W. Plishker, K. Ravindran, K. Keutzer: NP-Click: A Productive Software Development Approach for Network Processors. IEEE Micro, Vol. 24 No. 5 (2004) 45–54
14. W. Xu and L. Peterson: Support for Software Performance Tuning on Network Processors. IEEE Network Vol. 17, No. 4, (2003) 40–45

On the Placement of Active Monitor in IP Network

Xianghui Liu, Jianping Yin, Zhiping Cai, and Shaohe Lv

School of Computer Science, National University of Defense Technology,
Changsha City, Hunan Province, 410073, PRC
LiuXH@tom.com

Abstract. There is increasing interest in concurrent active measurement at multiple locations within an IP network. In this paper, we consider the problem of where to place monitors within the network and how to make measurement strategy of each monitor which edges in its measurement tree are measured by the monitor. To address the tradeoff between measurement cost and measurement coverage, we consider several optimization problems on the placement of network monitor. We show that all of the defined problems are NP-hard and propose approximation algorithm to these problem.

1 Introduction

As Internet growing rapidly, more and more network applications need to know the network traffic information to monitor network utilization and performance. Knowledge of traffic is critical for numerous important network management tasks, including usage-based accounting, traffic engineering, and attack/intrusion detection. Yet the importance of traffic measurement capabilities is compounded by the fact that IP networks do not maintain per-flow state. By contrast, in circuit-switched networks, the traffic information is essentially "observable for free", because per-call state exists along each node on the call's path. In a sense, the scalability of the stateless IP networks has been bought at the expense of observables, especially in large scale [1] [2] [3] [4].

Recently, many measurement tools have been developed to monitor network traffic information. In general, conventional measurement schemes to measure network flow are classified into two types:

1. Passive measurement is usually monitor network traffic information at some special position, which not imposes extra traffic or modify packet. This means that this method will not affect network traffic. But we need extra schemes to gather and harmonize data from the distributed measurement elements [5] [6] [7].

2. Active measurement monitors network traffic by sending probe packets. This method uses the probe-packet stream to determine the network traffic indirectly. This means that we implicitly assume that performance of networks is the same as the values measured from active probe packets. Up to the present these active measurement schemes will result in significant volumes of additional network traffic. The overhead that schemes impose on the underlying router can be significant and can adversely impact the router's throughput when used extensively [8] [9].

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 1181 – 1187, 2005.

By active measurement method, as we known, when placing a measurement tool at a certain node would only guarantee measurements along the edges of a measurement tree rooted at that node. Thus, to monitor all active paths in the network, the measurement tools should be placed at a set of network nodes such that the measurement trees rooted at these nodes cover all edges of the network[9] [10] [11]. To reduce the network overhead caused by network measurement traffic, one needs to find minimum cost of deploying and maintaining network nodes such that their measurement trees cover all network edges. The problem has several interesting variations. In this paper, we consider the problem of placement of network monitor under active measurement scheme. In practice, when an active monitor is placed, it may send packets carried by the links along the edges of its measurement tree depending on some specific configuration (such as routing configuration). In order to observe all/some active paths in the network, we need to deploy lots active monitor concurrently and gather and harmonize data from any single measurement point. Placing monitor into routers results in some deployment cost and maintenance cost. Deployment cost includes fixed cost components such as the monitor's hardware and software cost, and maintenance cost include the dynamic monitor's operation cost such as sending packets. Therefore, we focus efficient placement of network monitor for active measurement on minimizing the total measurement cost as less as possible. The measurement cost is defined as the sum of deployment cost and maintenance cost. When we choose measurement cost as optimization target and can form several optimization problems on the placement of network monitor. We show all of the problems are NP-hard. Therefore, we propose algorithm and analysis the algorithm approximation ratio.

The remainder of this paper is structured as follows. In the next section, we define a graph-based model of the measurement problem, the key ideas, and the performance metrics which include deployment cost, maintenance cost, and measurement target. Then we formulate various measurement problems along with their complexity analysis. At last we conclude with a summary of our results and a discussion of future work.

2 Problem Description

We represent the IP network of our measurement domain as an undirected graph $G(V,E)$, where V denotes the set of nodes and $E \subseteq V \times V$ denotes the set links. We refer to the routing reachable tree for $v(v \in V)$ as measurement tree for v and denote it by T_v. Note that tree T_v defines the routing paths from node v to some other nodes in V. Without losing generality, we limit the maximum hop count for active measurement as $H_{\max}$. So for any node $u \in V(T_v)(u \neq v)$, the length of this path (u,v) denoted as $l(u,v)$ satisfies that $l(u,v) \leq d_{\max}$.

The solution to an optimal placement of network monitor consists of two parts: (1) a set of nodes $S \subset V$ on which a monitor been placed, (2) measurement strategy of each monitor at the relative node indicates which edges in its measurement tree are measured by the monitor. And we should consider the tradeoff of measurement cost

and satisfy the measurement target. And we are interested in the following measurement target:

(All-Edges-Cover-Problem): given a graph $G(V,E)$, select a minimum set of nodes $S \subset V$ and for each node such that the union of selected measurement edges in its measurement trees covers all edges of *G*.

We define the monitor deployment and Maintenance costs respectively.

Deployment cost: The deployment cost indicates the cost of deploying a measurement device named monitor. We use d_i to denote the deployment cost of a monitor at node *i*. Hence, the total deployment cost is $C_D = \sum_{i \in V} d_i y_i$, where the binary y_i indicates whether a monitor is deployed at node $i(i \in V)$.

Maintenance cost: The maintenance cost is decided by edges which are measured by the monitor. We use c_i to denote the unit Maintenance cost of monitor *i*. This could represent the cost of sending a single packet at monitor *i*. The values of c_i at each monitor can differ, e.g., because of monitor speed. We also use binary x_{ij} to denote the link *j* measured by a monitor *i* and t_j to denote the traffic demand for measuring link *j* respectively. The total Maintenance cost is:

$$C_M = \sum_{i \in V} y_i c_i \sum_{j \in E} x_{ij} t_j$$

The problem efficient placement of network monitor can be expressed as an integer programming formulation. Below we formulate various measurement problems in various situations along with their complexity analysis.

3 The All Edges Cover Problem Without Maintenance Cost

In this section we provide a heuristic for the all edges cover problem without maintenance cost, and point out that our heuristic is the best possible polynomial time approximation algorithm for the problem.

For each node *v* of graph $G(V,E)$ we construct a set U_v of unavoidable by *v* edges in *G*. It is easy to see that for a given *v*, the set U_v can be obtained in time $O(|E|)$. Consider now an instance of the weighted set cover problem $(E,\{U_v : v \in V\})$, where E is the universe of elements and $\{U_v : v \in V\}$ is the collection of subsets. We have the following conclusion.

A set $S(S \subset V)$ is an optimal solution to the all edges cover problem without maintenance cost on a graph $G(V,E)$ if and only if $\{U_v : v \in S\}$ is an optimal solution to the corresponding weighted set cover problem.

The well-known greedy heuristic for the set cover problem translates into a greedy heuristic for the Problem. According to [12], the greedy algorithm is a $(\ln D + 1)$ -

approximation algorithm for the set cover problem, where D is the size of the biggest subset. Since in our case, for any $v \in V$, U_v cannot contain more edges than a measurement tree rooted at v has, we have the following result.

The Greedy algorithm computes a $\left(\ln|V|+1\right)$ -approximation for the All Edges Cover Problem without Maintenance Cost and the worst-case time complexity of the Greedy algorithm can be shown to be $O\left(|V||E|\right)$.

4 The All Edges Cover Problem with Maintenance Cost

In this section we consider the all edges cover problem with maintenance cost. It can be shown that this problem is NP-hard by directly mapping this problem to the well-known uncapacitated facility location problem (FLP). The uncapacitated FLP is defined as follows. We are given a set of locations $N=\{1,2,\cdots,n\}$ with the distances between them denoted as $c_{ij}\left(i,j=1,2,\cdots,n\right)$. We may open a facility at potential facility locations $F \subseteq N$ with building a facility at location $i \in F$ has an associated non-negative cost f_i. We also have a set of demand points that must be assigned to an open facility, denoted as $D \subseteq N$; for each demand point $j \in D$, we have a positive integral demand d_j that must be shipped to its assigned location. The cost of assigning location i to an open facility at j is c_{ij} per unit of demand shipped. We assume that these costs are non-negative, symmetric, and satisfy the triangle inequality. The objective is to determine the set of locations to open facilities and an assignment of demand to the opened facilities, in order to minimize the total cost that is the sum of facility opening cost and the total shipping cost.

We can map the all edges cover problem with maintenance cost to the uncapacitated facility location problem in the following way. Let $N = V \bigcup E$, where N is a set of locations in FLP. Since monitor can be deployed only on nodes and we only measure links, $F = V$ $D = E$ where F and D are subsets of locations in FLP. Although the original FLP problem definition requires symmetry and the triangle inequality properties, these are not of concern to us because F and D are disjoint in our special case. The deployment cost of a facility f_i is defined as the deployment cost of monitor at node i; the demand d_j is defined as t_j to denote the traffic demand for measuring link j. The distance c_{ij} is defined as follows. If link j is measured by monitor on node i, then c_{ij} is defined as unit maintenance cost of monitor i, otherwise $c_{ij}=1$.

This problem can be formulated as the following integer program. We want to find an assignment to the variables y_i and x_{ij} , such that the objective functions are minimized:

$$\min \sum_{i \in V} d_i y_i + \sum_{i \in V} y_i c_i \sum_{j \in E} x_{ij} t_j \qquad \text{(obj)}$$

$$\text{subject to: } x_{ij} \le y_i \left(i \in V, j \in T_i\right) \tag{1}$$

$$\sum_{i \in V} x_{ij} = 1 \left(j \in E\right) \tag{2}$$

$$x_{ij} \in \{0,1\} \left(i \in V, j \in T_i\right) \tag{3}$$

$$y_i \in \{0,1\} \left(i \in V\right) \tag{4}$$

Constraint 2 makes sure that each link is measured by exactly one monitor. Constraint 1 makes sure if link $j\left(j \in T_i\right)$ is measured by node *i* then node *i* must be a monitor.

D. Shmoys et al. [13] proposed a polynomial-time approximation algorithm that finds a solution within a factor of $1+2/e$ of the optimal, where $1+2/e \approx 1.736$. The approximation solution is obtained by rounding an optimal fractional solution to a linear programming relaxation.

5 Budget-Constrained Problem

In some case, we only observe some not all active paths in the network and are interested in the tradeoff between measuring cost and measuring coverage. We present this problem called Budget-Constrained with Some Edges Cover Problem.

Formally, the problem goal is to find a set of nodes $S \subset V$ to place monitor in and measurement strategy of each monitor at the relative node indicates which edges in its measurement tree are measured by the monitor such that the number of edges be measured is maximum, with the measurement cost being subjected to the constraint that $\left(\sum_{i \in V} d_i y_i + \sum_{i \in V} y_i c_i \sum_{j \in E} x_{ij} t_j\right) \le B$ for some budget B.

We next present an IP formulation for the problem.

$$\max \sum_{i \in V} \sum_{j \in E} x_{ij} \tag{obj}$$

$$x_{ij} \le y_i \left(i \in V, j \in T_i\right) \tag{5}$$

$$\left(\sum_{i \in S} d_i y_i + \sum_{i \in S} y_i c_i \sum_{j \in E} x_{ij} t_j\right) \le B \tag{6}$$

$$x_{ij} \in \{0,1\} \left(i \in V, j \in T_i\right) \tag{7}$$

$$y_i \in \{0,1\} \left(i \in V\right) \tag{8}$$

This is a covering integer program defined by Stavros G. Kolliopoulos and Neal E. Young [14] and give an algorithm that produces an $\left(O\left(1+\log\left(m\right)/W\right),1+\varepsilon\right)$ - approximate solution w.r.t. the standard LP optimum for any $\varepsilon \geq 0$: (The constant in the order notation depends on $1/\varepsilon$) where m is the maximum number of constraints any variable appears in.

Another Budget-Constrained Problem is Deployment Cost Budget-Constrained with Minimum Maintenance Cost Problem. The problem goal is to find a set of nodes $S \subset V$ to place monitor in and a mapping $\varphi : E \to S$ such that $\sum_{i \in V} y_i c_{\varphi(i)} \sum_{j \in E} x_{ij} t_j$ is minimum, with the deployment cost being subjected to the constraint that $\sum_{i \in V} d_i y_i \leq B$ for some budget B. It is not hard to see the k-median problem is a special case of this problem, and the IP formulation for the problem is:

$$\min \sum_{i \in V} y_i c_{\varphi(i)} \sum_{j \in E} x_{ij} t_j \tag{obj}$$

$$\text{Subject to: } x_{ij} \leq y_i \left(i \in V, j \in T_i\right) \tag{9}$$

$$\sum_{i \in V} d_i y_i \leq B \tag{10}$$

$$x_{ij} \in \{0,1\}\left(i \in V, j \in T_i\right) \tag{11}$$

$$\sum_{i \in V} d_i y_i \leq B \tag{12}$$

The basic idea behind this problem approximation algorithm of Jain and Vazirani for the k-median problem is to change the objective function to $\sum_{i \in V} y_i c_{\varphi(i)} \sum_{j \in E} x_{ij} t_j + \lambda \sum_{i \in V} d_i y_i$ and removing the constraint (2) from the IP. The resulting IP is then the well known uncapacitated facility location problem [12]. Further note that a large value of λ ¸ forces lesser number of facilities to be opened while a smaller value leads to larger number of facilities to be opened. The algorithm essentially does a binary search on λ to get the 'best' possible k-median solution. For more details see [15].

6 Conclusion

In this paper, we have presented near-optimal monitor placement and measurement strategies in a distributed measurement system. Each solution determines (1) a set of nodes in which a monitor been placed, (2) measurement strategy of each monitor at the relative node indicates which edges in its measurement tree are measured by the monitor. More specifically, we first introduced novel measurement cost and the

measurement target models for a distributed passive measurement system, which can accommodate all situations. Based on these models, we formulated a set of placement problems assuming different constraints. We also showed that various placement problems are NP-hard. We proposed approximation algorithms to determine placement locations.

Our research in the future will focus on evaluating our algorithm on more network topologies and considering the case of route changes caused by link failures. In addition, we will investigate whether a tighter bound of the approximation ratio of the solution can be found.

References

[1] P. Ferguson and G. Houston.: Quality of Service: Delivering QoS on the Internet and in Corporate Networks. John Wiley & Sons, (1998)

[2] Z.Wang.: Internet QoS: Architectures and Mechanisms for Quality of Service. Morgan Kaufmann, (2001)

[3] C.S. Chang.:Performance Guarantees in Communication Networks. Springer-Verlag, IEEE, New York, (2000)

[4] I. Stoica, H. Zhang.: Providing guaranteed services without per flow management. In: Proceeding of SIGCOMM Symposium on Communications Architectures and Protocols, ACM, Boston, MA, (1999)

[5] Y. Breitbart, C. Chan, and et al.: Efficiently monitoring bandwidth and latency in IP networks. In: Proceeding of. IEEE INFOCOM 2001, IEEE, New York, (2001)

[6] Y. Bejerano and R. Rastogi.: Robust monitoring of link delays and faults in IP networks. In: Proceeding IEEE INFOCOM 2003, IEEE, New York, (2003)

[7] C. Fraleigh, C. Diot, B. Lyles, S. Moon, P. Owezarski, K. Papagiannaki, and F. Tobagi.: Design and deployment of a passive monitoring infrastructure. In: Proceeding of Passive and active measurement workshop. IEEE, New York, (2001)

[8] M. Sharma, G. Iannaccone, and S. Bhattacharrya.: On the placement of monitoring devices in an IP network. Sprint ATL Research Report RR03-ATL-112424, (2004)

[9] INMON Corp.: sFlow accuracy and billing. http://www.inmon.com/pdf/sFlow Billing. pdf, (2001)

[10] J. Horton and A. Ortiz.: On the number of distributed measurement points for network tomography. In: Proceeding of ACM Internet measurement conference, (2003).

[11] Liu XH, Yin JP, Tang LL, Zhao JM.: Analysis of efficient monitoring method for the network flow. In: Journal of Software, Volume14, No2, (2003), 300~304.

[12] Chvátal, V.: A greedy heuristic for the set covering problem. In: Journal of Math. Operation Research, Vol4, (1979), 233-235.

[13] F. Chudak and D. Shmoys.: Improved approximation algorithms for the unpacacitated facility location problem. In: ACM SIAM Journal on computing, No1, (2003).

[14] Stavros G. Kolliopoulos, Neal E. Young.: Tight Approximation Results for General Covering Integer Programs. In: Proceeding of IEEE Symposium on Foundations of Computer Science, IEEE, New York, (2001), 522-528.

[15] K. Jain and V. Vazirani.: Approximation Algorithms for Metric Facility Location and k-Median Problems Using the Primal-Dual Schema and Lagrange Relaxation. In: Journal of ACM, Vol. 48 (2001) 274-296.

An Adaptive Edge Marking Based Hierarchical IP Traceback System*

Yinan Jing, Jingtao Li, and Gendu Zhang

School of Information Science & Engineering, Fudan University,
Shanghai, China 200433
{jingyn, lijt, gdzhang}@fudan.edu.cn

Abstract. IP traceback is one of the most effective techniques to defeat the denial-of-service attacks and distributed denial-of-service attacks. And in terms of previous research fruits, the technique based on probabilistic packet marking (PPM) has been proven that it has more advantages than other IP traceback techniques. In this paper, we present a hierarchical IP traceback system, which is more practical and can be implemented and deployed more conveniently and securely than previous end-host schemes. We also present an improved edge marking algorithm called adaptive edge marking scheme (AEMS), which not only can shorten the convergence time, but also be more stable and robust. And detailed theoretical analysis and simulation results have also been presented to show the advantage and efficiency of this scheme.

1 Introduction

Denial-of-service (DoS) attacks can deny or degrade services to legitimate users by over-consuming the resources of a victim host or network. And distributed denial-of-service (DDoS) attacks can cause more significant damage than DoS attacks by leveraging a group of zombie hosts. Moreover such attacks can be easily launched, because the hacker tools are readily available in the Internet. And they are very difficult to prevent, because spoofed IP source addresses are usually used to disguise the attackers' location [1]. So DoS and DDoS attacks have become one of the major threats to the Internet today.

Unfortunately, existing traditional countermeasures, such as firewalls and intrusion detection systems (IDS), only can mitigate the impact by detecting and tolerating these attacks as they occur, but can not eliminate the problem thoroughly.

IP traceback is to identify the sources of DoS or DDoS attacks in the presence of IP spoofing. Figure 1 depicts the system model of research on IP traceback. In this figure, V represents either a victim host or the border device of a victim network. A_i represents an attacker or a zombie host. The dashed line with arrow represents that attack packets traverse from upstream node to downstream node. The attack path is the ordered list of

* Supported by the National Natural Science Foundation of China under Grant No. 60373021.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 1188 – 1197, 2005.

routers from A_i to V. And the attack graph is composed of all attack paths. The IP traceback problem is to determine attack paths to reconstruct the attack graph and identify the attack origin.

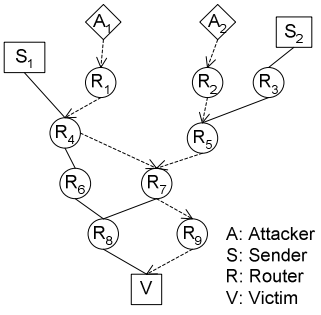


Fig. 1. System model of IP traceback

However, it is very difficult to identify attackers' true location only by IP traceback, because some subtle attackers often use a fair number of stepping-stones to conceal their true location. Although IP traceback is not a complete solution, it can identify the nearest routers to attackers. Then some countermeasures such as filters can be deployed in good season. And an efficient IP traceback system can hold attackers in awe. So nevertheless IP traceback is an effective countermeasure to defeat DoS and DDoS attacks by far.

In this paper, we make two contributions. First, we propose a more practical hierarchical IP traceback system, which can be implemented and deployed more conveniently and securely. Second, we present an adaptive edge marking scheme for IP traceback, which is more stable and robust.

The rest of this paper is organized as follows. In section 2, we describe related work on IP traceback. Section 3 presents the architecture of the hierarchical IP traceback system and the adaptive edge marking scheme for IP traceback. Theoretical analysis of the algorithm is provided in section 4. And section 5 provides simulation results. In Section 6, we will discuss how to implement making authentication in our traceback system. Finally, we make a conclusion for this paper in section 7.

2 Related Work

Research on IP traceback has been rather active since DDoS attacks came forth in the late 1999. A number of approaches have been proposed for IP traceback.

Ferguson et al. proposed ingress filtering which blocks packets with illegitimate source addresses [3]. Burch et al. proposed a link-testing technique called controlled flooding [4]. Unfortunately this approach itself is a DoS attack. So it is unpractical.

Sager [5] and Stone [6] proposed log-based schemes, however which consume enormous resources of routers. Snoeren et al. [7] improved the log-based scheme, but it still has a huge overhead for the router to compute hash values for every packet.

Bellovin [8] proposed a new ICMP-based scheme (iTrace) for IP Traceback. And an intention-driven iTrace was also introduced to reduce unnecessary iTrace messages and thus improve the performance of iTrace system [9].

Probabilistic packet marking (PPM) was firstly proposed by Savage et al. [10]. In this approach, routers probabilistically mark packets with partial path information before forwarding. The victim then reconstructs the complete graph after receiving a modest number of marking packets. By comparison on management cost, additional network and router load, and the ability to trace multiple simultaneous attacks, PPM has more advantages than previous approaches [2][10].

In terms of the difference of marking information, PPM has two classes: node sampling algorithm and edge sampling algorithm. Because the node sampling algorithm could not trace DDoS attacks, the edge sampling algorithm is more prevalent now. And it includes two typical schemes: fragment marking scheme (FMS) proposed by Savage et al. [10] and advanced marking scheme (AMS) proposed by Song and Perrig [11].

Recent research has suggested that less than 0.25% of packets are fragmented [12] and modern network stacks implement automatic MTU discovery to prevent fragment, so the rarely-used 16 bit identification field in IP header is usually used to encode the marking information. However, it is clear that the 16-bit free space is not enough to encode one node (IP address of one router, 32 bit) or one edge (IP addresses of two neighborhood routers, 64 bit).

To solve this problem, FMS subdivides each edge into some number of fragments and marks them into separate packets. But this approach has a very high computation overhead for the victim to reconstruct the attack path and gives a large number of false positives when under multiple simultaneous attacks [11].

AMS is more communication and computation efficient than FMS. Based on the assumption that the victim knows the map of its upstream network, AMS can reconstruct the attack graph by only encoding the hash value of one edge instead of encoding the full IP addresses in FMS.

Although PPM has more advantages than other IP traceback approaches, however, existing PPM schemes have two major shortcomings. First, existing PPM schemes are end-host schemes. That is evidence collection and reconstruction of attack paths are accomplished by victim itself. End-host scheme not only increase the computation and storage overhead for the victim, but also make deployment of the traceback system more complex. For instance, in AMS each potential victim must save a copy of upstream network map and update it periodically to ensure the accuracy of AMS. The hierarchical traceback system proposed in this paper will solve this problem.

Second, the previous PPM schemes use a fixed marking probability in every router. Thus there is a greater likelihood that the marked packets will be overwritten by downstream routers. So much more time is required to reconstruct an attack path. In this paper, we will propose a scheme using adaptive marking probability to reduce time. From theoretical analysis and simulation results given in the following sections, we can believe the new scheme is more stable and robust.

3 Hierarchical IP Traceback System

Hierarchical IP traceback includes three processes. First, the packets are sampled and marked by traceback-enabled routers. Second, the victim receives and collects those

marking information in packets as evidence for attacks and then dispatches a traceback request to the traceback system. Finally, the traceback system reconstructs the attack graph and takes some countermeasures for the victim.

3.1 Architecture

The processes of packet marking, evidence collection, and attack graph reconstruction are dispersed among three separate components in the Hierarchical IP Traceback System (HITS), which is depicted in Figure 2. Each traceback-enabled router has a Marking Agent (MA) associated with it. The MA can be implemented as a software agent, a plug-and-play interface card, or a separate device connected to the router. MA samples the packets passing through it and encodes the partial attack path information into the packets.

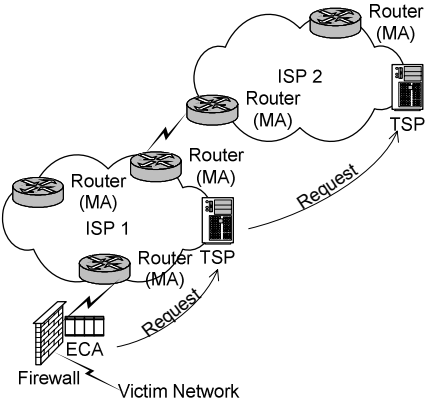


Fig. 2. Hierarchical IP Traceback System (HITS)

MAs are managed by the Traceback Service Provider (TSP). One TSP is responsible for a particular region of the network, such as one ISP network. That is each ISP has its own TSP. TSP can enable or disable the MAs in its own region.

The Evidence Collection Agent (ECA) can be implemented as software or hardware installed into the victim. It is responsible for collecting marking information as evidence for attacks. After collecting a modest number of marking packets, it sends a traceback request with evidence to TSP by the given API, Web Service interface, or simple HTTP/HTTPS protocols. Upon receipt of the request, the TSP cryptographically verifies its authenticity and integrity in order to prevent new DoS attacks. After successful verification, the TSP then performs the traceback operation. And for load balance, one ISP can have more than one TSP.

3.2 Marking Information Encoding

In order to make path reconstruction more convenient, the XOR-based compressed edge algorithm [10] [11] will no longer be used in our scheme. We make the most of the

free space in IP head and encode the head and tail of one edge separately. As above mentioned, packets are rarely to be fragmented in modern network, so both identification field and offset field are rarely used. Hence, we encode the head and tail of one edge into these two fields. Figure 3 shows the encoding policy.

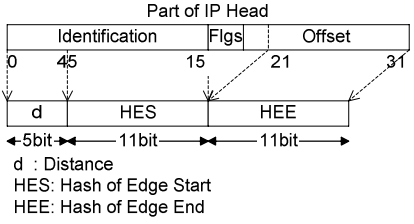


Fig. 3. Marking Information Encoding

One edge can be represented as a triple <d, HES, HEE>. The 5-bit distance (d) field represents the hops from the head of the edge to the victim. 5 bits can be used to represent 32 hops, which are sufficient for almost all paths. HES and HEE fields save the 11-bit hash values of the head and tail of one edge.

3.3 Adaptive Edge Marking Scheme

In order to reduce the probability that the information marked by an upstream router is overwritten by downstream routers, we improve the edge marking algorithm of AMS. Instead of using fixed marking probability, MA will adjust the marking probability according to distance field (d) in the packet. First, we assume that the initial sampling and marking probability (p) of each router is equal. Upon sampling a packet, the MA firstly judges whether this packet has been marked or not. If marked, the MA will use lower probability p/(d+1) to re-mark the packet, i.e. reduce the overwriting probability. This adaptive algorithm is namely the Adaptive Edge Marking Scheme (AEMS) as depicted in Algorithm 1.

Algorithm 1. **AEMS Algorithm**

```
Marking process of MA in R_i:
For each packet P
Let r be a random number from [0,1)
  If ( r < p ) {
       If ( P.d > 0 ) {
          Let r be a random number from [0,1)
          If ( r < p/(P.d+1) ) {
             P.HES = Hash(R_i);
             P.d = 1;
          }
          else{
             if (P.d == 1 )  P.HEE = Hash(R_i);
             P.d = P.d +1;
          }
```

```
        }
        else {
           P.HES = Hash(R_i);
           P.d = 1;
        }
    }
    else {
        if ( P.d == 1 )  P.HEE = Hash(R_i);
        if ( P.d > 0 )  P.d = P.d+1;
    }
```

3.4 Traceback and Attack Graph Reconstruction

After receiving a modest number of marking packets, the ECA can obtain an edge set (*E*). Then the ECA dispatches a traceback request with *E* to the TSP in its ISP's network. After verifying the request, the TSP performs the traceback operation. First, it concatenates the edges in *E* according to the *d* of each edge and then constructs the path set (*P*). Each path in *P* is an ordered list of hash values of routers sorted by distance from the victim. For example, as Figure 1, *E*= {<1, H_9, V>, <2, H_7, H_9>, <3, H_4, H_7>, <4, H_1, H_4>, <3, H_5, H_7>, <4, H_2, H_5>}, and H_i represents the 11-bit hash value of the IP address of R_i. Then the TSP can get *P*. *P*= {<H_9, H_7, H_4, H_1,>, <H_9, H_7, H_5, H_2,>}. Only by hash values in *P* the TSP could not reconstruct attack paths.

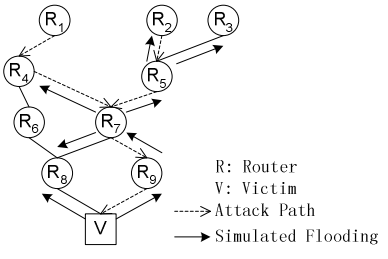


Fig. 4. Simulated Reverse-Path Flooding

Second, the TSP checks every path in *P* using topology information. As we all known, it is difficult for one victim host to get an accurate map of its upstream topology, but it is easier for the TSP to get the topology information of its own ISP network. The TSP simulates reverse-path flooding to check whether paths in *P* match with the topology map. Figure 4 depicts how to use simulated reverse-path flooding to check the path <H_9, H_7, H_5, H_2,> on topology information. If matches completely, the true attack path can be obtained. If matching terminates at the border of this TSP's region, then the TSP will produce a new request to other neighborhood TSPs with remained path information, which will be checked continuously by other TSPs until no match is found.

Finally, after reconstructing the entire attack graph by simulating reverse-path flooding, the TSP can give a response to the ECA. And furthermore the TSP can install or activate some filters in MAs to defeat attacks for the victim.

4 Analysis

In this section, the theoretical analysis about AEMS algorithm will be presented. From the analysis results, we will see that AEMS not only converges more quickly, but also is more stable than AMS. We first give the definition of the convergence time of one marking algorithm.

Definition 1. **Convergence Time.** The convergence time of a marking algorithm is defined as the least number of packets, X, required to reconstruct an attack path.

Let p denote the initial sampling and marking probability of each MA. And let q denote the probability of receiving a marking packet marked by the upstream router d hops away from the victim. If we restrict p to be identical at each MA, then we can get the following equation for AMS:

$$q_{AMS} = p(1-p)^{d-1} \tag{1}$$

According to the Coupon Collector Problem, the expected value of the convergence time of AMS algorithm [11] is:

$$E(X)_{AMS} \approx \frac{\ln(d)}{p(1-p)^{d-1}} \tag{2}$$

And we can further get the following equation for AEMS:

$$q_{AEMS} = p(1-\frac{p}{2})(1-\frac{p}{3})\cdots(1-\frac{p}{d}) = p\prod_{i=2}^{d}(1-\frac{p}{i}) \tag{3}$$

Similarly according to the Coupon Collector Problem, the expected value of the convergence time of AEMS algorithm is:

$$E(X)_{AEMS} \approx \frac{\ln(d)}{p\prod_{i=2}^{d}(1-\frac{p}{i})} \tag{4}$$

Obviously, we can further infer that $q_{AMS}<q_{AEMS}$ and $E(X)_{AMS}>E(X)_{AEMS}$ when $d>1$. That is we can believe that AEMS algorithm converges more quickly than AMS algorithm by theoretical analysis.

Furthermore, we can figure out the numeric curves of $E(X)_{AMS}$ and $E(X)_{AEMS}$ in terms of equation (2) and (4) to compare the stability of these two algorithms. Figure 5 depicts those numeric curves.

Figure 5(a) shows that the convergence time of AMS fluctuates dramatically along with the increase of p. However, AEMS is more stable than AMS. And when $p>0.05$, variation of p has little influence on E(X).

Figure 5(b) depicts that the E(X) of AMS fluctuates dramatically along with the increase of d. However, AEMS is more stable than AMS.

If we let $p=1$ identically, then we get the following equations: $q_{AEMS}=1/d$ and $E(X)_{AEMS}=d\ln(d)$. In this instance, AEMS degrades into the scheme proposed by Tao Peng [14]. We should note that AEMS is not optimal when $p=1$. However, it is not vital

because p has little influence on the performance. The following section will give the simulation results to validate above theoretical analysis conclusions.

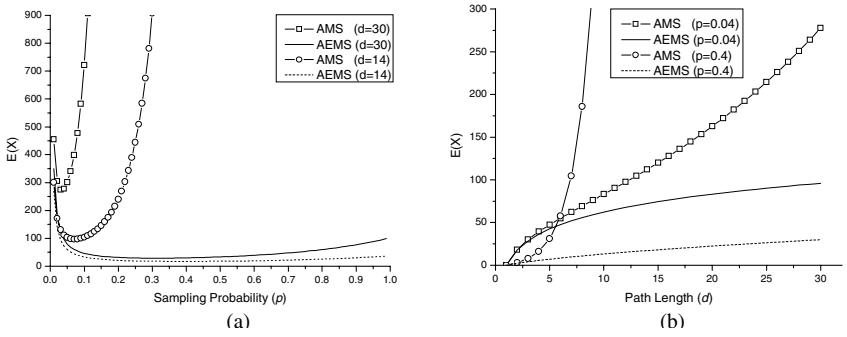


Fig. 5. Comparison between AMS and AEMS on Stability of Algorithm

5 Simulation Results

We have implemented AEMS and AMS algorithms on NS-2 (Version 2.27) [15] to compare the performance of these two algorithms. Before discussing the simulation results, we first define the following concepts as baseline for comparison.

Definition 2. Complete Convergence. After receiving a modest number of marking packets, if we can get all the information of an attack path, we call it is in complete convergence state.

Definition 3. Complete Convergence Rate (CCR). After receiving a certain number of packets and repeating some times of this experiment independently, we call the ratio of the number of experiments arriving at complete convergence state to the total number of experiments as complete convergence rate (CCR). The variation of CCR reflects the convergence speed and the stability of one marking algorithm.

Because the reconstruction of each path is independent, the number of packets needed to reconstruct all paths is a linear function of the number of attackers [10]. Hence, we conducted experiments on simulated attacks using a single attack path from the Skitter dataset on June 29, 2004 [13]. And the path length d is 14.

Figure 6 depicts the variation of complete convergence rate of AEMS and AMS when we let p be different value from 0 to 1, given d=14. In this figure, the value on the vertical axis of every data point represents the CCR when 100 times of independent experiments repeated.

Figure 6(a) shows that the curves live close with each other when we let p be different value in AEMS. That is different values of p have little influence on the convergence speed of AEMS. And on the other hand, Figure 6(b) shows that the convergence speed of AMS algorithm varies dramatically when we let p be different value from 0 to 1. Hence, p could not be arbitrary value in AMS and $E(X)_{AMS}$ is minimized when p=1/d. So we usually set p=1/25= 0.04 in AMS. However, because

different values of p have little influence on AEMS, we can periodically change the initial sampling probability p of each MA to prevent some subtle attackers from faking marking packets by guessing the value of p. Thus, the traceback system using our AEMS algorithm is more secure and robust.

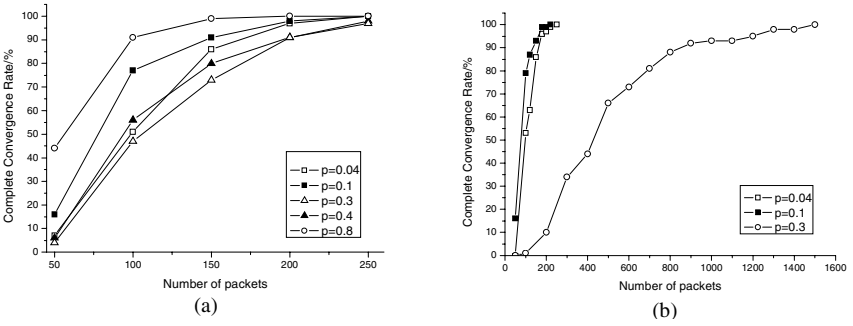


Fig. 6. Variation of Complete Convergence Rate of AEMS and AMS

In addition, in Table 1 we made a comparison of the convergence time of AEMS and AMS when we let p be different value. From Table 1, we can show that the convergence time of AEMS is always not more than 300 packets. However, the convergence time of AMS increases dramatically to 8500 packets when p equals to 0.4. So, we can make a conclusion that AEMS is more stable than AMS.

Table 1. Comparison of Convergence Time of AMS and AEMS

	0.04	0.07	0.3	0.4
AMS	250	250	1500	8500
AEMS	250	250	300	300

The simulation results given above sufficiently validate the theoretical analysis results presented in the previous section.

6 Discussion

Although AEMS is more stable and robust, there is a shortcoming in HITS. That is some powerful attackers can forge marking information to prevent from reconstructing true attack paths, because the packet marking are not authenticated. A simple mechanism to solve this problem is to use Message Authentication Code (MAC) technique. But it is obviously impractical in end-host schemes, because it requires each router to share a secret key with each potential victim.

However, we believe MAC is practical in the HITS, in which it only requires each TSP to share a secret key with each MA. Fortunately, because each MA is managed by

the TSP in its region, therefore it is convenient to share a secret key within them. And moreover, some vicious victims could not tamper the evidence collected by ECA, because they have no secret keys. So the MAC-enhanced HITS is secure enough to prevent from forging packet marking and tampering evidence.

Hence, we believe that it is more convenient to deploy HITS securely and incrementally than end-host schemes.

7 Conclusion

In this paper, at first we have proposed a practical hierarchical IP traceback system, which can be implemented and deployed more conveniently than previous end-host schemes. And then we proposed an adaptive edge marking scheme, which has been proven to be more stable and robust than previous schemes by theoretical analysis and simulation results. Finally, we discussed how to implement packet marking authentication in our traceback system.

References

1. D Moore, G Voelker, S Savage. Inferring Internet denial-of-service activity. The 10th ACM USENIX Security Symposium, Washington D C, 2002-11.
2. Xia Chunhe , Wang Haiquan *et al*. Research on tracing attacks. Journal of Computer Research and Development (in Chinese) , 2003 , 40 (7) : 1021-1027.
3. P. Ferguson and D. Senie. Network ingress filtering: Defeating denial of service attacks which employ ip source address spoofing. RFC 2267, January 1998.
4. Hal Burch and Bill Cheswick. Tracing anonymous packets to their approximate source. Unpublished paper, December 1999.
5. Glenn Sager. Security fun with ocxmon and cflowd. Presentation at the Internet 2 Working Group, November 1998.
6. Robert Stone. Centertrack: An IP overlay network for tracking DoS floods. In Proc. USENIX Security Symposium '00, Aug. 2000.
7. Alex C. Snoeren, Craig Partridge, Luis A. Sanchez, and *et al*. Hash-based IP traceback. In Proceedings of the 2001 ACM SIGCOMM Conference, California, U.S.A., August 2001.
8. Steve Bellovin. The ICMP traceback message. http://www.research.att.com/~smb, 2000.
9. Mankin, D. Massey, C. Wu, S. F. Wu, and L. Zhang. On Design and Evaluation of Intention-Driven ICMP Traceback, In Proceedings of IEEE International Conference on Computer Communications and Networks, 2001.
10. Stefan Savage, David Wetherall, Anna Karlin *et al*. Practical network support for IP traceback1. In Proc. ACM SIGCOMM Conf '00. Stockholm, Sweden, 2000 : 295~306.
11. D X Song , A Perrig. Advanced and authenticated marking schemes for IP traceback. In Proc. IEEE INFOCOM 2001. Alaska, USA, 2001.
12. I Stoica, H Zhang. Providing guaranteed services without per flow management. In Proceedings of the 1999 ACM SIGCOMM Conference, Boston, MA, 1999 : 81~94
13. CAIDA. Skitter. http://www.caida.org/tools/measurement/skitter/index.xml, 2004-07-08/ 2004-12-30.
14. Tao Peng, C. Leckie, R Kotagiri, Adjusted Probabilistic Packet Marking for IP Traceback, In Proceedings of the Second IFIP Networking Conference (Networking 2002). Pisa, Italy, May 2002. 697-708.
15. Network Simulator 2. http://www.isi.edu/nsnam/ns, 2004-07-10.

FAOM: A Novel Active Queue Management with Fuzzy Logic for TCP-Based Interactive Communications

Jin Wu[1,2] and Karim Djemame[3]

[1] School of Computer Science and Engineering, Beihang University,
100083 Beijing, China
jinwu@buaa.edu.cn
[2] Sino-German Joint Software Institute, Beihang University,
100083 Beijing, China
[3] School of Computing, University of Leeds,
LS2 9JT Leeds, United Kingdom

Abstract. Given the fact that the current Internet is getting more difficult in handling those interactive traffics which are rapidly increasing, new techniques are required. In this paper, a new Active Queue Management (AQM) algorithm for interactive communications that uses fuzzy logic is proposed. We analyse the relationship between the end-to-end performance and switch nodes' operation state in packet-switched networks. Based on this analysis, guidelines for performing control actions at gateways are given. A new AQM algorithm, Fuzzy Adaptive Optimized Marking (FAOM), is also introduced. We then compare FAOM with RED (Random Early Detection) by simulation using NS-2 to show that FAOM does improve the end-to-end performance for real-time interactive communications.

Keywords: Fuzzy, Congestion Control, TCP/IP, Active Queue Management, Communication.

1 Introduction

The current Internet has been used in a wider context than before. Beside the file transferring and emailing, many other real-time and interactive applications are also been used in the Internet. Most researches have a strong impression that real-time applications are supplied with the User Datagram Protocol (UDP). However, there are still a significant part real-time and Interactive applications, e.g. Net-game and on-line charting, need reliable transmission that have to be supplied with the Transmission Control Protocol (TCP). The research of improving the performance of TCP-based interactive communication is a useful and valuable topic but has not been adequately studied before. In this paper, we focus on the designing of a novel Active Queue Management (AQM) algorithm that can improve the TCP performance for real-time interactive communication. A potential application for this new AQM algorithm in the industries is to implement it into the access networks. It can be activated when game-players are accessing. Fuzzy logic is a departure from classical Boolean logic in

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 1198 – 1207, 2005.

that it implements soft linguistic variables on a continuous range of truth values which allows intermediate values to be defined between conventional binary. Since fuzzy logic can handle approximate information in a systematic way, it is ideal for controlling nonlinear systems and for modelling complex systems for which an inexact model exists or systems where ambiguity or vagueness is common. RED is one of the most widely researched AQM algorithms. However, RED is not efficient enough to eliminate congestion, especially the queuing delay when the communication network is under heavy load. By using analytical models and fuzzy logic, we introduce in this paper a new AQM algorithm called Fuzzy Adaptive Optimized Marking (FAOM), which employs a totally different design principle against current AQM algorithms. Interactive communications in a congested Internet will benefit from FAOM. It is shown through simulation that the performance of real-time interactive communications significantly improves under a wide range of traffic load when deploying FAOM compared to RED at bottleneck switch nodes.

This paper is organized as follows. Related works are introduced in Section 2. In Section 3, the relation between end-to-end performance and switch-node running state is illustrated. From that, a new AQM algorithm called FAOM is introduced in Section 4, which improves the network performance by controlling the queue length at switch-nodes approaching an ideal length. In Section 5, simulation experiments are designed for comparing FAOM and RED to show that FAOM can indeed improve the performance of real-time interactive communications under network congestion situations. Also, we show using the simulation results that FAOM is capable of providing more stable queuing despite the variation of traffic load. We end this paper with some concluding remarks in Section 6.

2 Related Work

Extensive research has been done on Random Early Detection (RED) since it was introduced by Floyd and Jacobson [3]. There are many arguments on whether RED can improve end-to-end performance. The Internet Engineering Task Force (IETF) recommended its deployment [8], while some researchers showed evidence of opposing its use [4, 13]. Many research papers using mathematical modelling [11, 20, 21] and simulation experiments [19] to evaluate the performance of RED are found in the literature. Other papers treat essentially design guidelines of AQM algorithms [15, 16]. Several RED variations are found in the literature: FRED [17], SRED [12], BRED [18], and ARED [1] are among those that received most attention. FRED, SRED and BRED use per flow queuing algorithms. REM [15, 16, 19] is a new proposed AQM algorithm that received increase attention by the research community. REM uses a (mathematical) duality model to simulate the network congestion control process. REM has two drawbacks that make it hardly fit in current network environments: 1) REM needs to revise the source algorithm when deployed; and 2) REM cannot work along current AQM algorithms such as RED. Model abstraction for communication networks has attracted great interest among the research community. Classic control theory is also used for network modelling [4], [11]. The network congestion control process is converted into a close loop system. This model works well in single switch node system, but not in large-scale networks as there are more model

parameters affecting end-to-end performance. Although works on the application of intelligent control to communication networks is found in e.g. [5, 6], they are more alike pure adaptive algorithms that do not prove the use of knowledge structure supporting machine intelligence. Current AI research in the communication networks mostly focuses on human interaction and the application layer. Although not quite active, there is some research on applying AI in the field of Transmission subnet management. Reference [2] presents an Intelligent Agent Architecture and a Distributed Artificial Intelligent based approach for Network Management (NM) where a NM system based on intelligent agents, claimed to be more elastic than conventional centralized approaches, is proposed. Applying fuzzy logic in the congestion control process is also an area that researchers are looking at. References [10, 23] apply fuzzy logic in the Available Bit rate (ABR) congestion control in ATM networks. Reference [22] uses fuzzy logic for the design of an AQM algorithm and proposes Fuzzy RED.

3 Optimizing End-to-End Performance

The Internet was designed for non real-time communications. The only congestion signal for this kind of communication network is packet loss [5]. This kind of architecture could not meet the developing requests for interactive and real-time applications. A new method needs to be used to evaluate the network performance. As we know, the total delay an end-to-end connection suffers is the time length between the information began to send by source and this information been well received by receiver, which is the sum of transmission time and transmission delay. Transmission time can be measured by the quotient of information size and useful transmission rate (in terms of throughput in some applications). Theoretically, the transmission delay of any network system is composed by two parts, propagation delay and queuing delay. Propagation delay only relates to the properties of data link layer and physical layer, and can be roughly considered as a constant at the level of network layer if the route has been setup. Therefore the variance of the transmission delay is only affected by queuing delay. The queuing delay is the quotient of queue length and transmission rate. Therefore the total delay that suffers the end-to-end connection l of sending an information block can be given as

$$T^l(D) = \frac{D}{Throughput^l} + \sum_{x \subset l} \frac{Queuelength_x}{Capacity_x * Usage_x} + t^l_{propagation} \tag{1}$$

where D is the size of block that transmitted by sender, $Throughput^l$ is the transmission rate that received by the connection l, $Queuelength_x$ is the queuing of the multiplexer at switch-node x, $Capacity_x$ and $Usage_x$ are the output port's capacity and its efficiency of the multiplexer at switch-node x respectively. $t^l_{propagation}$ is a constant that represents for the total propagation delay that connection l suffers. For and switch-node x, $x \subset l$ when connection l passes through x. While for the multiplexer in switch-node x, if the ingress data of the multiplexer is fluctuated, a function G exists for

$$G(Queuelength_x, Usage_x, \eta) = 0$$

where η is the distribution of ingress data. Consider the situation where no massive packets lost takes place. Therefore, the following equation exists.

$$\sum_{l \subset x} Throughput^{l} = Capacity_{x} \times Usage_{x} \tag{2}$$

Then, the T^l can be considered as the performance measurement of end-systems' application in data networks. When T^l is reduced, the receiver is able to obtain information it needs earlier for which benefits the performance of real-time applications. Moreover, lower T^l also benefits interactive communications. Therefore the target of congestion control process is to minimise it. In a communication network, suppose n connections are accessing one bottleneck link. The performance evaluation of the TCP connection L is P_L. As defined above,

$$P_L = E[T^l] \tag{3}$$

The overall performance of all connections that access switch-node x can be represented as:

$$P' = \frac{\sum_{L=x}^{n} P_L}{n} \tag{4}$$

In order to improve performance, the value of P' needs to be minimised. Ideally P' can be reduced after some configuration performed at the network switch node, this is the so-called process of network optimization.
It can be easily shown that,

$$P' = E[\frac{\sum D}{\sum Throughput_{x}}] + E[\frac{Queuelength}{Usage * capacity}] + Const \tag{5}$$

It is known that Queue length affects the switch-node's performance while long queue encourages queuing delay and short queue wastes the bandwidth. Ideal queue length optimising the switch-node's performance, therefore we define the running cost of the switch-node as $J(queue)$.

$$J(queue) = \alpha * \frac{1}{Usage(queue)} + \beta * \frac{queue}{Capacity * Usage(queue)} \tag{6}$$

Then, equation (5) can be converted as:

$$P' = \frac{E[\sum D]}{Capacity} * \frac{1}{Usage(queue)} + \frac{E[queue]}{E[Usage] * Capacity} + Const \tag{7}$$

If defined

$$\frac{\alpha}{\beta} = \frac{E[\sum D]}{Capacity} \tag{8}$$

then,

$$P' = J + Const \tag{9}$$

So, the performance evaluation J is linked with the criterion of the congested switch-node P' through equation (9), which means that the performance of the end-to-end connections can be improved through the configuration done at switch-nodes. The network is running at an optimized state when J is minimized by tuning queue length.

4 Fuzzy Adaptive Optimised Marking

It has been proved in the Queuing Theory that the function $z=J(queue)$ is a concave function. So,

$$\frac{d^2J}{dqueue^2} > 0 \tag{10}$$

Defined a variable $\Delta t > 0$ and a small constant $\varepsilon > 0$. A function is defined as

$$F = J(q'+\Delta t+\varepsilon) - J(q'+\Delta t) \tag{11}$$

From Taylor Expansion,

$$J(q'+\Delta t+\varepsilon) = \frac{dJ}{dqueue}\cdot(\Delta t+\varepsilon) + \frac{d^2J}{dqueue^2}\cdot\frac{(\Delta t+\varepsilon)^2}{2!} + o(|\Delta t+\varepsilon|) \tag{12}$$

$$J(q'+\Delta t) = \frac{dJ}{dqueue}\cdot\Delta t + \frac{d^2J}{dqueue^2}\cdot\frac{\Delta t^2}{2!} + o(|\Delta t|) \tag{13}$$

Then, from (11), (12), and (13)

$$\frac{dF}{d\Delta t} = \varepsilon\cdot\frac{d^2J}{dqueue^2} + o(|\Delta t+\varepsilon|) \approx \varepsilon\cdot\frac{d^2J}{dqueue^2}$$

So based on (10), roughly, a conclusion can be given as follows.

$$\frac{dF}{d\Delta t} > 0 \tag{14}$$

In switch-nodes, Δt represents for the offset between existing queue length and ideal queue length. The value of F can be directly measured in switch-node, so that value of F can be used to depict the value of offset. The Explicit Congestion Notification (ECN) [8] mechanism is used in this research to control the queue length. Naturally, the ECN bit is marked with higher probability to put down the queue length and marked with lower probability to encourage queuing. From above analysis, the ECN marking probability needs to be increased when the value of F is positive and decreased reversely. Since no clear equations can be generated from above analysis, the idea of fuzzy logic is applied to organise the queue management process. Therefore, a linguistical logic table is given below. Defined the ECN marking probability as p.

Table 1. Logical Table in Determine Control Actions

F	NL	NS	0	PS	PL
Δt	NL	NS	0	PS	PL
d*p*/d*t*	NG	NW	IV	PW	PG

NL: Negative Large
NS: Negative Small
0: Zero
PS: Positive Small
PL: Positive Large
NG: Negative Strong
NW: Negative Weak
IV: Invariable
PW: Positive Weak
PG: Positive Strong

Five linguistic values, w_1, w_2, w_3, w_4, and w_5, are defined. As previously studied, the following membership function for the fuzzy controller can be generated accordingly.

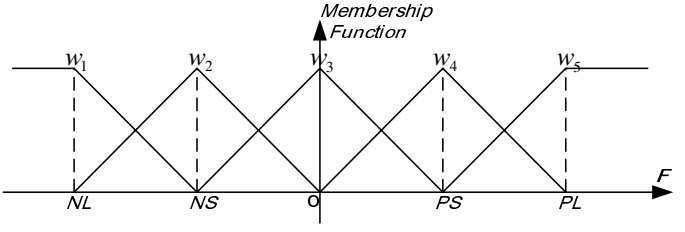


Fig. 1. Membership Function for the Fuzzy Controller

The deffuzzification process is defined as Centre-average. Therefore, the controller output can be given as:

$$Output = p + h_{NL} \cdot M_{w_1} + h_{NS} \cdot M_{w_2} + h_{PS} \cdot M_{w_4} + h_{PL} \cdot M_{w_5}$$

h_{NL}, h_{NS}, h_{PS} and h_{PL} are four predefined step length values respectively stand for "negative large", "negative small", "positive large", and "positive small". Adaptive mechanism can be applied to tune those values. Then, we come up with a new AQM algorithm, Fuzzy Adaptive Optimized Marking (FAOM) based on FIFO which runs on TCP/IP networks. As the traffic on the Internet is highly fluctuated, there is a

if((now – last_update) > update_interval)
{ $J = \frac{\alpha * Capacity_k}{Throughput_k} + \frac{\beta * Queue_k}{Throughput_k}$
$F = J(q'+\Delta t+\varepsilon) - J(q'+\Delta t)$
PL = 0.67*maxF;
PS = 0.33*maxF;
NS = 0.33*minF;
NL = 0.67*minF;
if (NL ≤ F)&(F < NS)
{p = p - kl*t_2*(NS-F)/(NS-NL); }
if (NS ≤ F)&(F < 0)
{p = p - kl*t_2* F/(-NS); }
if (0 ≤ F)&(F < PS)
{p = p + kl*t_1* F/PS; }
if (PS ≤ F)&(F < PL)
{p = p - kl*t_2*(NS-F)/(NS-NL); }
last_update = now;
}

C: Cost at switch-node
p: Marking probability
α: Bandwidth price
β: Delay price
kl, ks: small constant & kl>ks>0
update_interval: Time slot between two p updates
now: current time
last_update: last update time
e and ε: small value constant.
Δt: small time interval.
maxF: maximum value of F.

Fig. 2. FAOM Algorithm

trade-off between link utilization and queue length. It is reasonable to assume that ingress traffic to be a stationary process. Long queues at switch nodes will result in a high link utilisation as well as a large queuing delay. Conversely, a lower number of packets queuing will decrease the queuing delay but harm the link utilization. The idea behind FAOM is to balance this trade-off and optimize the end-to-end performance. Detailed algorithm of FAOM is shown in Figure 2. There are five parameters that need to be set for FAOM, which are: *freeze_time*, *t1*, *t2*, α and β. *freeze_time* is set to control the marking probability p update frequency. FAOM maintains a single probability, p to mark the Explicit Congestion Notification (ECN) bit [7] when the packets are dequeued. As we assumed that all sources are running TCP. *t1* and *t2* determine the amount by which p is increased or decreased. In the decision process, as calculated by the reward structure, α and β are respectively the bandwidth price and delay price associated with a network object.

5 Simulation Experiments

In this section, we compare through simulation FAOM and RED in terms of queue length and link utilization at routers. We also compare the end-to-end performance of passing through an FAOM capable gateway against a RED capable switch-node. This is to prove that the end-to-end performance can indeed be improved by tuning the trade-off between queue occupancy and link utilization. In these experiments, we trace the queue length and link utilization at congested gateway and the Tl for the TCP connection accessing it. The bottleneck gateway is set to different congestion states to prove that FAOM can successfully improve the end-to-end performance for interactive communications under various congestion situations.

5.1 Simulation Environment

The simulation study is based on the network simulator ns-2 [6]. Simulation is performed on the network shown in Figure 3 where an interactive communication shares a bottleneck link with cross traffic. The bottleneck link is set to 15Mb with 10ms' propagation delay. Other links are all set to 20Mb bandwidth with 10ms' propagation

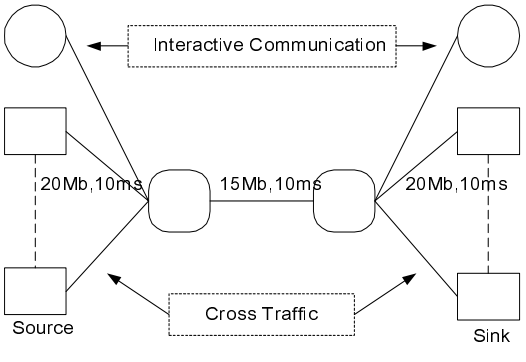


Fig. 3. Network Topology

delay. The cross traffic is modelled using an N sources configuration consisting of N identical TCP sources and sinks. All the sources and sinks are connected to a router with N TCP connections passing through the bottleneck link. All cross traffic sources are supplied by FTP (File Transfer Protocol) applications. The TCP default packet size is 1KB. The buffer capacity at the router is 100 packets. Packets are served in FIFO order and are marked with ECN bit using the probability of the AQM algorithms.

5.2 Simulation Scenario

In this experiment, RED parameters are set as follows: maximum threshold = 60, minimum threshold = 20, maximum probability = 0.1 and w = 0.002. For FAOM, we set $\alpha = 20$, $\beta = 100$, t1 = 0.001 t2 = 0.0001. kl and ks are set to 2 and 0.5. The freeze_time is set to 10ms, D is set to 20 Kb. We trace the queue length and link utilization at gateway as well as the end-to-end throughput and transmission delay. The simulation is run for 110 seconds. All the TCP sources start transmitting at the same time. The first 10 seconds are considered as simulation warm-up time, thus the simulation measurements start at time t = 10 seconds. We set the number of cross traffic connection to 10, 100 and 200 respectively to represent different congestion states.

5.3 Simulation Results and Discussion

The simulation results are shown in Tables 2 and 3. We apply the Relative Percentage Deviation (RPD) over RED to measure FAOM's performance improvement. Table 2 illustrates the average for link utilization and queue length at FAOM and RED capable gateway under different cross traffic states. Table 3 is the average value for the end-to-end performance comparison of passing through FAOM capable network with RED capable network under different cross traffic states at valid running time.

Table 2. RPD Analysis of FAOM over RED at Isolated Gateway

		RED	**FAOM**	**RPD**
Usage	10	0.975	0.96	-1.54%
	100	0.997	0.96	-3.71%
	200	0.997	0.96	-3.71%
Queue Length	10	22.65	10.16	55.14%
	100	53.01	10.27	80.74%
	200	58.34	10.74	82.25%
J	10	21.75	21.39	1.66%
	100	22.75	21.40	5.93%
	200	23.18	21.43	7.55%

Table 3. End-to-end RPD Analysis

Connection	**RED**			**FAOM**			**RPD**
	Delay	Thput	T^l	Delay	Thput	T^l	
10	42.4	1.461	98.49	35.61	1.44	49.50	49.74%
100	58.33	0.1497	250.27	35.67	0.144	174.56	30.25%
200	61.20	0.0748	389.78	35.69	0.072	313.45	19.58%

Table 2 illustrates the fact that although the queue length was under a proper control, the Link Utilization has decreased when deploying FAOM. But we can also observe from Table 3 that there is a significant performance enhancement in terms of T^l when comparing FAOM and RED, which means that FAOM gateway keeps a better trade-off between Queuing and Link Usage than RED, and can provide a more swift delivery for data block D. This result supports our design principle that an interactive time AQM algorithm is needed. It proves that 1) our analysis in section 3 is correct in pragmatic, and 2) fuzzy logic has an alternative way to apply in AQM design other than those in ref. [20, 21, and 22].

6 Conclusions

In this paper, several contributions have been made. 1) We illustrated the relationship between switch-node tuning and end-to-end connections' performance and therefore proposed a new AQM algorithm for interactive applications. 2) We applied fuzzy logic in the AQM algorithm. 3) From the simulation comparison between FAOM and RED, we can observe a significant performance enhancement for real time end-to-end communications when deploying FAOM.

Acknowledgement

The author acknowledges the support from the National Natural Science Foundation of China under key project 90412011 and the support from the National 863 Program of China under grant number 2004AA119030.

References

[1] Jin Wu, Karim Djemame. Simulation Comparison of AOM and RED. In Processdings of SPECTS 2002: 319-26, USA, July 2002.
[2] W. Feng, D, Kandlur, and K. Shin. A self-configuring RED gateway. In Proceedings of INFOCOM' 99: 1320-28, USA, March 1999.
[3] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. IEEE/ACM Trans. on Networking, Vol.1 No.4:397-41, August 1993.
[4] C. V. Hollot, Vishal Misra, Don Towsley and Wei-bo Gong. On designing improving controllers for AQM Routers Supporting TCP Flows. In Proceedings of INFOCOM'01: 1726-34, USA, April, 2001.
[5] R.J.Gibbens, F.P.Kelly. Resource Pricing and the Evolution of congestion control. Automatica, 35, 1999.
[6] "ns-2 Network Simulator," Obertain via http://www.isi.edu/nsnam/ns/
[7] K. Ramakrishnan, S. Floyd, A Proposal to add Explicit Congestion Notification (ECN) to IP, RFC 2481, January, 1999.
[8] B. Braden and etc. "Recommendations on Queue Management and Congestion Avoidance in the Internet", RFC 2309, April 1998.

[9] Jitendra Padhye, Victor Firoiu, Donald Towsley, and Jame Kurose. Modeling TCP Reno Performance: A Simple Model and Its Empirical Validation, IEEE/ACM Transaction on Networking, Vol.8, No.2, April 2000.

[10] T.Bonald, and M.May, Analytic evaluation of RED performance, INFOCOM' 00: 1415-24, Tel Aviv, 2000.

[11] C.V.Hollot, Vishal Misra, Don Towsley and Wei-Bo Gong, A Control Theoretic Analysis of RED. In Proceedings of INFOCOM 2001, USA, 2001

[12] T. J. Ott, T. V. Lakshman, and L. H. Wong. SRED: Stabilized RED. In Proceedings of IEEE INFOCOM, 1999.

[13] M.May, J.Bolot, C.Diot, and B.Lyles. Reasons Not to Deploy RED. In Proc. of IWQoS'99, June 1999.

[14] M.Allman, V.Paxson, W.Stevens. TCP Congestion Control, RFC 2581, April 1999.

[15] Steven H. Low. A Duality Model of TCP and Queue Management Algorithms. under doing, from: http://netlab.caltech.edu

[16] Steven H. Low, David Lapsley. Optimization Flow Conrol. IEEE/ACM Transactions on Networking, Vol.7 No.6 861-75, Dec. 1999.

[17] D. Lin and R. Morris, Dynamics of random early detection. SIGCOMM' 97:127-37, September 1997.

[18] F. Anjum and L. Tassiulas, Fair Bandwidth sharing among adaptive and non-adaptive flows in the Internet. Infocom99:1412-20, USA, March 1999.

[19] Sanjeewa Athuraliya, and Steven Low. Simulation Comparison of RED and REM. ICON00, Singapore 2000.

[20] A. Pitsillides, et. Al., Effective Control of Traffic Flow in ATM Networks Using Fuzzy Explicit Rate Marking, IEEE JSAC, Vol. 15 Issue 2, pp. 209-25, 1997

[21] A. Pitsillides, et. al., Fuzzy Logic Based Congestion Control, COST 257: Impacts of new services on the architecture and network performance of broadband networks, Cyprus, 1999,

[22] Y. A. Sekercioglu, A. Pitsillides. Fuzzy Control of ABR Traffic Flow in ATM LANs, in Proc. of IEEE ISCC'95, pp 227-32, 1995.

[23] R. Loukas, S. Kohler, P. Andreas, T-G Phuoc. Fuzzy RED: Congestion Control for TCP/IP Diff-Serv, in Proc. of IEEE MeleCon2000 Vol.1, pp19-22, 2000.

A CORBA-Based Dynamic Reconfigurable Middleware

Wanjun Huang[1], Xiaohua Fan[2], and Christoph Meinel[1]

[1] Hasso-Plattner-Institut, University of Potsdam,
Postfach 900460, D-14440 Potsdam, Germany
{huang, meinel}@hpi.uni-potsdam.de
[2] Dandong Information Center,
No. 99, 11 Jingjie, Dandong, Liaoning, P.R. China
fan@dandong.net

Abstract. The widespread Internet and mobile applications demand increasing requirements for easy and flexible to reconfigure a deployed system during run time. The middleware proposed to help programmer developing distributed application automatically inherits these demands and requirements. In his paper we present a CORBA based middleware system that adopts our technology of Routing Based Workflow (RBW). RBW has modeled the execution environment of cooperative components. Within RBW component instances are temporally bound to routing for their functionality execution. It is the temporal binding makes the dynamic reconfiguration of software components easy to realize and greatly simplify the hard problems of preserving consistency.

1 Introduction

To decrease the development cycle and alleviate the burden of distributed application developer from tedious non-business programming, middleware is proposed and widely applied for establishing large scaled distributed business applications. Application programmer just gets what he needs from middleware through provided application programming interface (API) or services, and does not care about how these functionalities are implemented. This simple mechanism of black box has brought middleware great success. But as development of Internet and mobile application, middleware is also required to improve to adapt new requirements of distributed applications. When designing and implementing a middleware application, designer always try to complete all functionalities and services in advance. But if all components are loaded to run when server starts, the fat server will cost much unnecessary memory and CPU resources for some seldom used or even never used components. Configurable middleware can customize components to provide specific services according to different circumstances and application areas at start time. But the kind of offline configurability is still not enough to satisfy the increasing requirements. Dynamic reconfiguration is gradually being an indispensable requirement for large scaled system, especially for the system that provides crucial continuous services and mobile services. In some case it is impracticable or will cause big loss if the system shut down or restart. However, faults are nearly unavoidable, so component has to be replaced by its new version. Also, new component will be

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 1208 – 1217, 2005.

requested to integrate into the system. So, if system has capability of dynamic reconfiguration, the loss can be decreased to minimum.

In this paper we propose a CORBA based middleware system that adopts the technology of Routing Based Workflow [2] which provides a flexible mechanism to assemble software components and reaches a high capability for dynamic reconfiguration. In next section we first introduce the motivation of our new approach and the work mechanism of routing based workflow. Then we explain how to construct a CORBA based middleware system using technology of routing based workflow. After that we give a discussion and analysis for our approach. Related works are also given to narrate some related research activities and indicate the differences to our proposal.

2 Motivation

Serials of Architecture Description Language (ADL) approaches give their solutions for configurable distributed system [1]. But all of these ADLs have no or limited dynamism. There are also plenty of proposals directly for dynamic reconfiguration. Most of these approaches for dynamic reconfiguration adapt a strategy of "*waiting until safe state*" [3], [4], [5]. This strategy first hold the new coming request, and wait all concerned components to go into a safe state which means component finishes the processing of task and is kept in a state of waiting. After all involved components go into a safe state, the reconfiguration operations then begin to be performed, and the held requests will resume to be processed after reconfiguration. The strategy of "*waiting until safe state*" works well in normal case, but it may collapse in an extreme case. For example, if the concerned component involves into a long time interaction, the processing of reconfiguration have to wait for long time, and all relevant services have to be stopped for long time.

All of approaches for dynamism try to separate the functionality of component from its structural and management dependency with other components. The maximum independency of component gives the possibility for maximum flexibility of dynamism. Through the analysis of software system we get to know that every software components live in its execution environment. When the execution environment of component is provided, the component can be executed. For multiple cooperated components, there is also a global execution environment. All involved components can run and interact with each other when their global execution environment is available. In our core idea of routing based workflow, global execution environment is modeled as routing. The component instance is just temporal bound to execution environment, namely routing, to execute its functionality. After execution component instance will be unloaded from its routing and keep independence again. The execution environment can be duplicated, modified and replaced, and these changes result in new interactive way of involved components. When reconfiguration operation comes, what need to do is to modify the routing, and the hard issues of consistence maintaining is then naturally simplified to synchronization of routing updating because all the change operations are effected on routing, rather on components.

3 Routing Based Workflow

In this section we describe how to construct the global execution environment, namely routing, and explain how it works for temporal binding of component.

3.1 Overview of Routing Based Workflow

RBW has modeled the software components from structure to running time state. The modeling of cooperated components in different states can be reflected into three tiers: routing schema, bound routing and active routing, shown as in Fig.1. Routing schema describe the structure relation of cooperated components, and specific features and properties of each components. Bound routing is an idle execution environment in which all components are instantiated and all the interfaces of components and their communication path of data are also established and tested. In other words a bound routing is ready to serve for request. The third tier, active routing is the execution environment where the control, management and execution activities of involved cooperating components occur. In fig.1 the framework of routing based workflow depicts briefly the relation and work mechanism among different routing tiers. Workflow Manager is responsible for general management task between routing schemas, bound routings and active routings. *Schema Configurator* not only parses XML based offline configuration to schema object or decode schema object back to offline configuration. The real operations that realize dynamic reconfiguration on software components are implemented and performed by *Routing Configurator*. Because the reconfiguration operations are effected on bound routing, not on routing schema or component itself, the efficiency of reconfiguration is much higher than traditional solutions. For the management of functionality execution, a module of *Execution Engine* is designed to dispatch concrete request to the relevant bound routing, turn the state of routing to be active and guide the execution.

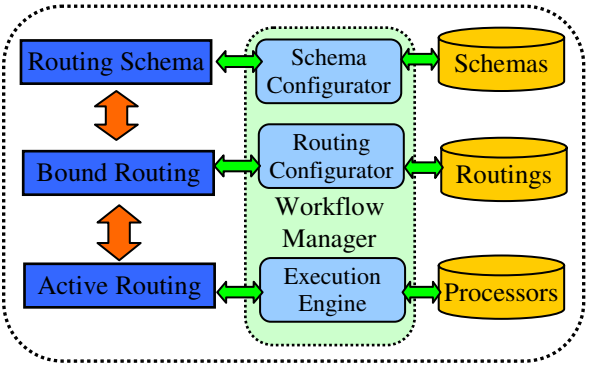


Fig. 1. Three Layers Modeling of Routing Based Workflow

3.2 Routing Composition

Fig. 2. describe the structure of routing based workflow. The key elements are component delegate, component container, component processor and communication ports that take charge of the communication among component delegates and processors.

Component Delegate

Component Delegate encapsulates IO behaviors of component that it delegates, and represents the component in term of all management activities. Only when it is time to execute the functionality of component, the delegate will ask for its component processor to execute. After this execution finishes, delegate will get results and return component processor back to component container. Dependences management is another task of which delegate has to take charge. There are two dependences existed in routing based workflow. One is data-flow dependence has already been realized by binding of communication ports. The other one is control dependence that indicates the control relationship between current component and its neighboring ones. The implementation of control dependence adopts the pattern of *Component Configurator*.

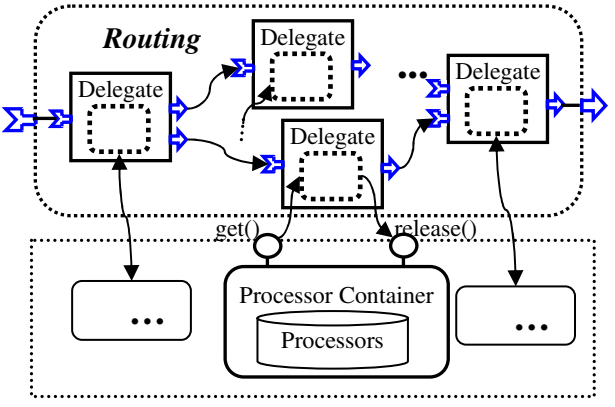


Fig. 2. Structure of Routing Based Workflow

Component Container and Processor

Component Container is used to manage component processor pool and keep contact with routing. *Component processor* is the wrapper for real functional entity that implements the functionality what it provides. Additional to be wrapper of functional entity, *Component Processor* need complete another tasks: triggering the execution of functional entity on arrival of input data.

Communication

Communication ports are designed to communicate among different components. From IO direction communication port can be identified as in port and out port. Communication port is the unique data exchange media among different components. So these ports are used both in component processor and component delegate. For component processor communication ports are fixed after the design and implementation of this component finish. But for component delegate, its ports are detachable, and this is part reason resulting in high flexible dynamic reconfiguration. In a routing all transported data has to be encapsulated into a *Named Object* which contains the information of name, data type, state and value etc.

3.3 Routing Execution

Routing undergoes different states from its instantiation to execution. Once routing is created and instantiated from routing schema, it goes into initial state in which each

component delegate is separated from other delegates. When routing completes the operation of virtual binding, it goes into virtual bound state that means routing is ready to be executed. If a specific request is dispatched to a routing for execution, the routing goes into real bound state. The relation of different states is shown in Fig.3.

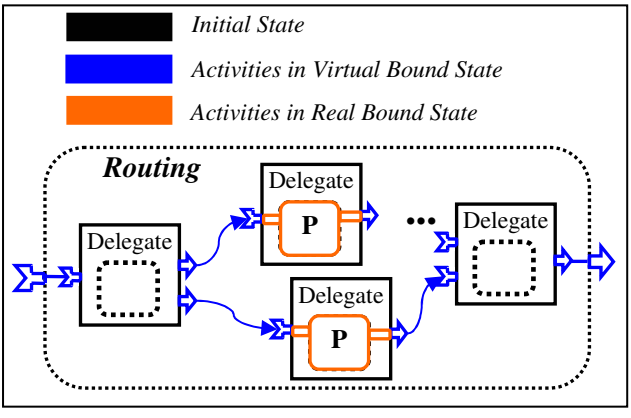


Fig. 3. Different States during Routing Execution

Temporal Binding for Execution

The procedure of routing execution is a serial of temporal binding of component processor instance with routing. It is the temporal binding that enables the maximum independence of component with others and results in flexibility of dynamic reconfiguration on software components. The procedure of routing execution can be divided into two stages: *Virtual Binding* and *Real Binding*.

Virtual binding is a serial of operations to make the component delegates of routing connected. One key task of virtual binding is ports binding for delegates, which means connecting delegates according the provided or automatically detected binding pairs of coupled ports. If all ports of all delegates in a routing have been bound, a dataflow pipeline of routing from input set to output set has been created. After virtual binding, the communication path between delegate and its processor is still separated, that is the responsibility of real binding. For a component processor, only when it is its turn to execute, the operation of real binding will be processed and the component processor will run to execute the functionality. After execution, the processor should also process the operation of real unbinding to unload processor instance from routing. So it is possible for a processor to serve in several active routing at the same time. Of course, the component processor has to be synchronized to ensure different routings can orderly acquire an independent time slice for it.

The separation of virtual binding and real binding increases the independence of component from its cooperated components, and create a completed idle running environment for all involved components – virtual bound routing which provides a perfect operation environment for dynamic reconfiguration.

4 Secure Middleware System – Smart Data Server Version 3.0

We have developed a secure middleware system based on the technology of routing based workflow – Smart Data Server Version 3.0, abbreviated as SDS3 and depicted in Fig.4. SDS3 is built on modified CORBA communication platform to provide a dynamic reconfigurable secure solution for middleware architecture. In SDS3 the security is not provided as middleware services, but integrated into core components of SDS3. In our design just part of core components, such as authentication, access control components etc., are integrated to reconfigurable part managed by RBW. The rest are non-reconfigurable part, such as underlying communication infrastructure – Object Request Broker.

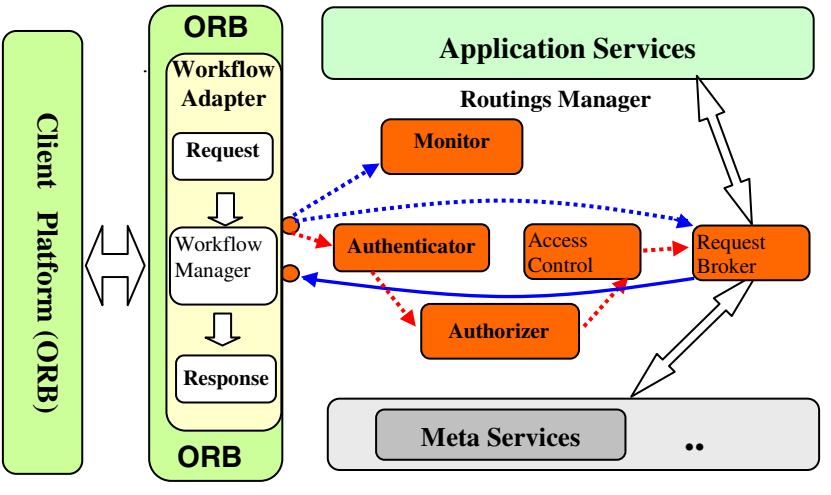


Fig. 4. Architecture of Smart Data Server Version 3.0

4.1 Underlying Infrastructure – Modified Object Request Broker

CORBA is a standard and specification of middleware computing paradigm from Object Management Group (OMG) [11]. Here we adopt the open source project openORB [12] and utilize part of CORBA – Object Request Broker, instead of the entire platform. To transfer our routing based request and response we make some modification for the ORB. The modifications are involved in Dynamic Invocation Interface for the client end and Object Adapter of the server side.

Wrapper for Dynamic Invocation Interface

In CORBA Dynamic Invocation Interface (DII) is a flexible, dynamic object access method. For the reason of flexibility and simplicity, DII is chosen as the base of object access method in SDS3. In the original DII client program has to obtain an object reference through additional transferring of object ID or invocation of *Name Services* of CORBA that still need the support of running of entire CORBA. In SDS3 we create a wrapper for DII that enables the creation of object reference in local

computer. But the price is that the remote information of object, such as host name, port and object name etc, have to be specified when instantiating a object reference. The DII wrapper also enables transferring of routing and security information, such as signature etc.

Wrapper for Object Adapter

In CORBA DSI collaborates with Object Adapter entity, namely Basic Object Adapter (BOA) or Portable Object Adapter (POA), to transfer a request to object implementation. In SDS3 request has to be intercepted from ORB to Routing Based Workflow (RBW) to execute functionality implementation. Between ORB and RBW we design a *Workflow Adapter* to take charge of the task similar to BOA or POA. *Workflow Adapter* is also registered as the default Object Adapter for all deployed objects, so it is in fact the only one adapter for all deployed services.

4.2 Reconfigurable Part – Components for Security Control

The reconfigurable parts of SDS3 are organized and managed by routing based workflow. Now there are four components available, namely *Monitor*, *Authenticator*, *Authorizer*, *Access Control* and *Request Broker*. The indispensable component is *Request Broker* which is responsible to invoke application. *Monitor* is used to register and supervise the accessing of clients. *Authenticator*, *Authorizer* and *Access Control* components work in a model of Role Based Access Control (RBAC) [10] to enhance the security of invocation. *Authenticator* employs Public Key Certificate (PKC) technology to authenticate whether current user is valid one or not. Attribute Certificate (AC) [9] based *Authorizer* component checks which role belongs to current user and judges whether this role is valid or not. *Access Control* checks whether this user can make this invocation or not according to a policy that records role hierarchy and access policy information. All these components work together to give different rights to different users for different accessing. The simplest routing contains only one component, *Request Broker,* where there is no restriction for access. Four other components are able to freely add to form different routings during run time as long as their ports are matched and the operation gets permission. For example, integration of four secure components can provide the strictest security control.

5 Analysis and Discussion

Routing Based Workflow (RBW) provides high dynamic reconfigurable capabilities for middleware components. The reconfiguration of routing will result in new interactive way of cooperated components, but these changes do not affect the functional processing of component and also successfully simplify the hard issue of consistency preserving for the temporal binding of component instance with routing.

5.1 Procedure of Dynamic Change

In RBW routing represents the structure and cooperation relation of software components. So change of routing means changing the inner structure of software components. Fig.5. describe the detailed steps to make change on a routing. For each request, *Execution Engine* has to acquire a specific routing to execute it. So even when *Execution Engine* is executing a request with *routing A, Routing Configurator* can also accept a request to make changes on it. What it has to do is just get a copy of *routing A*, and make changes on the copy, then update the copy on the repository of *Bound Routings*. Anytime *Execution Engine* can continue accept request for *routing A*. The difference is that it gets the old *routing A* before the updating and gets the new one after the updating. During the procedure of dynamic change, there is no any operation concerning on components. So the traditional hard issue of state transfer is avoid here, and issue of consistency preserving is also simplified to synchronization of routing updating in *Bound Routings* repository, which can be addressed much more easily.

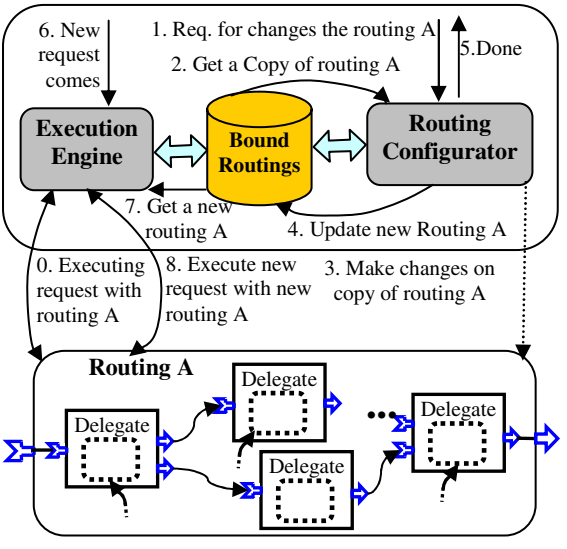


Fig. 5. Dynamic Change Steps

5.2 Dynamic Change Capabilities

Operations for reconfiguration in RBW can be classified into three categories: routing change, delegate change and port change. Routing change contains the operations for adding a new routing and deleting a routing. Here we have not offered operation for modifying a routing because it can be achieved by the delegate change and port change. Delegate change includes operations for adding a delegate, deleting a delegate etc. Operations for port change are enabling a port and disabling a port. When deleting a delegate it is only deleted from current routing. Only when the component is not used in any routing, the component processor and its container will then be deleted from repository. For ports of component processor are fixed after the completion of programming. So port updating only refers to the port of component delegate. When disabling a port of a delegate, it means to hide the port of delegate from other delegates. Component processor will adopt the default value as inputting of disabled port when performing execution. Only when a port is disabled, the operation of enabling for this port is available.

6 Related Works

Allen et al [3] separate the dynamic re-configuration behavior of architecture from its non-reconfiguration functionality, and reconfiguration occur only at points in the computation permitted by the participating components and connectors. Allen's approach based on the Architecture Description Language (ADL) is only passive to know when it's better to make the reconfiguration and has not addressed the problem of consistency maintaining. Goudarzi et al [5] present an approach to preserving consistency, and give a description of the intended changes, automatically identify and forces the affected components into a safe state. In [4], Almeida et al employ an approach based on Goudarzi et al, and they propose concrete solution to address dynamic issues for CORBA, such as structural integrity and mutual consistency etc. In [6], [7], Kon et al design a C*omponent Configurator* that is responsible for storing the runtime dependencies between a specific component and application components and other system. Through the communication and event contact between each component with hooked components and its client, dynamic reconfiguration is enabled for components that are already running. *Component Configurator* just records the dependences of component, but it has no consideration of the consistency. In our approach the design of control dependency are affected by the concept of *Component Configurator*. Rather similar idea with us can be found in [8], Shrivastava et al present a model based on workflow for distributed applications. In their model, workflow schema is used to represent the structure of tasks in a distributed application with respect to task composition and inter-task dependencies. Task controller is an expressive enough object to represent temporal dependences between constituent tasks and also used to guide the workflow execution. In our approach the design of component delegate are much effected by the idea of task controller. Because Shrivastava et al have not modeled the execution environment and they also directly bind the component implementation to task controller, so the there still exists the problem for consistency preserving.

7 Conclusion

Dynamic reconfiguration is increasingly demanded for large scaled distributed system supporting continuous services and reactive mobile computing. However there is still no perfect solution that addresses the hard issues caused by dynamic changes. In this paper we propose a Routing Based Workflow (RBW) and apply this technology to construct a high flexible and secure CORBA based middleware system - Smart Data Server Version 3.0. In RBW the component instance is only temporally bound to routing, the modeled execution environment for cooperated components. The capability of dynamic reconfiguration is realized by the change on bound routing, instead of components themselves. So the hard problems, like preserving consistency etc., are skillfully simplified to synchronization of routing updating. However, in our current solution all the component and their component are hosted on local computer. RBW is not able to manage the components distributed on remote computers. Now we are investigating to extend the *Component Delegate* local delegate and remote delegate which are expected to enable RBW applicable for distributed components.

References

1. N. Medvidovic, R. N. Taylor, "A framework for classifying and comparing architecture description languages". Proceedings of the 6th European conference held jointly with the 5th ACM SIGSOFT international symposium on Foundations of software engineering. Zurich, Switzerland. P.60-76, Sep. 22-25, 1997.
2. W.Huang, X.Zhang, U.Roth and Ch.Meinel. Routing Based Workflow for Construction of Distributed Application. The 9th IEEE International Symposium on Computers and Communications. Alexandria, Egypt. June 29 –July 1. 2004, pages 80-85.
3. R. Allen, R. Douence and D. Garlan. Specifying and analyzing dynamic software architectures. In Fundamental Approaches to Software Engineering, volume 1382 of LNCS, pages 21-37. Springer-Verlag, 1998
4. J.P.A. Almeida, M. Wegdam, M. van Sinderen and Lambert Nieuwenhuis. Transparent Dynamic Reconfiguration for CORBA. In Proceedings of 3rd International Symposium on Distributed Object & Applications (DOA 2001). Rome, Italy. 17-20 September, 2001.
5. K. Moazami-Goudarzi. Consistency-Preserving Dynamic Reconfiguration of Distributed Systems. PhD thesis, University of London, Department of Computing, Imperial College of Science, Technology and Medicine, 180 Queen's Gate, London SW7 2BZ, UK, 1997.
6. Fabio Kon, Manuel Román, Ping Liu, Jina Mao, Tomonori Yamane, Luiz Claudio Magalhães, and Roy H. Campbell. Monitoring, Security, and Dynamic Configuration with the dynamicTAO Reflevtive ORB, IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing (Middleware'2000). New York. April 3-7, 2000.
7. Fabio Kon and Roy H. Campbell, Dependence Management in Component-Based Distributed Systems. IEEE Concurrency, 2000. 8(1): p. 26-36.
8. Shrivastava, S. and Wheater, S., Architectural Support for Dynamic Reconfiguration of Large Scale Distributed Applications. The 4th International Conference on Configurable Distributed Systems (CDS'98), Annapolis, Maryland, USA, May 4-6 1998.
9. S.Farrell and R.Housley. An Internet Attribute Certificate Profile for Authorization. RFC 3281. http://www.ietf.org/rfc/rfc3281.txt. April. 2002.
10. R.S.Sandhu, E.J. Coyne, H.L.Feinstein, C.E. Youman. Role-Based Access Control Models. IEEE Computer, Volume 29, Number 2, Feb. 1996, pages 38-47.
11. Object Management Group, Common Object Request Broker Architecture, http:// www.corba.org/.
12. Open Source Project – The Community openORB Project, http://openorb.sourceforge.net/.

An Integrated Architecture for QoS-Enable Router and Grid-Oriented Supercomputer*

Chunqing Wu** and XuejunYang***

School of Computer, National University of Defense Technology,
410073 Changsha, China
xixiwu2001@yahoo.com.cn

Abstract. The overall performance of networking computation is determined by the grid node computer and the network infrastructure under it. But the grid node computer currently in use is not designed for grid applications. At same time, router capacity is getting more and more powerful. To Combine high performance router with traditional supercomputer will be a feasible way for both supercomputing and high performance QoS routing. At one side, it can release the node computer from the networking overhead and focus on computing and data processing, At the other side, it bring more powerful computing power to router and enable it to meet stream-based QoS and security processing requirement. This paper proposed an innovative integrated architecture AQRGS(Architecture for QoS-enable Router and Grid–oriented Supercomputer), which makes the CPUs focus on their high performance computing and data processing and leave the network communication jobs to the more specialized network processors in the high performance router. Thus the computing and communication capability of the grid node computers will be both highly improved and the grid application will run much faster. This paper discussed the related issues of building a grid oriented parallel computer integrating the network processing, which can also be benefit for QoS processing when used as a high performance router.

1 Introduction

After more than 10 years research, grid supercomputing is still a hot topic when people talk about the grid research. Lots of effort has been put in the Globus-liked grid middleware researches which make the grid supercomputing possible. But

* This work is supported by National Natural Science Foundation of China (NSFC), under agreement no 90104001, National Basic Research Priorities Program of China, under agreement no.2003CB314802.

** Female, Asociated professor.Research interesting is network processor and high performance core router.

*** Male, Professor. Research interesting is high performance supercomputer, parallel supercomputing,etc.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 1218 – 1226, 2005.

performance is still the most important issue that blocks the real supercomputing jobs running on the grid. Besides of the networking infrastructure, the supercomputers itself is another important factor that affects the overall grid computing performance. Thus designing and making more powerful supercomputers which can combine the computing and communication together may be an efficient way to solve the performance problems.

In traditional supercomputers, the CPUs have to consume a lot of cycles on networking processing when the computer needs to communicate with other computers on the net. For the communication intensive applications, it will cause a lot of overhead that will obviously slow down the CPUs. For the scalable, cost-effective value-added services supporting different traffic priorities (voice, video, data), you need to guarantee the quality of service (QoS)[1]. Thus you have to apply intelligent traffic engineering management schemes which will dramatically slow down the computation intensive applications. In the case of online massively multi-player games (MMGs), you may need peer-to-peer overlay support[2] to address the large amount of players playing the sophisticated high-performance MMGs.

So combining the computing and communication capabilities tend to be a very important issue in making grid-designated supercomputers and QoS-enable high performance router.

2 Related Work

Lots of efforts have been put in the approaches to combine the computing and communication together. Active technology is one of the most important directions. Some related works are described bellow.

2.1 ASAN[3]

Georgia Institute of Technology's ASAN(Active System Area Network) is one of the important projects dealing with the topic of combining the networking with computing together. They have studied the ways to build a much more powerful network interface which is capable of doing a lot of networking related computing job (i.e. the stream based computation) such as computing checksums on packets, data encryption and data compression and network related services such as firewalls, intrusion detection, or denial of service policies. Their main idea is to utilize the new type of computing and transceiving capable FPGAs to construct active SANs that can perform the stream oriented computations together with communications when the data is in transit. They have got some good results by making experiments on a Myrinet based system[4] which applying a new structure implemented on FPGAs.

The structure is as in Fig. 1.

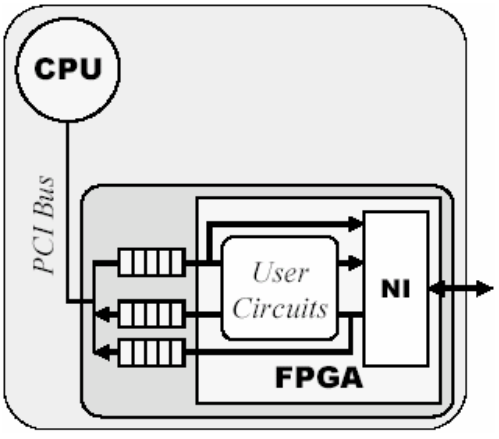


Fig. 1. Host with FPGA-Based NI

They use the new type FPGA's on-chip hardware such as memory, processor, transceivers to construct an enhanced NI(Network Interface) to off-load the host CPU from the communication functionalities. They also developed a connection-oriented programming model of communication where computations can be associated with connections and dynamically placed in the NI through the run-time reconfiguration of the FPGAs for application-specific, or connection-specific customizations. Their ultimate goal is the availability of a scalable communication and computation infrastructure for cluster-based data intensive computing.

2.2 Active Messages[5]

Active Messages is an asynchronous communication mechanism proposed in UC Berkeley while they study the large-scale multiprocessor computers. Their goal is to provide a cost effective way to expose the full hardware flexibility and performance of modern interconnection networks and coordinate the computing and communication without sacrificing processor cost/performance. They concluded that active messages is an efficient way to reduce the communication overhead dramatically on message driven machines. With this mechanism, overlap of the computing and communication is easily achieved and latency is successfully hided. And latency tolerance becomes a programming / compiling concern.

The basic idea of the Active Messages is that the control information at the head of a message is the address of a user-level instruction sequence that will extract the message from the network and integrate it into the on-going computation.

They have examined the message passing machines like Ncube/2 and CM5 and proofed the active messages mechanism is an order of magnitude faster than the traditional three-phase send/receive protocol.

They also developed a simple programming model that provides split-phase remote memory operations in the C programming language, i.e. Split-C, an experimental programming model using Active Messages. And they showed how the Active

Messages can be incorporated into a coarse-grain SPMD (single-program multiple data) programming language.

In addition, message driven architectures like J-machine and Monsoon and the major difference of the message driven computation and the Active Messages were discussed.

Active Messages shows its advantage over both the message passing and the message driven mechanism.

As for the hardware support for the Active Messages, they discussed two ways: improvement to network interfaces and modification to the processor to facilitate execution of message handlers.

2.3 Active Networks[6, 7, 8]

Active Networks are a novel approach to network architecture in which the switches of the network perform customized computations on the messages flowing through them. This approach is motivated by user applications, which perform user-driven computation at nodes within the network, and the emergence of mobile code technologies that make dynamic network service innovation attainable. These networks are active in the sense that nodes can perform computations on, and modify, the packet contents.

Its goal is to solve the problems of today's network which has difficulties of integrating new technologies and standards into the shared network infrastructure, poor performance due to redundant operations at several protocol layers, and difficulties of accommodating new services in the existing architectural model.

2.4 Summary

Active SAN, Active Messages and Active Networks are three different ways to combine computing and communication. But the Active SAN focuses on the SAN architecture to enhance the performance of cluster computers only. Active Messages emphasizes improving the inter-processor communications within the massive parallel architecture itself, and Active Networks works on the networking infrastructure to expand the switch or router with computation capability.

3 The Architecture of the AQRGS

In this paper, we focus our effort on constructing a new architecture for the emerging grid applications and at the same time, improving the performance of the grid node supercomputers.

Grid is the future computing environment in which we can share computing, storage, and various information resources all over the world. And one of our dream is collective supercomputing over the grid which will involve a number of supercomputers connected to the grid. Today's supercomputers do not have the designated support for the inter-supercomputer communication mechanism to coordinate the collective computing. Usually, it takes the traditional supercomputers too much time to send or receive a message when they need to communicate each other. This is mostly due to the fact that they have to use their CPUs to perform the networking operations where the big latency will obviously occur.

Progress on network processors in recent years makes it possible to construct a grid-oriented architecture in a cost-effective way to satisfy the emerging application needs in the near future.

AQRGS (Architecture for QoS-enable Router and Grid–oriented Supercomputer) is our approach utilizing the sophisticated network processor technology to offload the communication overhead from the CPUs and improve the computing perfomance of router.

3.1 The Structure of AQRGS

The basic idea of AQRGS is to combine a NP (network processor) via processor bus interface with each of the traditional PNs (Processing Nodes, i.e. the CPUs or super-nodes) in a massive parallel supercomputer, where the PNs are hooked to the interconnection network. In the issues, we call the network processor as NCP (Network Communication Processor), which work as a co-processor to PN. When it is used as a router, each PN is called as RCP(Routing Computing co-Processor, usually one CPU for one node in the issue). NPs are connected via switch interface to switch fabric. NP communicate with network environment via various network interface, such as 10Gbps POS/WAN/LAN, 2.5Gbps or 1Gbps Ethernet. To connect it with storage, the router may also have some InfiniBand-like network interfaces. Fig. 2 is the architecture of AQRGS.

The bandwidth between PNs is much wider than PCI or other processor bus. The bandwidth among NPs are close to bandwidth among PNs. Within this structure, we can easily see the interconnection bandwidth is nearly doubled and connectivity between the PNs is much better.

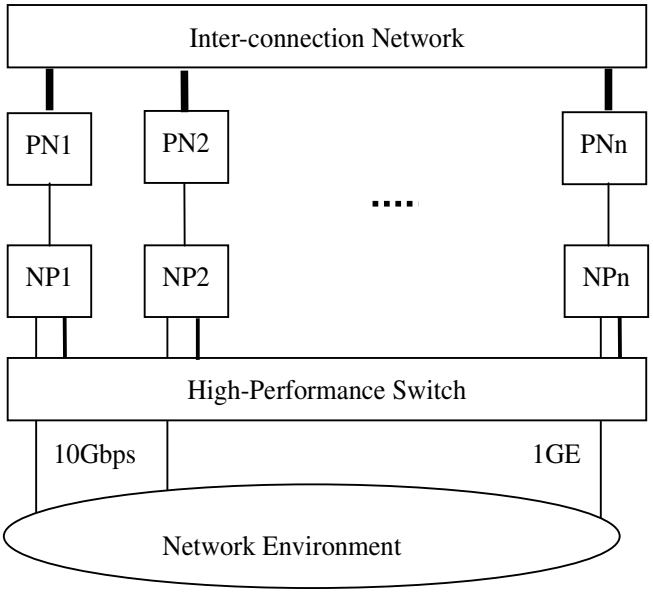


Fig. 2. AQRGS Structure

When it is viewed as a supercomputer architecture, it has three advantages over legacy architecture. At first, we can achieve the short messages exchange and control message communication among PNs via NP. Secondly, we can offload processing overload to NP when we use NP as a mean of short message communication, which usually consumes a lot of CPU cycles. As a result, PNs are dedicated to computing. Thirdly, when the supercomputer needs to communicate with another one on the grid, the related CPU can send an active message though its NCP and get back to the work it is doing. And the NCP processes the message and prepares all the necessary information for the message transfer.

When it is viewed as a new QoS-enable router, it brings the router powerful computing performance. It will have two advantages. Firstly, stream-oriented processing or QoS processing based on Diff-serv model need more and more computing performance, the traditional one CPU for one core router can not meet the requirement. Secondly, today's core router usually have only one CPU. As a result, the network infrastructure become low efficiency since core routers suffering from DDOS attacks.

You can also perform the Active SAN mechanism on the new network when the Active Message mechanism is running on the traditional interconnection network. This will indeed improve the system performance and at the same time, makes the grid communication much easier.

3.2 The Challenges of the AQRGS Architecture

The challenges in constructing AQRGS are the following:

- The communication mechanism between PN and its NCP.
- Programming model[9] The challenge is a programming model that enables users to develop applications that can effectively make use of the NCP. And this programming model will support both the grid wide computing and the computing within a grid node. Further more, the model will make the most use of the SAN resource to enhance the communication between the PNs.
- Connection resource management: a methodology of how to choose a right connection, i.e. whether to go through interconnection network or the SAN like network switch, is needed and thus the shortest route algorithm should be designed.
- Cost-performance evaluation of the architecture, what is the tradeoff between performance improvement and extra hardware and software.
- The task assign algorithm between network processor and high performance CPU.
- The collaboration mechanisms among high performance CPUs when it is used as a QoS-enable router.

Proof of the concept for this work will be established by constructing a prototype system using commercially available resources: network processors inside a high performance router, existing traditional supercomputer, and commercial product.

4 The Prototype System Implementation

The ideal communication between PN and its NCP could be a dual core semiconductor on which PN and NCP share the cache or memory. But in the prototype system, it is enough to construct it by the existing parts and equipments.

As the Fig. 3, we use an existing supercomputer with 16 P5 CPUs, a router with 16 forward cards, each incorporated with a network processor and an internet simulator to construct the test bed. In the approach, we use one gigabit Ethernet port to act the channel between PN and NP.

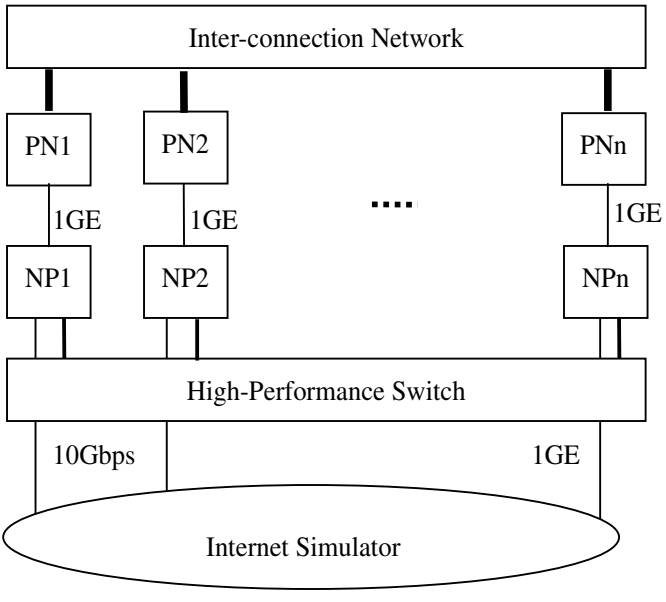


Fig. 3. The Prototype System

In the prototype system, supercomputer's CPU board has spare PCI slot which can be used to plug a gigabit Ethernet interface card. The NCP module here is a modification of the forward board developed for a high performance core router. On the forward board there is an IBM NP4GS3 network processor, 386MB memory and G-bit Ethernet ports, etc. and is well suited to the prototype needs.

The NCP module runs an embedded operating system and performs all the networking functionalities acting as a peripheral device of the CPUs.

The Internet/Grid Simulator is constructed based on the software router technology.

Two kinds of experiment are evaluated on the test bed, grid-oriented network supercomputing and QoS processing of high performance router. The first kind experiments evaluate the issues about reducing communication overhead with NP as a

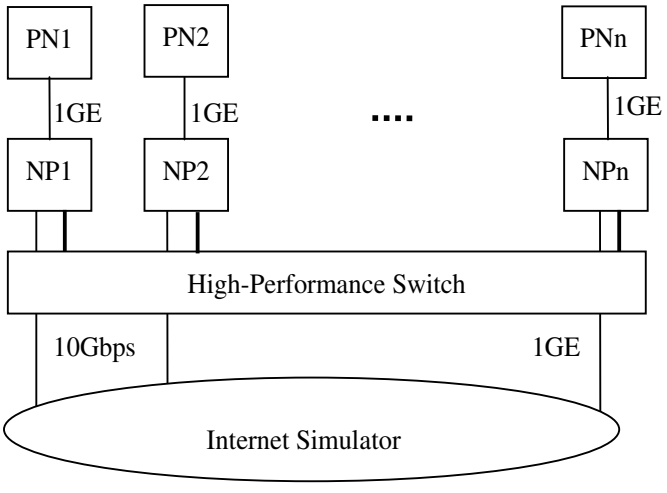


Fig. 4. Architecture for router's computing performance enhanced evaluation

communication coprocessor to CPU. The second kind experiments evaluate the issues about computing performance of router enhanced by the CPUs. The architecture is simplied as Fig. 4.

The experiment results show that communication overhead can be great reduced by the introduction of network processor into each node of supercomputer. And routing computing performance are enhanced by attaching the CPU to each NP, which make the router behave well, especially when the router suffering DDOS attack and other kinds attack.

We will also try the Active Message over the interconnection network and try the Active SAN though the switching router.

5 Conclusion

Grid computing is a promising area that needs to be studied and developed in depth. Constructing the Grid oriented supercomputers is one of the interesting directions. But we have to know what the Grid really need from the supercomputers, and what the supercomputers can do for the Grid applications.

This paper proposes a novel structure utilized a network processor technology that trying to integrate the computing and communication. At the same time, the work is undergoing to construct a prototype system which not only proof the advantages of the new architecture, but also act as a test bench and experimental platform for Grid supercomputing applications. Together with the internet/grid simulator, it will help us to know much more about both the Grid and future supercomputer and to do the right thing to cope with the emerging Grid supercomputing application challenges. At the same time, core router can beneficed from the similar architecture when it is used in a environment suffering from DDOS and other network attacks.

References

1. Maher Ali, Girish Chiruvolu, and An Ge, Alcatel, Traffic Engineering in Metro Ethernet. IEEE Network Magazine,March/April 2005.
2. Björn Knutsson,Honghui Lu,Wei Xu,Bryan Hopkins, Peer-to-Peer Support for Massively Multiplayer Games,Infocom 2004.
3. Craig Ulme,Chris Wood, and Sudhakar Yalamanchili,Active SANs:Hardware Support for Integrating Computation and Communication Proceedings of the Workshop on Novel Uses of System Area Networks at HPCA (SAN 2002)
4. N. Boden, D. Cohen, R. Felderman, A. Kulawik, C. Seitz, J. Seizovic, and W. Su. Myrinet: A Gigabit-per-second Local Area Network, in IEEE Micro, vol. 15, no.1, 1995.
5. T. Eichen, D. Culler, S. Goldstein, and K. Schauser, Active Messages: A Mechanism for Integrated Communication and Computation, in Proceedings of The 19th Annual International Symposium on Computer Architecture, pp.256-266,1992.
6. D. L. Tennenhouse, J. M. Smith, W. D. Sincoskie, et al. A survey of active network research. IEEE Communication Magazine, 1997, 35(1):80~86.
7. D. Buntinas, J. Duato, P. Sadayappan, and D. K. Panda, NIC-Assisted Broadcast/Multicast on Myrinet, Workshop on Communication and Architectural Support for Network-Based Parallel Computing (CANPC), Jan 2000
8. Dhabaleswar K. Panda: Active Network Interface: Opportunities and Challenges, LCN 2002: 605-608 (27th Annual IEEE Conference on Local Computer Networks (LCN 2002), 6-8 November 2002, Tampa, FL, USA, Proceedings. IEEE Computer Society 2002, ISBN 0-7695-1591-6)
9. George Bosilca, Aurelien Bouteiller, Frank Cappello, etc. MPICH-V: Toward a Scalable Fault Tolerant MPI for Volatile Nodes, SC2002, 2002.

APA: An Interior-Oriented Intrusion Detection System Based on Multi-agents

Dechang Pi [1], Qiang Wang[2], Weiqi Li[3], and Jun Lv [4]

[1,2] College of Information Science and Technology,
Nanjing University of Aeronautics and Astronautics, Yudao Street 29, Nanjing, Jiangsu 210016, PR China
[3,4] G.E International Software-System Co.Ltd, Nanjing 210002, PR China
[1] dechang_pi@hotmail.com
[2] nuaacs@yahoo.com.cn
[3] liweiqi@public1.ptt.js.cn

Abstract. Considering some employees in the department abuse their privilege for personal gain through the local network, in the paper, we present a distributed intrusion detection system named APA (Application Process Audit), which tackles the interior violation. APA provides a multi-agents system to set up tailored intrusion detection systems for real-time applications. Data mining technologies have been applied to the alerts file and audit logs in order to find some interesting audit rules, at the same time the rules base can be automatically extend with these rules. The whole system has six kinds of agent, which cooperate with each other to implement the monitor. Now APA has been applied to several security departments and has received a good reputation.

Keywords: Intrusion Detection; Multi-Agents; Audit; Network Security.

1 Introduction

As the network-based computer systems play a vital role increasingly today, they have become the target of the intrusions. Fraudulent employees abuse their privilege for personal gain. As a result, intrusion detection systems (IDSs) have received increasing attention in recent years.

Intrusion detection systems have proved to be an effective instrument for security. Systems with real-time capabilities provide automated protection of computer and network resources and allow the detection of ongoing security violations. Intrusion detection systems are currently one of the few reactive security mechanisms to counter threats on the communication infrastructure.

In the paper we present a distributed intrusion detection infrastructure of APA that is developed at Nanjing G.E International Software-System Co.Ltd. The objective of the APA approach is to get a module system to flexibly set up intrusion detection systems for real-time applications. The APA concept is based on the use of agents which can be combined to set up a tailored intrusion detection system, and it meets the requirements of a given application environment.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 1227 – 1233, 2005.

Extensive researches have been done in this field [1-7], but they have been mainly designed for external attackers. Guy Helmer et al. [1] designed and implemented an intrusion detection system prototype based on mobile agents, which travel between monitored systems in a network of distributed systems to obtain information. Sung Baik and Jerzy Bala [6] present their preliminary works on an agent-based approach applied to intrusion detection domain. Ricardo S. Puttini et al. [5] propose a distributed and modular architecture dedicated to a mobile ad hoc network environment. Oleg Kachirski and Ratan Guha [3] implement an efficient and bandwidth-conscious framework based on mobile agent. Moon Sun Shin et al. [2] put forward a false alarm classification model to reduce the false alarm rate using classification analysis. Tadeusz Pietraszek [7] describes an adaptive learner for alert classification, which can reduce the false positives in intrusion detection. P. Ramasubramanian and A. Kannan [4] use a combination of both statistical anomaly prevention and rule based misuse prevention in order to detect misuser.

APA system applies itself to interior detection in the local network and at the same time it aims at the application level. This is the main difference between APA and the other IDSs.

2 The Architecture

APA monitors the system by scanning the audit data that represent the users' operating trace (such as operate the application system, database, operating system and network). Fig.1 describes its configuration about monitoring, auditing and security.

The system has the following agents, which cooperate with each other to implement the whole monitor.

1. Data Collecting Agent (DCA). This kind of agent collects the running information about the user terminal and the network, and submits the messages to the EDA (Event Detection Agent).
2. Event Detection Agent (EDA). Once this kind of agent receive audit data sent by the DCA, it lookup the audit rule base to call the corresponding EA (Event Agent) to monitor the terminal.
3. Event Agent (EA). EA carries on the signature of the audit data and the audit rule to math in order to judge whether the event violate the rules or not.
4. Interdict Agent (IA). Once this kind of agent runs, the terminal-user's operation will be rejected.
5. Audit Agent (AA). It writes the audit logs and alerts to files in order to audit afterwards. These data is useful for Data Mining Agent.
6. Data Mining Agent (DMA). This kind of agent executes the data mining algorithms to find some interesting rules. These rules can be used to extend the audit rule base.

Knowledge Query and Manipulation Language (KQML) is a kind of communication language for agent. At the same time it is a descriptive protocol for information

and knowledge exchanging. The communication between agents in APA is peer to peer. The following is a typical message.

```
{
     ask-one
     :sender ProbeAgentID
     :content (MonitorContent)
     :receiver MonitorAgentID
     :reply-with AuditContent
     :language self-defined
     :ontology Monitor-mode
}
```

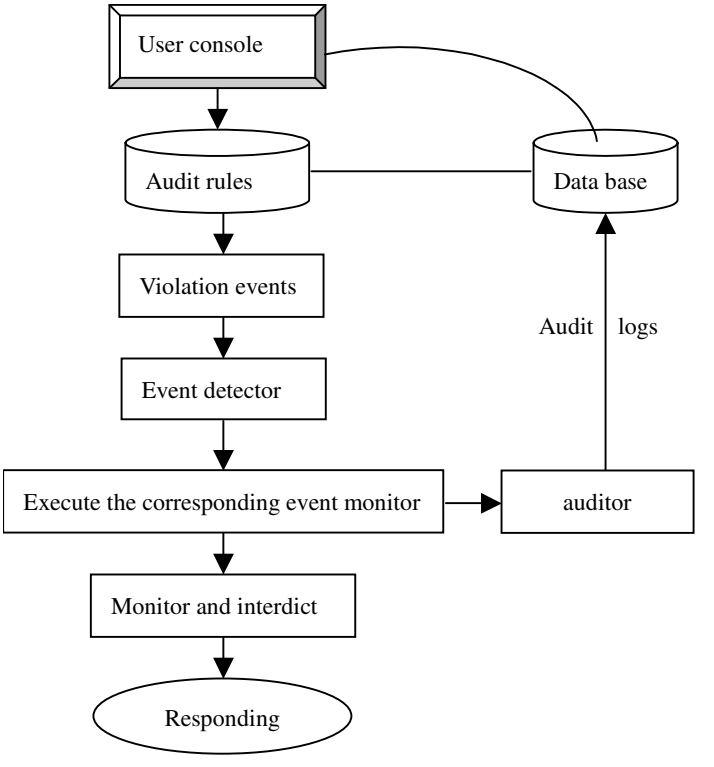


Fig. 1. The security idea of APA

The following is a KQML message of APA.

```
KQMLMSG   msg;                  // Message name
while (GetKQMLMessage (&msg, NULL, 0, 0))
{
   TranslateKQMLMessage (&msg); //Translate the message
   DispatchKQMLMessage (&msg);  //Dispatch the message
}
```

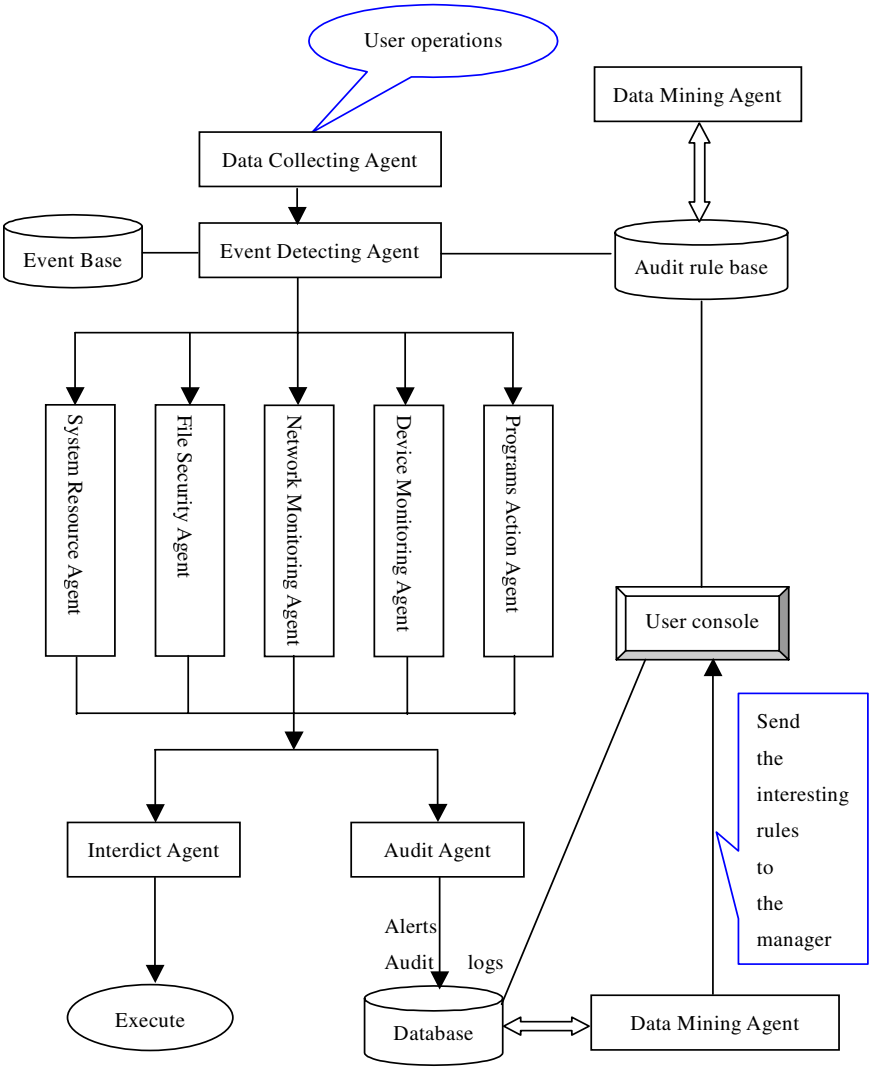


Fig. 2. The architecture of APA based on multi-agent

The APA architecture based on multi-agents is described in Fig.2. We now explain its two important agents, DCA and EDA.

2.1 Data Collecting Agent

DCAs (Data Collecting Agents) collect and handle the audit data. They aim at a fast reading, preprocessing and forwarding the audit data. DCAs can be placed at different points of the monitored hosts, which depends on the applied security policy.

DCA are located on each node of the network, and act as anomaly-based monitoring sensors at the user-level and system-level. These agents look for suspicious activities on the host node, such as unusual process memory allocations, CPU activity, I/O activity, user operations (invalid login attempts with a certain pattern, super user actions, etc). If an anomaly is detected with strong evidence, a DCA will terminate suspicious process or lock out the user and initiate re-issue of security keys for the entire network. If some inconclusive anomalous activity is detected on a host node by the monitoring agent, the node is reported to the decision agent of the same cluster that the suspicious node is a member of. If more conclusive evidence is gathered about this node from any source, the action is undertaken by the agent on that node.

2.2 Event Detecting Agent

After the fast capturing of the audit data by the DCAs, the second step to maximize the detection speed is to ensure an efficient analysis of these data. This is the task of the EDA. Beside the application of optimized analysis algorithms they use an appropriate distribution of data. This distribution is based on a classification of the signatures into local and distributed contexts. To detect signatures with a local context only local preprocessed data are analyzed, while for signatures with a distributed context data from various agents are demanded.

The most efficient way to perform such an analysis in a network is to apply a combined execution scheme. The detection of distributed attacks takes place on a central location. Unlike any other known system, however, APA applies this hybrid concept in a stringent manner to achieve a maximal local concentration and a minimal need for network traffic and delay. For this purpose we extended the notion of signature. In context of APA signatures are not only used for mapping complete security violation sequences. A signature can also represent a partial sequence of such an attack. This extension enables a hierarchy of agents to split the detection process for a distributed attack into a number of local sub-detections and a small amount of central combining.

3 Interface

The user interface of APA is graphical, which enables a security operator to perform several tasks in the context of a given APA intrusion detection system. The most important tasks are the configuration of the system and the visualization of the detection results. Furthermore, the user interface can act as a link between a security management and an intrusion detection system.

The whole of APA can be divided into three parts including user console, control server and agent. The console part is used by the manager, and coded in Delphi. It has a friendly interface, such as the Fig. 3. The agents run on the monitored terminals. After the users login, the agents automatically start and hide themselves. They begin to real-timely monitor the users' operators. If some violation appears, they will interdict the users from continuing operating. At the same time the audit data will be uploaded real-timely. The agent part is coded in c++, which has a high efficient.

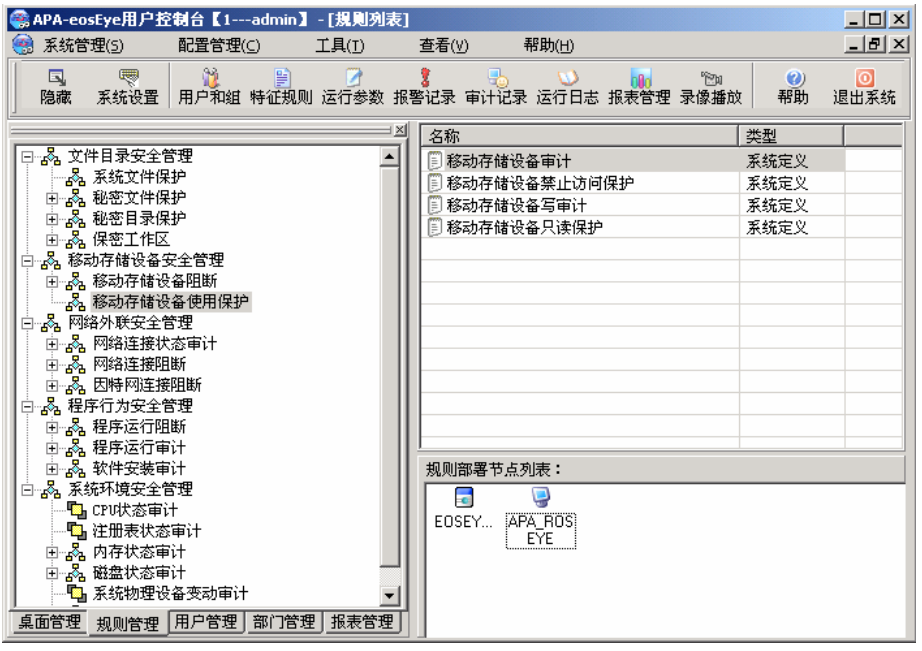


Fig. 3. The interface for the managers

4 Conclusions and Further Work

In the paper we have presented the intrusion detection infrastructure of APA. Its approach aims at a module system to set up efficient and tailored intrusion detection systems for local area networks. The module system provides a set of specialized agents for audit data capturing and flexible agents for data analyzing.

Future work will involve research into more robust and intelligent cooperative detection algorithms, as well as a choice of an anomaly detection model most appropriate for APA. We will investigate possible attacks on the system infrastructure of APA and on individual mobile agents in particular, and research effective means of defense.

Acknowledgements

Many thanks to three anonymous referees for their constructive comments.

References

1. Guy Helmer, Johnny S.K. Wong, Vasant Honavar, Les Miller, Yanxin Wang. Lightweight agents for intrusion detection. The Journal of Systems and Software 67 (2003) 109–122
2. Moon Sun Shin, Eun Hee Kim, Keun Ho Ryu. False Alarm Classification Model for Network-Based Intrusion Detection System. IDEAL 2004, LNCS 3177, pp. 259–265, 2004. Springer-Verlag Berlin Heidelberg 2004

3. Oleg Kachirski, Ratan Guha. Effective Intrusion Detection Using Multiple Sensors in Wireless Ad Hoc Networks. Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03) 0-7695-1874-5/03
4. P.Ramasubramanian and A. Kannan. Intelligent Multi-agent Based Database Hybrid Intrusion Prevention System. ADBIS 2004, LNCS 3255, pp. 393–408, Springer-Verlag Berlin Heidelberg 2004
5. Ricardo S. Puttini, Jean-Marc Percher , Ludovic Mé. A Modular Architecture for Distributed IDS in MANET. ICCSA 2003, LNCS 2669, pp. 91-113, 2003. Springer-Verlag Berlin Heidelberg 2003
6. Sung Baik, Jerzy Bala. A Decision Tree Algorithm for Distributed Data Mining: Towards Network Intrusion Detection. ICCSA 2004, LNCS 3046, pp. 206–212, 2004.Springer-Verlag Berlin Heidelberg 2004
7. Tadeusz Pietraszek. Using Adaptive Alert Classification to Reduce False Positives in Intrusion Detection. RAID 2004, LNCS 3224, pp. 102–124, 2004. Springer-Verlag Berlin Heidelberg 2004

Implementation of Ant Colony Algorithm Based-On Multi-agent System

Jian-min He[1], Rui Min[2], and Yuan-yuan Wang[1]

[1] Institute of Command Automation, PLA Univ. of Sci.& Tech.,
Nanjing 210007, China
Hejianmin2001@163.com
[2] Institute of Sciences, PLA Univ. of Sci. & Tech.,
Nanjing 211101, China
Minrui_he@163.com

Abstract. Ant colony algorithm (ACA) is a simulated evolutionary algorithm which was developed in recent years. ACA has attracted many researchers' attentions for the solving of combinatorial optimization problems. Agent-based simulation (ABS) is one of novel methods for the analysis of complex system. This paper introduces the basic principles of ACA and its method of design and implement in a multi-agent system (MAS). Computer simulation results of MAS based on ACA are introduced and discussed in this thesis. The results show that the reasonable combination of ACA and the simple local rules of agent can effectively improve the colony behaviors of agents.

1 Introduction

The aim of this paper is to present some preliminary work which applies Ant Colony Algorithm to Multi-Agent System to solve the problem of adaptive optimization for traveling route of agents in an unknown environment.

Combinatorial optimization (CO) problems are growing in importance for scientific research work. In order to obtain high quality solutions in a short run-time, heuristic approaches are one of the best alternatives. Ant Colony Algorithm (ACA) is one of the best heuristic approaches for solving CO problems. ACA is also called Ant Colony System (ACS) or Ant Colony Optimization (ACO), which was developed by Italian scholars, Prof. Marco Dorigo and his colleagues in early 1990s [1, 2]. ACA is a kind of evolutionary algorithm which was inspired by foraging behavior of ants in the real world. This behavior enables ants to find the shortest paths between food sources and their nest. Though a single ant is not clever enough to find the shortest way, the ant colony can solve the problem of optimal routing in a rather simple way. MAS is a new approach of analysis and design of complex system. Study on agent and MAS is becoming a focus of some interdisciplinary fields such as Artificial Life and Complexity Science. Agent-Based simulation (ABS) is considered as a promising way of analysis, design and implementation of complex system [3]. We apply the ideals of ACA to MAS and ABS and have got some elementary results.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 1234 – 1242, 2005.

2 Descriptions of ACA

When ants begin to search the foods, they explore the area surrounding their nest in a random manner. As soon as an ant finds a food source, it carries some foods and back to the nest. During the trip to the nest, the ant deposits some kind of chemical matter, which is called "pheromone", on the ground. The value of pheromone deposited on the path depends on the quantity and quality of the food. Then the pheromone trail guides other ants to the food source. More and more ants find the food source and deposit some pheromone during the back trip to their nest. Meanwhile pheromone will volatilize as time passing by. Intensity of pheromone on the shorter path is higher than other paths and therefore it attracts more ants follow it to find the food source. By using the indirect communication between the ants via the pheromone trail, the ant colony is able to find the proximate optimal path in a relatively short time. ACA imitates the process of finding the optimal path of real ants. We take the CO problem of Traveling Salesman Problem (TSP) as an example to explain the principles of ACA.

Suppose the path between city i and city j is denoted by r_{ij} and the distance of r_{ij} is d_{ij}. Total numbers of ants is denoted by $m = \sum b_i(t)$, where $b_i(t)$ represents the number of ants stayed at i^{th} city at time t. The pheromone value on path r_{ij} at time t is denoted by $\tau_{ij}(t)$, whose initial value is a constant C, that means $\tau_{ij}(0) = C$. When the discrete time step parading to t+1, the pheromone value on path r_{ij} becomes:

$$\tau_{ij}(t+1) = (1-\rho)\ \tau_{ij}(t) + \Delta\tau_{ij} \tag{1}$$

Where $\rho \in (0, 1)$, represents the volatility of pheromone. $\Delta\tau_{ij}$ is the increment of pheromone in the time step. Its value is calculated by:

$$\Delta\tau_{ij} = \sum \Delta\tau_{ij}^{k} \tag{2}$$

$\Delta\tau_{ij}^{k}$ is the pheromone value deposited by ant k on the path r_{ij} at time t. The initial value is $\sum \Delta\tau_{ij}^{k} = 0$. Then at time t we have:

$$\sum \Delta\tau_{ij}^{k} = \begin{cases} Q/\sum d_k & , \quad \text{if ant } k \text{ pass through } r_{ij} \\ 0 & , \quad \text{otherwise} \end{cases} \tag{3}$$

where Q is a constant, and $\sum d_k$ is the total length of all paths. The probability of ant k for choosing the next solution is defined as follows:

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t) \times \eta_{ij}^{\beta}}{\sum \tau_{iu}^{\alpha}(t) \times \eta_{iu}^{\beta}} & , \quad \text{if} \quad j \in A_k \\ 0 & , \quad \text{otherwise} \end{cases} \tag{4}$$

Here A_k is the subset of feasible cities which exclude those cities that the ant k has passed through. u is a city in the subset A_k. Parameter η_{iu} represents expectation of

willingness for ant k to transfer from city i to city u, and η_{iu} accordingly means the expectation to transfer from city i to city j. η_{iu} and η_{ij} are all some kind of heuristic information. In TSP we can simply set $\eta_{ij} = 1/d_{ij}$, and α, β are weighting functions. Variables α, β, ρ, Q and C make up of parameter set for the basic model of ACA. We should carefully choose their values according to the practical optimization problem.

From the descriptions of the algorithm we find that ACA has some meaningful mechanisms to ensure the validity of algorithm:

- Selection mechanism. The path which preserves more pheromone is likely to have a higher probability to be chosen by ants.
- Update mechanism. The intensity of pheromone on a path will increase as more and more ants pass through the path. Meanwhile it will decrease as the time passing by. The shorter path will accumulate more pheromone because it can transit more ants in a given time.
- Positive feedback mechanism. If there are more ants pass through a path, then will be more pheromone deposited on the path and it will guide more ants to the path.
- Collaboration mechanism. Ant colony is able to communicate and collaborate in an indirect way through the information preserved on the path.

As a result, a single ant is not "clever" enough to find the shortest path but the ant colonies are able to find the optimal or sub-optimal path in a short time. These mechanisms are very important for ants to adapt themselves to the complicated environments. Inspired by the basic instincts of ants, Dorigo et al put forward the Ant Colony Algorithm and it has been one of the preferred algorithms for solving CO problems in some certain areas.

3 Rules of Ants in MAS

We apply the ideas of ACA to the Multi-Agent System with the help of 2D Cellular Automata (CA). Here all the ants are independent agents. The agents want to accomplish a certain task obeying some local rules. They set out from the nest and try to find food source. Once have found the food source, they carry some foods and go back to their nest. When arrive at the nest, the agents put down foods and set out again to search the foods. The 2D CA provides us a virtual world which contains ants, nest, food source, paths, obstacles and pheromone. Ants can move on the free grids of CA but they have to avoid meeting the obstacles in front of them. Now we define some behavioral rules for ants.

Rule 1. If an ant does not carry food he must manage to search food source. If there is food source in his detectable range, he directly moves to the food source, otherwise he moves to the neighbor grid which has the highest intensity of pheromone. If the intensity of pheromone in his neighbor grids are all the same value that are greater than zero, then he will randomly choose a direction to move. If the pheromone values within his neighbor grids are all zero, then he will keep his advance direction.

We adopt the Moore's definition of neighborhood, which means, if the radium is denoted by r, then the agent will have $(2\ r + 1) \times (2\ r + 1)$ neighbors around him.

Usually we set r to 1. So the agent can detect the environment information in the range of 3×3 grids and move to one of the immediate adjacent grids during one run time step.

Rule 2. If an ant find a food source then he carry a certain mount of foods and go back to his nest at once. On his way home he will continually deposit pheromone on the path and the pheromone will evaporate with time.

We suppose the value of pheromone deposited on the path will decrease as the move distance increases. Because of the evaporation, the pheromone value will slowly decrease as time pass-by. We can calculate the evaporation with formula (**1**). When all foods are taken away by ants then the food source will no longer exist.

Rule 3. If there exists an obstacle ahead the ant's position, he will choose a direction for advance according to the pheromone value or by random selection.

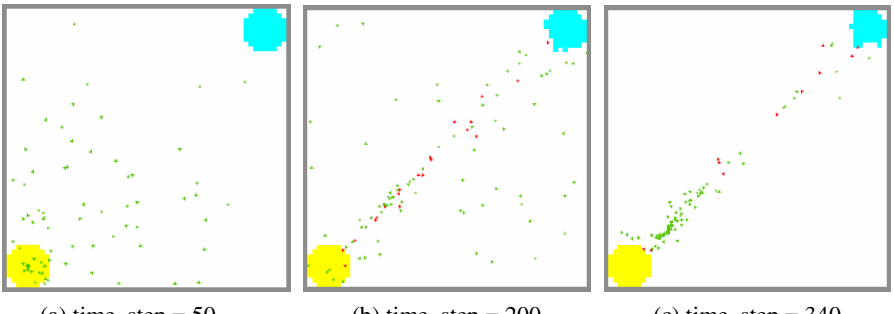
Drived by the simple rules, each ant's behavior is simple and predictable but the collective behaviors of agents are usually complex and unpredictable. It is the nonlinear interactions between the numerous agents that produce the high-level "emergent" behaviors. Now we apply the three simple local rules as the behavioral rules of agents and observe the high-level behaviors of agents.

4 Computer Simulations of Ant Colony

We construct a Multi-Agent System based-on ACA and the rules mentioned above to simulate the behaviors of ant colony. Users can easily change the setting of parameters such as the grid size of CA, the amounts of ants and food sources, rates of pheromone deposition and evaporation, etc. Furthermore, users can conveniently select obstacles and kinds of curves.

4.1 Behaviors of Ants on Free Space

There is no obstacle on the Cellular Automata now, so ants are allowed to move to any grids freely inside the CA space. The nest (solid circle in yellow) locates on the lower-left and the food source (in cyan) locates on the upper-right corner of CA. The CA size is set to 61×61 and the amount of ants is set to 100.



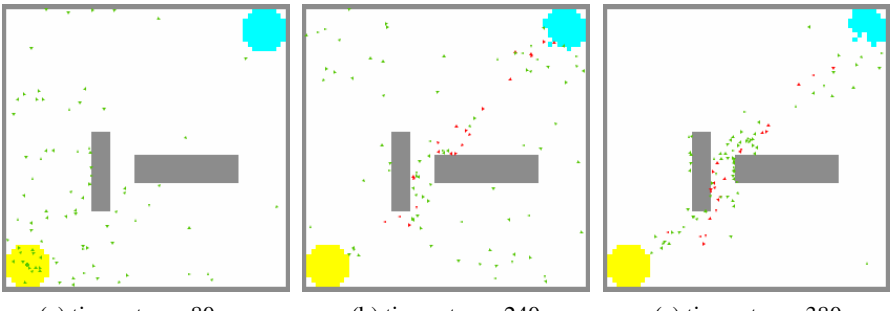
(a) time_step = 50 (b) time_step = 200 (c) time_step = 340

Fig. 1. Behaviors on free space

Fig. 1 is the screen snap in different stages of running. Fig. 1(a) shows that ants set out in succession from the nest to search the food source in random direction. After a period of time a few ants have found food source and carry some food and try to return to their nest, as Fig. 1(b) shows. The red dots in the figure represent the ants that are carrying foods. The green dots in the figure represent the ants who have not found the food source yet or who have put down foods in the nest. The pheromone deposited by the red ants attracts the nearby ants to search the food source along the direction that the intensity of pheromone increases. As the intensity of pheromone is not great enough, the ants in far distance are still walking in random direction. Fig. 1(c) shows that after a longer time, more and more ants are able to find the food source along the direction that the pheromone increases. They deposit more pheromone on their way home. Now the pheromone is intense enough to attract almost all the ants to the path constructed by the ants that are back nest. When the ants arrive at their nest, they put down foods and immediately set out again to search the food source. But now they will not walk randomly, instead, they can quickly find the food source directed by pheromone. The process keeps on like this until all foods in the source have been taken away.

4.2 Behaviors of Ants on the Space with Obstacles

We add some rectangular obstacles inside the CA space as shown in Fig.2. When an ant meets with obstacles he will turn round random angles and go ahead along the direction. So it is very hard for a single ant to find the optimal path in a space with obstacles. How about the ant colony? From Fig. 2(b) to Fig. 2(c) we find that there are more and more ants attracted to the space in the middle of two obstacles and construct the optimal path for ant colony.

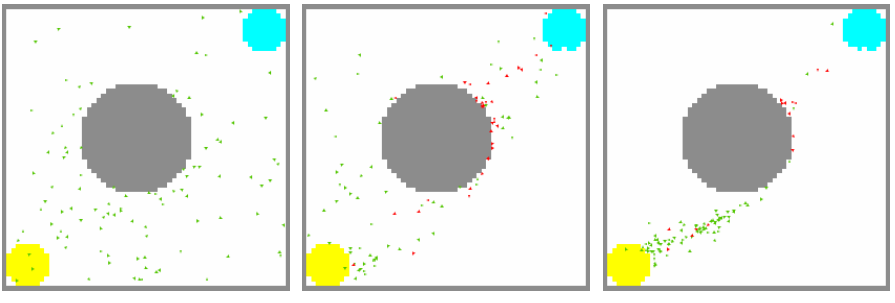


(a) time_step = 80 (b) time_step = 240 (c) time_step = 380

Fig. 2. Behaviors on space with rectangular obstacles

Now we change the rectangular obstacles to circular obstacle. If the coordinates of center point in CA is denoted by (0, 0), then the coordinates of the center of the solid circle are set to (-2, 2). The radius of circle is $r = 12$. Other parameters are the same with the last example.

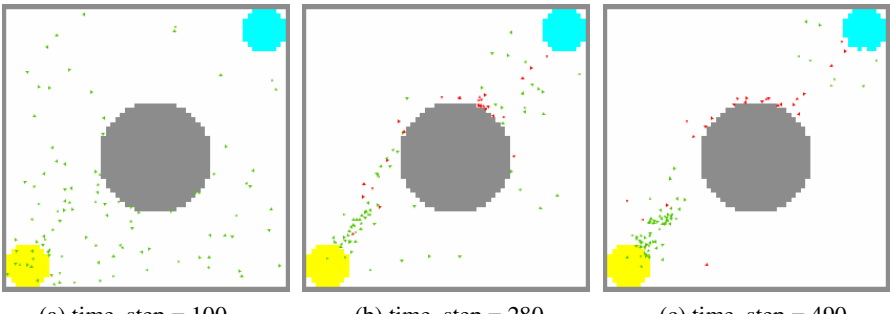
Fig. 3 shows three screen snaps in different steps of running. At the initial stage all ants set out from the nest in succession to search the food source in a random manner, as shown in Fig. 3(a). Then a few ants have found food source and carry food back to their nest. When they meet with the orbicular obstacle (i.e. the grey circle) they turn round a random angle so there are two queues around the circle as shown in Fig. 3(b). The situation is changed after a period of time. Due to the effect of pheromone, almost all the ants are attracted to go round the circular obstacle along with the lower-left path and therefore they form the optimal or sub-optimal path between the food source and the nest, as shown in Fig. 3(c).



(a) time_step = 100 (b) time_step = 270 (c) time_step = 460

Fig. 3. Ant colony goes round the circle in the lower direction

If we change the coordinates of center of the solid circle to (2, -2) and keep other parameters unchanged then we can get the simulation results as shown in Fig. 4. We find that, the ant colony makes correct choice again.



(a) time_step = 100 (b) time_step = 280 (c) time_step = 490

Fig. 4. Ant colony goes round the circle in the upper direction

The three instances listed above show that the Multi-Agent System based-on simple local rules exhibits “intelligence” to a certain extent. Although the environment has been changed, the agents are still able to find the optimal or sub-optimal solutions.

4.3 Behaviors of Ants with Two Food Sources

What will happen if there are two food sources nearby the nest? Fig. 5 is the simulation result of this situation.

The yellow circle in the middle-left is the ant-nest, and the two food sources locate in the upper-right (denoted by "food source 1") and lower-right (denoted by "food source 2") corner of the CA space respectively. Note that the distance between nest and the food source 2 is slightly shorter then the distance between nest and food source 1. In Fig. 5(a) all ants set out to search the food sources in a random manner. Fig. 5(b) shows there are two ant queues formed by the carrying-foods ants. For convenience, we call the two paths as "path 1" and "path 2" respectively. Fig. 5(c) and Fig. 5(d) show that more and more ants are led to the lower queue because the distance of path 2 is shorter so the intensity of pheromone is higher on this path. After all the foods have been taken away, ants begin to search new source of food, as shown in Fig. 5(e). The optimal path between nest and food source 1 has been formed in Fig. 5(f).

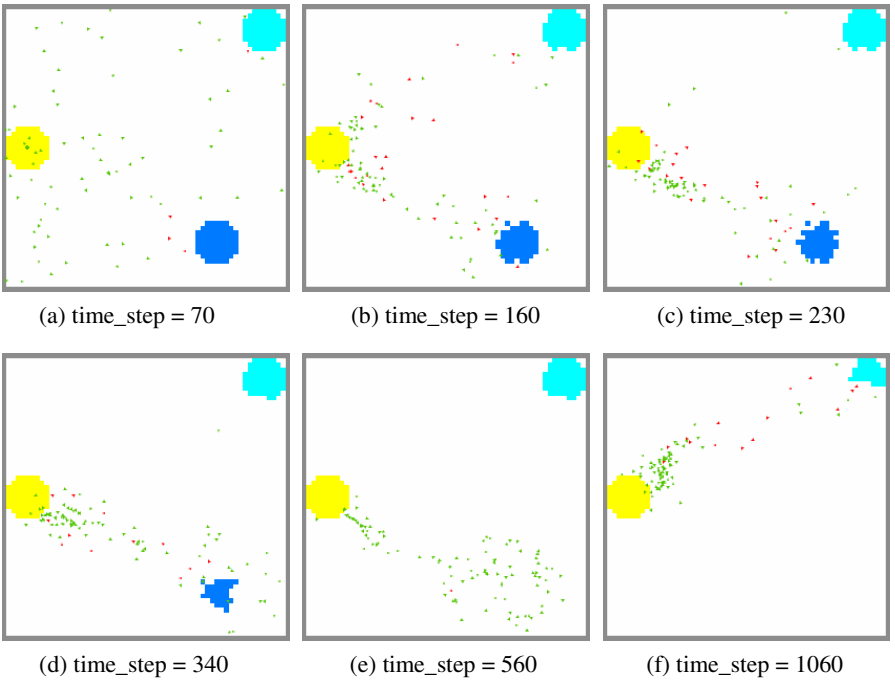


Fig. 5. Behaviors of ant colony with two food sources

Why the two queues formed in the initial stage rapidly reduce to one queue? It is the results of positive feedback mechanism of ACA. When ants begin to search food source, some of them found source 1 and some other found source 2. The ants deposit pheromone on their back-nest way to indicate the correct direction for other ants. Hence

two queues of ants are constructed at first. Since the distance of path 2 is shorter, the intensity of pheromone deposited on path 2 is higher in a given time. It attracts more ants along with this path and the intensity of pheromone is even higher. As a result, almost all the ants are attracted to path 2 after periods of time. Only after all the foods in source 2 have been taken away should the ants to look for new food source. To make it clear, we draw two curves to reflect the changing trends of foods amount in two sources, as shown in Fig. 6. This is very similar with the phenomenon we observed in the real world. If there are more than one food source around the ant nest, the ants typically taking away foods one source by one source, not taking away all the food source at the same time. Our simulation is a good explanation of this phenomenon.

Fig. 6. Foods stored in two sources

5 Conclusion

When Prof. Dorigo and his colleagues put forward Ant Colony Optimization model they solved the problem of TSP. Some other CO problems are solved afterwards. For example, Quadratic Assignment Problem (QAP), Job Shop Scheduling Problem (JSP), Vehicle Routing Problem (VRP), etc. References of some successful applications of ACA can be found in [4, 5, 6].

We study the ant colony algorithm in a novel point of view and implement it in Multi-Agent System. Besides the ACA described above, we use the ant model developed by the MIT Media Laboratory for reference [7], and combine them with local rules of ants. We consult the participatory simulations projects in NetLogo (version 2.1.0) when programming but improve it in many aspects especially in the ant's ability of evading obstacles. The "intelligent" behaviors emergent from ants are very interesting and valuable. The ant colonies exhibit the ability of adapting to the unknown and changing environments according to simple rules. It is very instructive for the study of complex systems. Cellular Automata, Evolutionary Computation have become focuses in computer science in recent years. They have the characters of self-organization, adaptation, and self-learning etc. We can find from the simulations described above that MAS has many advantages: simplicity, robust, scalability and so on. What is important is that the swarm intelligence indeed emergent from the local simple rules. These are embodiments of methodologies advocated by Artificial Life

and Complexity Science. It provides us a very valuable thoughtway for solving complex systems. It can be forecasted that the Ant Colony Algorithm will gradually becoming widely used in the areas of Agent-Based Simulation, Alife, Complex Adaptive System and so on.

References

1. Dorigo M., Caro G D, Gambardella L M.: Ant Algorithms for Discrete Optimization. Artificial life. 2(1999) 137–172
2. Dorigo M., Maniezzo V., Colorni A.: Ant system: Optimization by a colony of coorperating agents. IEEE Transactions on Systems, Man, and Cybernetics-Part B,1 (1996) 29–41
3. Jennings N R, Sycara K, Wooldridge M.: A Roadmap of Agent Research and Development. Autonomous Agent and Multi-Agent System, 1(1998)275–306
4. Stutzle T, Hoos H. In Meta-Heuristics: Advances and trends in local search paradigms for optimization. Boston: Kluwer Academic, (1999)313–329
5. Colorni A, Dorigo M. Maniezzo V et al. Belgian J. of Operations Research Statistics and Computer Science, 1(1994) 39–53
6. M. Dorigo, T. Stutzle: Ant Colony Optimization. MIT Press, Cambridge, MA, 2004.
7. Resnick M.: Turtles, Termites and Traffic Jams: Explorations in Massively Parallel Microworlds. MIT Press, Cambridge, MA. (1994)
8. NetLogo 2.1.0 User Manual. http://ccl.northwestern.edu/netlogo/. (2004)

Load Balancing Using Mobile Agent and a Novel Algorithm for Updating Load Information Partially

Yongjian Yang, Yajun Chen, Xiaodong Cao, and Jiubin Ju

College of Computer Science and Technology, Jilin University,
No.20 Nanhu Road, Changchun, Jilin Province, China 130012
yyj@jlu.edu.cn

Abstract. This paper introduces a model LBMA (Load Balancing using Mobile Agent) firstly. LBMA can resolve some problems in traditional load balancing, including structure of system, updating load information, and adjusting strategies of load balancing. Secondly, the paper analyses some traditional algorithms for updating load information and their disadvantages. Then, aiming at these disadvantages, we propose a novel algorithm for updating load information partially based on mobile agent and stochastic interval, named ULISI (Updating Load Information based on Stochastic Interval). Finally, from our simulation experiment results, we conclude that it is reasonable and feasible to introduce mobile agent to load balancing, and the performance of ULISI is improved.

1 Introduction

Two controlling manners of load balancing system (centralized controlling and distributed controlling) both have some problems [3],[5]. Centralized controlling has bottleneck and low reliability. Distributed controlling needs a number of messages to update load information and balance load among nodes. Moreover, in most load balancing systems, strategy of load balancing is single, so adaptability of system is low.

Based on the discussion above, we can find that there are some problems in structure, reliability, performance, and adaptability of current load balancing. To resolve these problems, we need to quest for new approaches [6].

Mobile agent is a novel technology originated from distributed network and artificial intelligence [8]. It has been used extensively, such as network management, electronic commerce. Mobile agent can move, and can take data and codes. It can execute tasks in destination nodes asynchronously, independently and automatically. Users can use mobile agent conveniently and flexibly. For example, in IBM Aglets [1], we can create, clone, dispatch, retract, activate, and destroy mobile agent.

Mobile agent can improve load balancing for following three reasons: (1) Improving efficiency and performance: Mobile agent can reduce data transmitting, save network bandwidth and overcome network latency, because it can move independently, and transfer computations into data fields. (2) Improving reliability: Mobile agent can be executed asynchronously and independently on destination nodes. (3) Improving adaptability: Mobile agent is intelligent, mobile, flexible, and active, so it

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 1243 – 1252, 2005.

can complete assigned tasks substituting origin host. Mobile agent can also apperceive the change of environment and respond to it [7].

This paper proposes a model named LBMA (Load Balancing based on Mobile Agent), and an algorithm ULISI (Updating Load Information based on Stochastic Interval). The rest of the paper is organized as follows: Section 2, 3 and 4 introduce system architecture, collecting and updating load information, and adjusting strategy in LBMA. Section 5 introduces ULISI. We simulate ULISI and present its results in section 6. We conclude for this paper and bring forward our future work in section 7.

2 System Architecture

LBMA adopts distributed controlling manner, which hasn't bottleneck. So every node can receive tasks independently. To improve efficiency, adaptability and extensibility of system, a controlling node is used. It can collect and update the load information, and monitor the running state of system. It would adjust the strategy and structure of system if necessary. These operations can be completed by mobile agent. In this way, extra cost is low, and the performance and adaptability can be improved. However, this manner may bring about a new bottleneck in controlling node. Fortunately, it can be overcome by mobile agent. Mobile agent can distribute computations into all nodes, and a mobile agent can accomplish the work in one time that needs many interactions among nodes in traditional ways. Figure 1 illustrates the network structure of LBMA.

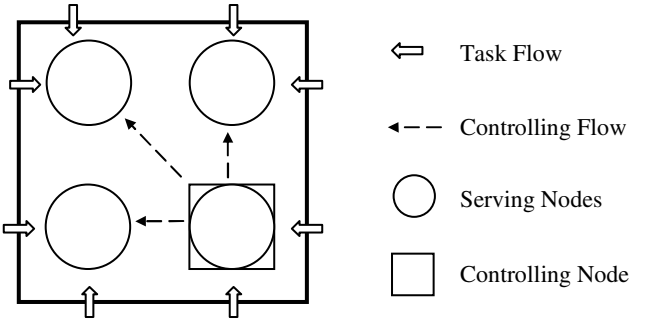


Fig. 1. Network structure of system

In this architecture, serving nodes can receive and execute tasks. The controlling node can control the whole system, and it is one and only. In LBMA, all controls are completed by mobile agent, so controlling node wouldn't be the bottleneck. To improve the reliability of controlling node, a standby node is adopted.

To achieve load balancing, every node must install LBMA system. Figure 2 shows node architecture of LBMA. Load Balancing Subsystem is similar with traditional load balancing system, which can complete some low level operations, such as processes migration. Mobile Agent Subsystem provides creating and executing environ-

ment. It can realize some functions, such as collecting and updating load information. Load Balancing Information Library saves the related information for balancing load. The former three parts are under the control of Main Controlling Program.

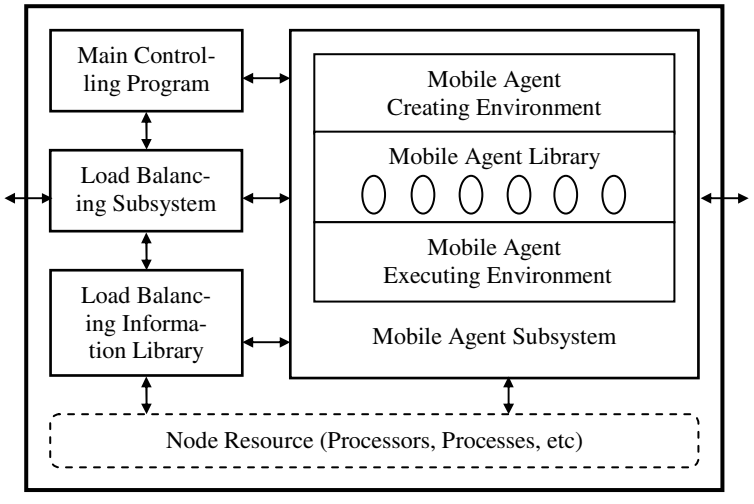


Fig. 2. Node architecture of LBMA

3 Collecting and Updating Load Information

Load information of each node should be collected first of all, to balance load among nodes. In most methods, load information of every node is collected by its local fixed agent LIC (Load Information Collector). But how can every node get the load infor-

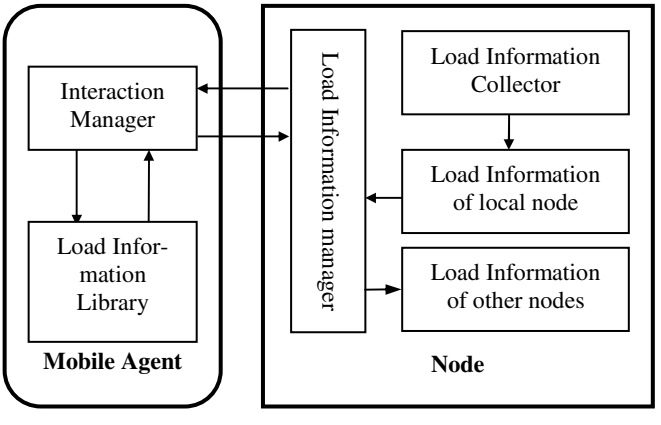


Fig. 3. Collecting load information using mobile agent

mation of others? Traditional methods act as followings usually. In centralized controlling manner, each node sends its load information to a load balancer, which is a central node to control the whole system. Then the balancer sends load information of all nodes to each node. In distributed controlling manner, every node broadcasts load information of itself to all the other nodes. However, there are some problems in both methods. In the former, the balancer will be the bottleneck. So the reliability is low. In the latter, so many messages to transfer load information will consume plenty of bandwidth. So, the performance is not high.

LBMA introduces the following method as figure 3: LIC of every node collects its current load information periodically. Mobile agent dispatched by controlling node collects load information of every node in every cycle. At the same time, mobile agent releases the load information of other nodes to every node, which is stored in mobile agent and up to date. Section 5 proposes a novel approach ULISI.

4 Adjusting Strategies of Load Balancing

LBMA can balance the load under the control of current load balancing strategy, after collecting load information. However there are many load balancing strategies can be chosen and used, such as sender-initiated, receiver-initiated, and so on. Every strategy can work perfectly only in its suited environment. So we need to adjust the load balancing strategies. Mobile agent can realize the adjusting of strategy, because it is intelligent and mobile. Moreover, it is asynchronous and can route dynamically, so it can improve the reliability of system. LBMA classifies adjusting strategies into three levels: tuning parameters, switching existent strategies, and adding new strategies.

(1) Tuning parameters: Every strategy has some parameters. So nodes can adjust them in order to adapt the current environment. If it is necessary to adjust some parameters, Controlling Node will dispatch a mobile agent for adjusting strategy to adjust the parameters to necessary nodes.

(2) Switching existent strategies: While designing a strategy of load balancing, we need to set up corresponding models and design appropriate strategies, according to the structure and tasks of system. No one strategy can satisfy all situations, and different situations need related strategies. So, to improve the adaptability of system, we must adjust the strategy itself (i.e. replace current strategy with another) other than adjusting parameters.

To switch strategies, evaluating strategies is needed. We can analyze the some load balancing strategies and their suited environment. For example, sender-initiated can acquire good performance, when the whole load of system is light (i.e. system-status). On the other hand, receiver-initiated can show its good advantage, when the whole load of system is high. So, LBMA need only analyze the whole load of system, and then choose the proper strategy as serving strategy. Figure 4 shows this principle. After deciding to switch current serving strategy into some other existent strategy, controlling node dispatches a mobile agent, which moves to every node to activate and start the chosen strategy from strategies library. In addition, the administrator can also choose strategy manually according to his experience.

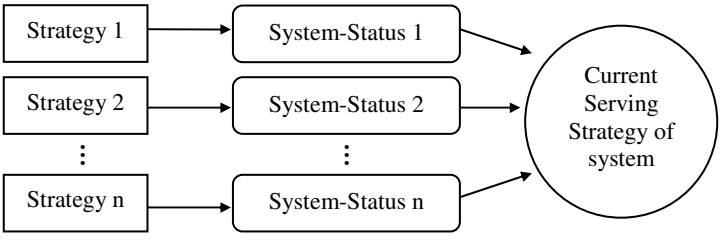


Fig. 4. Switching existent strategies of load balancing

(3) Adding new strategies: Sometimes, if we want to introduce some a new strategy, LBMA can add it into system while system is running, and can't affect the working. Mobile agent move to every node taking the new strategy, then load and install it on each node. Thus, the new strategy can be activated and used when necessary.

5 Updating Load Information Based on Mobile Agent

To prove the feasibility of using mobile agent in load balancing, we design a new algorithm (ULISI) for updating load information in subsection 5.2. Before this, we analyze two kinds of traditional algorithms in subsection 5.1. We also simulate the new algorithm in section 6.

5.1 Traditional Algorithms for Updating Load Information

According to the scope of updating load information among nodes, the algorithms of updating load information can be classified into ULIF (Updating Load Information Fully) and ULIP (Updating Load Information Partially).

(1) Updating Load Information Fully
In this algorithm, each node exchange load information between every other node. Every node broadcasts its load information to all other nodes in every interval. Therefore, in any time, every node can know the load information of others, and it can balance load easily. But, the number of messages to update load information is very large.

(2) Updating Load Information Partially
In order to decrease the number of messages, we can also only update the load information of partial nodes. ULIP is implemented as this. But how should I select these partial nodes? MOSIX and OpenMosix adopt a Probabilistic Dissemination Algorithm (PRODI) [2], [4], which is one of ULIP.

5.2 A Novel Algorithm for Updating Load Information Based on Stochastic Interval

In this subsection, we will introduce a novel algorithm called ULISI (Updating Load Information based on Stochastic Interval) which is also one of ULIP.

To make the problem simple, we suppose that every node had the same capability or resource. Before starting to state ULISI, we will define some parameters.

- In a system, there are n nodes which are marked by 1, 2, … , n.
- We abstract all resource in node i into one uniform value, including CPU, memory, disk and others. The percentage of resource utilization represents the load of node i which marked L[i]. L[i] can show the current load status of node i.
- L[i] is obtained and saved by node i periodically in every interval (T_n) for collecting local load information.
- Mobile agent circulates among all nodes in a system. The time of a circulation is Tm, which roughly equals to T_n.

5.2.1 Stochastic Interval

Often, stochastic values for a system characteristic are determined by examining a time series of previous values for that characteristic. Given data in the form of a time-series, the simplest way to represent the variability of a stochastic value is as an interval.

We define the interval of a stochastic value X to be the tuple.

$$X = [\underline{x}, \bar{x}] \ \{x \in X \mid \underline{x} \le x \le \bar{x}\} \tag{1}$$

The values $\underline{x}$ and $\bar{x}$ are called the endpoints of the interval. The value $\underline{x}$ is the minimum value over all $x \in X$ and is called the lower bound, and the value $\bar{x}$ is the maximum value, called the upper bound.

5.2.2 Description of Load Information Using Interval

From the above definition, we can have a description of load information using interval. As follows:

Assume that load of system is L, so L is a stochastic interval and is up to definition of interval. We can make a definition for L as that:

$$L = [\underline{l}, \bar{l}] \ \{l \in L \mid \underline{l} \le l \le \bar{l}\} \tag{2}$$

$$\underline{l} = \mathrm{Min}\{L_1, L_2, \cdots, L_n\}$$
$$\bar{l} = \mathrm{Max}\{L_1, L_2, \cdots, L_n\}$$

There L_1, … , L_n is a load state series of the node with a period of time. This period of time can be tuned.

We can make a safe conclusion that load state of every node in distributed system express as a two-tuples, but not a simple value. So there is a better description to load of every node.

5.2.3 Collection of Load Information Using Mobile Agents

There is a Node Load State Table on every node in distributed system, and the structure of this table is as follow:

```
strcut NodeLoadState
{
    int NodeNumber;
    float LoadMax;
    float LoadMin;
}
```

There is a Node Load State Series Window: $W = \{L_1, L_2, \ldots, L_n\}$, and this window will be updated at the end time of each collecting information periods.

At initialization, mobile agent will go out from a node and move between every node by engaged route policy. And it begins to collect the load information of the node. There is a Node Load State Table on mobile agent. When mobile agent moves to a new node, load state two-tuples on new node will be read by mobile agent. Node Load State Table on mobile agent will be updated and than Node Load State Table and Node Load State Series Window on the node will be updated by mobile agent. So after a period of mobile agent moved, the task of collecting load information of node finish.

5.2.4 Select Migrated-Task Node Matching Algorithms

Because we transform the load state value of node into the two-tuples, migrated-task node matching algorithms can be looked as the tow-tuples matching algorithms. This problem is hierachical cluster analysis at the math and can be conclude with degree of differ. The distance is the best method of calculated degree of difference. Generally, we use to calculate by Minkowski distance.

$$d(i,j) = \sqrt[q]{(|X_{i1} - X_{j1}|^q + |X_{i2} - X_{j2}|^q + \cdots + |X_{ip} - X_{jp}|^q)} \tag{3}$$

There $i = (X_{i1}, X_{i2}, \ldots, X_{ip})$ and $j = (X_{j1}, X_{j2}, \ldots, X_{jp})$are two p dimensionality Objects, q is a positive integer. It can make a better effect at compute distance with $q = 2$, and computational complexity is low.

So we can make a better matching algorithm:

1. Suppose that load state of every node is a two-tuples (L_{1i}, L_{1j}), (L_{2i}, L_{2j}), …, (L_{ni}, L_{nj}).
2. Calculate load of a suppositional middle node M.

$$M = (\frac{1}{n}\sum_{K=1}^{n} L_{ki}, \frac{1}{n}\sum_{k=1}^{n} L_{kj}) \tag{4}$$

3. Currently node will calculate Minkowski distance d with middle node M by load state two-tuples of itself.
4. Calculate Minkowski distance between other node and middle node respectively, and we have a distance series $d_1, d_2, \ldots, d_{n-1}$.
5. And than calculate $|d-d_1|$, $|d-d_2|$, … $|d-d_{n-1}|$, we can obtain a difference value series $S_1, S_2, \ldots, S_{n-1}$.
6. The best matching node P can be calculated:

$$p \in \{1,2,\cdots,n-1\} \cup S_p = Min\{S_1, S_2, \cdots, S_{n-1}\} \tag{5}$$

7. A migrated-task mobile agent will be started and migrate task to P node.

6 Simulated Experiment

This section provides the results of our simulation experiment. We present some comparison between Updating Load Information Based on Stochastic Interval (ULISI) and Probabilistic Dissemination Algorithms (PRODI).

To evaluate the performance of our proposed algorithm, we have used following performance metrics: degree of balance of all nodes, the execution time of all tasks, and number of messages. These three parameters can reflect the performance of algorithms. Our goals are to make the degree of balance of all nodes of ULISI higher than that of PRODI, to make execution time of all tasks in whole system of ULISI shorter than that of PRODI, and to reduce the messages among nodes.

Before our simulation experiment, we have tested some correlative parameters, including network delay, execution time of task and so on. The experiment is implemented in Linux installed OpenMosix and Aglets. Then we apply these parameters and some experiential values into the simulation.

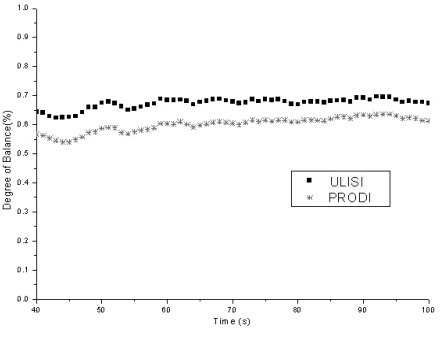


Fig. 5. Comparison for Degree of Balance between ULISI and PRODI

In our simulation, we compare that the average load rate of ULISI and PRODI. There are 100 nodes and 2000 tasks which are added to these nodes randomly. But the distribution of tasks on nodes is same in different algorithms. From Figure 5, we can conclude that the degree of balance of ULISI is higher about 2% than PRODI.

From figure 6, we can see that the total execution time of all tasks in ULISI is shorter than that in PRODI. It is more evident when the number of tasks is larger.

From figure 7, we can conclude that the more of nodes, the more of messages to update load information. But the increase in ULISI is slower than PRODI, because of the use of mobile agent.

As same as mentioned before, when there are adequate nodes in a system, messages flood will degrade the performance of whole system.

In figure 8, we can find that the average load calculated by mobile agent is very close to the real one. So we can prove ULISI is feasible.

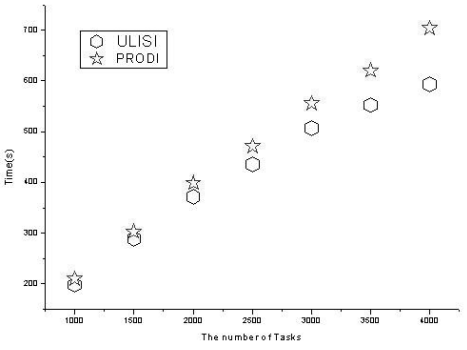


Fig. 6. Comparison for execution time of all tasks between ULISI and PRODI

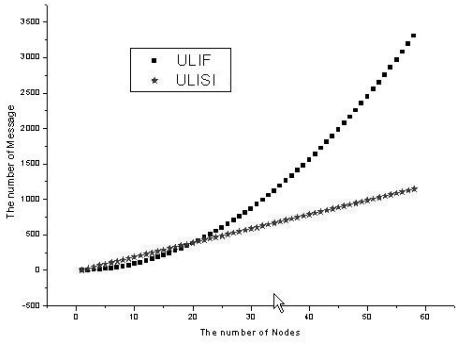


Fig. 7. Comparison for the number of messages between ULISI and ULIF

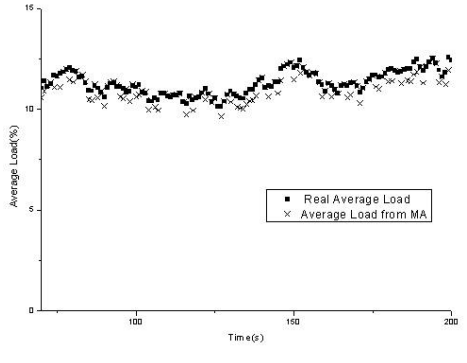


Fig. 8. The degree of fitting on average load of ULISI

7 Conclusions and Future Works

In this paper, we propose a model for load balancing using mobile agent named LBMA. LBMA can resolve some problems existed in traditional load balancing sys-

tem, such as bottleneck, messages flood, and adjusting strategies. It can improve the adaptability, efficiency, and reliability of load balancing.

Then we introduce a novel algorithm ULISI to update load information using mobile agent and stochastic interval. According to analysis to ULISI and its simulation experiment results, we can conclude that the performance of ULISI is better than PRODI. ULISI can avoid messages flood, shorten the whole execution time of tasks, and improve the performance of load balancing.

Our future works are to implement ULISI in real system, and to develop the platform for load balancing based on mobile agent.

Acknowledgements

This work was supported by ZHSTPP PC200320007 of China.

References

1. D. Lange, M. Oshima: Programming and Deploying Java Mobile Agents with Aglets. Addison-Wesley (1998)
2. A. Barak, O. La'adan: The MOSIX Multicomputer Operating System for High Performance Cluster Computing. Journal of Future Generation Computer Systems, March 1998, 13(4-5):361-372
3. J. He: An Architecture for Wide Area Network Load Balancing. Proc. 2000 IEEE Int'l Conf. on Communications (2000), 2:1169-1173
4. A. Barak, O. La'adan, A. Shiloh: Scalable Cluster Computing with Mosix for Linux. Proceedings of the Linux Expo, Raleigh, N.C. (1999), 95-100
5. Mohammed J. Zaki, Wei Li, Srinivasan Parthasarathy: Customized dynamic load balancing for network of workstations. In Proceedings of HPDC '96 (1996), 282-291
6. Jiannong Cao, Yudong Sun, et al: Scalable load balancing on distributed web servers using mobile agents. Journal of Parallel and Distributed Computing (2003), 63(10):996-1005
7. J. Gomoluch and M. Schroeder, Information agents on the move: A survey on load balancing with mobile agents. Software Focus (2001), 2(2)
8. V.A.Pham, A.Karmouch, Mobile Software Agents: An Overview. IEEE Communications Magazine (1998), 36(7):26-37

Online Internet Traffic Prediction Models Based on MMSE

Ling Gao[1], Zheng Wang[2,3], and Ting Zhang[4]

[1] Department of Computer Science, Xi'an JiaoTong University, 710041 Xi'an, China
gl@nwu.edu.cn
[2] Department of Computer Science, Northwest University, 710069 Xi'an, China
wangzheng@ieee.org
[3] China Research Lab, IBM, 200021 Shanghai, China
wangzheng@ieee.org
[4] Key Laboratory of Geographical Science in Jiangsu Province,
Nanjing Normal University, 210097 Nanjing, China
dearlollipop@sina.com.cn

Abstract. Traffic prediction model is critically important for network performance evaluation and services quality. Traditional traffic prediction models cannot reflect the characteristics of self-similar traffic. Current long-range prediction models, however, are too complex to be used as online traffic predictors. This paper presents two new traffic predictors which are MMSEP and NMSEP. They are based on minimum mean square error. Time series and control theory are used to build the mathematic models. By modifying the way of calculating the predicted error, MMESP and NMSEP can reflect the burst of self-similar traffic in multiple timescales. When compared with FARIMA model which is one of the best fractional predictor, numerical results of experiments show that MMSEP and NMSEP can achieve accuracy with less than 5% of errors while keeping simplify in computation and low memory used.

1 Introduction

Traffic prediction models are significant important in many domains, including network performance evaluation [1], buffer management [2], congestion control [3], [4], wireless network [5], and bandwidth allocation [6]. One of the key issues in measurement-based network control is to predict the variance of traffic in a next control time interval based on the online measurement of traffic characteristics. Our own focus in this paper is on providing online traffic prediction models for network management and for network devices to improve performance. A good traffic prediction model must be able to catch the statistical characteristic of real network traffic. Its implementation should be simple in order to get applied in online traffic prediction.

Traditional traffic prediction models are short-memory models, such as Markov process, AR, ARMA, ARIMA etc al. Recent research on network traffic points out that network traffic shows self-similarity and long range dependence [7]. Consequently, some long-range models have been put forward, including FBM (fractional Brownian

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 1253 – 1262, 2005.

motion) [8], FGN (fractional Gaussian noise) [8], FARIMA (fractional ARIMA model) [9], [10], and GARMA (generalized ARMA model) [11]. The common deficiency of those long-range models is they are complex in computation and are sensitive to parameters. Therefore, these models chiefly are used as offline traffic estimation algorithms.

In this paper, we look at the problem of self-similarity of network traffic, basing on MMSE (minimums mean square error) model and, time series analysis and control theory are used to build the network traffic predictor. We present two novel prediction models for online internet traffic prediction, namely MMSEP (minimums mean square error predictor) and NMSEP (normalize minimums mean square error predictor). These models aim to simplify computational complexity while keeping high estimated accuracy.

The rest of the paper is organized as follows: in section 2, a general traffic prediction model based on MMSE will be proposed. Based on this model, the definition of MMSEP and NMSEP will be presented. Probability constraint of these models also is described in section 2. Experiments and discussions including trace forecasting and accuracy comparison are adopted and described in section 3. Finally, concluding remarks are given in section 4.

2 Online Traffic Prediction Models Based on MMSE

Flow measurement of [12] shows that the number of end hosts pair in an hour to be as high as 1.7 million (Fixed-West) and 0.8 million (MCI). Even with aggregation, the number of flows in 1 hour in the Fix-West used by was as large as 0.5 million. The number of real network is large, therefore the traffic predictor should be simple in computation in order to keep up with the burst of network traffic and reduce the computational burden of network devices.

Study of [13] shows that the Hurst parameter refers to $0<H<1$ as the long-range indicator, and in real network trace rarely exceeds 0.85, which means that real traffic does not exhibit strong long-range dependence, therefore finite traffic information can achieve good estimated performance [1]. Ghaderi in [14] points out MMSE can be used in internet online traffic prediction. In this paper, we propose two prediction models based on MMSE, the difference of our work and [14] is that we modify the calculating method of predicted error to suit the changing trend of real network traffic, so our models can achieve better estimated accuracy.

2.1 Network Traffic Prediction Model

A prediction model for one-step-ahead prediction based on MMSE can be presented by the sum of infinite weighted sum and a random impulse:

$$\hat{X}_{t+l} = \sum_{j=l}^{\infty} \pi_j X_{t+l-j} + \alpha_{t+l} \tag{1}$$

where $\{X\}$ is a stochastic sequence and α is the random impulse. $\hat{X}$ is the predicted value and $\sum_{j=l}^{\infty} \pi_j = 1$.

Assume the best prediction is $\hat{X}_{t+l} = \pi_1^* X_{t+l-1} + \pi_2^* X_{t+l-2} + \pi_3^* X_{t+l-3} + ... + \alpha_{t+l}$, weighted parameters $\pi_1^*, \pi_2^*, ...$ are indefinite. The goal of MMSE model is to minimize the equation $E[X_{t+l} - \hat{X}_t]^2$ [15]. The relationship of real value X_{t+l} and the predicted value $\hat{X}_{t+l}$ at time $t+l$ can be written as $X_{t+l} = e_t(l) + \hat{X}_t(l)$, where $e_t(l)$ is the predicted error. We have $\alpha = e$ when l =1[15], l=1 means one-step-ahead prediction.

According to the control theory, the MMSE model can be viewed as sequence $\{X\}$ passes through linear filter and added a random impulse. The model of MMSE based on control theory is presented in figure 1.

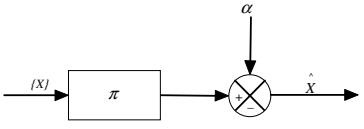


Fig. 1. The prediction model for MMSE

The deficiency of MMSE model mentioned above when applied to long-range dependent traffic is that it only considers the amendment of predicted error but the changing trend of traffic is not taken into account. In order to make up for the deficiency of MMSE, our models take the changing trend of traffic into account.

Our traffic model described by control theory can be seen in figure 2. Sequence $\{X\}$ is the arrival traffic, when it passes through filter π, we can get the weighted sum of $\{X\}$ as $\bar{X}_{t+l} = \sum_{j=1}^{k} \pi_j X_{t+l-j}$. Predicted error $e_t = X_t - \hat{X}_t$ is the input of filer f. The output of f is e^* and the prediction $\hat{X}$ is the output of φ. We can set $e_0 = 0$ and $\hat{X}_0 = X_0$ on the initial stage.

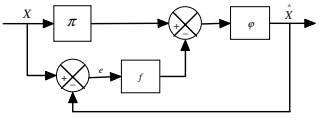


Fig. 2. The prediction model for MMSEP and NMSEP

The transfer function of figure 1 is $\hat{X} / X = (1 - f\pi) / \varphi(\pi - f)$. Based on our traffic prediction model, we present two internet traffic prediction models: MMSEP and NMSEP, which are described as following.

2.2 MMSEP Model

Consider *f* as a time-varying filter which satisfies:

$$f = \frac{e_t^3}{X_t^2} + \sum_{i=t-m}^{t} \omega_i (X_i - X_{i-1}), m = 0,1,2... \tag{2}$$

The predicted error and the changing trend of traffic are taken into account in equation 2. e^*, which is the output of f, and they can indicate the burst degree and decreasing trend of real traffic. Filter *f* ensures our model achieving good performance in self-similar environment. Our experimental results show that when $m \geq 4$, MMSEP can reflect the long-range dependence of network traffic efficiently.

Filter φ is $\varphi(\bar{X}_t, e^*) = \bar{X}_t + e^*$ and it can be written as $\varphi(\bar{X}_t, e^*) = \mathbf{W}\mathbf{Z}^T + e^*$. $\mathbf{W}$ is the weighted vector and $\mathbf{Z}$ is the vector of sequence $\{X\}$, which is $\mathbf{Z} = [X_t, X_{t-1}, ..., X_{t-k}]$. $\mathbf{W}$ can be acquired by solving the equation-- $\mathbf{W} = \mathbf{\Gamma}\mathbf{G}^{-1}$ [16]. $\mathbf{G}$ and $\mathbf{\Gamma}$ are autocorrelation matrix, which satisfies:

$$\mathbf{G} = \begin{bmatrix} \rho_0 & \rho_1 & \cdots & \rho_{k-1} \\ \rho_1 & \rho_0 & \cdots & \rho_{k-2} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \cdots & \rho_0 \end{bmatrix}, \mathbf{\Gamma} = [\rho_k, ..., \rho_1] \tag{3}$$

where $\rho_m = \frac{1}{k} \sum_{t=m+1}^{k} X_t X_{t-k}$.

2.3 NMSEP Model

MMSEP solves the weighted matrix by calculating an inverse matrix, which may weaken the practical ability of MMSEP when it is applied to some network devices, which are lack of computational abilities. In this section, we propose an approximate model for MMSE namely NMSEP, which aims to reduce the computational complexity of MMSEP.

The normalized MMSE method is an adaptive and recursive solution to compute weight vector for MMSE [16]. Based on this method, we present a new traffic prediction model namely NMSEP. NMSEP is computational simplifier than MMSEP and its practicability is improved because NMSEP does not require prior knowledge of the correlation structure of time series.

The recursive linear estimator for weight vector **W** in NMSEP is as follow:

$$\mathbf{W}_{t+1} = \mathbf{W}_t + \mu \frac{\mathbf{Z}}{\| \mathbf{Z} \|^2} e_t \tag{4}$$

where μ is the adaptation constant which determines the convergence speed. Equation4 is convergent in the mean square error sense if the adaptation μ satisfies [16]: $0 < \mu < 2$. We can let **W**=[1,0,0…] at the initial stage.

2.4 Probability Constraint of MMSEP and NMSEP

When MMSEP and NMSEP are used to as network admission schemes and the prediction value is lower than the real value, it may cause wrong decisions. For example, when reserving buffer is too small, it causes the drop of packets. In this section, we will look at the problem of confidence limits for MMSEP and NMSEP.

The variance of predicted error which is one-step-ahead at any time t is the mean of $e_{t+l}^2 = [X_{t+l} - \hat{X}_{t+l}]^2$. Assume predicted error e satisfies Gaussian distribution [17], the conditional probability distribution of X_{t+l}—$\Pr\{X_{t+l} | X_t, X_{t-1}, \ldots\}$ also satisfies Gaussian distribution, and the mean of X_{t+l} is $\hat{X}_{t+l}$ [15]. Let u be the probability whose real value is lower than predicted value, eventually, 1-u will be the probability for the condition whose real value is larger than predicted value. Let $\hat{X}_{t+l}^u$ be the sum of MMSE prediction and the offset ε, and $\hat{X}_{t+l}^u$ satisfies $\hat{X}_{t+l}^u = \hat{X}_{t+l} + \varepsilon$. The relationship between ε and u can be acquired in advance by solving $\Pr\{e_{t+l} \le \varepsilon\} = u$ where $0.5 \le u < 1$.

3 Experiments and Results

In this section, the predicted accuracy of MMSEP and NMSEP will be compared with several other prediction methods, which are: AR (6), ARIMA (10, 1, 0) and FARIMA (1, d, 1). Experiment trace is pAug.TL from Bell core [7]. pAug.TL is a typical self-similar trace and it is widely used as the experimental trace in [7], [13], [14] and [10]. The parameters of predictors are listed as table 1.

Table 1. Parameters of predictors. Model parameters come from [10] except MMESP

Model	Model Parameters
FARIMA(1,d,1)	$\theta_1 = -0.374$, $\phi_1 = -0.171$, $d = 0.294$
ARIMA(10,1,0)	$\phi_1 = -0.4736, \phi_2 = -0.4879, \phi_3 = -0.3929721, \phi_4 = -0.2892, \phi_5 = -0.2043$
	$\phi_6 = -0.2398, \phi_7 = -0.1701, \phi_8 = -0.1083, \phi_9 = -0.0693, \phi_{10} = -0.039$
AR(6)	$\phi_1 = 0.4840, \phi_2 = -0.034, \phi_3 = 0.069, \phi_4 = 0.081, \phi_5 = 0.066, \phi_6 = -0.051$
MMSEP	k=20, w_1=2.0, w_2=-1.6, w_3=-0.5, w_4=0.1
NMSEP	k=20, μ =0.618, w_1=2.0, w_2=-1.6, w_3=-0.5, w_4=0.1

3.1 Autocorrelation Functions

ACF (autocorrelation functions) can effectively reflect the accuracy of predictors. In this section, the ACF of predicted trace and real trace will be compared.

As we can see from figure 3 and figure 4, MMSEP and NMSEP get good performance both in long-range dependent and short-range dependent areas. The ACF

of MMSEP is more stable and closer to the ACF of pAug.TL than NMSEP's ACF; this is mainly because the response of NMSEP lags behind the response of MMSEP when the ACF of original trace is changing.

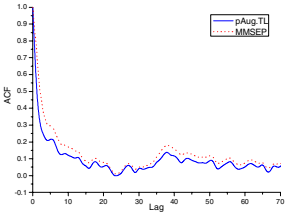


Fig. 3. Autocorrelation function of MMSEP

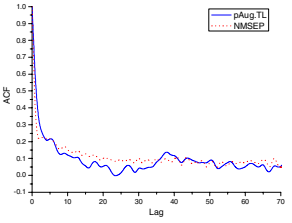


Fig. 4. Autocorrelation function of NMSEP

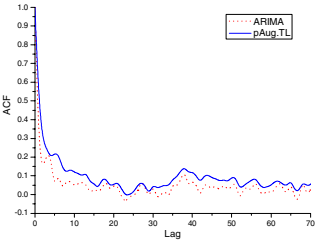


Fig. 5. Autocorrelation function of ARIMA

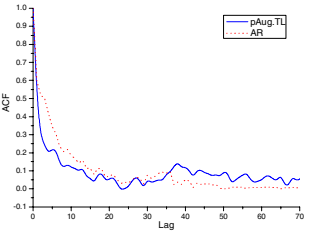


Fig. 6. Autocorrelation function of AR

According to figure 5, ARIMA has good performance in both long-range and short-rang dependent area, but its reaction for the changing of trace lags behind MMSEP. AR's estimated accuracy is lower than MMSEP and NMSEP as to figure 6, especially when it refers to long-range dependent area. In general, our traffic prediction models are more accurate than both ARIMA and AR according to the ACF.

The MSE (mean square errors) of ARIMA, AR, MMSEP and NMSEP are 0.0012174, 0.0031445, 0.0009732 and 0.0009747. As we can see from table 2, among 4 models, MMSEP, NMSEP and ARIMA can archive high accuracy. However, MMSEP and NMSEP need less parameters and more simplifier.

3.2 Traffic Forecasting

In order to verify the effectiveness of our models, MMSEP is used to fit the trace of pAug.TL in different timescales varying from 1s to 10ms. As we can see from figure 7 to figure 9, MMSEP can achieve good accuracy in different timescales, especially in large timescales.

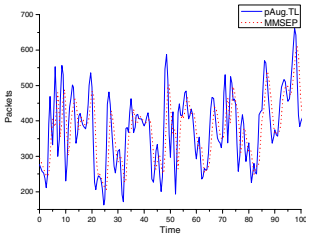


Fig. 7. Traffic forecasting uses MMSEP in 1s interval

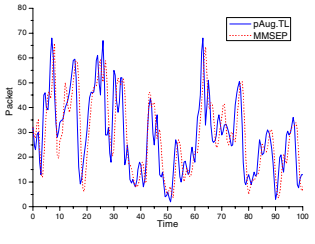


Fig. 8. Traffic forecasting uses MMSEP in 100ms interval

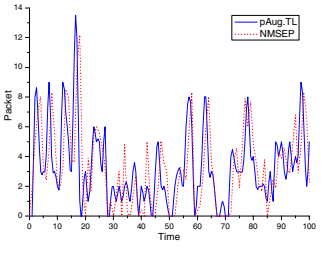


Fig. 9. Traffic forecasting uses MMSEP in 10ms interval

It is noticed that prediction models always have good performance when the traffic is smoothing. So, we believe our models can achieve better accuracy when they applied to actual network control than the results mentioned above, because the network backbone traffic exhibits low-frequency traffic variations and seldom presents strong self-similar character [1].

3.3 Prediction Accuracy

FARIMA is one of the best fractional predictors for self-similar traffic [16], [14]. The reverse of signal to noise ratio—SNR^{-1} is used as the accuracy measure to compare our models with FARIMA. SNR^{-1} satisfies $SNR^{-1} = \sum e^2 / \sum X^2$ and the smaller the SNR^{-1}, the more accurate the predictor. In this experiment, each predictor is used to forecast pAug.TL in 1s interval for one-step-ahead.

Experiment result shows that the accuracy of MMSEP and NMSEP is similar to FARIMA. We have the result list as the following

$$SNR^{-1}_{(MMSEP)} - SNR^{-1}_{(FARIMA)} \le 0.02$$
$$SNR^{-1}_{(NMSEP)} - SNR^{-1}_{(FARIMA)} \le 0.05$$

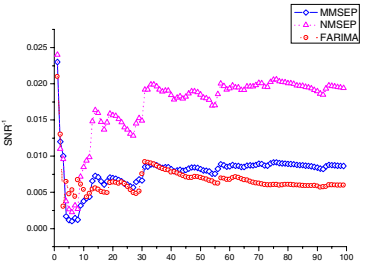


Fig. 10. SNR^{-1} comparison among MMSEP, NMSEP and FARIMA

According to our experiment, the accuracy of MMSEP and NMSEP is close to FARIMA while they are computational simplifier than FARIMA because they do not have to investigate Hurst of traffic from time to time. What's more, our models need less history data than FARIMA. When MMSEP and NMSEP are applied to network devices, they will need less physical memory than FARIMA and other long-range dependent models which are similar to FARIMA. Some studies point out that network devices have to use the cheapest memories: DRAMs because they cannot keep up with the number of flow (with or without aggregation); further, the gap between DRAM speeds (improving 79% per year) and link speeds (improving 100% per year) is only increasing [18]. Our models can achieve high accuracy even in using a small amount memory. Therefore, MMSEP and NMSEP can be easily applied to network devices with fast memories (SRAM).

4 Conclusion

Motivated by the studies show that traffic prediction is critical important for QoS and current long-rang prediction models are too computation complexity to be applied as online traffic prediction models, we propose two effective models for online traffic prediction, which are MMSEP and NMSEP. Based on the MMSE, the way of handling predicted error is amended in order to make our model suit long-range dependent traffic. Experiment results from applying our models in forecasting real trace show the suitability of our proposal for online prediction. According to our experimental results, the accuracy of MMSEP and NMSEP is close to FARIMA which is one of the best fractional predictors. But our models are much simpler than FARIMA because they do not need to identify the model parameters repeatedly. Further, our models use only a few memories, making them suitable for use in high speed routers.

There are some important practical implements of our study. Firstly, MMSEP and NMSEP can be used in congestion control to improve the effectiveness and accuracy of decisions. Secondly, our model can be useful for call admission control because it can reflect the changing trend of network traffic effectively. The application of our models is not limited to the network devices, but because of the character that the distributed applications use predictions of network traffic to sustain their performance by adapting their behavior [19], our models also can be applied into improving the adaptive performance of distributed applications.

References

1. Aimin, S., Li, S. Q.: A Predictability Analysis of Network Traffic. In: IEEE INFOCOM'00, Tel Aviv, Israel (2000) 342-351
2. Amenyo, J.T., Lazar, A.A., Pacifici, G.: Proactive Cooperative Scheduling and Buffer Management for Multimedia Networks. ACM/Springer Verlag Multimedia Systems, Vol. 1. ACM/Springer Verlag (1993) 37-49
3. Jacobson, V.: Congestion Avoidance and Control. In: ACM SIGCOMM'88, Stanford, CA, USA (1988) 314-329
4. Brakmo, L. S., Peterson, L. L.: Tcp Vegas: End to End Congestion Avoidance on a Global Internet. IEEE Journal on Selected Areas in Communications, Vol. 13. (1995) 1465-1480
5. Kim, M., Noble, B.: Mobile Network Estimation. In: ACM SIGMOBILE Seventh Annual International Conference on Mobile Computing and Networking, Rome, Italy (2001) 298-309
6. Chong, S., Li, S. Q., Ghosh, J.: Predictive Dynamic Bandwidth Allocation for Efficient Transport of Real-- Time Vbr Video over Atm. IEEE Journal on Selected Areas in Communications, Vol. 13. (1995) 12-23
7. Lend, W. E., Taqqu, M., Willinger, W. *Et Al.*: On the Self-Similar Nature of Ethernet Traffic (Extended Version). IEEE/ACM Transactions on Networking, Vol. 2. (1994) 1-15
8. Norros, I.: A Storage Model with Self-Similar Input. Queueing Systems, Vol. 16. (1994) 387-396
9. Granger, C. W. J., Joyexu, R.: An Introduction to Long-Memory Time Series Models and Fractional Differencing. Journal of Time Series Analysis, Vol. 1. (1980) 15-29

10. Shu, Y., Jin, Z., Wang, J., Yang, O. W.: Prediction-Based Admission Control Using Farima Models. In: IEEE ICC'00, New Orleans, USA (2000) 1325-1329
11. Ramachandran, R., Bhethanabotla, V. R.: Generalized Autoregressive Moving Average Modeling of the Bellcore Data. In: 25th Annual IEEE Conference on Local Computer Networks, Tampa, Florida, USA (2000) 654-661
12. Fang, W., Peterson, L.: Inter-as Traffic Patterns and Their Implications. In: IEEE GLOBECOM'99, Rio de Janeiro, Brazil (1999) 1859-1868
13. Willinger, W., Taqqu, M. S., Sherman, R., Etc Al.: Self-Similarity through High-Variability: Statistical Analysis of Ethernet Lan Traffic at the Source Level. IEEE/ACM Transaction on Networking, Vol. 5. (1997) 71-86
14. Ghaderi, Majid. On the Relevance of Self-Similarity in Network Traffic Prediction. University of Waterloo. (2002)
15. George, E. P., Gwilym, M. J., Gregory, C. R.: Time Series Analysis Forecasting and Control. 2rd edn. Prentice-Hall, New York (1994)
16. Haykin, S.: Adaptive Filter Theory. Book Adaptive Filter Theory 2rd edn., edited by Editor. Prentice-Hall, New York (1991)
17. Addie, G.: Traffic Will Be More Gaussian in Future. In: Proceedings of the Australian Telecommunication Networks & Applications Conference (ATNAC'96), Melbourne, Australia (1996)
18. Cristian, E., Geogre, V.: New Directions in Traffic Measurement and Accounting Focusing on the Elephants, Ignoring the Mice. In: ACM SIGCOMM Internet Measurement Workshop, San Francisco (2001) 75-80
19. Yi, Q., Jason, S., Peter, D.: Multiscale Predictability of Network Traffic. Computer Science Department, Northwestern University. NWU-CS-02-13 (2002)

Mobile Code Security on Destination Platform*

Changzheng Zhu, Zhaolin Yin, and Aijuan Zhang

College of Computer Science and Technology,
China University of Mining and Technology, XuZhou, JiangSu, China
ZIP: 221008
cz_z@163.com, {zhlyin, zaj}@cumt.edu.cn

Abstract. There are some security threats when mobile agent codes are loaded or running on destination host platforms. Those threats are detecting, draining or altering the agent's intention. In order to protect the code from these attacks, we discuss some measures to assure the original code security with agent code obfuscation, encryption, self-defining classloaders and rebuilding the JVM system classloader in this paper.

1 Introduction

The application of mobile agent technology in communication industry is compelling in recent years. Yet the security problem of mobile agent technology always disturbs its development. The lifecycle of a mobile agent can roughly be divided into transferring and running phases where the agent will be faced with security threats. This paper puts emphasis on discussing how to prevent mobile agent codes' intention from being detected, drained or altered on the destination platform in running phase.

Mobile agent exposes its code, status and data to the destination host agent platform that it is transferred to. Since a mobile agent can be running on any platform of different security domain, it is necessary to take some measures to ensure the integrity of a mobile agent's code, status and data, which means to protect the agent's code, status and data from being detected, drained or altered by some vicious agent platforms.

Code obfuscation is a kind of technology to prevent agent platform from detecting, draining or altering the agent's code by increasing the difficulty of comprehending the decompiled mobile agents' source code. The other aspect of this paper is to discuss how to improve the effect of code obfuscation.

Another kind of technology to prevent agent platform from detecting, draining or altering the agent's code is mobile agent code encryption. Most mobile agent codes are written in Java these days. So a user-defined classloader is going to be used to load the encrypted agent code to run on the destination platform. The user-defined classloader is also loaded by the system classloader of JVM, if the decryption algorithm is cracked, the agent's intention is still going to be detected, unless the user-defined classloader is encrypted, too. But the current JVM system classloader cannot load an encrypted user-defined classloader. So, to make the confidential code running in encrypted state, the system classloader of JVM must be rebuilt to be capable of loading either a normal class or an encrypted user-defined classloader.

* Funded by CUMT Scientfic Research Fundation

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 1263 – 1270, 2005.

Apparently, the combination of code obfuscation and mobile agent code encryption is also a kind of technology to prevent agent platform from detecting, draining or altering the agent's code.

2 Code Obfuscation

Mobile agent codes are mostly platform independent mid-state byte codes, transferred in the same form. There are tools to decompile these mid byte code to acquire the intention and information about the agent client. Vicious agent platforms or network clients can capture and decompile these byte codes that may be stored in files, buffers or network streams. In order to hold back altering mobile agent in limited time, code obfuscation technology can be used to transform the compiled byte code into a form that can only be recognized by the Virtual Machine to reduce the readability of the decompiled code, thus, to make it more difficult for illegal users to acquire the agent's intention.

Code obfuscation transforms code in an undetectable way. The execution of transformed code makes no difference to JVM. But it will be more difficult for illegal users to understand the program. For example, given a series of obfuscation transform function set T={T[1], …, T[n]}, and program P contains objects, classes, methods and variable declarations denoted as {S[1], ..., S[k]}, can be transformed into Q={R[1], …, R[k]} by expression Q={…, R[j]=T[i](S[j]), …}. The transformed program Q must have the conditions hereinafter:

1. Program Q must have the same function as program P, which means the transformed code must completely keep the semantics of original program.
2. Program Q must be obfuscated strongly to make it much more difficult to decompile or comprehend the intention of program Q than to achieve the same object directly with original program P.
3. The execution efficiency of program Q must be improved as high as possible. It cannot cost too much running time and storage pace to process obfuscation.

Code obfuscation methods differ from each other, although most of them process the obfuscation based on accidence. The typical process is to replace classes' names, methods' name, packages' names, objects' names and other identifiers in the program, even to rename identifiers of the programs in the whole project. Although the attacker may be confused by this kind of obfuscation process, they can still surmise the real purpose of the program with careful observations and tests. In order to perfect the code obfuscation technology, here to introduce obfuscation transform based on control and obfuscation transform based on data.

2.1 Obfuscation Transform Based on Control

The essential of obfuscation transform based on control is to hide the significations of predications, which means it will be difficult for the attacker to surmise the possible result of a predication after it is transformed. For example, if the result of predication Pr is always False, define it as Pr1. Also, define it as Pr2 if its result is always True, or Pr3 if its result is uncertain.

When the obscure predications are defined, the original program flow can be disrupted through obfuscation transform based on control. In figure 1, predication Pr2 is inserted into original program blocks AB(the designer knows the result of Pr2 is True, but the attacker cannot analyze it easily). As viewed from the attacker, program B seems to be executed only some conditions are met. In figure 1(b), program B is transformed into two different versions -- B and B1(B1 can be the result of obfuscation process based on accidence) while predication Pr3 is inserted after program A. The program flow goes to B when the result of Pr3 is True, B1 when it's False. Techniques like above mentioned can be taken to confuse the attacker.

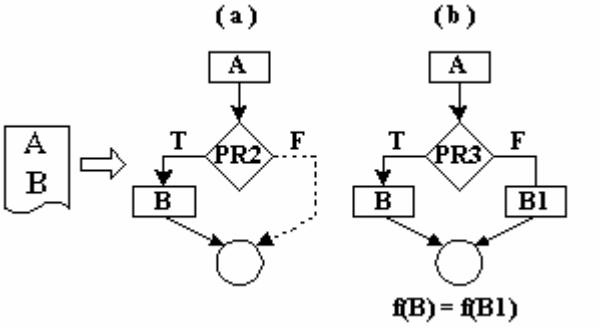


Fig. 1. Sketch map of control transformation with predication inserted

2.2 Obfuscation Transform Based on Data

Data can also be transformed when code obfuscation transformation is processed. The basic idea is to split one variable into several variables with the program being

(a)

g(V) p	g(V) q	f(p,q) V	2p+q
0	0	false	0
0	1	true	1
1	0	true	2
1	1	false	3

(b)

AND[A,B]	A=0	A=1	A=2	A=3
B=0	3	0	0	0
B=1	3	1	2	3
B=2	0	2	1	3
B=3	3	0	0	3

(c)

```
(1) bool A,B,C;                 (1') short a1,a2,b1,b2,c1,c2;
(2) B=false;                    (2') b1=0;b2=0;
(3) C=false;          T         (3') c1=1;c2=1;
(4) C=A&B;            ⇒         (4') AND[2*a1+a2,2*b1+b2];c1=x/2;c2=x%
(5) C=A&B;                      (5') c1=(a1^a2)&(b1^b2);c2=0;
(6) if(A)…;                     (6') x=2*a1+a2;if((x==1)||(x==2))…;
(7) if(B)…;                     (7') if(b1^b2)…;
```

Fig. 2. Sketch map of data transformation with Boolean variable split

reasonably transformed to keep the same semantic function while the variable's meaning is shielded to make the program difficult to understand.

For example, a Boolean variable is split into two integers p and q as figure 2 shows. The interesting thing is the same Boolean expression can be expressed in several different ways. In figure 2, expression 2 and expression 2' look different while they both assign False value to variables, expression 4 and expression 4' perform the same function "A&B"(AND[A,B]).

3 Mobile Agent Code Encryption

Although obfuscation technology can protect code from being comprehended to a certain extent, it is not impossible for programming expert to modify the decompiler to process the obfuscated class files. It cannot only depend on obfuscation technology to ensure the source code safety in situations that have higher security requests. The encrypted mobile agent code is decrypted while it is loaded after transferred to the destination platform. The loader can be looked as a just-in-time decoder. Since the decrypted byte code will never be saved in file system, it id difficult for the attacker to get the decrypted code. Without the key, the application security completely depends on the security of encryption algorithm.

The following will discuss how to integrate the encryption and decryption into JVM.

3.1 Rebuild the System Class Loader of JVM

3.1.1 The Architecture of System Class Loader of JVM

In JVM, every single class is loaded by its class loader which is also a class to be loaded by another class loader, the first started class loader in JVM is called original class loader or system class loader. The system class loader, generally written in local language (such as C), loads classes from local file system in platform dependent ways. The system class loader is to load classes of core Java API, such as classes defined in java.* package, which are very important for JVM and runtime system to perform their functions. JVM also defines other class loaders:

- java.lang.ClassLoader: defines the necessary interfaces for class loaders, doesn't implement on how to load the byte code of a class;
- java.security.SecureClassLoader: introduced in JDK 1.2, a subclass of java.lang.ClassLoader, provides security functions, still abstractive;
- java.net.URLClassLoader: a subclass of SecureClassLoader, responsible for loading classes according to the CLASSPATH.

The system class loader loads classes of core Java API, creates multi instances of subclass (i.e. the system class loader can create several class loader objects), at least one of them is an object of URLClassLoader to be used by JVM to load necessary classes (including application classes) from classpath when the program is running. Figure 3 shows the hierarchy of different types of class loaders in Java 2.

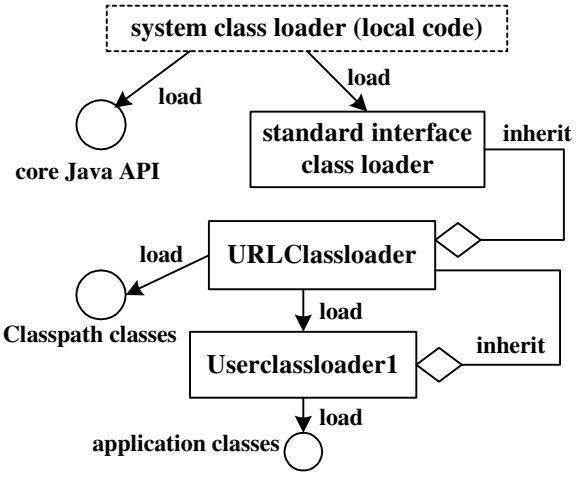


Fig. 3. The hierarchy of different types of class loaders in Java 2

3.1.2 Load the Encrypted Code

The process of loading and executing the encrypted classes on destination platform:

1. Use a certain algorithm to encrypt the class files and transform them to the destination host.
2. Use the specific user-defined class loader to load and parse the encrypted class files.

The user-defined class loader has to be encrypted to ensure it is safe. The problem is the system class loader to load the user-defined class loader doesn't support to load an encrypted class loader. So it is necessary to modify the system class loader with corresponding decryption algorithm to make the load method written in local code capable of decrypting and loading the encrypted user-defined class loader. The key to the problem is to modify the JVM class loader architecture to load either normal or encrypted class code and consequently to protect the mobile code security on destination platform.

3.2 Redefine the JVM Class Loader

Re-implement the JVM class loader as following steps:

1. Customize and encrypt the user-defined class loader MyClassLoader;
2. Modify the JVM to load either normal classes or encrypted user-defined class loader MyClassLoader dynamically;
3. Compile the modified JVM source code to generate the new executable java.exe.
4. Use the user-defined class loader to load encrypted mobile codes.

The architecture of redefined JVM is shown in figure 4.

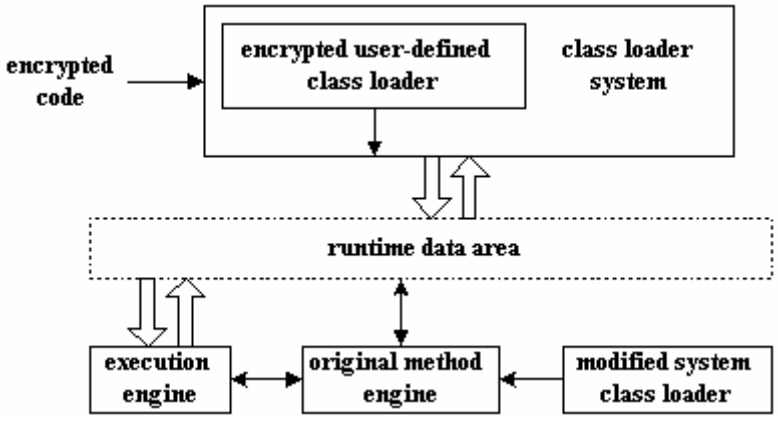


Fig. 4. The architecture of redefined JVM

3.2.1 Customize the User-Defined Class Loader

ClassLoader provides three main methods:

loadClass(String name):Class
findClass(String name):Class
defineClass(String name, byte[] b, int off, int len):Class

The process of using the user-defined class loader to load encrypted class files is shown in figure 5.

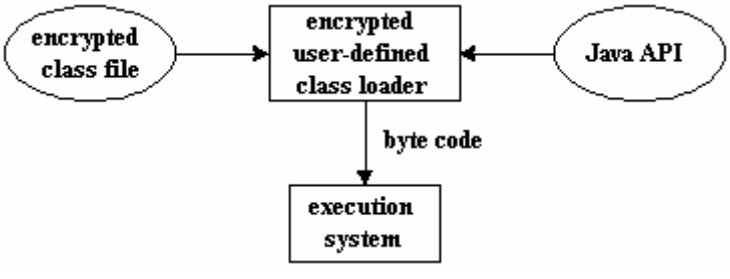


Fig. 5. The process of loading encrypted byte code

The class loader object makes programmers to be able to extend Java applications dynamically during runtime. Here it is to encrypt the user-defined class loader implemented with Java.

The functions of the user-defined class loader MyClassLoader includes reading encrypted class file from local file system, decrypting the class file, and invoking the method defineClass to return the decrypted data transformed as a Class object to method findClass( ). The following is the source code of method findClass( ):

```
public Class findClass(String name) throws
ClassNotFoundException{
  byte[] classData=null;
  classData=getdecrypteddata( );
  /* get decrypted byte code data, the implementation is
clipped*/
  if (classData != null) {
  Class=defineClass(name,classData,0,classData.length);
  }
  return x;
}
```

3.2.2 Encrypt the User-Defined Class Loader

Write an encrypting program in C to encrypt the user-defined class loader by XOR operating the file byte by byte.

3.2.3 Modify the System Class Loader of JDK and Re-compile JVM

Before the encrypted user-defined class loader gets to load an encrypted class, JVM parses and load this class loader to generate the object instance from binary code.

3.2.4 The User-Defined Class Loader Loads Encrypted Mobile Agent Code

The user-defined class loader has to invoke Reflection API to active the main( ) method of the instance after it gets the Class object of Java class, detailed steps as follows:

1. Create an object of the user-defined class loader;
2. Use the user-defined class loader object to load byte code. Specify the name of the Java class in the parameters of method loadClass( ) and invoke it. The return of this method is an object of Class, which can be used to
3. Invoke the instance's method main( ) though Reflection API.

4 Conclusion

This paper introduces some threats and solutions of detecting, draining or altering the agent's intention on destination host platform. Code obfuscation and mobile agent code encryption are the two measures to protect agent code from vicious attack discussed in this paper. The analysis and rebuilding of JVM system class loader is also presented, which make it capable of loading encrypted user-defined classes to load and execute encrypted mobile code. It is still a subject worth deeply research to protect agent code security radically.

References

1. Francisco Rodriguez Henriquez: Software Security Through Code Obfuscation. http://delta.cs.cinvestav.mx/~francisco/cripto2002/obsfuscation.ppt
2. Douglas Low: Protecting Java Code Via Code Obfuscation. http://www.acm.org/crossroads/xrds4-3/codeob.html
3. Greg Travis: Understanding the Java ClassLoader. http://www-106.ibm.com/developerworks/edu/j-dw-Javaclass-i.html

A Publicly Verifiable Authenticated Encryption Scheme with Message Linkages

Yin-Qiao Peng, Shi-Yi Xie, Yue-Feng Chen, Rui Deng, and Ling-Xi Peng

School of Information, ZhanJiang Ocean University, Zhanjiang 524088, P R China
pyinqiao@sohu.com

Abstract. In this study, an authenticated encryption scheme with public verifiability and message linkages is proposed. The new scheme requires smaller bandwidth and computational time as compared to previously proposed authenticated encryption schemes with message linkages. Furthermore, if the signer repudiates the signature, the recipient can prove the dishonesty of the signer to any verifier without disclosing the message by converting this signature into an ordinary one single.

1 Introduction

The drawback of new signature schemes with message recovery proposed by Nyberg and Ruepple is that the communication cost and computation cost are too high. Based on Nyberg and Ruepple's scheme, Horster et al.[3] using a one-way function proposed an authenticated encryption scheme with lower computation cost and communication cost. To remove the extra one-way function, Lee and Chang[4] proposed an other authenticated encryption scheme with the same lower computation but without the use of a one-way function. An authenticated encryption scheme can be regarded as the combination of data encryption scheme and digital signature scheme. In an authenticated encryption scheme, the signer may make a signature block for a message and then send it to a specified receiver, and only the receiver recover and verify the message. The advantage of an authenticated encryption scheme is that the scheme can be used for confidentiality as well as integrity. Comparison to straightforward approach employing the encryption and the signature schemes for a message separately, the authenticated encryption scheme requires smaller bandwidth of data communication for achieving privacy, integrity and authentication.

In order to recover message from an authenticated encryption scheme, the message cannot be hashed to reduce the size of the message. If the message is long, the message must be divided into a sequence message blocks, and each message block is encrypted and signed as a signature block individually, that is, if the message M is made up of the sequence $M_1, M_2,\ldots,M_t$, where each block M_i has the required length, the sender would encrypt the message by computing (r_1, s_1), $(r_2, s_2),\ldots(r_t, s_t)$[1-5]. Therefore, the mentioned authenticated encryption schemes above[1-5] have the following disadvantages: 1) The communication cost and the computation cost of the entire message are too high. 2) The recipient will not know if the message

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 1271 – 1276, 2005.

components are reordered, replicated, or partially deleted during transmission. 3) The receiver cannot prove the signer's dishonesty to anyone if the signer denies his/her signature. The second disadvantage can be remedied by adding some redundant bits to link up message blocks. However this approach will increase communication cost. To reduce the communication cost for employing the redundancy mechanism, some approaches was proposed. But these schemes still have communication cost and computation cost. Moreover, in these schemes, the receiver cannot show the dishonesty of signer to anyone if disputation occurs.

Araki et al.[6] proposed a convertible limited verifier signature scheme which can be recognized as a new type of authenticated encryption one. However, the conversion of the signature requires the signer to release one more parameter. This results in a further communication burden. More importantly, it might be unworkable if the signer is unwilling to cooperate[7].

In this study, the authors proposed a convertible authenticated encryption scheme with message linkages. The proposed scheme has the following properties:

1) The new scheme requires smaller bandwidth and computational time as compared to previously proposed authenticated encryption schemes with message linkages.
2) If the signer repudiates the signature, the recipient can prove the dishonesty of the signer to any verifier by converting this signature into an ordinary one without the cooperation of the signer.

2 The Proposed Scheme

In our scheme, the signature need only be recovered and verified by the recipient in the normal procedure. In case of later dispute, the recipient can reveal the converted signature for verifying. The converted signature is embedded in the authenticated encryption signature and thus the conversion does not require the cooperation of the signer.

The proposed scheme consists of four phases: the system initialization phase, the signature generation phase, the message recovery and verification phase, and the conversion and verification phase.

2.1 System Initialization Phase

The system authority (SA) chooses the following parameters[9]:

p: a large prime,
q: a large prime factor of (p-1),
g: a generator of order q over GF(p),
$h(.)$: a strong one-way hash function[10].

Then, SA publishes p,q,g and $h(.)$. Each user U_i owns a secret key $x_i \in Z_q^*$ and a public key $y_i = g^{x_i} \bmod p$.

2.2 Signature Generation Phase

Without loss of generality, assume that signer U_a wants to send U_b a large message M, and M is made up of the sequence M_1, M_2,…,M_n, where $1 \le M_I \le p-1$ and $1 \le I \le n$. Thus, the signer U_a carries out the following procedure to generate the signature blocks for the message M:

Step 1: Lets $r_0 = 0$ and choose a random number $k \in GF(q)$.

Step 2: Computes

$$r_i = M_i \cdot h(r_{i-1} \oplus y_b^{\,k}) \bmod p \tag{1}$$

for i=1,…,n, where "$\oplus$" denotes the exclusive operator.

Step 3: Computes

$$R = h(h(M) \| r \| g^k) \tag{2}$$

where "||" denotes the concatenation operator and $r = h(r_1 \| r_2 \| ... \| r_n)$.

Step 4: Computes

$$s = k - R \cdot x_a \bmod q \tag{3}$$

Finally, U_a then sends signature blocks (R,r,s,r_1,r_2,...,r_n) to U_b through an insecure channel where r_i is used as a linking parameter between ith and (i+1)th message blocks.

2.3 Message Recovery and Verification Phase

After receiving the set (R,r,s,r_1,r_2,...,r_n), U_b performs the verification procedure to recover the message blocks{M_1, M_2,...,M_n}.

Step 1: Compute $r' = h(r_1 \| r_2 \| ... \| r_n)$ and check if $r' = r$. If the relation holds, then do the following steps, else stop.

Step 2:Recover the message blocks {M_1, M_2,...,M_n} as follows:

$$M_i = r_i \cdot (h(r_{i-1} \oplus y_b^{\,s} (y_a^{\,x_b})^R))^{-1} \bmod p \tag{4}$$

for i=1,...,n and $r_0 = 0$.

Step 3: After recovering the message blocks {M_1, M_2,...,M_n}, U_b can firstly concatenate M=M_1||M_2||...||M_n. U_b then verifies the signature with the following equality $R = h(h(M) \| r \| g^s y_a^{\,R})$. If it holds, the signature is valid.

Later on, if the signer repudiates the signature, U_b can prove the dishonesty of the signer by revealing the parameters (R,r,s) and $M' = h(M)$. Anyone can verify the dishonesty of signer with

$$R = h(M' \| r \| g^s y_a^{\,R}) \tag{5}$$

Theorem 1. The message block $\{M_1, M_2,...,M_n\}$ can be obtained by computing $M_i = r_i \cdot (h(r_{i-1} \oplus y_b^{\,s}(y_a^{\,x_b})^R))^{-1} \bmod p$, where i= 1,....., n and $r_0 = 0$.

Proof.
$$\begin{aligned} & r_i(h(r_{i-1} \oplus y_b^{\,s}(y_a^{\,x_b})^R))^{-1} \bmod p \\ &= r_i(h(r_{i-1} \oplus y_b^{\,k-R\cdot x_a}(g^{x_a})^{R\cdot x_b}))^{-1} \bmod p && \text{(by Eq.(3))} \\ &= r_i(h(r_{i-1} \oplus y_b^{\,k-R\cdot x_a}(g^{x_b})^{R\cdot x_a}))^{-1} \bmod p \\ &= r_i(h(r_{i-1} \oplus y_b^{\,k}))^{-1} \bmod p \\ &= M_i && \text{(by Eq.(1))} \end{aligned}$$

Theorem 2. The converted signature can be verified by computing $R = h(M' \| r \| g^s y_a^R)$.

Proof.
$$\begin{aligned} & h(M' \| r \| g^s y_a^R) \\ &= h(M \| r \| g^{k-R\cdot x_a} y_a^{\,R}) && \text{(by Eq.(3))} \\ &= h(M \| r \| g^k (g^{x_a})^{-R} y_a^{\,R}) \\ &= h(M \| r \| g^k) \\ &= R && \text{(by Eq.(2))} \end{aligned}$$

3 Security Analysis

The security of the proposed scheme is based on well-known cryptographic assumptions: the intractability of reversing the one-way hash function (OWHF)[10] and solving the discrete logarithm problem(DLP)[9,11]. None of the following possible attacks against the proposed scheme can break this proposed scheme

1) An intruder tries to derive the user's secret key x_a from the corresponding public key $y_a = g^{x_a} \bmod p$. He will face the difficulty of computing the discrete logarithm problem. He also cannot derive the signer's secret key x_a from $s = k - R \cdot x_a \bmod q$, since the equation contains two unknown variables x_a and k, and k is protected under the OWHF and the DLP assumptions[12].
2) If an intruder knows one message block M_i ,the intruder might try to derive the shared key between U_a and U_b, i.e., $y_{ab}(= y_a^{\,x_b} = y_b^{\,x_a} \bmod p)$ .He first computes $h(r_{i-1} \oplus y_b^{\,k}) = M_i \cdot r_i \bmod p$. If he can obtain $y_b^{\,k}$, then y_{ab} can be derived from $y_b^{\,k} = y_b^{\,s} \cdot y_{ab}^{\,R} \bmod p$. But $y_b^{\,k}$ is protected under the one-way hash function, It is difficult to obtain.

3) If an intruder knows one message block M_i, the intruder will try to derive the other message blocks. Although he may obtain $h(r_{i-1} \oplus y_b^k) = M_i \cdot r_i \bmod p$, he cannot derive y_b^k, because y_b^k is protected under the one-way hash function. Thus, our scheme can withstand the known-plain text attack.
4) It is hard to reorder, replicate or partially delete message components during transmission. If any message component is modified, all the signature equation must be modified as well. because $r = h(r_1 \| r_2 \| ... \| r_n)$, it guarantees that all ciphertexts have not been replicated , modified or deleted.
5) An intruder tried to recover the message block M_i from the authenticated encryption signature. From the Eq. (3) and Eq.(4),we can know that the message M_i can be recovered by the one having the secret key x_a or x_b . Thus, the attack is infeasible since the secret key is protected under the DLP assumption.
6) An attacker tried to verify the signature before converted. It requires the message M to perform the signature verification of Eq.(5). From the discussion above, the attacker cannot obtain the message M before the signature is converted. Hence he cannot verify the signature.

4 Performance Analysis

We will discuss the computation cost and the communication cost. For convenience, we denote the following notations to facilitate the performance evaluation:

T_h: the time for performing a OWHF,
T_i: the time for performing a modular inverse computation,
T_m: the time for performing a modular multiplication computation,
T_e: the time for performing a modular exponentiation computation,
$|x|$: the bit-length of an integer x.

Note that the time for computation modular addition, subtraction and exclusive operation is ignored, because they are much smaller than T_h, T_i, T_m and T_e .The comparison of our scheme with previous scheme is listed in Table 1. As shown in Table 1, it is obvious that our proposed scheme is more efficient than previously proposed scheme in terms of the computation and the communication costs.

Table 1. Font sizes of headings. Table captions should always be positioned *above* the tables. The final sentence of a table caption should end without a period

cost	Hwang et al.'s scheme[8]	Lee-chang's scheme[13]	The proposed scheme
Communication cost	$n\|p\|+n\|q\|$	$n\|p\|+v\|q\|+\|h\|$	$n\|p\|+\|q\|+2\|h\|$
computation cost for signature generation	$n(T_e+T_i+T_h)+2nT_m$	$nT_e+T_h+(n+v)T_m$	$2T_e+(n+2)T_h+(n+1)T_m$
Computation cost for message recovery	$(2n+1)T_e+nT_h+3nT_m$	$(v+2)T_e+nT_i+T_h+(3n+v)T_m$	$3T_e+nT_i+(n+1)(T_h+T_m)$

5 Conclusions

In this paper, we proposed a convertible authenticated encryption scheme with message linkages. If the signer repudiates the signature, the recipient can prove the dishonesty of the signer by revealing an ordinary signature that can be verified by any verifier without the cooperation of the signer. Our proposed scheme is superior to the previously schemes.

References

1. Nyberg K.,Rueppel RA. A new signature scheme based on the DSA giving message recovery[R]. Proceeding of the First ACM Conf on computer and Communications Security, Fairfax, VA, 1993
2. Horster P, Michels M, Petersen H. Authenticated encryption schemes with low communication costs[J]. Electron letters 1994,30(15):1212-1213
3. Lee WB.,Chang CC.,Yang WP.. Authenticated encryption schemes without using a one way function[J]. Electron Letter,1995,31(19):1656-1657
4. Chen K.. Authenticated encryption schemes based on Quadratic residue[J]. Electronics Letters, 1998,34(22) :2115-2116
5. Ma CS.,Chen KF.. Publicly verifiable authenticated encryption[J]**.** Electronics Letters,2003,39(3): 281-282
6. Tseng YM.,Jan JM.. An efficient authenticated encryption scheme with message linkages and low communication costs[J]. Journal of Information Science and Engineering, 2002,18(1):41-46
7. Tseng YM.,Jan JM.. Digital signature with message recovery using self-certified public keys and its variants[J]. Applied Mathematics and Computation,2003,136(2): 203-214
8. Lee WB,Chang CC. Authenticated encryption schemes with linkage between message blocks[J]. Inform pross lett,1997,63(5):247-250
9. Araki S.,Uehara S.,Imamura K.. The limited verifier signature and its application[J]. ICICE Transactions on Fundamentals 1999,E82-A(1):63-68
10. Yang Yixian, Shen Wei, Niu Xinxin. New Theory of Modern Cryptography[M]. Beijing: Publishing House of Science,2002. 106-128
11. Li Zichen, Li Zhongxian, Yang YiXian. A new forgery attack on message recovery signatures[J]. Journal of China institute of communications,2000,21(5):84-87

Provable Security of ID-Based Proxy Signature Schemes*

Chunxiang Gu and Yuefei Zhu

Network Engineering Department, Information Engineering University, Zhengzhou, 450002, P.R. China
gcxiang5209@yahoo.com.cn

Abstract. In the last couple of years, identity based cryptography has got fruitful achievements [1,2,3,4]. This paper provides theoretical discussions for the provable-security of ID-based proxy signature primitive. First, we present a general security model for such schemes. Then we describe how to construct an ID-based proxy signature scheme with a secure ID-based signature scheme, and prove the construction's security in the standard model. At last, we analyze the ID-based proxy signature scheme proposed by Zhang et.al.[4], and show that this scheme can be proven to be secure in the random oracle model.

1 Introduction

1.1 Proxy Signature Schemes

Since Mambo, Usuda and Okamoto[5] first introduced the concept of proxy signature scheme, many new schemes have been proposed. Because of the complication of proxy signature's applications, the security of proxy signature scheme catches many people's eyes. The security requirements for proxy signature are discussed in [5,6]. That is, a secure proxy signature scheme should satisfy the following requirements: *Verifiability, Strong Unforgeability, Strong Identifiability, Strong Undeniability*, and *Prevention of misuse*.

While these requirements provide some intuition about the goals that a notion of security for proxy signature schemes should capture, their precise meanings are unclear. The fact is, with new security consideration and constructions have been proposed, old schemes have been broken. Readers can see [7] for good examples. Recently, a method called provable-security[8] has been developed and has been extensively used to support standards. Boldyreva et, al.[10] and Gu et, al.[9]use this theory to help the analysis of proxy signature primitive, and provide methods to prove the security of such schemes.

* Research supported by Found 973 (No. G1999035804), NSFC (No. 90204015, 60473021) and Elitist Youth Foundation of Henan in China (No. 021201400).

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 1277–1286, 2005.

1.2 ID-PKC and ID-Based Proxy Signature Scheme

ID-based proxy signature scheme (ID-PSS) is a special ID-based public key cryptography (ID-PKC). In 1984 Shamir[1] first proposed the idea of ID-based cryptography to simplify key management procedures of traditional certificate-based PKI. In ID-PKC, an entity's public key is derived directly from certain aspects of its identity, for example, an IP address belonging to a network host, or an e-mail address associated with a user. Private keys are generated for entities by a trusted third party called a private key generator (PKG). The direct derivation of public keys in ID-PKC eliminates the need for certificates and some of the problems associated with them.

In the last couple of years, a rapid development of ID-PKC has taken place. ID-based proxy signature schemes have also been proposed, and have found numerous practical applications. For example, they can be widely used in distributed systems, electronics transaction, and mobile agent applications.

However, the security arguments for ID-PSS have not been enough. We remark that the scheme in [4] has not been proved to get provable-security. In this article, we try to analysis this question followed the work in [9].

1.3 Contributions and Organization

This paper provides a general security model for ID-based proxy signature primitive. Then we provide a construction of ID-PSSs from any secure ID-based signature schemes, and prove its security in the standard model. At last, we analyze the ID-PSS proposed by Zhang et.al.[4], and show that the scheme can be proven to be secure in the random oracle model.

This paper is organized as follows: Some preliminary works are given in Section 2. A general security model is provided in Section 3. In Section 4, we present a construction of ID-PSSs with security proof in the standard model. Section 5 analyzes the Zhang's scheme and proves its security in the random oracle model. Finally, we conclude in Section 6.

2 Bilinear Pairing and ID-Based Signature Scheme

Let $(G_1, +)$ and $(G_2, \cdot)$ be two cyclic groups of order q for a large prime q. Let $\hat{e} : G_1 \times G_1 \to G_2$ be a map with the following properties:

1. Bilinear: $\forall P, Q \in G_1, \forall \alpha, \beta \in Z_q, \hat{e}(\alpha P, \beta Q) = \hat{e}(P, Q)^{\alpha\beta}$;
2. Non-degenerate: If P is a generator of G_1, then $\hat{e}(P, P)$ is a generator of G_2;
3. Computable: There is an efficient algorithm to compute $\hat{e}(P, Q)$ for any $P, Q \in G_1$.

Such a bilinear map is called an *admissible* bilinear map. Let P be a generator of G_1, and $a, b, c \in F_q$. We are interested in the following mathematical problems:

1. Decisional Diffie-Hellman problem(DDHP). Given (P, aP, bP, cP), decide whether $c = ab \mod q$.

2. Computation Diffie-Hellman problem(CDHP). Given (P, aP, bP), compute abP.

Generally speaking, an ID-based signature scheme consists of four polynomial-time algorithms[2]: **Setup**, **Extract**, **Sign** and **Verify**. The general known notion of security of an ID-based signature scheme is proposed by [2]. An ID-based digital signature scheme is said to be *existential unforgeable secure under adaptive chosen message and ID attacks* **(EUF-ACMIA)**, if on polynomial time adversary $\mathcal{A}$ has a non-negligible success probability in the following game:

1. A challenger $\mathcal{C}$ runs **Setup** of the scheme to generate the system parameters Ω and gives it to $\mathcal{A}$.
2. $\mathcal{A}$ can issue queries to the **Sign** oracle $S(.)$ and the **Extract** oracle $E(.)$ adaptively.
3. $\mathcal{A}$ outputs (ID, m, δ), where ID is an identity, m is a message, and δ is a signature, such that ID and (ID, m) are not equal to the inputs of any query to $E(.)$ and $S(.)$ respectively. $\mathcal{A}$ succeeds in the game if δ is a valid signature of m for ID.

3 Security Model for ID-PSS

In this paper, if there is no special statement, let A be the original signer with identity ID_A and private key d_A. He delegates his signing rights to a proxy signer B with identity ID_B and private key d_B. A warrant is used to delegate signing right.

3.1 Definition of ID-PSS

Definition 1. *An ID-based proxy signature scheme is specified by eight polynomial-time algorithms with the following functionalities.*

- **Setup:** *The parameters generation algorithm, takes as input a security parameter $k \in N$ (given as 1^k), and returns a master secret key s and system parameters Ω. This algorithm is performed by PKG.*
- **Extract:** *The private key generation algorithm, takes as input an identity $ID_U \in \{0,1\}^*$, and outputs the secret key d_U corresponding to ID_U. PKG uses this algorithm to extract the users' secret keys.*
- **Delegate:** *The proxy-designation algorithm, takes as input A's secret key d_A and a warrant m_ω, and outputs the delegation $W_{A\to B}$.*
- **DVerify:** *The designation-verification algorithm, takes as input ID_A, $W_{A\to B}$ and verifies whether $W_{A\to B}$ is a valid delegation come from A.*
- **PKgen:** *The proxy key generation algorithm, takes as input $W_{A\to B}$ and some other secret information z (for example, the secret key of the executor), and outputs a signing key d_p for proxy signature.*
- **PSign:** *The proxy signing algorithm, takes as input a proxy signing key d_p and a message $m \in \{0,1\}^*$, and outputs a proxy signature (m, δ).*

- **PVerify:** *The proxy verification algorithm, takes as input ID_A and a proxy signature (m, δ), and outputs 0 or 1. In the later case, (m, δ) is a valid proxy signature of A.*
- **ID:** *The proxy identification algorithm, takes as input a valid proxy signature (m, δ), and outputs the identity ID_B of the proxy signer.*

An ID-based proxy signature scheme should first be correct. However, in this paper, we focus all our attention on the provable-security, and do not discuss this question.

3.2 Security Model

We consider an adversary $\mathcal{A}$ which is assumed to be a probabilistic Turing machine which takes as input the global scheme parameters and a random tape.

Definition 2. *For an ID-based proxy signature scheme ID_PS. We define an experiment $Exp_{\mathcal{A}}^{ID\text{-}PS}(k)$ of adversary $\mathcal{A}$ and security parameter k as follows:*

1. *A challenger $\mathcal{C}$ runs **Setup** and gives the system parameters Ω to $\mathcal{A}$.*
2. *$C_{list} \leftarrow \phi$, $D_{list} \leftarrow \phi$, $G_{list} \leftarrow \phi$, $S_{list} \leftarrow \phi$.*
3. *Adversary $\mathcal{A}$ can make the following requests or queries adaptively.*
   - *$Extract(.)$: This oracle takes as input a user's ID_i, and returns the corresponding private key d_i. If $\mathcal{A}$ gets $d_i \leftarrow Extract(ID_i)$, let $C_{list} \leftarrow C_{list} \cup \{(ID_i, d_i)\}$.*
   - *$Delegate(.)$: This oracle takes as input the designator's identity ID and a warrant m_ω, and outputs a delegation W. If $\mathcal{A}$ gets $W \leftarrow Delegate(ID, m_\omega)$, let $D_{list} \leftarrow D_{list} \cup \{(ID, m_\omega, W)\}$.*
   - *$PKgen(.)$: This oracle takes as input the proxy signer's ID and a delegation W, and outputs a proxy signing key d_p. If $\mathcal{A}$ gets $d_p \leftarrow PKgen(ID, W)$, let $G_{list} \leftarrow G_{list} \cup \{(ID, W, d_p)\}$.*
   - *$PSign(.)$: This oracle takes as input the delegation W and message $m \in \{0,1\}^*$, and outputs a proxy signature created by the proxy signer. If $\mathcal{A}$ gets $(m, \tau) \leftarrow PSign(W, m)$, let $S_{list} \leftarrow S_{list} \cup \{(W, m, \tau)\}$.*
4. *$\mathcal{A}$ outputs (ID, m_ω, W) or (W, m, τ).*
5. *If $\mathcal{A}$'s output satisfies one of the following terms, $\mathcal{A}$'s attack is successful.*
   - *The output is (ID, m_ω, W), and satisfies: $DVerify(W, ID) = 1$, $(ID, .) \notin C_{list}$, $(ID, ., .) \notin G_{list}$ and $(ID, m_\omega, .) \notin D_{list}$. $Exp_{\mathcal{A}}^{ID\text{-}PS}(k)$ returns 1.*
   - *The output is (W, m, τ), and satisfies $PVerify((m, \tau), ID_i) = 1$, $(W, m, .) \notin S_{list}$, and $(ID_j, .) \notin C_{list}$, $(ID_j, W, .) \notin G_{list}$, where ID_i and ID_j are the identities of the designator and the proxy singer defined by W, respectively. $Exp_{\mathcal{A}}^{ID\text{-}PS}(k)$ returns 2.*

   Otherwise, $Exp_{\mathcal{A}}^{ID\text{-}PS}(k)$ returns 0.

Definition 3. *An ID-based digital signature scheme ID_PS is said to be existential delegation and signature unforgeable under adaptive chosen message and*

*ID attacks (**DS-EUF-ACMIA**), if for any polynomial time adversary $\mathcal{A}$, any polynomial $p(.)$ and big enough k,*

$$Pr[Exp_{\mathcal{A}}^{ID\_PS}(k) = 1] < \frac{1}{p(k)} \quad and \quad Pr[Exp_{\mathcal{A}}^{ID\_PS}(k) = 2] < \frac{1}{p(k)}$$

4 A Construction of ID-PSS

In this section, we present how to construct a secure ID-based proxy signature scheme from a secure ID-based signature scheme.

4.1 Description of the Construction

Let $ID\_Sign = \{Setup, Extract, Sign, Verify\}$ be an ID-based signature scheme, we can construct an ID-based proxy signature scheme $ID\_PSign = \{Setup, Extract, Delegate, DVerify, PKgen, PSign, PVerify, ID\}$, where,

- **Setup, Extract:** The two algorithms are the same as that of ID_Sign.
- **Delegate:** The original signer A generates signature on warrant m_ω and outputs the delegation $W_{A\to B} = (m_\omega, Sign(m_\omega, d_A))$.
- **DVerify:** B accepts the delegation if and only if $Verify(W_{A\to B}, ID_A) = 1$.
- **PKgen:** If B accepts $W_{A\to B}$, B sends $W_{A\to B}$ to **PKG**. If $Verify(W_{A\to B}, ID_A) = 1$, PKG extracts the secret key d_p of m_ω, and sends it to B by a secure channel as the proxy signing key on behalf of A.
- **PSign:** Let d_p be B's proxy signing key, for a message m, B computes $\tau = Sign(m_\omega \parallel m, d_p)$ and lets (m_ω, τ) be the proxy signature for m.
- **PVerify:** Given a proxy signature $(m, (m_\omega, \tau))$, a recipient first checks if the proxy signer and the message conform to m_ω. Then he verifies whether $Verify((m_\omega \parallel m, \tau), m_\omega) = 1$. If both steps succeed, the signature is a valid proxy signature on behalf of A.
- **ID:** The proxy signer's identity ID_B can be revealed by m_ω.

Here, we say that ID_PSign is an ID-based proxy signature scheme evolved from ID_Sign.

4.2 Proof of the Security

Theorem 1. *If ID_Sign is EUF-ACMIA, then ID_PSign is DS-EUF-ACMIA.*

Proof: Suppose that there is a polynomial-time adversary $\mathcal{A}$ who manages $Exp_{\mathcal{A}}^{ID\_PSign}(k)$ and gets nonzero return by un-negligible probability ε. From $\mathcal{A}$, we can construct an adversary $\mathcal{B}$ of ID_Sign under ACMIA.

1. A challenger $\mathcal{C}$ runs **Setup** and gives the system parameters Ω to $\mathcal{B}$.
2. $C_{list} \leftarrow \phi$, $D_{list} \leftarrow \phi$, $G_{list} \leftarrow \phi$, $S_{list} \leftarrow \phi$.
3. $\mathcal{B}$ gives $\mathcal{A}$ Ω and lets $\mathcal{A}$ manage $Exp_{\mathcal{A}}^{ID\_PSign}(k)$. During the execution, $\mathcal{B}$ emulates $\mathcal{A}$'s oracles as follows:

- $Extract(.)$: For input ID_i, B requests to his own $Extract(.)$ oracle, and lets the response be the reply to $\mathcal{A}$. Let $C_{list} \leftarrow C_{list} \cup \{(ID_i, d_i)\}$.
- $Delegate(.)$: For input the designator's identity ID_i and warrant m_ω, $\mathcal{B}$ requests to his own $Sign(.)$ oracle with (m_ω, ID_i). If the reply is δ, $\mathcal{B}$ lets $W = (m_\omega, \delta)$ be the reply to $\mathcal{A}$. Let $D_{list} \leftarrow D_{list} \cup \{(ID_i, m_\omega, W)\}$.
- $PKgen(.)$: For input the proxy signer's identity ID_j and $W = (m_\omega, \delta)$ with designator's identity ID_i, if $Verify(W, ID_i) \neq 1$, $\mathcal{B}$ replies with $\perp$. Otherwise, $\mathcal{B}$ requests to his $Extract(.)$ oracle with m_ω and lets the response d_p be the reply to $\mathcal{A}$. Let $G_{list} \leftarrow G_{list} \cup \{(ID_j, W, d_p)\}$.
- $PSign(.)$: For input $W = (m_\omega, \delta)$ and message m, B requests to his own $Sign(.)$ oracle with $(m_\omega \parallel m, m_\omega)$. If the response is τ, $\mathcal{B}$ lets (m_ω, τ) be the reply to $\mathcal{A}$. Let $S_{list} \leftarrow S_{list} \cup \{(W, m, \tau)\}$.

4. Let S'_{list} and E_{list} be the query&answer lists coming from $\mathcal{B}$'s $Sign(.)$ oracle and $Extract(.)$ oracle respectively during the attack.
   - If $\mathcal{A}$'s output is (ID_i, m_ω, W) and $Exp_{\mathcal{A}}^{ID\_PSign}(k) = 1$, lets $W = (m_\omega, \delta)$, $\mathcal{B}$ can output (ID_i, m_ω, δ) satisfying $Verify((m_\omega, \delta), ID_i) = 1$ and $(ID_i, m_\omega, .) \notin S'_{list}$, $(ID_i, .) \notin E_{list}$.
   - If $\mathcal{A}$'s output is (W, m, τ) with $W = (m_\omega, \delta)$ and $Exp_{\mathcal{A}}^{ID\_PSign}(k) = 2$, $\mathcal{B}$ can output $(m_\omega, m_\omega \parallel m, \tau)$ satisfying $Verify((m_\omega \parallel m, \tau), m_\omega) = 1$ and $(m_\omega \parallel m, m_\omega, .) \notin S'_{list}$, $(m_\omega, .) \notin E_{list}$.

So we can see, if $\mathcal{A}$ manages $Exp_{\mathcal{A}}^{ID\_PSign}(k)$ and gets nonzero return by an un-negligible probability ε, $\mathcal{B}$ will succeeds in his attack against ID_PSign with probability no less than ε. That is, if ID_Sign is EUF-ACMIA, then ID_PSign is DS-EUF-ACMIA.

5 The Zhang's Scheme and Its Security Proof

5.1 Description of the Scheme

We can describe the Zhang's scheme as follows:

- **Setup:** Takes as input a security parameter k, and returns a master key s and system parameters $\Omega = (G_1, G_2, q, \hat{e}, P, P_{pub}, H_1, H_2)$, where $(G_1, +)$ and $(G_2, \cdot)$ are two cyclic groups of order q, $\hat{e} : G_1 \times G_1 \rightarrow G_2$ is an admissible bilinear map, $P_{pub} = sP$, $H_1 : \{0,1\}^* \rightarrow G_1^*$ and $H_2 : \{0,1\}^* \times G_2 \rightarrow Z_q$ are hash functions.
- **Extract:** For a given identity ID_U, computes $Q_U = H_1(ID_U) \in G_1^*$, $d_U = sQ_U$. PKG returns d_U as the user's secret key. (In the following description, denote $Q_x = H_1(ID_x)$.)
- **Delegate:** For input secret key d_A and a warrant m_ω, A computes $r_A = \hat{e}(P, P)^k$, where $k \in Z_q^*$, $c_A = H_2(m_\omega, r_A)$, $U_A = c_A d_A + kP$, and outputs the delegation $W_{A \rightarrow B} = (m_\omega, r_A, U_A)$.
- **DVerify:** Once B receives $W_{A \rightarrow B} = (m_\omega, r_A, U_A)$, he computes $c = H_2(m_\omega, r_A)$. If $r_A = \hat{e}(U_A, P)(\hat{e}(Q_A, P_{pub}))^{-c}$, he accepts the delegation.

- **PKgen:** If B accepts the delegation $W_{A\to B} = (m_\omega, r_A, U_A)$, he computes the proxy signing key d_p as $d_p = H_2(m_\omega, r_A) \cdot d_B + U_A$.
- **PSign:** Let d_p be B's proxy signing key, for a message m, B chooses $k \in Z_q^*$ at random and computes $r_P = \hat{e}(P,P)^k$, $c_P = H_2(m, r_P)$, $U_P = c_P d_p + kP$, and lets $(m, \tau) = (m, r_P, U_P, m_\omega, r_A)$ be the proxy signature for m.
- **PVerify:** For a proxy signature $(m, r_P, U_P, m_\omega, r_A)$, a recipient first checks if the proxy signer and the message conform to m_w. Then he computes $c_P = H_2(m, r_P)$ and verifies whether $r_P = \hat{e}(U_P, P)(r_A \cdot \hat{e}(Q_A + Q_B, P_{pub})^{H_2(m_\omega, r_A)})^{-c_P}$. If both steps succeed, the proxy signature on behalf of A is valid.
- **ID:** The proxy signer's identity ID_B can be revealed by m_ω.

5.2 Proof of the Security

Theorem 2. *For Zhang's scheme ID_PS, if there is a polynomial-time adversary $\mathcal{A}$ who manages an $Exp_{\mathcal{A}}^{ID\text{-}PS}(k)$ and gets return 1 with un-negligible probability ε, there is an ACMIA adversary $\mathcal{B}$ succeeding in existential forgery of Hess's scheme with probability at least ε.*

Proof : From $\mathcal{A}$, we can construct an ACMIA adversary $\mathcal{B}$ of Hess's scheme, who can succeed in existential forgery with probability at least ε. The construction of adversary $\mathcal{B}$ is an analogy of that in the proof of Theorem 4.1.

Theorem 3. *Let ID_PS be a Zhang's scheme. In the random oracle mode, let $\mathcal{A}$ be a polynomial-time adversary who manages an $Exp_{\mathcal{A}}^{ID\text{-}PS}(k)$ within a time bound T, and gets return 2 by un-negligible probability ε. We denote respectively by n_{h_1},n_{h_2} and n_s the number of queries that $\mathcal{A}$ can ask to the random oracle $H_1(.)$, $H_2(.)$ and the proxy singing oracle $PSign(.)$. Assume that $\varepsilon \geq 10(n_s + 1)(n_{h_2} + n_s)n_{h_1}/q$, then there is an adversary $\mathcal{B}$ who can solve CDHP within expected time less than $120686 \cdot n_s \cdot n_{h_2} \cdot n_{h_1} \cdot T/\varepsilon$.*

To prove the theorem, we define a *generic digital signature*[8], called **GDS**, as follows:

- **Kgen:** Given a security parameter $k \in N$, generate the key pair.
  1. $(s, \Omega = (G_1, G_2, q, \hat{e}, P, P_{pub}, H_1, H_2)) \leftarrow Setup(1^k)$, where $P_{pub} = sP$. Pick randomly $Q, Q_A \in G_1^*$, and set $d_A = sQ_A, d = sQ$.
  2. Pick a random $m_\omega \in \{0,1\}^*$ and use Hess's scheme to compute the signature (m_ω, r_A, U_A) on m_ω with secret key d_A.
  3. Compute $e = H_2(m_\omega, r_A)$, $d_p = e \cdot d + U_A$.
  4. The public key is $(G_1, G_2, q, \hat{e}, P, P_{pub}, H_2, Q, Q_A, m_\omega, e, r_A)$. The private key is d_p.
- **Sign:** To sign on a message m, choose $k \in_R Z_q^*$, $r_P = \hat{e}(P,P)^k$, $c_P = H_2(m, r_P)$, $U_P = c_P \cdot d_p + kP$. Let $(m, r_P, U_P, m_\omega, r_A)$ be the signature for m.
- **Verify:** For a proxy signature $(m, r_P, U_P, m_\omega, r_A)$, a recipient computes $c_P = H_2(m, r_P)$ and verifies whether $r_P = \hat{e}(U_P, P)(r_A \cdot \hat{e}(Q_A + Q, P_{pub})^e)^{-c_P}$.

Lemma 1. *Given $(G_1, G_2, q, \hat{e}, P, P_{pub}, H_2, Q, Q_A, m_\omega, e, r_A)$, let $\xi = \hat{e}(Q_A + Q, P_{pub})^e$, the following distributions are the same.*

$$\delta = \left\{ (r,c,U) \left| \begin{array}{c} k \in_R Z_q^* \\ c \in_R Z_q \\ r = \hat{e}(P,P)^k \\ U = c \cdot d_p + k \cdot P \end{array} \right. \right\} \text{ and } \delta' = \left\{ (r,c,U) \left| \begin{array}{c} U' \in_R G_1 \\ c \in_R Z_q \\ U = U' \\ r = \hat{e}(U,P) \cdot (\xi \cdot r_A)^{-c} \\ r \neq 1 \end{array} \right. \right\}$$

Proof: First we choose a triple (α, β, γ) from the set of the signatures: let $\alpha \in G_2^*$,$\beta \in Z_q$,$\gamma \in G_1$ such that $\alpha = \hat{e}(\gamma, P)(r_A \cdot \hat{e}(Q_A + Q, P_{pub})^e)^{-\beta} \neq 1$. We then compute the probability of appearance of this triple following each distribution of probabilities:

$$Pr_\delta[(r,c,U) = (\alpha,\beta,\gamma)] = \Pr_{k \neq 0} \begin{bmatrix} \hat{e}(P,P)^k = \alpha \\ c = \beta \\ c \cdot d_p + k \cdot P = \gamma \end{bmatrix} = \frac{1}{q(q-1)}.$$

$$Pr_{\delta'}[(r,c,U) = (\alpha,\beta,\gamma)] = \Pr_{r \neq 1} \begin{bmatrix} \alpha = r = \hat{e}(U',P) \cdot (\xi \cdot r_A)^{-c} \\ c = \beta \\ U = U' = \gamma \end{bmatrix} = \frac{1}{q(q-1)}.$$

Proof of Theorem 5.2: Without any loss of generality, we may assume that for any ID, $\mathcal{A}$ queries $H_1(.)$ with ID before ID is used as (part of) an input of any query to $Extract(.)$, $Delegate(.)$, $PKgen(.)$ and $PSign(.)$, by using a simple wrapper of $\mathcal{A}$.

From the adversary $\mathcal{A}$, we can construct a probabilistic algorithm $\mathcal{B}$ such that $\mathcal{B}$ computes aQ on input of any given $P, aP, Q \in G_1^*$ as follows:

1. A challenger $\mathcal{C}$ runs $Setup(1^k)$ to generate $\Omega = (G_1, G_2, q, \hat{e}, P, P_{pub}, H_1, H_2)$ and gives Ω to $\mathcal{B}$.
2. $\mathcal{B}$ sets $P_{pub} = aP$ and $i = 1$.
3. $C_{list} \leftarrow \phi$, $D_{list} \leftarrow \phi$, $G_{list} \leftarrow \phi$, $S_{list} \leftarrow \phi$.
4. $\mathcal{B}$ picks randomly $t, 1 \leq t \leq n_{h_1}$ and $x_i \in Z_q, i = 1, 2, ...n_{h_1}$.
5. $\mathcal{B}$ gives $\mathcal{A}$ Ω and lets $\mathcal{A}$ manage $Exp_{\mathcal{A}}^{ID\text{-}PS}(k)$. During the execution, $\mathcal{B}$ emulates $\mathcal{A}$'s oracles as follows:
   - $H_1(.)$: For input ID,$\mathcal{B}$ checks if $H_1(ID)$ is defined. If not, he defines $H_1(ID) = \begin{cases} Q & i = t \\ x_i P & i \neq t \end{cases}$, and sets $ID_i \leftarrow ID, i \leftarrow i+1$. $\mathcal{B}$ returns $H_1(ID)$ to $\mathcal{A}$
   - $H_2(.)$: If $\mathcal{A}$ makes a query (m, r) to random oracle $H_2(.)$, $\mathcal{B}$ checks if $H_2(m, r)$ is defined. If not, it picks a random $c \in Z_q$, and sets $H_2(m, r) \leftarrow c$. Then he returns $H_2(m, r)$ to $\mathcal{A}$.
   - $Extract(.)$: For input ID_i, if $i = t$, then abort. Otherwise, $\mathcal{B}$ lets $d_i = x_i \cdot P_{pub}$ be the reply to $\mathcal{A}$ and sets $C_{list} \leftarrow C_{list} \cup \{(ID_i, d_i)\}$.
   - $Delegate(.)$: For input ID_i and warrant m_ω, if $i \neq t$, $\mathcal{B}$ uses $d_i = x_i P_{pub}$ as the private key to sign on m_ω with Hess's scheme[3] and gets (r_0, U_0). Otherwise, $\mathcal{B}$ simulates ID_t's proxy-designation as follow:
     - Pick randomly $U_0 \in G_1$, $c_0 \in Z_q$.
     - Compute $r_0 = \hat{e}(U_0, P)(\hat{e}(Q, P_{pub}))^{-c_0}$,

- If $\mathcal{A}$ has made the query (m_ω, r_0) to $H_2(.)$, then abort (a collision appears). Otherwise, set $H_2(m_\omega, r_0) = c_0$.

Let $W = (m_\omega, r_0, U_0)$ be the reply, and set $D_{list} \leftarrow D_{list} \cup \{(ID_i, m_\omega, W)\}$.

– $PKgen(.)$: For input proxy signer's ID_j and delegation $W = (m_\omega, r_0, U_0)$, if $j = t$, then abort. Otherwise, $\mathcal{B}$ computes $d_p = H_2(m_\omega, r_0)x_j P_{pub} + U_0$ as the reply to $\mathcal{A}$. Let $G_{list} \leftarrow G_{list} \cup \{(W, ID_j, d_p)\}$.

– $PSign(.)$: Let the input be $W = (m_\omega, r_0, U_0)$ and message m, designator's identity be ID_i and proxy signer's identity be ID_j. If $j \neq t$, $\mathcal{B}$ computes the proxy signature (r_P, U_P) on m with secret signing key $d_p = H_2(m_\omega, r_0)x_j P_{pub} + U_0$, and return $(m, \tau) = (m, r_P, U_P, m_\omega, r_0)$ as the reply. Otherwise, $\mathcal{B}$ simulate ID_t's proxy signature on behalf of ID_i as follow:

- Pick randomly $U \in G_1$, $c \in Z_q$.
- Check whether $H_2(m_\omega, r_0)$ is defined. If not, request oracle $H_2(.)$ with (m_ω, r_0). Let $H_2(m_\omega, r_0) = e$.
- Compute $r = \hat{e}(U, P)(r_0 \cdot \hat{e}(x_i P + Q, P_{pub})^e)^{-c}$.
- If $\mathcal{A}$ has made the query (m, r) to $H_2(.)$, then abort(a collision appears). Otherwise, set $H_2(m, r) = c$.
- Let $(m, \tau) = (m, r, U, m_\omega, r_0)$ be the reply of $PSign(.)$.

(Using Lemma 5.1, the simulation is indistinguishable from the real one.)
Let $S_{list} \leftarrow S_{list} \cup \{(W, m, \tau)\}$.

6. If $\mathcal{A}$'s output is $(W, m, \tau) = ((m_\omega, r_0, U_0), m, (r, U, m_\omega, r_0))$ with designator's identity ID_i and proxy signer's identity ID_j, satisfying: $PVerify((m, \tau), ID_i) = 1$, $(W, m, .) \notin S_{list}$, $(ID_j, .) \notin C_{list}$, $(ID_j, W, .) \notin G_{list}$, and $j = t$, $\mathcal{B}$ can get a forgery (m, r, U, c) of **GDS** scheme corresponding to private key $d_p = eaQ + U_0$, where $e = H_2(m_\omega, r_0)$ and $c = H_2(m, r)$.
7. If $\mathcal{B}$ have got two **GDS** signatures corresponding to private key $d_p = eaQ + U_0$: (m, r, U, c) and (m, r, U', c'), $\mathcal{B}$ can computes and outputs aQ as follow:

$$\begin{aligned} \xi_1 &\leftarrow (c - c')^{-1} \mod q \\ \xi_2 &\leftarrow e^{-1} \mod q \\ d_p &\leftarrow \xi_1 \cdot (U - U') \\ aQ &\leftarrow \xi_2 \cdot (d_p - U_0) \end{aligned}$$

Otherwise, set $H_2(m_\omega, r_0) = e$, $i = 1$, and goto step 5.

During $\mathcal{B}$'s execution, if $\mathcal{A}$ manages an $Exp_{\mathcal{A}}(k)$ and gets return 2, collisions appear with negligible probability, as mentioned in [8]. So $\mathcal{B}$'s simulations are indistinguishable from $\mathcal{A}$'s oracles. Because t is chosen randomly, $\mathcal{B}$ can output a forgery of **GDS** scheme corresponding to private key $d_p = eaQ + U_0$ within expected time T with probability ε/n_{h_1}. GDS scheme is a generic digital signature, based on the **Forking lemma**[8], $\mathcal{B}$ can produce two valid signatures (m, r, U, c) and (m, r, U', c') such that $c \neq c'$ within expected time less than $120686 \cdot n_s \cdot n_{h_2} \cdot n_{h_1} \cdot \frac{T}{\varepsilon}$. So $\mathcal{B}$ can output aQ. Thus we prove the theorem.

Theorem 4. *In the random oracle model, Zhang's scheme is DS-EUF-ACMIA under the assumption of hardness of the CDHP.*

6 Conclusion

This paper provides theoretical discussions about the provable-security of ID-based proxy signature primitive. First, we present a general security model for such schemes, which defines the security against existential delegation and signature forgery on adaptive chosen message and ID attacks (**DS-EUF-ACMIA**). Then we provide a construction of ID-based proxy signature schemes from any secure ID-based signature schemes, and prove that it is DS-EUF-ACMIA in the standard model. Although the schemes of the construction are efficient and can be proven to be secure. These schemes need PKG to extract the secret proxy signing keys. In practice, the proxy signer may want to generate the secret proxy signing key by himself. The ID-based proxy signature scheme proposed by Zhang et.al.[4] satisfies this requirement. In this article, we also analyze the security of this scheme, and show that this scheme with Hess's ID-base signature scheme used for proxy signature, can be proved to be DS-EUF-ACMIA in the random oracle model, under the assumption of hardness of the CDHP.

References

1. A. Shamir. Identity-based cryptosystems and signature schemes. In Advances in Cryptology - CRYPTO'84, volume 196 of LNCS, pages 47-53. Springer-Verlag, 1984.
2. J.C. Cha and J.H. Cheon. An identity-based signature from gap Diffie-Hellman groups. In Y. Desmedt, editor, Public Key Cryptography - PKC 2003, volume 2567 of LNCS, pages 18-30. Springer-Verlag, 2002.
3. F. Hess. Efficient identity based signature schemes based on pairings. In K. Nyberg and H. Heys, editors, Selected Areas in Cryptography 9th Annual International Workshop, SAC 2002, volume 2595 of LNCS, pages 310-324. Springer-Verlag, 2003.
4. F. Zhang and K. Kim, Efficient ID-based blind signature and proxy signature from bilinear pairings, ACISP 03, LNCS 2727, pp. 312-323, Springer-Verlag, 2003.
5. M. Mambo, K. Usuda, E. Okamoto. Proxy signatures for delegating signing operation. In: 3rd ACM Conference on Computer and Communications Security (CCS'96), pp. 48-57. New York: ACM Press, 1996.
6. B. Lee, H. Kim, and K. Kim. Strong proxy signature and its applications. In: Proc. of the 2001 Symposium on Cryptography and Information Security (SCIS'01), vol 2/2 pp. 603-608. Oiso, Japan, Jan. 23-26, 2001.
7. H.-M. Sun and B.-T. Hsieh. On the security of some proxy signature schemes. Cryptology ePrint Archive, Roport 2003/068.
8. D.Pointcheval and J.Stern. Security arguments for digital signatures and blind signatures. Journal of Cryptology, 13(3):361-369,2000.
9. Chunxiang Gu, Yuefei Zhu, Provable Security of Proxy Signature Schemes, Proceedings of 16th international conference on computer communication, pp.1059-1064. Sept. 2004, Beijing, China.
10. A. Boldyreva, A. Palacio, and B. Warinschi. Secure proxy signature schemes for delegation of signing rights. http://eprint.iacr.org/2003/096

A Practical Scheme of Merging Multiple Public Key Infrastructures in E-commerce

Heng Pan[1], JingFeng Li[1,2], YueFei Zhu[1], and DaWei Wei[2]

[1] Institute of Information Engineering, Information Engineering University, Zhengzhou, China 450002
[2] Institute of Electronic Technology, Information Engineering University, Zhengzhou, China 450004
panhengpan@hotmail.com

Abstract. With the development of E-commerce, many companies have built their own *Public Key Infrastructure* (PKI) to support various security services. When consolidation or combine of different companies happens, the multiple PKIs deployed by different companies should be reconstructed into a new one. The ordinary ways depend on using cross-certificates, and the complexity of the certificate path construction and validation is still difficult to avoid. Considering the specialties of the context in E-commerce, without using cross-certificates, our paper proposes a practical merging scheme, which is based on hierarchy structure. Making a small change of those existing PKIs, the whole merging process is quick and low-cost. Moreover, compared with using cross- certificates, the path construction is much more efficient and convenient. Additionally, in order to describe clearly, some conceptions used in our scheme are formalized.

1 Introduction

The *Public Key Infrastructure* (PKI) is an important secure technology for the E-commerce and it provides secure services for many companies [2]. In the business world, sometimes a certain great enterprises will consolidate some small companies and sometimes several small companies will combined into a great one. At that time, those multiple PKIs deployed by different companies should be merged and interoperated. So the problem of "*Merging Multiple PKI in E-commerce*" (MMPE) is significant.

Till now, the cross certificates are commonly used to resolve the merging issue such as the mesh CA model and the Bridge CA model [3][4][8]. Nevertheless, these traditional methods have some conspicuous shortcomings. For example, the more cross-certificates are used, the more complex the certificate path processing is. Moreover, since the original companies take different certificate policies (CP), after merging by the cross-certificates, it is hard to combine them into a new CP. Thus, the CP of new PKI still remains as a trouble.

Furthermore, as a particular case in E-commerce, in contrast with the ordinary merging problem, MMPE problem has some special characteristics.

X. Lu and W. Zhao (Eds.): ICCNMC 2005, LNCS 3619, pp. 1287 – 1294, 2005.

After merging, the previous company has become one of the departments in the new enterprise. Generally speaking, there are administrative levels among these departments in the new enterprise.

In spite of these, as a department in the new enterprise, every original company still remains integrity in dealing the business. Thus, the traffic in the same department is much higher than that between the different departments.

In addition, MMPE problem is usually happened in some great enterprises. That means, the new PKI will have hundreds and thousands users. The cost of MMPE also needs to be considered [2].

According to these specialties, this paper proposes a practical scheme based on hierarchy structure to solve MMPE problem. Such hierarchy model suits to the new enterprises' administrative structure. The reconstruction process is quick and low-cost. With an efficient trust path construction and validation, the new PKI performs well. Besides these, in order to describe our scheme clearly, some conceptions used in PKI are formalized.

The rest of the paper is organized as follows. In section 2, first, the definitions of some conceptions in PKI are formalized. Then, our proposed scheme is described. Section 3 discusses the certificate path construction in detail. A performance analysis about the scheme is given in Section 4. Finally, our future work is also discussed in the section of conclusion.

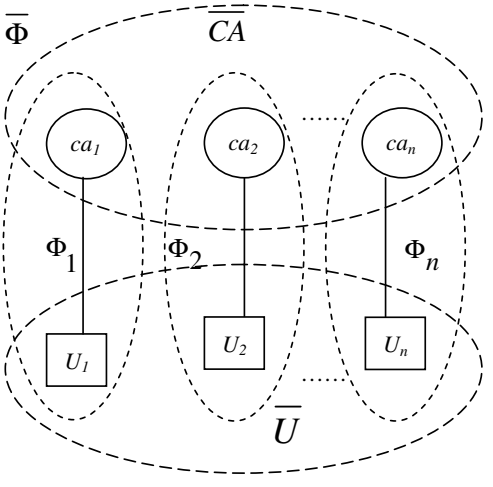


Fig. 1. The Model of Previous PKIs

2 Proposed Scheme

In this section, our proposed scheme to solve MMPE is thoroughly discussed. Compared with those common ways, the whole merging process is quick and simple.

2.1 Formalized Definition

With natural language, it is not easy to accurately describe the structure and functions of PKI. Using formal approach, the system can easily be illuminated and evaluated.

Till now, there are many ways trying to formalize PKI. An approach to the formal specification of the structure and behavior of a certificate management system was provided in [6]. In paper [7], a private key's life cycle was modeled as a finite state machine. And various kinds of trust structures were formalized in [9]. In this paper, for clarity, some concepts used in MMPE are formalized and some basic assumptions are primarily made.

Here give the assumptions: there are n ($n \geq 2$) previous PKIs deployed by different companies need to be merged together. Let every PKI consists of one certificate autho`rity ($ca_i (1 \leq i \leq n)$) and a set of users $U_i (1 \leq i \leq n)$, where $|U_i| = m, (1 \leq i \leq n)$. Every previous PKI is denoted by $\Phi_i = ca \cup U_i (1 \leq i \leq n)$. So the whole system is composed of a CA set $\overline{CA} = \{ca_1, ca_2, \cdots, ca_n\}$ and a users' set $\overline{U}$ ($\overline{U} = U_1 \cup U_2 \cup \cdots \cup U_n$, $|\overline{U}| = mn$). Let $\overline{\Phi}$ ($\overline{\Phi} = \Phi_1 \cup \Phi_2 \cup \cdots \cup \Phi_n$) be the set of all the entities in this system. Fig 1 shows the model of n previous PKIs deployed by n different companies before merging.

Definition 1. A certificate has the following form: $Cert = Cert(I, S, D, pk_s, Sig_I)$, where I is the issuer, S is the subject of the certificate, pk_s is the public key of S , D is the validity period of the certificate and Sig_I is the signature of the issuer I .

Definition 2. Let $\downarrow$ be a trust relationship over set $\overline{\Phi}$, where $a \downarrow b = \{(a,b) \mid a,b \in \overline{\Phi}, \exists Cert = cert(a,b,D,pk_b,Sig_a)\}$.

The trust relationship $\downarrow$ has the following properties: (1) If $a \downarrow b$ then b trusts a, b takes the certificates issued by a. (2) The trust relationship $\downarrow$ can be transferred, i.e., if $a \downarrow b, b \downarrow c$, then $a \downarrow c$.

Following these definitions and properties, the set $\overline{CA}$ can be described as $\overline{CA} = \{ca \mid \exists u \in \overline{U}, ca \downarrow u\}$. What we should pay attention to is that in the hierarchy trust model, the root certificate authority (rca) is a special element, which has the property of $rca \downarrow rca$.

2.2 The Merging Process of MMPE

Our proposed scheme includes two phases. In the first phase, a special certificate authority will be selected as the new *root certificate authority* (rca) from the set $\overline{CA}$. This rca should be trusted by all the elements of $\overline{\Phi}$. How to establish such a trust relationship will be described in detail. In the second phase, all the *subordinate certificate authorities* (sca) must apply to rca for their new certificates. However, these new certificates still use the previous public keys. Moreover, the users' certificates need not be changed and still be the old one. Fig.2. shows the new PKI model after merging.

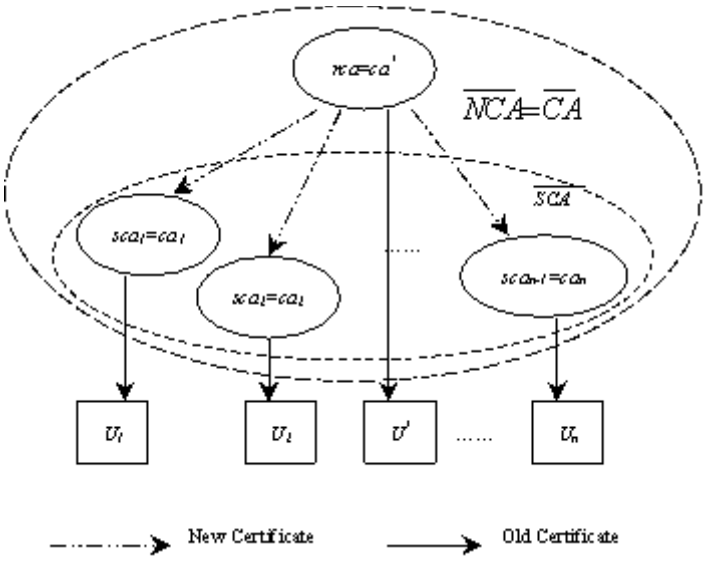


Fig. 2. The New PKI model after Merging

2.2.1 Select a New Root Certificate Authority

Normally, when MMPE happens, selecting a *rca* from the set $\overline{CA}$ is an administrative action, which needs to be decided concerning the reality of the new enterprise. As soon as the *rca* is determined, all the left elements in set $\overline{CA}$ will become the subordinate certificate authorities of the *rca* . Meanwhile, these *scas* should trust *rca* and store the certificate of *rca.* In order to make all the entities in $\overline{U}$ trust *rca*, the certificate of *rca* must be securely stored by every entity.

The main steps in this phase are as follows:

1) According to the certificate policies (CP) made by the new enterprise, a special element ca^* in set $\overline{CA}$ is selected as the new root certificate authority (denoted by *rca*). At the same time, all the left elements in set $\overline{CA}$ become the subordinate certificate authorities (denoted by $sca_i\,(1 \le i \le n-1)$) of *rca* . All these $n-1$ sca_i make up of a new set $\overline{SCA}$. However, the new certificate authorities' set is denoted by $\overline{NCA} = \{rca,\ sca_1,\ sca_2, \cdots, sca_{n-1}\}$.
2) Distribute the certificate of *rca* ($Cert_{rca} = Cert_{ca^*} = Cert(rca, rca, D, pk_{rca}, Sig_{rca})$) to every $sca_i\,(1 \le i \le n-1)$ in set $\overline{NCA}$.
3) Each $sca_i\,(1 \le i \le n-1)$ issues a temporary certificate $\overline{Cert}_{rca} = Cert\ (sca_i,\ rca,\ D,\ pk_{rca}, Sig_{sca_i})$ to *rca* . Then, the sca_i broadcasts the $Cert_{rca}$ and $\overline{Cert}_{rca}$ to every user in set U_i .

4) As soon as getting $\overline{Cert}_{rca}$, the user verifies its signature by using the corresponding public key of sca_i. If the signature is true, the user can acquire *rca*'s public key pk_{rca}.
5) The user verifies the signature of $Cert_{rca}$ using pk_{rca}. If it is true, the user will store $Cert_{rca}$ and trust *rca*.

Fig.3(a) shows these steps. Where, *Newrca( )* denotes selecting a *rca* from the set $\overline{CA}$. *Subordinate( )* makes all the elements (except the ca^*) to be the subordinate certificate authority of *rca*. *Issue( )* generates and signs a new certificate. *Apply( )* realizes the certificate application; *Distribute( )* can send the certificate to the user. *Broadcast( )* sends the certificates to a set of users at the same time. *Verify( )* determines whether the signature of a certificate is true. *Store( )* denotes store the certificates and keys securely.

```
1. ca* ← newra(CA);
   rca = ca*;
   for i=1 to n
      do{
          if (ca_i ≠ ca*)
          then  sca_i = subordinate(ca_i);
         }

2.  for i=1 to n-1
      do{
          distribute (Cert_rca, sca_i);
          store (Cert_rca, sca_i);
         }

3.  for i=1 to n-1
      do{
          Cert_rca ← issue(sca_i, rca);
   broadcast (Cert_rca, U_i);
   for j=1 to m
      do {
          verify (Cert_rca, pk_sca_i);
          store( pk_rca, U_i);
         }
   broadcast (Cert_rca, U_i);
   for j=1 to m
      do {
          verify( Cert_rca, pk_rca);
          store (Cert_rca, U_i);
         }
   }
```

Fig. 3(a)

```
for i=1 to n
  do {
      apply( sca_i, rca, pk_sca_i);
      Cert_sca_i ← issue(rca, sca_i);
     }
```

Fig. 3(b)

Fig. 3. The Merging Process of MMPE

2.2.2 $sca_i\ (1 \le i \le n-1)$ Applies to the *rca* for a New Certificate

The main steps in this phase are as follows:

1) $sca_i\ (1 \le i \le n-1)$ sends a message including its previous public key to the *rca* to apply for a new certificate:

$$sca \rightarrow rca : sca, pk_{sca_i} = pk_{ca_i}$$

2) rca creates and issues a new certificate $Cert_{sca_i} = (rca, sca_i, D, pk_{sca_i}, Sig_{rca})$ to each sca_i $(1 \le i \le n-1)$. Here we should emphasize that for all the users in set U_i, their certificates are still managed by sca_i $(1 \le i \le n)$. That is, in the new PKI, all the users still remain and use their previous certificates, which can be denoted by:

$$Cert_u = (ca_i, u, D, pk_u, Sig_{ca_i}) = Cert\ (\ sca_i, u, D, pk_u, Sig_{sca_i})(u \in U_i, 1 \le i \le n-1)$$

Fig.3 (b) illustrates these steps.

3 Certificate Path Processing in New PKI

Certificate path processing includes certificate path construction and verification. Verifying a given certificate involves verifying the identities of the certificate issuer and the certificate owner, verifying the validity date of the certificate, verifying the signature on the certificate, and verifying the certificate against the latest issuer's CRL list to make sure it has not been revoked. For this paper doesn't focus on certificate verification, such procedures are omitted. Here, we only concentrate on the certificate path construction.

From the above description, we can clearly see that the new scheme can turn from the previous PKIs smoothly. Because of every original PKI changes little, our scheme is fit for the special requirements of MMPE (see section1). That is, the certificate path processing within one department is much simpler than that in different department. Moreover, in order to ease the path processing in different department, several caches are set in the new PKI.

3.1 Certificate Path Processing Within One Department

Here, user $a \in U_i$ and $b \in U_i$ want to communicate with each other. When a gets the certificate $Cert_b$ from b, it will verify $Cert_b$. According to the definitions made in section 2.1, we can get $sca_i \downarrow a, sca_i \downarrow b$. From it, the trust path $a \to sca_i \to b$ can easily be built and the $Cert_b$ can be verified through this path.

3.2 Certificate Path Processing in Different Department

Suppose $b \in U_j$ transfers its certificate $Cert_b$ to $a \in U_i$ $(i \ne j)$ and a wants to verify the $Cert_b$. In this case, the certificate path processing is not as easy as 3.1.

To improve the efficiency of the system, a cache is set at every sca_i $(1 \le i \le n-1)$. This cache is used to store the certificate tags marking the paths that have been verified recently. Once there is a certificate need to be verified, the sca_i will first check its local cache. If the certificate has been marked in the cache, the trust path can quickly be made. Otherwise, a new trust path needs to be built step by step. For the sake of lessening storing amount, the cache needs to be updated regularly.

According to the definitions in section 2.1, the trust path $sca_i \downarrow a$, $rca \downarrow sca_i, rca \downarrow rca, rca \downarrow sca_j, sca_j \downarrow b$ can be made. Therefore, there is a trust path from a to b, that is $a \to sca_i \to rca \to sca_j \to b$. Then the $Cert_b$ will be validated

through this path. If the verification successes, a and b will trust each other. Then sca_i will store the tag of $Cert_b$ into its cache marking this path. From then on, while any $u \in U_i$ wants communicate with b, the path will become $u \to sca_i \to b$. Thereby, the cache makes the path processing much more quick and effective.

4 Performance Analysis

4.1 Discussion on the Merging Process

1) Certificate policy (CP)[11] defines the rules of certificate management and certificate applications. No matter which way will be chosen to solve MMPE, a new CP must be formulated to constrain the operations of the new PKI. As we have mentioned in section1, when using cross-certificate to solve MMPE, the new CP is difficult to make. However, our proposed scheme is depend on hierarchy structure and there is only one root CA. The distinct CPs of different previous PKIs can easily be amended and unified based on the CP of the new root CA.
2) The paper [5] provides a merging method, which also uses the hierarchy structure. According to the assumptions in section 2.1, that method needs to issue mn ($m >> n$) users' certificates. Considering the specialties of MMPE(see section1), few enterprises can afford such huge amount of certificate issuances. In addition, in mesh CA model, $n(n-1)$ cross-certificates are required to be issued. In the bridge CA model, the number of cross-certificates is $2n$. On the contrary, our proposed scheme only needs to issue n certificates to the elements in set $\overline{SCA}$.
3) The whole merging process can be regarded as establishing a new trust relationship among the elements in set $\overline{\Phi}$. Essentially, the new scheme doesn't change any existing trust relationship. It enables $u \in U_i$ to trust rca, relying on the original trust relationship between u and $sca_i (1 \le i \le n-1)$. Thus, the trust relationship can be built quickly and securely.

4.2 Discussion on the New PKI

1) The certificate management in the new PKI is relatively simple. For the user $u \in U_i$, its public key certificate is still managed by $sca_i (1 \le i \le n-1)$. On the other hand, the rca only takes charge of managing the certificates of the elements in set $\overline{SCA}$.
2) Compared with the mesh CA model and Bridge CA model, the hierarchy model has a shorter certificate path. In addition, the cache is set in every $sca_i (1 \le i \le n-1)$, which can strengthen the efficiency of the path processing.

5 Conclusions

This paper discusses the problem of merging multiple PKI when mergers or acquisitions are performed (MMPE). According to the characteristics of the MMPE, a prac-

tical scheme based on hierarchy structure is proposed. The merging process is quick and low-cost. The certificate path processing is much more simple and efficient than using cross-certificate. In addition, some conceptions used in this scheme are formalized. In our future work, the back-up CA technology will be taken into account for enhancing the security and reliability.

References

1. R. Perlman.: An Overview of PKI Trust Models, IEEE Network 11/12 ,(1999),38-43
2. A.Nash, W.Duane, C.Joseph, and D.Brink, PKI: Implementing and Managing E-security, McGraw-Hill Companies, New York, (2001)
3. S. Lloyad.: CA-CA Interoperability, White Paper, http:// www. Pki.forum.org/pdfs/ ,(2004)
4. Levi,A.: Design and Performance Evaluation of the Nested Certification Scheme and Its Application in Public Key Infrastructure, Ph.D. Thesis, Bogazici University, Dept. of Computer Engineering, (May1999).
5. S.Koga, K.Sakurai: A Merging Method of Certification Authorities without Using Cross-Certificates, Proceedings of the 18th International Conference on Advanced Information Networking and Application. AINA, IEEE Computer Society, ISBN 0-7695-2051-0/2004, Volume2,(2004),174-177
6. C. Liu, M. Henderson, and T. Cant.: A State-Based Model for Certificate Management Systems. In: PKC2000, Lecture Notes in Computer Science, Vol. 1751. Springer-Verlag, Berlin Heidelberg New York, (2000),75-92
7. A. Wiesmaier, M. Lippert, and V. Karatsiolis: The Key Authority-Secure Key Management in Hierarchical Public Key Infrastructures. In Proceedings of the International Conference on Security and Management,CSREA Press,(June 2004)89-93
8. S.Lloyd.: PKI Interoperability Framework, PKI Forum, http://www.pkiforum.org/pdfs/, (2001)
9. M.Henderson, R.Coulter, E.Dawson and E.Okamoto.: Modeling Trust for Public Key Infrastructures. In: The 7th Australasian Conference on Information Security and Privacy (ACISP 2002), Lecture Notes in Computer Science, Vol. 2348. Springer-Verlag, Berlin Heidelberg New York, (2002) ,56-70
10. .J.Lee, M.Lee and J.Gu. : New Adaptive Trust Models against DDos: Back-up CA and Mesh PKI, In: Lecture Notes in Computer Science, Vol. 2713. Springer-Verlag, Berlin Heidelberg New York, (2003),731-737
11. Stallings, W.: Cryptography and Network Security Principle and Practice, third edition, Chapter15, Prentice-Hall ,(2003)

Author Index