

LNCS 4517

Fernando Boavida  
Edmundo Monteiro  
Saverio Mascolo  
Yevgeni Koucheryavy (Eds.)

# Wired/Wireless Internet Communications

5th International Conference, WWIC 2007  
Coimbra, Portugal, May 2007  
Proceedings



Springer

*Commenced Publication in 1973*

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

## Editorial Board

David Hutchison

*Lancaster University, UK*

Takeo Kanade

*Carnegie Mellon University, Pittsburgh, PA, USA*

Josef Kittler

*University of Surrey, Guildford, UK*

Jon M. Kleinberg

*Cornell University, Ithaca, NY, USA*

Friedemann Mattern

*ETH Zurich, Switzerland*

John C. Mitchell

*Stanford University, CA, USA*

Moni Naor

*Weizmann Institute of Science, Rehovot, Israel*

Oscar Nierstrasz

*University of Bern, Switzerland*

C. Pandu Rangan

*Indian Institute of Technology, Madras, India*

Bernhard Steffen

*University of Dortmund, Germany*

Madhu Sudan

*Massachusetts Institute of Technology, MA, USA*

Demetri Terzopoulos

*University of California, Los Angeles, CA, USA*

Doug Tygar

*University of California, Berkeley, CA, USA*

Moshe Y. Vardi

*Rice University, Houston, TX, USA*

Gerhard Weikum

*Max-Planck Institute of Computer Science, Saarbruecken, Germany*

Fernando Boavida Edmundo Monteiro  
Saverio Mascolo Yevgeni Koucheryavy (Eds.)

# Wired/Wireless Internet Communications

5th International Conference, WWIC 2007  
Coimbra, Portugal, May 23-25, 2007  
Proceedings

## Volume Editors

Fernando Boavida  
Edmundo Monteiro  
University of Coimbra, Departamento de Engenharia Informatica  
Polo II, Pinhal de Marrocos, 3030-290 Coimbra, Portugal  
E-mail: {boavida, edmundo}@dei.uc.pt

Saverio Mascolo  
Politecnico di Bari, Dipartimento di Elettrotecnica ed Elettronica  
Via Orabona 4, 70125 Bari, Italy  
E-mail: saverio.mascolo@gmail.com

Yevgeni Koucheryavy  
Tampere University of Technology  
P.O. Box 553, 33101 Tampere, Finland  
E-mail: yk@cs.tut.fi

Library of Congress Control Number: 2007926907

CR Subject Classification (1998): C.2, D.4.4, D.2, H.3.5, H.4, K.6.4

LNCS Sublibrary: SL 5 – Computer Communication Networks and  
Telecommunications

ISSN 0302-9743  
ISBN-10 3-540-72694-2 Springer Berlin Heidelberg New York  
ISBN-13 978-3-540-72694-4 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media

springer.com

© Springer-Verlag Berlin Heidelberg 2007  
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India  
Printed on acid-free paper SPIN: 12068319 06/3180 5 4 3 2 1 0

# Preface

WWIC 2007 was organized by the University of Coimbra, Portugal, and it was the fifth event of a series of International Conferences on Wired/Wireless Internet Communications, addressing research topics such as the design and evaluation of protocols, the dynamics of the integration, the performance trade-offs, the need for new performance metrics, and cross-layer interactions. Previous events were held in Berne (Switzerland) in 2006, Xanthi (Greece) in 2005, Frankfurt (Germany) in 2004, and Las Vegas (USA) in 2002.

As in 2005 and 2006, WWIC was selected as the official conference by COST Action 290 (Wi-QoS-Traffic and QoS Management in Wireless Multimedia Networks).

WWIC 2007 brought together active and proficient members of the networking community, from both academia and industry, thus contributing to scientific, strategic, and practical advances in the broad and fast-evolving field of wired/wireless Internet communications.

The WWIC 2007 call for papers attracted 257 submissions from 36 different countries in Asia, Australia, Europe, North America, and South America. These were subject to thorough review work by the Program Committee members and additional reviewers. The selection process was finalized in a Technical Program Committee meeting held in Malaga, Spain, on February 15, 2007.

A high-quality selection of 32 papers, organized into 8 single-track technical sessions made up the WWIC 2007 main technical program, which covered transport layer issues, handover and QoS, traffic engineering, audio/video over IP, IEEE 802.11 WLANs, sensor networks, protocols for ad-hoc and mesh networks, and OFDM systems. The technical program was complemented by two keynote speeches, by Henning Schulzrinne (Columbia University, New York, USA) and Nitin Vaidya (University of Illinois at Urbana-Champaign, USA), on New Internet and 4G Wireless Networks, and Multi-Channel Wireless Networks, respectively.

In addition to the main technical program, the two days preceding the conference were dedicated to two workshops: the 1st ERCIM workshop on eMobility (<http://www.emobility.unibe.ch/workshop>) and the 1st WEIRD workshop on WiMax, Wireless and Mobility (<http://workshop.ist-weird.eu/>).

We wish to record our appreciation of the efforts of many people in bringing about the WWIC 2007 conference: to all the authors that submitted their papers to the conference, we regret that it was not possible to accept more papers; to the Program Committee and to all associated reviewers for their careful reviews; to our sponsors and supporting institutions; to the University of Malaga, for hosting the TPC meeting. Finally, we would like to thank all the people that helped us at the University of Coimbra and all the volunteers from the Laboratory of Communications and Telematics.

May 2007

Fernando Boavida  
Edmundo Monteiro  
Saverio Mascolo  
Yevgeni Koucheryavy

# Organization

## Executive Committee

**General Chairs** Edmundo Monteiro, University of Coimbra, Portugal  
Yevgeni Koucheryavy, Tampere University of Technology, Finland

**Program Chairs** Fernando Boavida, University of Coimbra, Portugal  
Saverio Mascolo, Polytechnic of Bari, Italy

## Local Organizing Committee

Jorge Sá Silva, University of Coimbra, Portugal  
Paulo Simões, University of Coimbra, Portugal  
Marília Curado, University of Coimbra, Portugal  
Fernando Velez, University of Beira Interior, Portugal

## Steering Committee

Torsten Braun, University of Bern, Switzerland  
Georg Carle, University of Tübingen, Germany  
Giovanni Giambene, University of Siena, Italy  
Yevgeni Koucheryavy, Tampere University of Technology, Finland  
Peter Langendoerfer, IHP Microelectronics, Germany  
Ibrahim Matta, Boston University, USA  
Vassilis Tsaooussidis, Demokritos University, Greece  
Nitín Vaidya, University of Illinois, USA

## Supporting and Sponsoring Organizations

Project COST290  
Cisco Europe  
Alvarion  
Advanced Resources  
Critical Software  
Fundação Calouste Gulbenkian  
Câmara Municipal de Coimbra  
University of Coimbra

## Program Committee

Rui Aguiar, University of Aveiro, Portugal  
Bengt Ahlgren, SICS, Sweden  
Ozgur B. Akan, Middle East Technical University, Turkey  
Khalid Al-Begain, University of Glamorgan, UK  
Manuel Alvarez-Campana, UPMadrid, Spain  
Chadi Barakat, INRIA, France  
Carlos Bernardos, Universidad Carlos III de Madrid, Spain  
Bharat Bhargava, Purdue University, USA  
Fernando Boavida, University of Coimbra, Portugal  
Torsten Braun, University of Bern, Switzerland  
Wojciech Burakowski, Warsaw University of Technology, Poland  
Maria Calderon, University Carlos III de Madrid, Spain  
Georg Carle, University of Tübingen, Germany  
Hermann de Meer, University of Passau, Germany  
Ruy de Oliveira, CEFET-MT, Brazil  
Michel Diaz, LAAS-CNRS, France  
Sonia Fahmy, Purdue University, USA  
Giovanni Giambene, University of Siena, Italy  
Geert Heijenk, University of Twente, The Netherlands  
Marc Heissenbüttel, Swisscom Mobile, Switzerland  
Markus Hofmann, Columbia University, USA  
Andreas Kessler, Karlstads University, Sweden  
Ibrahim Khalil, RMIT University, Australia  
Daniel Kofman, ENST, France  
Yevgeni Koucheryavy, Tampere University of Technology, Finland  
Rolf Kraemer, IHP Microelectronics, Germany  
Peter Langendoerfer, IHP Microelectronics, Germany  
Remco Litjens, TNO, The Netherlands  
Pascal Lorenz, University of Haute Alsace, France  
Christian Maihöfer, DaimlerChrysler, Germany  
Saverio Mascolo, Politecnico di Bari, Italy  
Paulo Mendes, DoCoMo Euro-Labs, Germany  
Ingrid Moerman, Ghent University, Belgium  
Dmitri Moltchanov, Tampere University of Technology, Finland  
Edmundo Monteiro, University of Coimbra, Portugal  
Liam Murphy, University College Dublin, Ireland  
Marc Necker, University of Stuttgart, Germany  
Guevara Noubir, Northeastern University, USA  
Philippe Owezarski, LAAS-CNRS, France  
George Pavlou, University of Surrey, UK  
George Polyzos, AUEB, Greece  
Utz Roedig, University of Lancaster, UK  
Alexandre Santos, University of Minho, Portugal  
Jochen Schiller, Free University Berlin, Germany  
Andrew Scott, Lancaster University, UK

Patrick Sénac, ENSICA, France  
 Dimitrios Serpanos, University of Patras, Greece  
 Jorge Silva, University of Coimbra, Portugal  
 Vasilios Siris, University of Crete / ICS-FORTH, Greece  
 Dirk Staehle, University of Würzburg, Germany  
 Burkhard Stiller, University of Zurich / ETH Zurich, Switzerland  
 Ivan Stojmenovic, University of Ottawa, Canada  
 Phuoc Tran-Gia, University of Wuerzburg, Germany  
 Vassilis Tsaoussidis, Demokritos University, Greece  
 J.L. van den Berg, TNO / University of Twente, The Netherlands  
 Fernando Velez, University of Beira Interior, Portugal  
 Giorgio Ventre, University of Naples, Italy  
 Thiemo Voigt, SICS, Sweden  
 Miki Yamamoto, Kansai University, Japan  
 Chi Zhang, Florida International University, USA  
 Martina Zitterbart, University of Karlsruhe, Germany

## Additional Reviewers

|                      |                      |                   |
|----------------------|----------------------|-------------------|
| Imad Aad             | Wei Chai             | Robert Henjes     |
| Janet Adams          | Roman Chertov        | Kin-Hon Ho        |
| Mifdaoui Ahlem       | Luca Cicco           | Hans-Jochim Hof   |
| Ivano Alocci         | Paolo Costa          | Richard Holzer    |
| Paul Amer            | Marilia Curado       | Tobias Hossfeld   |
| Mina Amin            | Saumitra Das         | Michael Howarth   |
| Houyou Amine         | Daniel Dietterle     | Michael Howarth   |
| Berl Andreas         | Stylianios Dimitriou | Christian Hübsch  |
| Omar Ashagi          | Roman Dunaytsev      | Bernhard Hurler   |
| Baris Atakan         | Barbara Emmert       | Talha Isik        |
| Urtzi Ayesta         | Ernesto Exposito     | Dedinski Ivan     |
| Alper Bereketli      | Sílvia Farraposo     | Aravind Iyer      |
| Giuseppe Bianchi     | Pedro Ferreira       | Jakub Jakubiak    |
| Andreas Binzenhoefer | Andreas Festag       | Oberender Jens    |
| Erik-Oliver Bläß     | John Fitzpatrick     | Ljupco Jorguseski |
| Roland Bless         | Frances Frances      | Guy Juanole       |
| Thomas Bohnert       | Bob Gaglianella      | Eleni Kamateri    |
| Nafeesa Bohra        | Thomas Gamer         | Alexander Klein   |
| Alister Burr         | Thierry Gayraud      | Gerald Koch       |
| Orlando Cabral       | Vasken Genc          | T. Kontonikolas   |
| Tiago Camilo         | Stelios Georgoulas   | Jukka Koskinen    |
| Maxweel Carmo        | Wagenknecht Gerald   | Andreas Kuntz     |
| Brian Carrig         | O. Gonzalez-Duque    | Jerome Lacan      |
| Ed Casey             | Alfredo Grieco       | Gabriela Leao     |
| Tim Casey            | Cataldo Guaragnella  | António Lebres    |
| Marcel Castro        | Karim Guennoun       | Emmanuel Lil      |
| Francisco Cercas     | Antonis Hadjiantonis | Luis Loyola       |
| Eduardo Cerqueira    | Jarmo Harju          | Andreas Maeder    |



|                    |                      |                      |
|--------------------|----------------------|----------------------|
| Maurizio Maggiora  | Augusto Neto         | Bilhanan Silverajan  |
| Damien Magoni      | Simon Oechsner       | Paulo Simoes         |
| Apostolos Malatras | Olga Ormond          | Siva Sivavakeesar    |
| Lefteris Mamatas   | Evgeni Osipov        | Rute Sofia           |
| Zoubir Mammeri     | Vittorio Palmisano   | Dirk Staehle         |
| Dimitri Marandin   | Manuel Pedro         | Mikael Sternad       |
| Brogle Marc        | Tanguy Perennou      | Werner Teich         |
| Portman Marius     | P. Papadimitriou     | Staub Thomas         |
| Anwander Markus    | Giorgos Papastergiou | Christina Thorpe     |
| Wälchli Markus     | Wuechner Patrick     | Ageliki Tsioliaridou |
| Wulff Markus       | Steffen Peter        | Francesco Vacirca    |
| Antonio Marques    | Krzysztof Piotrowski | Luc Vandendorpe      |
| Rüdiger Martin     | Rastin Pries         | Luis Veloso          |
| I. Martinez-Yelmo  | Ioannis Psaras       | Lars Völker          |
| Saverio Mascolo    | Andre Rodrigues      | Nicolas Wambeke      |
| Volkamer Melanie   | Christian Rohner     | N. Wang              |
| Abdelhamid Mellouk | Frank Roijers        | Ning Wang            |
| Michael Menth      | Sabyasachi Roy       | Ning Wang            |
| Dmitri Moltchanov  | Christos Samaras     | Wang Wenbo           |
| Andreas Müller     | Venkatesh Saranga    | Joerg Widmer         |
| Christian Müller   | Julian Satran        | Yan Wu               |
| John Murphy        | Michael Scharf       | Santiago Zapata      |
| Sean Murphy        | Isaac Seoane         | Xuan Zhong           |
| Marc Necker        | Pablo Serrano        | Andre Zimmermann     |

# Table of Contents

## Transport Layer Issues

|                                                                                                                                             |    |
|---------------------------------------------------------------------------------------------------------------------------------------------|----|
| TCP Contention Control: A Cross Layer Approach to Improve TCP Performance in Multihop Ad Hoc Networks .....                                 | 1  |
| <i>Ehsan Hamadani and Veselin Rakocevic</i>                                                                                                 |    |
| Providing Relative Service Differentiation to TCP Flows over Split-TCP Geostationary Bandwidth on Demand Satellite Networks .....           | 17 |
| <i>Wei Koong Chai, Merkourios Karaliopoulos, and George Pavlou</i>                                                                          |    |
| An Analytical Comparison of the Slow-but-Steady and Impatient Variants of TCP NewReno .....                                                 | 30 |
| <i>Roman Dunaytsev, Konstantin Avrachenkov, Yevgeni Koucheryavy, and Jarmo Harju</i>                                                        |    |
| SCTP Performance Issue on Path Delay Differential .....                                                                                     | 43 |
| <i>Yuansong Qiao, Enda Fallon, Liam Murphy, John Murphy, Austin Hanley, Xiaosong Zhu, Adrian Matthews, Eoghan Conway, and Gregory Hayes</i> |    |

## Handover and QoS

|                                                                                                           |    |
|-----------------------------------------------------------------------------------------------------------|----|
| Handover for Seamless Stream Media in Mobile IPv6 Network .....                                           | 55 |
| <i>Yi Liu, Mingxiu Li, Bo Yang, Depei Qian, and Weiguo Wu</i>                                             |    |
| A Secure Handover Protocol Design in Wireless Networks with Formal Verification .....                     | 67 |
| <i>Sun-Hee Lim, Ki-Seok Bang, Okyeon Yi, and Jongin Lim</i>                                               |    |
| Seamless Handover for Multi-user Sessions with QoS and Connectivity Support .....                         | 79 |
| <i>Eduardo Cerqueira, Luis Veloso, Paulo Mendes, and Edmundo Monteiro</i>                                 |    |
| QoS and Authentication Experiences in a Residential Environment Within a Broadband Access Framework ..... | 91 |
| <i>Iván Vidal, Francisco Valera, Jaime García, Arturo Azcorra, Vitor Pinto, and Vitor Ribeiro</i>         |    |

## QoS and Traffic Engineering

|                                                                                        |     |
|----------------------------------------------------------------------------------------|-----|
| Security and Service Quality Analysis for Cluster-Based Wireless Sensor Networks ..... | 103 |
| <i>Emrah Tomur and Y. Murat Erten</i>                                                  |     |

|                                                                                                                      |     |
|----------------------------------------------------------------------------------------------------------------------|-----|
| Admission Control for Inter-domain Real-Time Traffic Originating from Differentiated Services Stub Domains . . . . . | 115 |
| <i>Stylianos Georgoulas, George Pavlou, Panos Trimintzios, and Kin-Hon Ho</i>                                        |     |
| Fault Tolerant Scalable Support for Network Portability and Traffic Engineering . . . . .                            | 129 |
| <i>Marcelo Bagnulo, Alberto García-Martínez, and Arturo Azcorra</i>                                                  |     |
| Class-Based OSPF Traffic Engineering Inspired on Evolutionary Computation . . . . .                                  | 141 |
| <i>Pedro Sousa, Miguel Rocha, Miguel Rio, and Paulo Cortez</i>                                                       |     |

## Audio/Video over IP

|                                                                                      |     |
|--------------------------------------------------------------------------------------|-----|
| An Experimental Investigation of the Congestion Control Used by Skype VoIP . . . . . | 153 |
| <i>Luca De Cicco, Saverio Mascolo, and Vittorio Palmisano</i>                        |     |
| A Quality Adaptation Scheme for Internet Video Streams . . . . .                     | 165 |
| <i>Panagiotis Papadimitriou and Vassilis Tsaoussidis</i>                             |     |
| Performance Analysis of VoIP over HSDPA in a Multi-cell Environment . . . . .        | 177 |
| <i>Irene de Bruin, Frank Brouwer, Neill Whillans, Yusun Fu, and Youqian Xiao</i>     |     |
| Feasibility of Supporting Real-Time Traffic in DiffServ Architecture . . . .         | 189 |
| <i>Jinoo Joung</i>                                                                   |     |

## IEEE 802.11 WLANs

|                                                                                                        |     |
|--------------------------------------------------------------------------------------------------------|-----|
| Multi-rate Relaying for Performance Improvement in IEEE 802.11 WLANs . . . . .                         | 201 |
| <i>Laura Marie Feeney, Bilge Cetin, Daniel Hollos, Martin Kubisch, Seble Mengesha, and Holger Karl</i> |     |
| Performance Analysis of IEEE 802.11b Under Multiple IEEE 802.15.4 Interferences . . . . .              | 213 |
| <i>Dae Gil Yoon, Soo Young Shin, Jae Hee Park, Hong Seong Park, and Wook Hyun Kwon</i>                 |     |
| Characterization of Service Times Burstiness of IEEE 802.11 DCF . . . . .                              | 223 |
| <i>Francesco Vacirca and Andrea Baiocchi</i>                                                           |     |
| ID-Based Multiple Space Key Pre-distribution Scheme for Wireless Sensor Networks . . . . .             | 235 |
| <i>Tran Thanh Dai and Choong Seon Hong</i>                                                             |     |

## Sensor Networks and Location-Aware Systems

|                                                                                                         |     |
|---------------------------------------------------------------------------------------------------------|-----|
| Distributed Event Localization and Tracking with Wireless Sensors . . . . .                             | 247 |
| <i>Markus Wälchli, Piotr Skoczylas, Michael Meer, and Torsten Braun</i>                                 |     |
| Cross-Layer Distributed Diversity for Heterogeneous Wireless Networks . . . . .                         | 259 |
| <i>Hooman Javaheri, Guevara Noubir, and Yin Wang</i>                                                    |     |
| Location-Aware Signaling Protocol for WWAN and WLAN Interworking . . . . .                              | 271 |
| <i>SungHoon Seo, SuKyoung Lee, and JooSeok Song</i>                                                     |     |
| Performance Evaluation of Non-persistent CSMA as Anti-collision Protocol for Active RFID Tags . . . . . | 279 |
| <i>E. Egea-López, J. Vales-Alonso, A.S. Martínez-Sala, M.V. Bueno-Delgado, and J. García-Haro</i>       |     |

## Protocols for Ad Hoc and Mesh Networks

|                                                                                                                |     |
|----------------------------------------------------------------------------------------------------------------|-----|
| Multicast Overlay Spanning Tree Protocol for Ad Hoc Networks . . . . .                                         | 290 |
| <i>Georgios Rodolakis, Amina Meraihi Naimi, and Anis Laouiti</i>                                               |     |
| Detection of Packet Forwarding Misbehavior in Mobile Ad-Hoc Networks . . . . .                                 | 302 |
| <i>Oscar F. Gonzalez, Michael Howarth, and George Pavlou</i>                                                   |     |
| Reliable Geographical Multicast Routing in Vehicular Ad-Hoc Networks . . . . .                                 | 315 |
| <i>Maria Kihl, Mihail Sichitiu, Ted Ekeroth, and Michael Rozenberg</i>                                         |     |
| A Flexible Tree-Based Routing Protocol with a Mesh Relaying Node in Multi-hop Wireless Mesh Networks . . . . . | 326 |
| <i>Hyun-Ki Kim and Tae-Jin Lee</i>                                                                             |     |

## Efficient Spectrum Use

|                                                                                                                  |     |
|------------------------------------------------------------------------------------------------------------------|-----|
| An Interference-Robust Transmission Method for OFDMA Uplink . . . . .                                            | 338 |
| <i>Ki-Chang Lee and Tae-Hyun Moon</i>                                                                            |     |
| An Efficient Heterogeneous Wireless Access Based on Region and Time Partitioning in D-TDD OFDM Systems . . . . . | 346 |
| <i>Nak-Myeong Kim, Hee-Jeong Chung, Mee-Ran Kim, and Hye-In Yu</i>                                               |     |
| Cognitive Radio Based Bandwidth Sharing Among Heterogeneous OFDM Access Systems . . . . .                        | 358 |
| <i>Mee-Ran Kim, Eun-Ju Kim, Yeon-Joo Kang, and Nak-Myeong Kim</i>                                                |     |

|                                                                                |     |
|--------------------------------------------------------------------------------|-----|
| CDMA Power Control Using Channel Prediction in Mobile Fading<br>Channels ..... | 370 |
| <i>Sangho Choe</i>                                                             |     |
| <b>Author Index</b> .....                                                      | 381 |

# TCP Contention Control: A Cross Layer Approach to Improve TCP Performance in Multihop Ad Hoc Networks

Ehsan Hamadani and Veselin Rakocevic

Information Engineering Research Centre  
School of Engineering and Mathematical Sciences  
City University, London EC1V 0HB, UK  
{E.hamadani, V.rakocevic}@city.ac.uk

**Abstract.** It is well known that one of the critical sources of TCP poor performance in multihop ad hoc networks lies in the TCP window mechanism that controls the amount of traffic sent into the network. In this paper, we propose a novel cross layer solution called “TCP Contention Control” that dynamically adjusts the amount of outstanding data in the network based on the level of contention experienced by packets as well as the throughput achieved by connections. Our simulation results show TCP Contention Control can drastically improve TCP performance over 802.11 multihop ad hoc networks.

**Keywords:** Contention, Multiple ad hoc Networks, TCP Congestion Control.

## 1 Introduction

Multihop ad hoc networks are autonomous systems of mobile devices connected by wireless links without the use of any pre-existing network infrastructure or centralized administration. During recent years ad-hoc networks have attracted considerable research interest thanks to their easy deployment, maintenance and application variety. To enable seamless integration of ad hoc networks with the Internet (for instance in ubiquitous computing applications), TCP seems to be the natural choice for users of ad hoc networks that want to communicate reliably with each other and with the Internet. However, as shown in many papers (e.g. [1,2]), TCP exhibits serious performance issues such as low and unstable throughput, high end-to-end delay and high jitter. This is because most TCP parameters have been carefully optimized based on assumptions that are specific to wired networks. For instance, since bit error rates are very low in wired networks, nearly all TCP versions assume that packet losses are due to congestion and therefore invoke their congestion control mechanism in response to such losses. On the other hand, because of wireless medium characteristic and multihop nature of ad hoc networks, such networks exhibit a richer set of packet losses, including medium access contention drops, random channel errors and route failure where in practice each are required to be addressed differently. In particular, as we have shown in [3], when TCP runs over 802.11 MAC in multihop ad hoc networks, frequent channel contention losses at the MAC layer are wrongly

perceived as congestion and are recovered through TCP congestion control algorithm. This phenomenon severely degrades the performance of TCP as it leads to unnecessary TCP retransmission, unstable and low throughput, unfairness, high end-to-end delay, and high jitter. As we concluded there, a high percentage of MAC layer contention drops can be eliminated by decreasing the amount of traffic load in the network. This observation in addition to the results derived in [2,4], motivated us to propose a novel cross layer solution called “*TCP Contention Control*” that will be used in conjunction with TCP Congestion Control algorithm. In simple words, when TCP Contention and TCP Congestion Control are used together, the amount of outstanding data in the network is tuned based on the level of contention and channel utilization as well as level of congestion in the network. More precisely, while TCP Congestion Control adjusts the TCP transmission rate to avoid creating congestion in the intermediate network buffers, TCP Contention Control adjusts the transmission rate to minimize the level of unnecessary contention in the intermediate nodes. Therefore, when two algorithms are used jointly in the network, the TCP sender sets its transmission rate not merely based on the amount of congestion in the network and available buffer size at the receiver but also by the level of medium contention in intermediate nodes along the data connection. Our simulation results over a variety of scenarios confirm that the proposed scheme can dramatically improve the TCP performance in multihop networks in addition to substantial decrement in number of packet retransmission in the 802.11 link layer.

The rest of the paper is organized as follows. In section 2, we will give a brief overview of TCP congestion control algorithm. In section 3, the main problem of TCP congestion control in ad hoc networks are discussed in fine details. Then based on the drawn facts, we propose the new cross layer solution in section 4, which aims to improve TCP performance in multihop ad hoc networks. This is followed by the simulation model and the key results obtained by simulating the proposed model against the default TCP protocol in section 5. Finally, in section 6, we conclude the paper with some outlines towards future work.

## 2 TCP Congestion Control

TCP Congestion Control was added to TCP in 1987 and was standardized in RFC2001 [5] and then updated in RFC2581 [6]. In a broad sense, the goal of the congestion control mechanism is to prevent congestion in intermediate router’s buffer by dynamically limiting the amount of data sent into the network by each connection. To estimate the number of packets that can be in transit without causing congestion, TCP maintains a congestion window (*cwnd*) that is calculated by the sender as follows: when a connection starts or a timeout occurs, *slow start* is performed where at the start of this phase, the *cwnd* is set to one MSS (Maximum Segment Size). Then the *cwnd* is increased by one MSS for each acknowledgment for the new data that is received. This results in doubling the window size after each window worth of data is acknowledged. Once *cwnd* reaches a certain threshold (called the slow start threshold, *ssthresh*), the connection moves into the *congestion avoidance* phase. Ideally, a TCP connection operating in this phase puts a new packet in the network only after an old one leaves. The TCP in congestion avoidance also probes the network for resources that might have become available by continuously increasing the window, albeit at a lower rate

than in slow start. In the start of this phase, TCP gently probes the available bandwidth by increasing the *cwnd* by one packet in every round trip time (Additive Increase). During this time if the TCP detects packet loss through duplicate acknowledgments, it retransmit the packet (fast retransmit) and decreases the *cwnd* by a factor of two (Multiplicative Decrease) or it goes to slow start according to the TCP version used. Alternatively, if the sender does not receive the acknowledgment within retransmission time out (RTO), it goes to slow start and drops its window to one MSS. In both occasions, the *ssthresh* is set to half the value of *cwnd* at the time of loss.

After calculating the current value of *cwnd*, the effective limit on outstanding data (i.e. flight size), known as ‘send window’ (*swnd*), is set as the minimum of the *cwnd* and available receiver window (*rwnd*). The *rwnd* is the amount of available buffer size in the receiver side and is taken into account in order to avoid buffer overflow at the receiver by a fast sender (flow control). Therefore:

$$swnd = \min\{rwnd, cwnd\} \quad (1)$$

### 3 Problem Description

As we mentioned in section 2, the performance of TCP directly depends on the *swnd*. It is well known that the optimal value for *swnd* should be proportional to the bandwidth-delay product of the entire path of the data flow [4]. It is important to note that the excess of this threshold does not bring any additional performance enhancement, but only leads to increased buffer size in intermediate nodes along the connection. As shown in [1,7,8], the bandwidth-delay product of a TCP connection over multihop 802.11 networks tends to be very small. This is mainly because in 802.11, the number of packets in flight is limited by the per-hop acknowledgements at the MAC layer. Such property is clearly quite different from wireline networks, where multiple packets can be pushed into a pipe back-to-back without waiting for the first packet to reach the other end of the link. Therefore, as compared with that of wired networks, ad hoc networks running on top of 802.11 MAC, have much smaller bandwidth-delay product. However, as shown in [2], TCP grows its congestion window far beyond its optimal value and overestimates the available bandwidth-delay product. To get a better understanding of TCP overestimation of available bandwidth-delay product in ad hoc networks, consider a simple scenario in fig.1 where all nodes can only access their direct neighbors. Here a TCP connection is running from node A to E and all nodes have at least one packet to send in the forward direction.

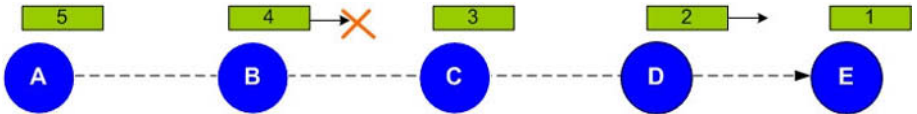


Fig. 1. 4 hop chain topology

Let us assume nodes B and D initially win the channel access and start to transmit their data into the network at the same time. Soon after both stations start transmitting their data, the packet from B to C is collided with the interference caused



by D→E transmission. Following this case, node A is very likely to win the access to the channel and starts transmitting several consecutive packets towards B before releasing the channel [9]. Meanwhile, since B is unable to access the channel it buffers the new packets in addition to packet(s) already in its buffer and starts building up its queue (figure 2).

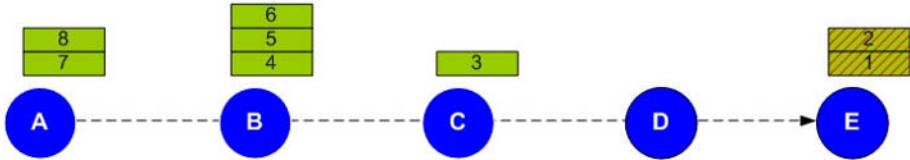


Fig. 2. Queue build up in network

This results in an artificial increase of the RTT delay measured by the sender as node B now becomes the bottleneck of the path. Such situation leads to an overestimate of the length of available data pipe and therefore an increase of the TCP congestion window and hence network overload in the next RTT. To have a better understanding of the effect of network overload on the TCP performance, fig.3 summarizes the chain of actions that occur following a network overload. In particular, increasing the network overload causes higher amount of contention among nodes as all of them try to access the channel (stage 2). On the other hand, when the level of contention goes up, more packets need to be retransmitted as the probability of collision increases with the increasing level of contention (stage 3). This in turn introduces extra network overload and therefore closing the inner part of the cycle (stage 1→stage2→stage3→stage1).

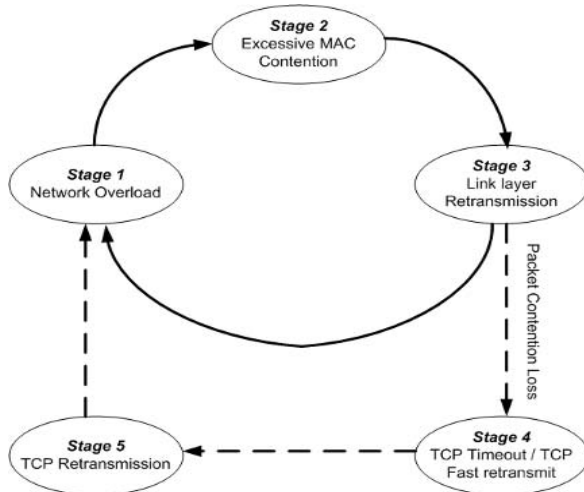


Fig. 3. TCP Instability cycle

This cycle is continued until one or more nodes cannot reach its adjacent node within a limited tries (specified by MAC\_Retry\_Limit in 802.11 MAC standard [10]) and drop the

packet (packet contention loss). This packet loss is then recovered by the TCP sender either through TCP fast retransmit or through TCP timeout (stage 4). In both cases, TCP drops its congestion window resulting in a sharp drop in number of newly injected packets to the network (stage 5) and therefore giving the network the opportunity to recover. However, soon after TCP restarts, it creates network overload again by overestimating the available bandwidth-delay product of the path, and the cycle repeats.

Fig.4 shows the change of *cwnd* and the instances of TCP retransmission in a 4 hop chain topology as shown in figure 1 using 802.11 MAC. Here, the only cause of packet drop in the network has been set to contention losses to verify the problem of TCP and link layer interaction in ad hoc networks. The results fully support the above argument and confirm that TCP behavior towards overloading the network causes extensive packet contention drops in the link layer. These packet drops are wrongly perceived as congestion by the TCP and result into false trigger of TCP congestion control algorithm and frequent TCP packet retransmissions.

This observation is also confirmed in many studies such as [1,2,11] by showing that TCP with a small congestion window (e.g., 1 or 2) tends to outperform TCP with a large congestion window in 802.11 multihop networks. To enforce the congestion window to a small value, the authors in [4] showed that the bandwidth-delay product of ad hoc networks is limited to round trip per hop count (RTHC). They then refine this upper bound based on the 802.11 MAC layer protocol, and show that in a chain topology, a tighter upper bound of approximately 1/5 of the round trip hop count of the path outperforms in comparison to default TCP. The authors in [2] impose a hard limit of 1/4 of chain length based on transmission interference in 802.11.

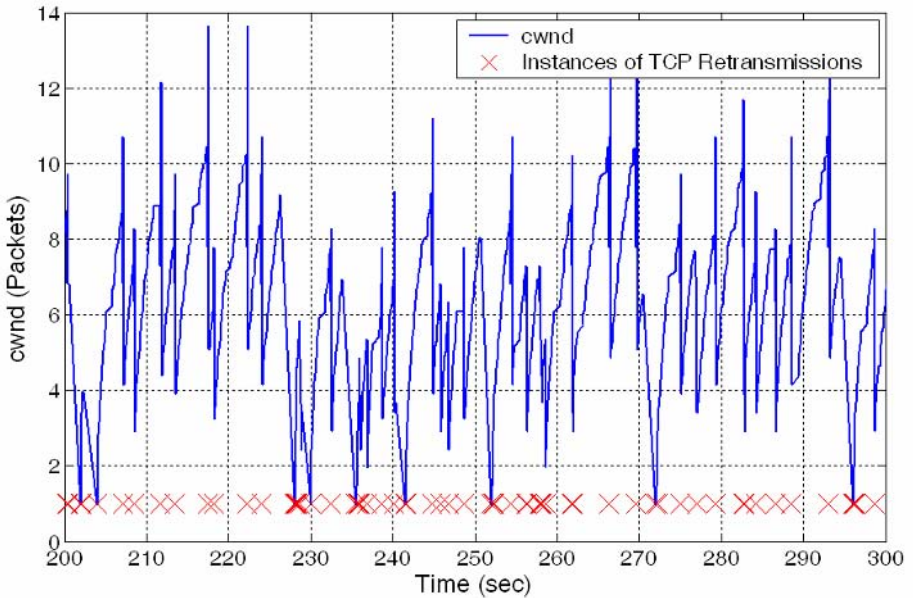


**Fig. 4.** Change of *cwnd* and the instances of TCP retransmission in a 4 hop chain topology

The main issue with all above algorithms is that they are confined to a single connection running over a chain of hops. In addition, the clamp is imposed by a sender regardless of the level of contention around intermediate nodes in the network. In the next section, we address the above issues by integrating TCP Contention Control into the default TCP and show how the proposed modification can dramatically improve TCP throughput and end-to-end delay in different topologies and flow patterns.

## 4 TCP Contention Control

To control the network overload and the consequent problems discussed in section 3, we propose a novel cross layer algorithm called TCP Contention Control which is implemented by the TCP receiver. The basic idea behind TCP Contention Control algorithm is quite simple. In each RTT, TCP Contention Control monitors the effect of changing the number of outstanding packets in the network on the achieved throughput and the level of contention delay experienced by each packet (we will shortly explain how the contention delay is measured by TCP Contention Control). Then, based on these observations, the TCP Contention Control estimates the amount of traffic that can be sent by the sender to get a balance between the maximum throughput and the minimum contention delay by each connection. To achieve this, TCP Contention Control defines a new variable called TCP\_Contention. The value of TCP\_Contention is determined according to the TCP Contention Control stages defined below:

**Fast Probe:** When a TCP connection is established, the TCP Contention Control enters the Fast Probe stage where the TCP\_Contention is increased exponentially. This is very similar to TCP slow start algorithm implemented by TCP sender to probe the available bandwidth in a short time. Thereafter, Fast Probe is generally entered after the network recovers back from Severe Contention stage explained shortly.

**Slow probe:** Slow probe is entered when the TCP Contention Control realizes that both the value of throughput and packet contention delay has decreased compared to last RTT. In this situation, the TCP Contention Control concludes the network is being underutilized and tries to gradually increase the amount of newly injected data into the network by adding one MSS to TCP\_Contention every RTT (additive increase).

**Light Contention:** If after changing the amount of injected data to the network, both the throughput and the level of packet contention delay is increased, the TCP Contention Control enters Light Contention stage. This means, despite the throughput increase during the last RTT, the network is in early stages of overload. Therefore the TCP Contention Control slowly decreases the TCP\_Contention by one MSS per RTT to control the amount of the outstanding data in the network while avoiding unnecessary reduction in the TCP throughput by implementing additive decrease.

**Severe Contention:** This stage is entered whenever the TCP Contention Control sees an increase in the level of contention delay while the achieved throughput has been decreased. This is a clear sign of network overload since it shows the push of more

data into the network has just increased the amount of contention experienced by individual packets without increasing the throughput seen by the receiver. This situation can also happen if suddenly the level of contention in the network increases (e.g. a second connection starts using the intermediate nodes). To combat this, the TCP Contention Control sets its TCP\_Contention to  $2 \times \text{MSS}$  to force the sender to minimize its transmission rate.

The pseudo code shown in fig.5, shows the detailed implementation of calculating TCP\_Contention in different stages.

```

if (DeltaThroughput >= 1)
{
  if (DeltaContention > 1)
    TCP_Contention = TCP_Contention -  $\frac{\text{MSS} \times \text{MSS}}{\text{TCP\_Contention}}$  /* Light Contention */
  else
    TCP_Contention = TCP_Contention + MSS /* Fast Probe */
}
else
{
  if (DeltaContention > 1)
    TCP_Contention =  $2 \times \text{MSS}$  /* Severe Contention */
  else
    TCP_Contention = TCP_Contention +  $\frac{\text{MSS} \times \text{MSS}}{\text{TCP\_Contention}}$  /* Slow Probe */
}

if (TCP_Contention <  $2 \times \text{MSS}$ )
  TCP_Contention =  $2 \times \text{MSS}$ 

```

**Fig. 5.** Pseudo code of calculating TCP\_Contention in different stages

As it can be seen in the code, the stages are entered depending on the value of two parameters named *DeltaThroughput* and *DeltaContention*. *DeltaThroughput* which is calculated as in formula 2, simply compares the amount of throughput received by the receiver in current RTT ( $\text{RTT}_{\text{new}}$ ) and the last RTT ( $\text{RTT}_{\text{old}}$ ).

$$\text{DeltaThroughput} = \frac{(\text{data received})_{\text{RTT}_{\text{new}}} * (\text{RTT}_{\text{old}})}{(\text{data received})_{\text{RTT}_{\text{old}}} * (\text{RTT}_{\text{new}})} \quad (2)$$

To measure the *DeltaContention*, we assume the presence of a new field, known as *ContentionDelay* in the MAC Protocol Data Unit (MPDU) that keeps the value of *Contention Delay* (CD). CD is calculated to be equal to the time from the moment the packet is placed at the beginning of buffer until it leaves the buffer for actual transmission on the link layer. Therefore, the CD does not record the queuing delay experienced by each packet. This is an important feature of contention delay as it

helps the TCP to distinguish between network congestion losses and network contention losses and therefore react properly as we explain later in this section. Then each packet alongside the connection records the CD experienced in each node and add the new CD to the ContentionDelay field. In this manner, the total contention delay experienced by each packet along the path are collected at the MAC layer and are delivered to the TCP receiver. The TCP receiver then calculates the Contention Delay per Hop (CDH) by dividing the CD by total number of hops traversed by that specific packet. Finally the receiver derives the Average Contention Delay per Hop (ACDH) by calculating the mean value of CDH received during one RTT. Having the value of ACDH, the DeltaContention is calculated as the value of ACDH in current RTT ( $ACDH_{RTT\_new}$ ) divided by the ACDH measured in last RTT ( $ACDH_{RTT\_old}$ ).

$$DeltaContention = \frac{ACDH_{RTT\_new}}{ACDH_{RTT\_old}} \quad (3)$$

We should also note that because of TCP Delayed ACK algorithm which generates an ACK every other received segment, we set the minimum TCP\_Contention to  $2 * MSS$  to make sure at least 2 segments are in the network and can trigger the transmission of TCP ACK at the receiver without waiting for maximum ACK delay timer to expire.

Having calculated the TCP\_Contention by TCP Contention Control, the important question that needs to be answered now is how we propagate the value of TCP\_Contention (which is calculated by the receiver) back to the sender. To do that, let us recall from section 2 in which we mentioned the TCP sender cannot have a number of outstanding segments larger than the rcwnd which is advertised by its own receiver. By default, the TCP receiver advertises its available receiving buffer size, in order to avoid saturation by a fast connection (flow control). We propose to extend the use of rcwnd to accommodate the value of TCP\_Contention in order to allow the receiver to limit the transmission rate of the TCP sender also when the path used by the connection exhibits a high contention and frame collision probability. Therefore, when TCP Contention Control is used, the new value of rcwnd becomes the minimum of TCP\_Contention and the available buffer size in the receiver ( $available\_receiver\_buffer$ ).

$$rcwnd = \min\{available\_receiver\_buffer, TCP\_Contention\} \quad (4)$$

It is important to note that the value of TCP\_Contention in every other RTT. In between of each change, the TCP\_Contention remains fixed to make sure the packets received by the receiver are sent into the network after the sender has applied the changes imposed by the receiver in the last RTT.

## 5 Results

### 5.1 Simulation Model

The simulations were performed using OPNET simulator [12]. The transmission range is set to 100m according to the 802.11b testbed measurements presented in [13].

In physical layer Direct Sequence Spread Spectrum (DSSS) technology with 2Mbps data rate is adopted and the channel uses free-space with no external noise. Each node has a 20 packet MAC layer buffer pool and in all scenarios, the application operates in asymptotic condition (i.e., it always has packets ready for transmission). The scheduling of packet transmission is FIFO. Nodes use DSR as the routing protocol. In transport layer, TCP NewReno flavor is deployed and the TCP advertised window is set its maximum value of 64KB so the receiver buffer size does not affect the TCP congestion window size. TCP MSS size is assumed to be fixed at 1460B. RTS/CTS message exchange is used for packets larger than 256B (therefore no RTS/CTS is done for TCP-ACK packets). The number of retransmission at MAC layer is set to 4 for packets greater than 256B (Long\_Retry\_Limit) and 7 for other packets (Short\_Retry\_Limit) as has been specified in IEEE 802.11 MAC standard. All scenarios unless otherwise stated, consist of nodes with no mobility.

## 5.2 Simulation Results

### *a) chain topology*

To evaluate the performance of TCP Contention Control, we first use a chain topology with changing the number of hops from 1 to 7. The importance of the results obtained in a chain topology is that the successive transmissions of even a single TCP flow interfere with each other as they move downstream towards the receiver as well as facing the flow of TCP acknowledgments from receiver towards sender., resulting in link-layer contention and packet drops. Therefore, as we will see in this section controlling the number of outstanding data in the network can substantially affect the amount of contention and hence the TCP performance.

In fig.6, we compare different TCP metrics in default TCP (TCP Congestion) with our proposed algorithm (TCP Congestion + TCP Contention).

As it can be seen from fig.6a and 6b the introduction of TCP Contention to TCP congestion algorithm has resulted in TCP throughput improvement and substantial decrease in number of TCP retransmissions. This is very important as it shows the TCP contention algorithm avoids false trigger of TCP congestion control in the sender and therefore decreases unnecessary TCP retransmissions. This is mainly because in default TCP, contention losses were often misinterpreted by TCP as congestion and resulted to TCP retransmission. However, when TCP Contention is added, the level of contention and hence the number of contention losses become negligible where most of them can be recovered within 802.11 retransmission recovery procedure. Therefore, very few contention losses result are recovered through TCP retransmission. On the other hand, figure 6c and 6d depict that TCP contention algorithm greatly decreases the value of RTT and RTT deviation, respectively. This means incorporating TCP Contention to default TCP decreases the application response time (by minimum factor of 2.3 as shown in figure 6c) while it guarantees smoother packet delays change (jitter) over a course of time.

After showing the improvement of end-to-end measurements using TCP Contention scheme, we investigate the effect of the algorithm on each individual nodes in the network. In the link layer, investigate the effect of our algorithm on the link layer performance. Table 1 shows that with the proposed approach the value of

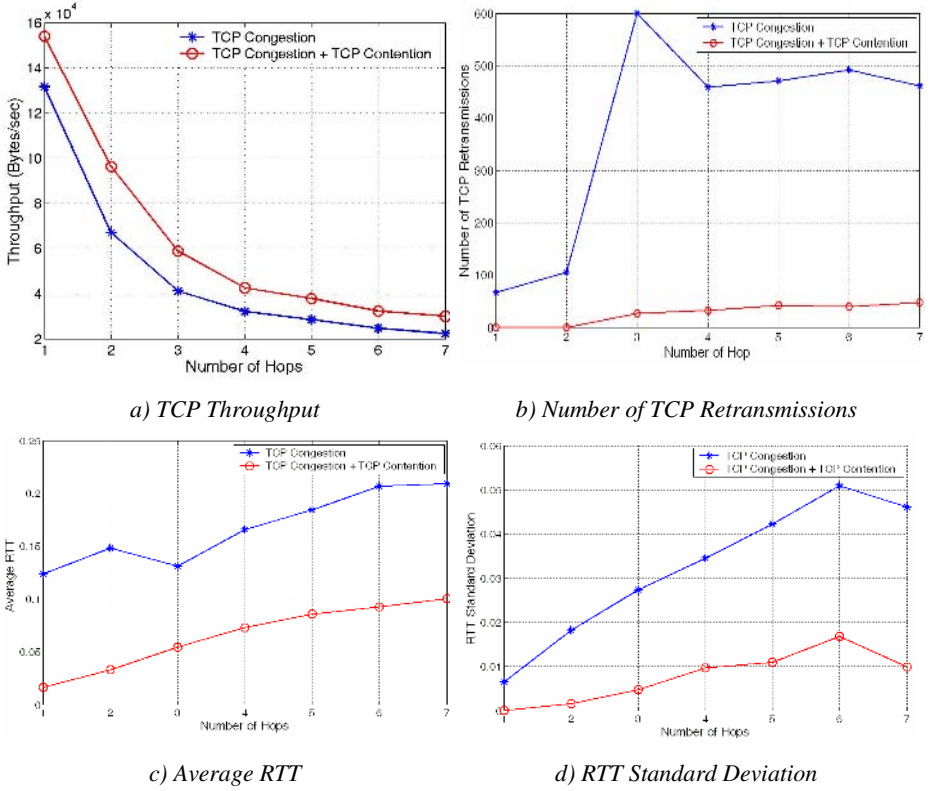


**Fig. 6.** TCP measurements in a chain topology

**Table 1.** The Average Link layer Attempt (ALA) in chain topology

| Number of Hops | TCP Congestion | TCP Congestion + TCP Contention |
|----------------|----------------|---------------------------------|
| 1              | 1.0301         | 1.0002                          |
| 2              | 1.3209         | 1.0183                          |
| 3              | 1.8536         | 1.3394                          |
| 4              | 1.8524         | 1.4914                          |
| 5              | 1.7944         | 1.4157                          |
| 6              | 1.7492         | 1.3660                          |
| 7              | 1.6869         | 1.2988                          |

Average Link layer Attempt (ALA) to successfully transmit a packet is strongly reduced. Also in this case, the reason is that by controlling the level of contention (and hence reducing the number of outstanding segments), the number of frame

collisions in the network are reduced. This effect has significant impact on the energy consumption of wireless node which are usually battery supplied devices. Thus, the developed scheme can be exploited when energy saving is an issue to be addressed.

*b) Grid Topology (flows starting at the same time)*

To further verify the performance of the TCP Contention, we extend our study to a grid scenario of four TCP flows shown in figure 7. This enables us to evaluate the algorithm in the presence of parallel connections as well as cross connections where nodes are subjected to extra contention in the network. Here all 4 connections start their transmission at the same time.

Table 2 presents the TCP throughput and total number of TCP retransmissions in each connection as well as the aggregated values in all connections using TCP

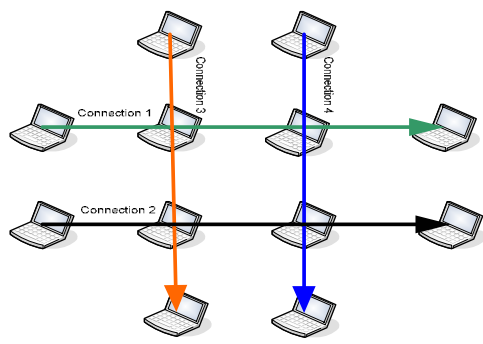


**Fig. 7.** 4x4 Grid topology

**Table 2.** TCP Throughput and Total Number of TCP Retransmissions in grid topology

|             | TCP Throughput (Bytes/sec)        |                                | Number of TCP Retransmissions       |                                 |
|-------------|-----------------------------------|--------------------------------|-------------------------------------|---------------------------------|
|             | TCP Congestion                    | TCP Congestion +TCP Contention | TCP Congestion                      | TCP Congestion + TCP Contention |
| Connection1 | 10030                             | 14236                          | 427                                 | 133                             |
| Connection2 | 12660                             | 12597                          | 383                                 | 130                             |
| Connection3 | 8009                              | 14157                          | 387                                 | 137                             |
| Connection4 | 11648                             | 10395                          | 338                                 | 145                             |
|             | 42347                             | 51385                          | 1535                                | 545                             |
|             | Aggregated Throughput (Bytes/sec) |                                | Total Number of TCP Retransmissions |                                 |



Contention and default TCP. It is clear TCP Contention algorithm also outperforms default TCP in case of multiple flows by reducing the overall number of TCP retransmissions in all connections while increasing the throughput by around 20%.

To measure the ALA in a grid topology, we use formula 5, where C is the total number of connections and N is the number of nodes in each connection.

$$ALA = \frac{\sum_{i=1}^N \sum_{j=1}^C (Transmitted\ Packets)_{i,j}}{\sum_{i=1}^N \sum_{j=1}^C (Successfully\ Transmitted\ Packets)_{i,j}} \quad (5)$$

The results presented in table 3 shows a decrease of 17% in overall number of packet retransmission in the link layer. We should note that ALA can be even further reduced by designing a more efficient collision avoidance technique such as the one we have proposed in [14], since even when TCP contention is adopted nearly every 1 out of 3 packets transmitted are collided, causing considerable number of unnecessary packet retransmission.

**Table 3.** The Average Link layer Attempt (ALA) in grid topology

|                                                  | TCP Congestion | TCP Congestion + TCP Contention |
|--------------------------------------------------|----------------|---------------------------------|
| Total Number of Successfully Transmitted Packets | 53662          | 57388                           |
| Total Number of Retransmitted Packets            | 47228          | 32302                           |
| Average Link layer Attempt (ALA)                 | 1.8862         | 1.5628                          |

On the other hand, to show the effectiveness of the TCP Contention scheme in the link layer performance and also TCP end-to-end delay in case of multiple flows, we measured the average number of packets buffered in all nodes during the simulation time. As shown in fig.8, the proposed algorithm keeps the average number of buffered packets in each node close to 1 comparing to much larger and time-varying number buffered packets using default TCP. This matches very closely with the objective of an ideal scenario explained in section 3 in which we stated the best scenario to keep the “right” amount of outstanding data in the network, would be the case where each node along the path holds exactly one packet at a time.

### c) Grid Topology (flows starting in different time)

In the next set of simulation, we use the topology showed in figure 7 with the change that while connection 1 and 2 are running from the beginning of the simulation, connection 3 and connection 4 start time is set to 300 seconds. Our main goal of conducting this simulation is to see how TCP Contention reacts when the level of

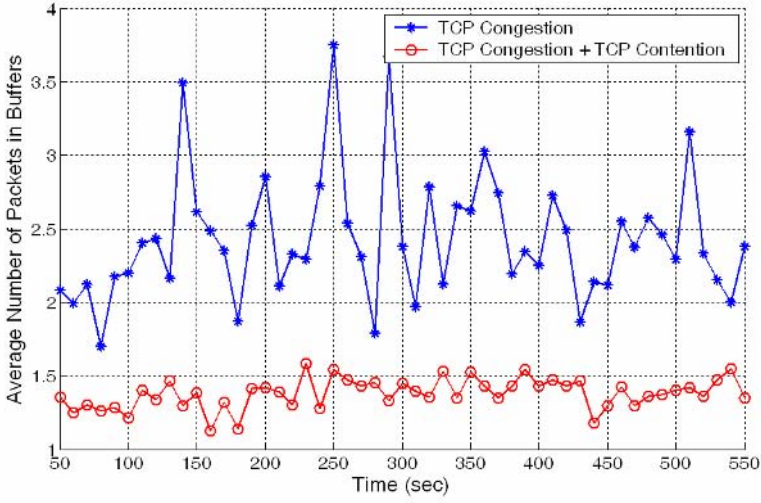


**Fig. 8.** Average number of packets buffered in all nodes in a 4x4 grid topology

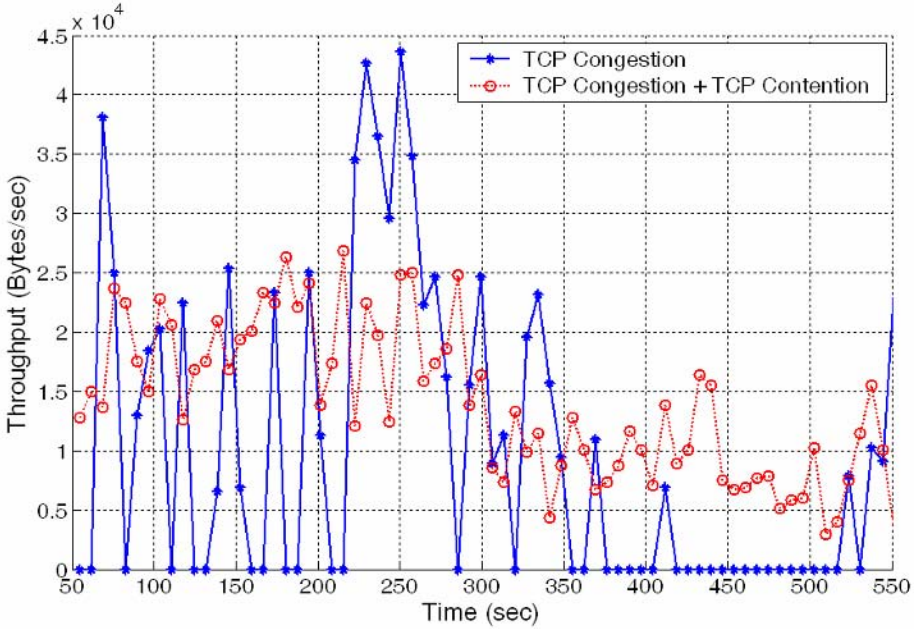


**Fig. 9.** TCP Throughput in connection 1 using a grid topology with different start time

contention around intermediate node is changed in the middle connection. Figure 9 depicts the TCP throughput seen by connection 1 before and after the contention in intermediate nodes is increased by time 300 sec. It is obvious that in both situations, the TCP which uses the TCP Contention algorithm achieves a higher and more stable

throughput. This is very promising as it shows one of the main causes of TCP instability in ad hoc networks can lie in the TCP window mechanism itself which controls the amount of traffic sent into the network.

Similar to the results presented in figure 8 for simultaneous connections in a grid topology, figure 10 compares the average number of packets which are buffered in all nodes when connection 3 and 4 start their transmission at time 300 sec. To have a better understanding of the queue size changes in this scenario, the average buffer size values are replaced with a smoothed values that is calculated by the moving average filter.

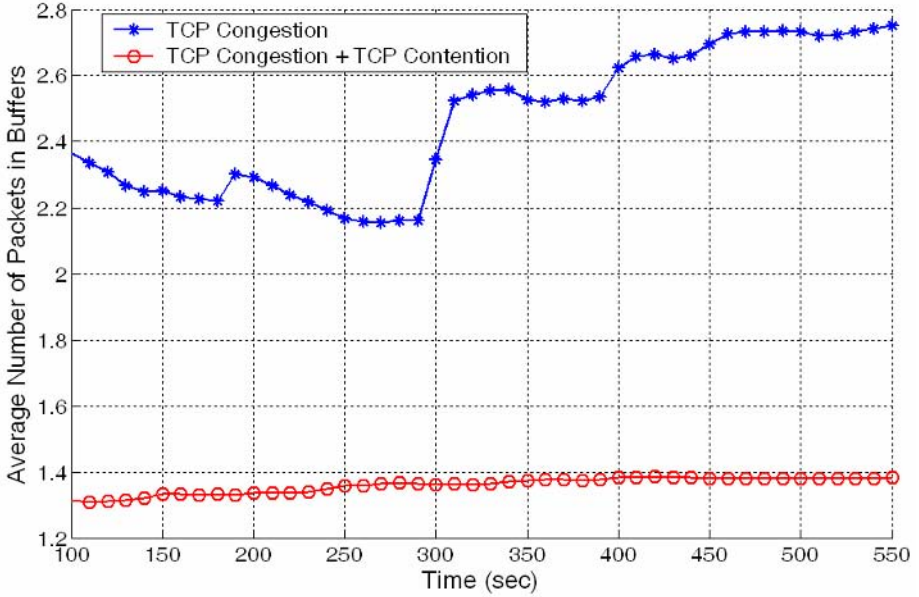


**Fig. 10.** Average number of packets buffered in all nodes in a 4x4 grid topology (different start time)

Here we can see that in default TCP, as soon as connection 3 and 4 start in the middle of the transmission by connection 1 and 2, the network starts getting overloaded and there is a sharp increase in number of packets that are buffered. On the other hand, in TCP Contention scheme since TCP receivers continuously adjust and control the amount of outstanding data, the average number of packets queued in the network remains almost constant.

Therefore, from the graphs in figure 9 and 10 we can conclude that the other important feature of TCP Contention Control is its flexibility to adopt quickly to different network conditions.

#### *d) Random Topology*

We also run a simulation in a random topology where 50 nodes are distributed randomly in a area of 1000m x 1000m and 10 pairs of TCP source and TCP receiver are chosen randomly. To setup the simulation environment more realistically, the rest of nodes (30 nodes) run CBR traffic in their background. The results summarized in

table 4 shows the TCP Contention Control outperforms the default TCP both in TCP layer and link layer. It is interesting to note that the scale of improvement in link layer is not as significant as we have seen in earlier scenarios. We believe this is mainly due to large overhead of routing protocol messages that are flooded across the network as the number of nodes is increased. The routing messages create considerable amount of contention and therefore packet drop in the network that cannot be eliminated by TCP Contention Control.

**Table 4.** Summary of measurements in a random topology

|                                                  | TCP Congestion | TCP Congestion + TCP Contention |
|--------------------------------------------------|----------------|---------------------------------|
| Average Throughput per Connection<br>(Bytes/sec) | 9893.6         | 11261.1                         |
| Total Number of TCP Retransmissions              | 2597           | 685                             |
| Average RTT (sec)                                | 0.31           | 0.12                            |
| Average Link Layer Attempt                       | 1.9645         | 1.6535                          |

## 6 Conclusion and Future work

Improving the performance of TCP over 802.11 multi-hop ad hoc networks is truly a cross-layer problem. As we showed in this paper, one of the critical sources of lowering TCP throughput lies in the TCP window mechanism which controls the amount of traffic sent into the network. To tackle the problem, we proposed a cross layer algorithm called TCP Contention Control that it adjust the amount of outstanding data in the network based on the level of contention experienced by packets as well as the throughput achieved by connections. The main features of this algorithm is its flexibility and adaptation to the network conditions. Furthermore, TCP Contention Control is compatible with all TCP versions and it does not require any changes in TCP congestion control algorithm since it simply uses the existing TCP to throttle the amount of outstanding data in the network. This can be very useful in heterogeneous networks (wire + wireless) where the same TCP can be used in both networks. In future we are planning to conduct more simulation to evaluate the capability of TCP Contention Control when the connection is expanding from wire to wireless networks.

## References

- [1] Z. Fu, X. Meng, and S. Lu, "How Bad TCP Can Perform in Mobile Ad Hoc Networks", IEEE Symposium on Computers and Communications, 2002
- [2] Z. Fu and others, "The Impact of Multihop Wireless Channel on TCP Performance," IEEE Transactions on Mobile Computing, vol. 4, no. 2, pp. 209-221, 2005.

- [3] E.Hamadani and V.Rakocevic, "Evaluating and Improving TCP Performance Against Contention Losses in Multihop Ad Hoc Networks", IFIP International Conference (MWCN), Marrakech, Morocco, 2005
- [4] K. Chen and others, "Understanding Bandwidth-Delay Product in Mobile Ad Hoc Networks," Computer Communications, vol. 27, no. 10, pp. 923-934, 2004.
- [5] W.Stevens, "RFC 2001 - TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms,".Technical Report, Jan.1997.
- [6] M.Allman, V.Paxson, and W.Stevens, "RFC 2581 - TCP Congestion Control,".Technical Report, Apr.1999.
- [7] S.Xu and T.Saadawi, "Does the IEEE 802.11MAC protocol work well in multihop wireless ad hoc networks", 39 ed pp. 130-137.2001
- [8] K. Chen and K. Nahrstedt, "Limitations of Equation-Based Congestion Control in Mobile Ad Hoc Networks", Proceedings - 24th International Conference on Distributed Computing Systems Workshops, Mar 23-24 2004, 2004
- [9] C. Ware and others, "Unfairness and Capture Behaviour in 802.11 Adhoc Networks", IEEE International Conference on Communications, 2000
- [10] "IEEE Standards for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY),Part 11:Technical Specifications" .1999
- [11] K. Xu and others, "TCP Behavior Across Multihop Wireless Networks and the Wired Internet", Proceedings of the Fifth ACM International Workshop on Wireless Mobile Multimedia (WOWMOM 2002), Sep 28 2002, 2002
- [12] OPNET simulator, <http://www.opnet.com>
- [13] G.Anastasi, M.Conti, and E.Gregori, "IEEE 802.11b Ad Hoc Networks: Performance Measurements", 2003
- [14] E Hamadani and V Rakocevic, "Enhancing Fairness and Stability in Multihop Ad Hoc Networks Using Fair Backoff Algorithm", Submitted to IEEE International Conference on Communications, 2006

# Providing Relative Service Differentiation to TCP Flows over Split-TCP Geostationary Bandwidth on Demand Satellite Networks

Wei Koong Chai<sup>1</sup>, Merkourios Karaliopoulos<sup>2</sup>, and George Pavlou<sup>1</sup>

<sup>1</sup> Centre for Communication Systems Research, University of Surrey, GU2 7XH, UK  
{W.Chai, G.Pavlou}@surrey.ac.uk

<sup>2</sup> Department of Computer Science, University of North Carolina, Chapel Hill, USA  
mkaralio@cs.unc.edu

**Abstract.** We propose a combined transport – medium access control (MAC) layer scheme to provide *relative* service differentiation to Transmission Control Protocol (TCP) flows over a geostationary (GEO) bandwidth on demand (BoD) satellite networks. Our approach involves the joint configuration of TCP-Performance Enhancing Proxy (TCP-PEP) agents at the transport layer and the scheduling algorithm controlling the resource allocation at the MAC layer. The scheme is independent of the TCP variant used in the network. Extensive simulation results show that the two mechanisms exhibit complementary behavior in achieving the desired differentiation throughout the traffic load space: the TCP-PEP agents control differentiation at low system utilization, whereas the MAC scheduler becomes the dominant differentiation factor at high load.

**Keywords:** relative service differentiation, satellite network, TCP, Bandwidth on demand.

## 1 Introduction

Satellite networks are vital components of the next-generation Internet. In order to seamlessly integrate with the global information infrastructure, they have to adjust to the technologies and trends that are adopted in terrestrial networks. For example, even if the DVB/MPEG-2 stack is now the basis for many operational systems, the dominance of Internet Protocol (IP) renders native-IP systems more attractive. All the same, Transmission Control Protocol (TCP) is the Internet de-facto transport protocol for communications and has to be efficiently supported over satellite networks.

Pretty much the same considerations dictate that satellite networks provide service differentiation to different types of traffic in agreement with the Internet quality of service (QoS) framework. The research effort on Internet QoS has been tremendous during the last 15 years and has sparked endless discussions addressing even its necessity as such. More recently, the efforts have been oriented towards frameworks that provide *relative* service differentiation, compromising effectively the hard, quantitative guarantees of Integrated Services (*IntServ*) with the scalability of Differentiated Services (*DiffServ*). Proportional Differentiated Services (PDS) [1] are one of the well-received proposals in this direction.

How well can satellite networks satisfy the requirements of these lighter QoS frameworks? Which are those functions and mechanisms within the network that can help achieve that? In this paper, we focus on providing *relative* service differentiation to TCP flows over a Geostationary (GEO) Bandwidth on Demand (BoD) satellite network. We assume that the satellite network is equipped with Performance Enhancing Proxies (PEP), which have been widely deployed in satellite networks despite security and scalability concerns [2]. Our approach involves both the transport and medium access control (MAC) layers. TCP flows are divided in a number of classes over the network. To provide them with *relative* service differentiation, we jointly configure the TCP-PEP agents and BoD scheduling algorithm.

The contribution of our work is highly methodological and adds to a broader study of service differentiation mechanisms over GEO satellite networks for all types of traffic [3]. We illustrate the advantages that modest use of cross-layer approaches can have in satellite network engineering. Likewise, we add some arguments in favor of the utility of TCP-PEPs in the same context. However, we equally insist on the evaluation of our proposal via extensive simulations that aim at showing its potential but also its weaknesses.

We organize the paper into six sections. We present the reference system architecture with details on the BoD operation in Section 2. In Section 3, we outline the design requirements for our combined transport–MAC layer differentiation scheme and detail its implementation. We evaluate our proposal in Section 4, showing how the mechanisms at the two layers can be complementary in realizing the desired differentiation objective. Related work is reviewed in Section 5. We conclude the paper in section 6.

## 2 Network Architecture and BoD Process

### 2.1 Network Architecture

The system architecture under consideration is a broadband GEO satellite IP network with resource allocation mechanisms analogous to the Digital Video Broadcasting - Return Channel via Satellite (DVB-RCS) standard [4]. We consider satellites with on-board processing allowing mesh terminal connectivity and satellite terminals equipped with TCP-PEPs. Note, however, that our approach is not limited to DVB-RCS like networks. Fig. 1 illustrates the main nodes of the architecture:

- Satellite – the scheduler is assumed onboard.
- Satellite Terminal (ST) – STs are equipped with TCP-PEP agents that split the TCP connections into a terrestrial (wired) component and a satellite (wireless) component. The STs may serve one (residential) or more users (collective).

The multiple access scheme in the satellite link is multi-frequency TDMA (MF-TDMA). The basic unit of link capacity in a MF-TDMA frame is the timeslot (TS), with multiple TSs grouped in TDMA frames along several frequency carriers.

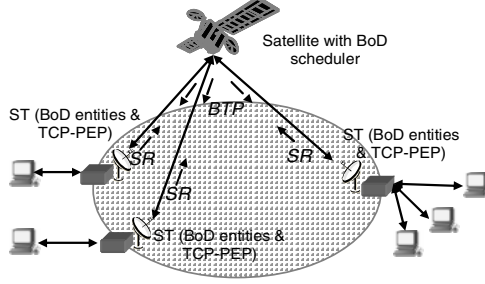


Fig. 1. Reference satellite network configuration

## 2.2 Bandwidth on Demand (BoD) Process

The BoD process used is drawn from [5] and consists of two functions that are executed periodically, namely the resource request estimation and resource allocation processes. The main entities involved are the BoD entity and the BoD scheduler. The BoD entity is located at the ST and handles all packets of the same class that share the same queue; there will be  $x$  BoD entities in a ST supporting  $x$  traffic classes. When there are new packet arrivals at their queues, BoD entities send slot requests (SRs) to the BoD scheduler with a period of  $n_s$  TDMA frames. If  $q(k)$  are the queued packets at the BoD entity at the start of the  $k^{\text{th}}$  allocation period, then the SR sent to the BoD scheduler is given by

$$SR(k) = \left\lceil \left( q(k) - n_s \cdot a(k) - n_s \cdot \sum_{j=1}^{L_s-1} SR(k - L_s + j) - n_s \cdot w(k) \right) / n_s \right\rceil^+ \quad (1)$$

In Eq. 1,  $a(k)$  denotes the number of TS per frame already allocated to the BoD entity for the coming  $n_s$  frames.  $w(k)$  are the owed TSs by the scheduler from previous resource allocation periods, when the BoD entity requests were partially or not at all satisfied and  $L_s$  is the *nominal system response time* (BoD cycle) in frames, namely the time elapsing from the moment a BoD entity submits a request till the moment it should receive its allocation from the scheduler. The  $L_s$  parameter accounts for the propagation, transmission and processing delays at the BoD entity and the BoD scheduler. The actual system response time may well be higher than the nominal one, if the request cannot be served in the first encountered resource allocation period but rather has to be queued and served in subsequent resource allocation periods.  $[v]^+ = v$  if  $v > 0$  and 0 otherwise. This ensures that no SR will be submitted if it is zero or negative.

Upon reception of SRs, the BoD scheduler allocates TSs to each requesting BoD entity based on the scheduling discipline and policies set by the network operator. It then constructs the burst time plan (BTP) that contains the allocation information and broadcasts it to the BoD entities. Fig. 2 outlines the time evolution of the BoD process and the timing of the resource request submission and allocation tasks.

At the STs, TCP-PEPs split the TCP connections between the terrestrial and satellite domains. The proxies will cache TCP segments and prematurely acknowledge their arrival. On the satellite network side, they are required to execute our transport layer differentiation mechanism.



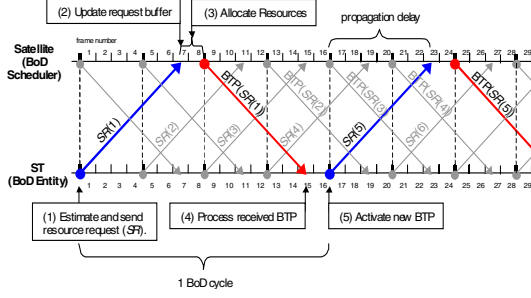


Fig. 2. BoD timing diagram

### 3 PDS Provision to TCP Traffic over Satellite

#### 3.1 Design Requirements

Our design follows the objective of the PDS model [1]. The model is especially suitable for satellite networks as it is lightweight and does not require complex mechanisms such as admission control or provisioning.

The network considered supports  $N$  service classes indexed by  $i$ ,  $i \in I \equiv \{1..N\}$ . Each class is assigned a differentiation parameter (DP)  $r_i$  controlling the performance gap between service classes. If  $\sigma_i$  denotes the performance metric of interest for class  $i$ , then the PDS model requires that

$$\frac{\sigma_i}{\sigma_j} = \frac{r_i}{r_j} \quad \forall i, j \in \{1, \dots, N\}. \quad (2)$$

In this paper, class 1 is the highest priority class with its DP set to unity. We normalize all DPs with reference to it so that

$$0 < r_N < r_{N-1} < \dots < r_2 < r_1 = 1$$

The metric of interest here is the average TCP throughput. We want to provide different throughput to TCP flows that are classified under different service classes. Applying the PDS model into our problem, we would like to have

$$\frac{\overline{thr}_i}{\overline{thr}_j} = \frac{r_i}{r_j} \quad \forall i, j \in \{1 \dots N\} \quad (3)$$

where  $\overline{thr}_i$  is the average throughput that traffic flows of class  $i$  obtain in their lifetime. From here onwards, we refer to the ratio  $\overline{thr}_i / \overline{thr}_j$  simply as throughput ratio.

#### 3.2 Combined Transport-MAC Layer Differentiation

To control TCP throughput, we first need to consider what parameters affect it. Existing analytical approximations for TCP throughput under congestion loss, e.g. [6], [7], are most useful to this end

$$thr = \frac{W \cdot MSS}{RTT} = \frac{k}{\sqrt{p}} \cdot \frac{MSS}{RTT} \quad (4)$$

where  $W$  is the average TCP send window,  $MSS$  is the maximum segment size of the TCP connection,  $p$  is the loss probability and  $k$  is a constant depending on the nature of loss and the use of the Delayed Acknowledgements option [6]. The term  $RTT$  denotes the overall round-trip time related to the TCP loop, including the propagation delay and the queuing/processing delays suffered by TCP segments in the forward direction and ACK packets in the return direction.

In practical implementations, the upper bound on the TCP window size is set by  $W_{\max}$ , which is dictated by the availability of socket buffer sizes at the two TCP endpoints and application-specific configuration. The throughput equation is then written as

$$thr = \min \left( \frac{W_{\max} \cdot MSS}{RTT}, \frac{k}{\sqrt{p}} \cdot \frac{MSS}{RTT} \right). \quad (5)$$

Eq. 5 suggests that the TCP throughput is basically dependent on (a) packet loss probability and (b) queuing delays since the propagation and processing delay contributions to  $RTT$  may be assumed constant for a given connection path. Since the physical layer in broadband satellite systems is dimensioned to yield bit error rates in the order of  $10^{-10}$ , we may assume  $p \rightarrow 0$  for the satellite component of the end-to-end connection. Therefore, the throughput achieved by the satellite components of the TCP connections can be written

$$thr = \frac{W_{\max} \cdot MSS}{RTD + dq_F + dq_R} \quad (6)$$

where  $dq_F$ ,  $dq_R$  are the queuing delays experienced in the forward and return (ACK) path of the TCP connections in the satellite network.  $RTD$  here refers to the round-trip delay excluding queuing delays.

Eq. 6 suggests that one class of TCP connections may obtain better performance than another in either or under combination of the following cases: a) when they experience lower delays at the MAC scheduler during the resource (i.e. slot) allocation process, b) when the upper bound of the TCP window of their satellite component is set to a higher value. The difficulty arises when we want to control, i.e. quantify, the relative performance differentiation the classes of connections obtain.

In principle, there are several mechanisms and configuration options that can yield the desirable result at the MAC and transport (PEP) layer. We have chosen to deploy the Satellite Waiting Time Priority (SWTP) scheduler for BoD scheduling [8] and vary the  $W_{\max}$  assigned to TCP connections of different service classes at TCP-PEP.

The SWTP scheduler [8] is an adaptation of the WTP scheduler for satellite networks. SWTP schedules  $SR$ s from BoD entities rather than individual packets. We have shown that SWTP can provide proportional queuing delay to several classes of MAC frames in the context of the BoD environment. We briefly describe its main aspects below.

**Resource request:** Formally, if  $Q_i$  is the set of newly arrived packets at the queue of BoD entity  $i$ , i.e. packets that came within the last resource allocation period,  $q$  the set cardinality, and  $\tau_j$  the arrival time of packet  $j$ ,  $1 \leq j \leq q$ , indexed in increasing order of arrival times, then the BoD entity  $m$  computes at time  $t$  the SR timestamp  $ts_i^m$  according to the arrival time of the last packet that arrived in the queue during the last resource allocation period, namely:  $ts_i = t - \tau_q$ .

**Resource allocation:** The BoD scheduler computes the priority of each SR,  $P_i^m(k)$  at  $k^{th}$  resource allocation period as

$$P_i^m(k) = r_i^D \cdot (w_i^{SR}(k) + \alpha) \quad (7)$$

where  $\alpha$  accounts for the propagation delay of BTP and the processing delay in BoD entities, while  $w_i^{SR}(k) = t - ts_i^m$  and  $ts_i^m$  is the timestamp encoded in each SR.  $r_i^D$  denotes the delay differentiation parameter (DDP). Each one of the  $M$  MAC layer classes features a specific  $r_i^D$ ,  $1 \leq i \leq M$ .

At the allocation instance, the SWTP allocates TSs by considering requests in decreasing priority order. Requests are fully satisfied as long as they do not exceed the available capacity. All unsatisfied requests will be buffered for the next allocation period. At the next allocation instance, the priorities of the buffered SRs will be recalculated to account for the additional waiting time of the request at the scheduler.

At the transport layer, we use  $W_{max}$  as the differentiation parameter for the satellite component of the TCP connections. Note that this is different than what several TCP variants, proposed with wireless networks in mind, have done. TCP Peach [9], Westwood [10], and Hybla [11], to mention but a few, actually intervene in the additive increase, multiplicative decrease congestion control mechanism of the protocol. What we do instead is to control  $W_{max}$  and this feature exactly makes the scheme independent of the actual TCP variant.  $W_{max}$  categorizes persistent TCP connections into two types [3]:

- *Capacity-limited* connection where  $W_{max} \geq \text{Path Bandwidth-Delay Product, BDP}$  ( $\text{BDP} = \text{RTT} \cdot C$  where  $C$  is the bottleneck link capacity).
- *Receive window-limited* connection where  $W_{max} < \text{Path BDP}$ .

Change of the  $W_{max}$  parameter has an impact on TCP throughput in the second case, whereas in the first case the TCP throughput may only be increased with an increase of the bottleneck link capacity.

The split-TCP connections are grouped into  $L$  transport layer classes, each mapped to a single  $W_{max}$  value. Then we can write

$$\frac{W_{\max i}}{W_{\max j}} = \frac{r_i^T}{r_j^T} \quad \forall i, j \in \{1 \dots L\} \quad (8)$$

where  $\{r_i^T\}$  is the set of throughput differentiation parameters (TDP). Normalizing with reference to the TDP of the lowest-priority class so that

$$r_1^T > r_2^T > \dots > r_{N-1}^T > r_N^T = 1$$

TCP-PEPs at the border of the network will set the  $W_{max}$  of the satellite component of each class  $i$  TCP connections to

$$W_{\max(i)} = r_i^T \cdot \min rwnd \quad (9)$$

where  $\min rwnd$  is the minimum value of  $W_{max}$  over all classes. For simplicity, we assume that the same number of service differentiation levels is defined at both MAC and transport layers, which equal the total number of TCP traffic classes that are supported within the satellite network; i.e.,  $L$  equals  $M$  and both equal  $N$ . Under these assumptions, the problem we face can be stated as follows:

*“In a split-TCP capable BoD satellite network, how should one jointly set the TDPs at the TCP-PEPs and the DDPs at the return link MAC scheduler, so that for a given set of DPs,  $\{r_i\}$ , the PDS model objective of Eq. 3 can be achieved?”*

## 4 Performance Evaluation

### 4.1 Simulation Setup

We extend the *ns2* to support BoD satellite networks and add an implementation of the SWTP scheduler. The network topology used is shown in Fig. 1, where the bottleneck is assumed to be at the satellite part of the topology. For all simulations, the terrestrial links are configured to be 2048 kbps while the satellite up/downlinks are set to 512 kbps. The packet size is 576 bytes and is fragmented into 48 byte MAC frames. Each TDMA frame period is 24 ms. We use out of band signaling with SRs submitted in pre-assigned slots. The rate granularity is 16 kbps. The effect of our solution is demonstrated by considering persistent TCP sources. Unless explicitly stated otherwise, the network serves three classes with DP set  $\{r_i\} \equiv \{1, 1/2, 1/4\}$ , i.e., the target throughput ratio between two successively ordered TCP traffic classes is 0.5. If the achieved throughput ratio is smaller than 0.5, then the actual performance spacing is greater than desired and vice versa. We fix  $\min rwnd$  to 8 kb in Eq. 9.

### 4.2 Impact of Traffic Load on the Two Service Differentiation Mechanisms

We first gain some insight into the impact of each of the two mechanisms separately on TCP throughput.

**Transport Layer Differentiation only:** We run simulations by switching off the SWTP and then taking a sample TCP connection from each service class for evaluation. Fig. 3 (a) and (b) shows the achieved throughputs under low and high traffic load respectively. The figures suggest that controlling  $W_{max}$  at the TCP proxies can yield proportional differentiation only when the system load is low. At low load most of the TCP connections in the network are *receive window-limited* connections and our transport layer configuration suffices to control the differentiation. However, when the network is highly loaded, the TCP connections are no longer limited by their respective  $W_{max}$ . Instead, they are *capacity-limited*. Slow start events take place, rendering the receive window constraints inactive. In PDS terminology, the *controllability*

and *predictability* of this transport-level differentiation mechanism are highly sensitive to the traffic load. Under low link utilization, the relative differentiation is achieved at steady-state after Slow-Start phase is over.

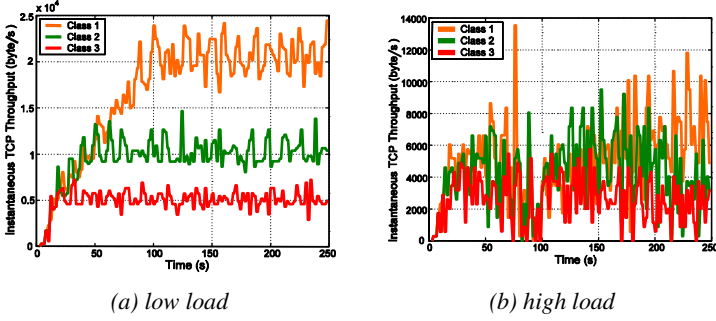


**Fig. 3.** Impact of send window control at TCP-PEPs, First-Come First-Served request scheduling at MAC layer

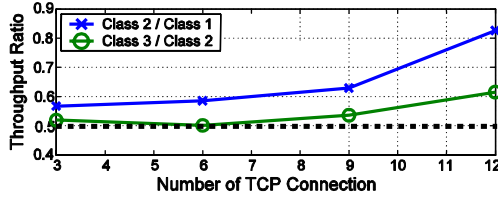


**Fig. 4.** As the number of TCP connections rises, the PEP loses control of the differentiation

An alternative demonstration of this behavior is shown in Fig. 4 which plots the mean relative performance differentiation versus input traffic load (i.e., number of TCP connections sharing the satellite link). It is clear that as the load increases, the achieved throughput ratios deviate from the target value (0.5). We also see that when the link is saturated, the system practically does not differentiate amongst the three classes, which is unacceptable for satellite operators designing satellite network radio bearer services.

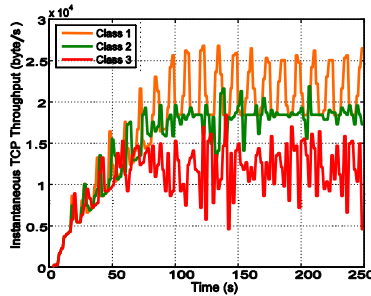


**Fig. 5.** TCP throughputs differentiated by SWTP; the performance spacing is not proportional

**MAC Layer Differentiation only:** We then run the simulations with TCP-PEPs switched off while SWTP is switched on. Fig. 5 shows that SWTP alone cannot provide the desired performance spacing even when the load is low. However, the SWTP has been shown in [8] to be increasingly effective when the load is increasing. This property fits in perfectly to complement our transport layer differentiation mechanism, which has been shown to work effectively at low load.

### 4.3 Evaluation of Full Differentiation Scheme

In Section 4.2, we showed that the two differentiation mechanisms are complementary. By deploying both transport and MAC layer differentiation mechanisms, we have a full service differentiation scheme that works throughout the load range. At low system load, the MAC layer differentiation mechanism is inactive, letting the transport layer differentiation mechanism alone provide the desired performance ratio as needed by the PDS model. As the system load increases, the transport layer differentiation mechanism slowly loses the differentiation control; at the same time, the SWTP scheduler starts taking effect by providing the required additional differentiation, so that the target performance ratio can be maintained.

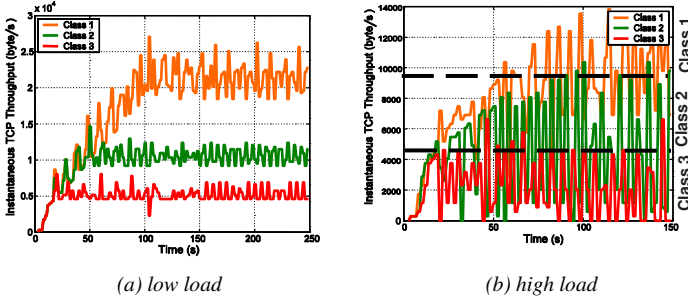
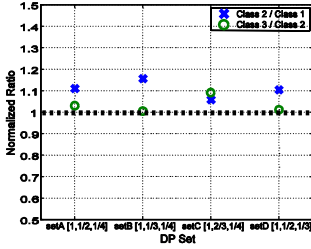


Fig. 6. Performance of our integrated approach for achieving the PDS model

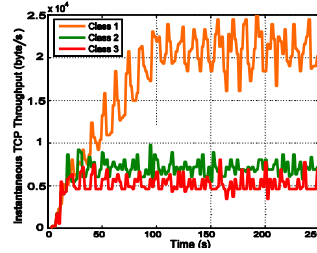
We first test the integrated scheme with 3 classes of TCP traffic and TDP set = DDP set =  $\{1, \frac{1}{2}, \frac{1}{4}\}$ . Fig. 6(a) implies that at low system utilization, the SWTP scheduler does not affect the differentiation provided by the proxy configuration. Fig. 6(b) shows the instantaneous TCP throughput for high load with our integrated approach. It shows improvement when compared with Fig. 3(b), in that now there are distinct throughput levels for each service class. However, there is still considerable fluctuation at individual TCP flow level. We found that the achieved throughput ratios under satellite link saturation are 0.4094 for  $\frac{\text{class2}}{\text{class1}}$  and 0.2793 for  $\frac{\text{class3}}{\text{class2}}$ , yielding a total deviation of 0.3113 from the target ratio (i.e. 0.5). This implies that the differentiation given by the combination of SWTP and the proxies has exceeded what is required. We run extensive simulations by varying both DDP and TDP sets. Table 1 shows selected results for the achieved ratios in the form of  $\{\frac{\text{class2}}{\text{class1}}, \frac{\text{class3}}{\text{class2}}\}$  throughput ratios. In this case, the optimal settings are DDP set =  $\{1, 0.6, 0.36\}$  and TDP set =  $\{1, 0.6, 0.36\}$  yielding a total deviation of only 0.0595.

**Table 1.** Achieved TCP throughput ratios between successively ordered classes under different combinations of DDP and TDP sets

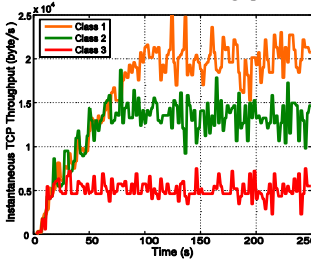
| DDP Set        | TDP Set        |                |                |                |                |
|----------------|----------------|----------------|----------------|----------------|----------------|
|                | {1, 0.4, 0.16} | {1, 0.4, 0.2}  | {1, 0.5, 0.25} | {1, 0.6, 0.3}  | {1, 0.6, 0.36} |
| {1, 0.4, 0.16} | 0.3294, 0.2605 | 0.3204, 0.3223 | 0.3648, 0.2230 | 0.3883, 0.3131 | 0.3888, 0.2991 |
| {1, 0.4, 0.2}  | 0.3168, 0.2548 | 0.2771, 0.3991 | 0.3574, 0.3008 | 0.3787, 0.2314 | 0.3772, 0.3202 |
| {1, 0.4, 0.24} | 0.2735, 0.3743 | 0.2810, 0.4354 | 0.3536, 0.4148 | 0.3779, 0.4135 | 0.3823, 0.3502 |
| {1, 0.4, 0.32} | 0.3213, 0.3834 | 0.3159, 0.4505 | 0.3510, 0.5050 | 0.3829, 0.5186 | 0.3883, 0.5499 |
| {1, 0.5, 0.2}  | 0.3530, 0.2862 | 0.3573, 0.3082 | 0.4149, 0.2372 | 0.4370, 0.2736 | 0.4675, 0.2850 |
| {1, 0.5, 0.25} | 0.3123, 0.3144 | 0.3606, 0.3215 | 0.4094, 0.2793 | 0.4249, 0.2446 | 0.4366, 0.3455 |
| {1, 0.5, 0.3}  | 0.3446, 0.3700 | 0.3139, 0.4633 | 0.3276, 0.5491 | 0.4358, 0.4298 | 0.3287, 0.5992 |
| {1, 0.5, 0.4}  | 0.3670, 0.3443 | 0.3726, 0.4267 | 0.4060, 0.5062 | 0.2831, 0.7912 | 0.7132, 0.5489 |
| {1, 0.6, 0.24} | 0.4077, 0.2910 | 0.3996, 0.3606 | 0.4823, 0.2816 | 0.4909, 0.4067 | 0.3976, 0.4616 |
| {1, 0.6, 0.3}  | 0.4283, 0.3042 | 0.4271, 0.3636 | 0.4765, 0.3888 | 0.4097, 0.4387 | 0.5378, 0.3396 |
| {1, 0.6, 0.36} | 0.4116, 0.3312 | 0.4137, 0.3667 | 0.3728, 0.5668 | 0.5401, 0.4422 | 0.5365, 0.5230 |
| {1, 0.6, 0.48} | 0.4472, 0.3333 | 0.4199, 0.3947 | 0.4845, 0.4534 | 0.5561, 0.3978 | 0.5620, 0.4897 |



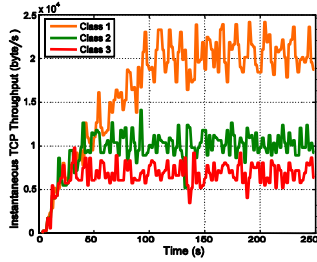
(a) Normalized mean throughput ratio



(b) PDS set 2



(c) PDS set 3



(d) PDS set 4

**Fig. 7.** Normalized mean and instantaneous throughput ratios under different DP sets

To assess further the capability of our integrated approach to accurately control the spacing between service classes, four sets of DPs have been defined: Set A = {1,  $\frac{1}{2}$ ,  $\frac{1}{4}$ }, Set B = {1,  $\frac{1}{3}$ ,  $\frac{1}{4}$ }, Set C = {1,  $\frac{2}{3}$ ,  $\frac{1}{4}$ }, Set D = {1,  $\frac{1}{2}$ ,  $\frac{1}{3}$ }. Four sets of simulation runs have been conducted based on the DP sets above with DP = TDP = DDP. Fig. 7(a) plots the *normalized* throughput ratios, i.e., the actual throughput ratios divided over the respective target ratios, for all four cases. The ideal value for the normalized throughput ratio is unity. Whereas, Figures 6(a) and 7(b–d) depict the

instantaneous throughput of TCP flows of different service classes under the four DP sets. It can be concluded that our approach allows much flexibility in the control of the spacing between the service classes.

## 5 Related Work

The provision of relative service differentiation to TCP flows has been primarily investigated in the context of wired networks. In [12], the authors rely on traffic metering, queue management and scheduling mechanisms to do so. The differentiation objective is achieved via marking algorithms used in tandem with explicit congestion notification (ECN) for regulating TCP traffic in the context of class-based service differentiation in [13] and via exploiting the receiver's advertised window (*rwnd*) of TCP connections in [14], where a weighted proportional window control mechanism is proposed. However, since the proposals above are mainly designed for wired network, they either implicitly or explicitly assume that the link capacities are constant. This is not the case for satellite and wireless networks.

Regarding support of QoS over satellite networks, there have been studies mainly on the implementation of the *DiffServ* framework. In [15], the authors assume a fully-fledged ATM switch onboard with buffer management capacity but do not consider the impact of the satellite MAC layer. A gateway architecture to achieve *DiffServ* for satellite networks via a joint resource management and marking approach is proposed in [16]. Their objectives are to minimize bandwidth wastage while satisfying QoS requirements. In [17], the authors compare several buffer management policies for satellite onboard switching to differentiate real time and non-real time traffic.

To the best of our knowledge, this is the first study on addressing *relative* service differentiation for TCP flows in BoD satellite environments. Previously in [18], we look at how to provide relative throughput differentiation for TCP flows in both GEO and non-GEO satellite networks via the use of congestion pricing. Meanwhile, the possibility of the joint use of transport-level and MAC-level service differentiation mechanisms has been demonstrated via analysis and extensive simulations in [19], in the context of split-TCP BoD satellite networks but with Strict Priority MAC level scheduling in mind. The work takes into account the impact of MAC layer in satellite networks when providing service differentiation. We extend this work here by demonstrating that the joint configuration of the two layers can realize a more demanding QoS model over the satellite network, which requires quantitative rather than qualitative relative service differentiation at class level. We not only want to ensure that the performance of one class is better than a lower priority one, we also try to control the performance gap between the two classes.

## 6 Conclusion

Our paper describes an integrated approach to the provision of proportional throughput differentiation to persistent TCP flows over BoD GEO satellite networks. The approach combines split-TCP proxy agents at transport layer together with a BoD scheduling algorithm, SWTP, exploiting their complementary behavior over the



network traffic load range. At low load, the differentiation can be controlled solely by the transport layer differentiation mechanism. At high load, the SWTP scheduling takes over to provide the desired quality spacing between classes, whereas the transport layer mechanism is less effective. With proper configuration of the two mechanisms, one can control the spacing between classes.

The paper contributions are largely methodological. Firstly, it demonstrates that it is feasible to provide differential treatment to TCP flows via combination of transport layer mechanisms and/or MAC layer scheduling algorithms in a BoD environment. Secondly, it describes how to achieve a specific QoS framework (here the PDS model) through our integrated approach.

In the real-world, the transport and MAC layers should be configurable in an automated manner according to the input load. The problem would benefit from analytical methods that can yield the correct parameterization of the two layers for a given traffic mix. We are currently [20] investigating analytical approximations that could assist with this task and we intend to report on our findings in the future.

**Acknowledgments.** This work is performed within the framework of the SatNEx project, funded by European Commissions (EC) under the Framework Program 6. The financial contribution of the EC towards this project is greatly appreciated.

## References

1. C. Dovrolis, D. Stiliadis and P. Ramanathan, "Proportional differentiated services: delay differentiation and packet scheduling," *IEEE/ACM Transactions on Networking*, vol. 10, no. 1, pp. 12-26, Feb. 2002
2. Y. Zhang, "A multilayer IP security protocol for TCP performance enhancement in wireless networks," *IEEE JSAC*, vol. 22, no. 4, pp. 767-776, May 2004
3. M. Karaliopoulos, "Support of elastic TCP traffic over broadband geostationary satellite networks," PhD. Dissertation, University of Surrey, Feb. 2004
4. EN 301 790 V1.3.1, "Digital Video Broadcasting (DVB); Interaction channel for satellite distribution systems," ETSI European Standard
5. G. Açar, "End-to-end resource management in geostationary satellite networks," PhD. Dissertation, University of London, Nov 2001
6. M. Mathis et al., "The microscopic behaviour of the TCP congestion avoidance algorithm," *Computer Communications*, vol. 3, July 1999
7. J. Padhye et al. "Modelling TCP throughput: a simple model and its empirical validation," in *Proc. ACM SIGCOMM*, Vancouver, 1998
8. W. K. Chai, M. Karaliopoulos and G. Pavlou, "Scheduling for proportional differentiated service provision in geostationary bandwidth on demand satellite networks," *Proc. of IEEE GLOBECOM*, USA, 2005
9. I. Akyildiz, G. Morabito and S. Palazzo, "TCP-Peach: a new congestion avoidance control scheme for satellite IP networks," *IEEE/ACM Transactions on Networking*, vol. 9, pp. 307-321, 2001
10. R. Wang et al., "TCP with sender-side intelligence to handle dynamic, large, leaky pipes," *IEEE JSAC*, vol. 23, no. 2, pp. 235-248, Feb 2005
11. C. Caini and R. Firrincieli, "TCP Hybla: a TCP enhancement for heterogeneous networks," *Int'l Journal of Satellite Communications and Networking*, vol. 22, pp. 547-566, 2004

12. T. Soetens, S. D. Cnodder and O. Elloumi, "A relative bandwidth differentiated service for TCP micro-flows," *Proc. of IEEE/ACM Int'l Symposium on Cluster Computing and the Grid*, pp. 602-609, 2001
13. N. Christin and J. Liebeherr, "Marking Algorithms for Service Differentiation of TCP Traffic," in *Computer Communications, Special Issue on End-to-End Quality of Service Differentiation*, 2005
14. J. Aweya et al., "Weighted proportional window control of TCP traffic," *Int'l Journal of Network Management*, vol. 11, pp. 213-242, 2001
15. A. Durresi et al., "Achieving QoS for TCP Traffic in Satellite Networks with Differentiated Services," *Space Communications*, Vol. 17, 2001
16. L. S. Ronga et al., "A gateway architecture for IP satellite networks with dynamic resource management and DiffServ QoS provision," *Int'l Journal of Satellite Communications and Networking*, no. 21, pp. 351-366, 2003
17. N. Courville, "QoS-oriented traffic management in multimedia satellite systems," *Int'l Journal of Satellite Communications and Networking*, no. 21, 2003, pp. 367-399
18. W. K. Chai, K.-H. Ho and G. Pavlou, "Achieving Relative Differentiated Services for TCP Traffic over Satellite IP Networks through Congestion Pricing", *Proc. of NEW2AN*, Russia, 2006
19. M. Karaliopoulos et al., "Providing differentiated service to TCP flows over bandwidth on demand geostationary satellite networks," *IEEE JSAC*, vol. 22, no.2, pp. 333-347, Feb 2004
20. W. K. Chai, M. Karaliopoulos and G. Pavlou, "Proportional Differentiated Service for TCP Flows over Geostationary Bandwidth on Demand Satellite Networks", Technical report, University of Surrey. Available at <http://www.ee.surrey.ac.uk/Personal/W.Chai/publications>

# An Analytical Comparison of the Slow-but-Steady and Impatient Variants of TCP NewReno

Roman Dunaytsev<sup>1</sup>, Konstantin Avrachenkov<sup>2</sup>, Yevgeni Koucheryavy<sup>1</sup>,  
and Jarmo Harju<sup>1</sup>

<sup>1</sup> Institute of Communications Engineering, Tampere University of Technology  
P.O. Box 553, FIN-33101, Tampere, Finland  
{dunaytse, yk, harju}@cs.tut.fi

<sup>2</sup> INRIA Sophia Antipolis, MAESTRO/MISTRAL Project  
2004, route des Lucioles, B.P. 93, FR-06902, Sophia Antipolis, France  
k.avrachenkov@sophia.inria.fr

**Abstract.** Current standard defines two variants of TCP NewReno: the Slow-but-Steady and Impatient. While the behavior of various TCP implementations has been extensively studied over the last years, little attention has been paid to performance analysis of different variants of TCP NewReno. In this paper, we first develop an analytical model of the Impatient variant, which being combined with the earlier proposed model of the Slow-but-Steady variant gives a comprehensive analytical model of TCP NewReno throughput. We then make an analytical comparison of the Impatient and Slow-but-Steady throughputs in the presence of correlated losses. We show that, although neither of the two variants is optimal, the Impatient variant provides the same throughput as the Slow-but-Steady one in a wide range of network conditions and significantly outperforms it in case of large windows and multiple packet drops. This could be of special interest for networks with large bandwidth and long delay.

**Keywords:** TCP NewReno; TCP throughput modeling; correlated losses.

## 1 Introduction

Transmission Control Protocol (TCP) is the prevalent transport layer protocol on the Internet. Most Internet applications, such as WWW, e-mail, FTP, and peer-to-peer file sharing, use TCP to provide reliable data transfer over unreliable “best-effort” service of IP. Since these applications are the dominant applications in today’s Internet, as a result, TCP controls a large fraction of bytes, packets, and flows traversed over the Internet. As long as TCP performance has a significant impact on user-perceived quality of service and the performance of the overall Internet, numerous implementations and modifications of TCP have been proposed during the last years, among the most known are TCP Tahoe, TCP Reno, TCP NewReno, TCP SACK, etc. Recent measurements [1] show that the most widely used TCP implementation in today’s Internet is TCP NewReno and the deployment of TCP NewReno has increased significantly in the last few years.

For a long time, the reference TCP implementation has been TCP Reno first deployed in the 4.3BSD-Reno and specified in [2]. This document defines four intertwined congestion control algorithms: slow start, congestion avoidance, fast retransmit, and fast recovery. Later it was found that TCP Reno has serious performance problems when multiple segments in the same window are lost. In particular, since only one of the lost segments can be recovered by each invocation of fast retransmit algorithm, the rest are often recovered using slow start after a usually lengthy retransmission timeout (RTO). TCP NewReno is a subsequent modification of the basic TCP Reno implementation and incorporates slow start, congestion avoidance, and fast retransmit from [3] with modified fast recovery algorithm. This modification concerns the sender's behavior during fast recovery when a partial acknowledgement (ACK) is received that acknowledges some but not all of the segments sent before the fast retransmit. While in TCP Reno the reception of partial ACK takes the sender out of fast recovery, in TCP NewReno the sender stays in fast recovery until all of the segments outstanding by the time the fast recovery was entered have been acknowledged, and, consequently, avoids multiple reductions of congestion window (*cwnd*) or unnecessary slow start algorithm invocation. Current standard [4] specifies two variants of TCP NewReno implementation: the Slow-but-Steady and Impatient. The only difference between them lies in the retransmission timer resetting scheme in response to partial ACKs. In the Slow-but-Steady variant the sender resets the retransmission timer after each partial ACK, whereas in the Impatient variant the sender performs resetting only after the first partial ACK. Depending on the given operational conditions (number of lost segments, delay variation, etc.) either one or the other may provide better performance.

While TCP performance modeling has received a lot of attention during the last years, the majority of the proposed models were developed for TCP Reno implementation (see references in [5], [6]). In the absence of sufficient analytical background, [4] recommends the Impatient variant of TCP NewReno based only on simulation results. Although simulation is a powerful tool in protocol analysis and design, it is a difficult task to simulate the protocol behavior across the range of all possible operational conditions and protocol parameter settings. In that case, an analytical model is extremely useful because it not only allows to apply the "what if" test to the different scenarios, but also to explore the protocol performance over the entire parameter space. To the best of our knowledge, the only analytical study explicitly addressed to comparison of TCP NewReno variants was recently presented in [7]. The authors studied TCP NewReno variants both analytically and using simulations and argued that the Slow-but-Steady variant is superior to the Impatient one in all but the most extreme network conditions and recommended it as the preferred variant of TCP NewReno, contrary to [4]. Unfortunately, whereas decision-making in protocol design requires very careful consideration and detailed analysis, the model in [7] is based on a number of crude approximations and simplifications as, for instance, infinite receiver buffer, oversimplified representation of fast retransmit/fast recovery phase, etc. Thus, the objective of our work is two-fold. We first develop a more accurate analytical model of steady state throughput of the Impatient variant of TCP NewReno, which together with the previously developed model of the Slow-but-Steady variant [8] will give us a comprehensive model of TCP NewReno throughput. We then evaluate TCP NewReno variants and define the most preferable one. Since the difference between

them only appears when a large enough number of segments in a window are lost, we focus our analysis on case of bursty losses inherent to a Drop-Tail environment.

The rest of the paper is organized as follows. In Section 2 we briefly describe the modeling assumptions. In Section 3 the detailed mathematical analysis is presented. Section 4 contains an analytical comparison of the steady state throughputs of the Impatient and Slow-but-Steady variants. Finally, some conclusions are summarized in Section 5.

## 2 Assumptions

While constructing our model we use the following assumptions. We assume that the sender uses the Impatient variant of TCP NewReno [4], always has data to send, and sends full-sized segments whenever *cwnd* and receiver window (*rwnd*) allow. We consider TCP behavior in terms of “rounds”, where a round starts when the sender begins the transmission of a window of segments and ends when the sender receives the first ACK for one or more of these segments; thus, the duration of a round is approximately equal to round trip time (RTT). We also assume that the time needed to send a window of segments is smaller than RTT and segment loss happens only in the direction from the sender to the receiver. It is assumed that RTT and probability of segment loss are independent of the window size. This assumption is justified for high-speed, large BDP links. Finally, we assume that a segment is lost in a round independently of any other segments lost in other rounds, but at the same time segment losses are correlated within a round (i.e., if a segment is lost, then all the subsequent segments in that round are lost as well). Such bursty loss model is a simplified representation of IP-packet loss process in routers with FIFO Drop-Tail queue management, which is still prevalent enough [9], [10].

## 3 Model Building

According to [4], a segment loss (if the sender is not already in fast recovery phase) can be detected in one of two ways: either by the reception at the sender of three duplicate ACKs or via retransmission timer expiration. In the latter case the sender enters slow start and recovers what appears to be the lost segment(s) using Go-Back-N strategy, but not fast recovery is performed. Since the fast recovery algorithm is the distinctive feature of TCP NewReno, we will focus on “pure” TCP NewReno behavior, i.e. when all loss detections are due to “triple-duplicate” ACKs. However, the model can be easily extended to capture loss detections via RTO expiration by following approach proposed in [11].

Let us consider steady state TCP NewReno behavior as a sequence of renewal cycles, where a cycle is a period between two consecutive loss events detected via the reception of three duplicate ACKs. For the  $i$ -th cycle ( $i = 1, 2, \dots$ ) let  $Y_i$  be the total number of segments sent during the cycle,  $A_i$  be the duration of the cycle, and  $W_i$  be the window size at the end of the cycle. Considering  $\{W_i\}_i$  as a regenerative process with a renewal reward process  $\{Y_i\}_i$ , we can define the steady state throughput as  $B = E[Y]/E[A]$ .

Let  $\delta_{i-1}$  denote the number of segments lost in the cycle  $(i-1)$ . In contrast to the Slow-but-Steady variant, where the sender recovers one lost segment per round and the number of rounds in fast retransmit/fast recovery (FR) phase of the  $i$ -th cycle can be defined as  $A_i^{FR} = \delta_{i-1}$ , in the Impatient variant the sender resets the retransmission timer only after the first partial ACK, hence the number of rounds in fast retransmit/fast recovery phase of the  $i$ -th cycle can be expressed as  $A_i^{FR} = \min(\delta_{i-1}, 1 + RTO/RTT)$ . Fig. 1 presents an example of transmission of segments during the  $i$ -th cycle for the Slow-but-Steady (Fig. 1a) and Impatient (Fig. 1b) variants.

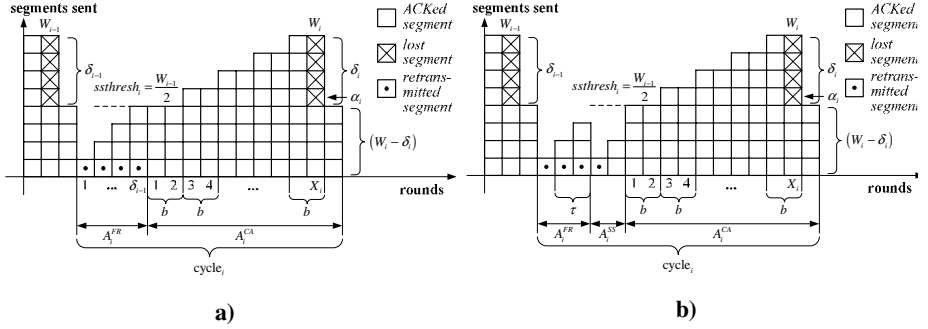


**Fig. 1.** Segments sent during the  $i$ -th cycle for the Slow-but-Steady and Impatient variants

Let us define  $\bar{\delta}$  to be the average number of segments lost in a row per loss event (also known as the average loss burst length) and  $\tau$  to be the ratio between average RTO and RTT values ( $\tau = RTO/RTT$ ). We assume that if  $\bar{\delta} < \tau + 1$ , managing the retransmission timer during fast recovery phase has negligible effect and the steady state throughput of the Impatient variant is identical to the Slow-but-Steady one. On the other hand, if  $\bar{\delta} \geq \tau + 1$ , it is expected that the sender will not recover all lost segments during fast recovery phase and after RTO expiration it will invoke slow start algorithm. Since for  $\bar{\delta} < \tau + 1$  the steady state throughput of the Impatient variant can be found from [8], to obtain a comprehensive model we need only to determine the steady state throughput of the Impatient variant for the case when  $\bar{\delta} \geq \tau + 1$ .

### 3.1 The Impatient Variant

Consider the  $i$ -th cycle as in Fig. 1b. After the reception of the third duplicate ACK, the sender enters fast retransmit/fast recovery phase and sets the slow start threshold as

$$sssthresh_i = W_{i-1}/2. \quad (1)$$

Since we assume that  $\bar{\delta}_{i-1} \geq \tau + 1$ , the sender's retransmission timer will ultimately expire and the sender will invoke slow start algorithm. Therefore, the number of rounds

in fast retransmit/fast recovery phase of the  $i$ -th cycle can be defined as  $A_i^{FR} = \tau + 1$ . Then the expected number of rounds in fast retransmit/fast recovery phase is

$$E[A_i^{FR}] = \tau + 1. \quad (2)$$

To simplify the subsequent derivation, we assume that  $\tau$  is integer.

In each round of fast retransmit/fast recovery phase the sender recovers one lost segment and transmits new segments, if allowed by new value of  $cwnd$ . Then the total number of segments sent during fast retransmit/fast recovery phase of the  $i$ -th cycle can be found as a sum of retransmitted segments and new ones:

$$Y_i^{FR} = \tau + 1 + \sum_{k=1}^{\tau+1} \left[ \max \left( 0, \frac{W_{i-1}}{2} - \delta_{i-1} + k - 1 \right) \right]. \quad (3)$$

From (3) we have that the expected number of segments sent during fast retransmit/fast recovery phase can be defined as

$$E[Y^{FR}] = \begin{cases} \tau + 1 + (\tau + 1) \left( \frac{E[W] + \tau - 2\bar{\delta}}{2} \right), & \text{if } \bar{\delta} \leq \frac{E[W]}{2}, \\ \tau + 1 + \frac{E[W]}{4} \left( \frac{E[W]}{2} - 1 \right) + (\tau + 1 - \bar{\delta}) \left( \frac{E[W] + \tau - \bar{\delta}}{2} \right), & \text{otherwise.} \end{cases} \quad (4)$$

After retransmission timer expiration, the values of  $cwnd$  and  $ssthresh$  are set as  $W = 1$  and  $ssthresh_{new} = FlightSize / 2$ , and slow start phase begins. Here  $FlightSize$  denotes the number of outstanding segments (i.e., that have been sent but not yet acknowledged) when fast retransmit/fast recovery phase was exited. Although new value of  $ssthresh$  may slightly differ from the value when fast recovery was entered, we found that this new value can be safely approximated by (1). The number of segments sent during slow start phase of the  $i$ -th cycle can be closely approximated by a geometric series, i.e.  $Y_i^{SS} = 1 + 2 + 2^2 + \dots + 2^{N_i-1} = 2^{N_i} - 1$ . Consequently, the required number of slow start rounds to send  $Y_i^{SS}$  segments can be expressed as  $N_i = \log_2(Y_i^{SS} + 1)$ . Since in slow start phase of the  $i$ -th cycle  $cwnd$  increases exponentially from 1 up to  $W_{i-1}/2$ , we get that  $W_{i-1}/2 = 2^{N_i-1}$  and  $N_i = \log_2(W_{i-1})$ . As in [8], we incorporate the last round of slow start phase into congestion avoidance phase, so we need to extract it from slow start phase. Thus, we obtain

$$E[A^{SS}] = \log_2(E[W]) - 1 \quad (5)$$

and

$$E[Y^{SS}] = 2^{E[A^{SS}]} - 1 = \frac{E[W]}{2} - 1. \quad (6)$$

Once the value of  $ssthresh$  is reached, the sender enters congestion avoidance phase. In this phase  $cwnd$  grows linearly with a slope of  $1/b$  segments per round until the first segment loss occurs. The value of  $b$  depends on the acknowledgement strategy of the receiver:  $b = 1$  if the receiver immediately acknowledges segments or

$b = 2$  if the receiver uses delayed acknowledgements algorithm as recommended in [12]. Let us denote by  $\alpha_i$  the first lost segment in the  $i$ -th cycle and by  $X_i$  the round of congestion avoidance phase where this loss occurs (see Fig. 1b). According to the sliding window algorithm, after the segment  $\alpha_i$ ,  $(W_i - 1)$  more segments are sent before the loss will be detected and the  $i$ -th cycle ends. Therefore, altogether  $Y_i = \alpha_i + W_i - 1$  segments will be sent during the  $i$ -th cycle. On the other hand, the total number of segments sent during  $A_i = A_i^{FR} + A_i^{SS} + A_i^{CA}$  rounds of the  $i$ -th cycle can be found as  $Y_i = Y_i^{FR} + Y_i^{SS} + Y_i^{CA}$ . It follows that

$$\begin{cases} E[Y] = E[\alpha] + E[W] - 1 \\ E[Y] = E[Y^{FR}] + E[Y^{SS}] + E[Y^{CA}] \end{cases} \quad (7)$$

The expected number of segments sent in a cycle up to and including the first lost segment is given in [11] as

$$E[\alpha] = P[\alpha = k]k = \sum_{k=1}^{\infty} (1-p)^{k-1} pk = \frac{1}{p}, \quad (8)$$

where  $p$  is the probability of the first segment loss in a cycle. In other words, parameter  $p$  captures the loss event rate, where a loss event corresponds to one or more segment losses within a round.

The number of rounds in congestion avoidance phase of the  $i$ -th cycle can be defined as  $A_i^{CA} = X_i + 1$ . To simplify the subsequent derivation, we assume that  $W_{i-1}/2$  and  $X_i/b$  take on only integer values, while  $\{X_i\}_i$  and  $\{W_i\}_i$  are mutually independent sequences of i.i.d. random variables. Then

$$E[A^{CA}] = E[X] + 1 = b \left( \frac{E[W]}{2} + 1 \right) + 1 \quad (9)$$

and

$$E[Y^{CA}] = \frac{3bE[W]}{4} \left( \frac{E[W]}{2} + 1 \right) + E[W] - \bar{\delta}. \quad (10)$$

Solving (7) for  $E[W]$  and taking into account that  $E[W]$  is positive by definition, we get

$$E[W] = \begin{cases} -\left(\frac{3b+2\tau+4}{3b}\right) + \sqrt{\frac{8}{3bp} + \left(\frac{3b+2\tau+4}{3b}\right)^2 + \frac{4(4\bar{\delta}+2\bar{\delta}\tau-\tau^2-3\tau-2)}{3b}}, & \text{if } \bar{\delta} \leq \frac{E[W]}{2}, \\ -\left(\frac{3b+2\tau+3-2\bar{\delta}}{3b+1}\right) + \sqrt{\frac{8}{p(3b+1)} + \left(\frac{3b+2\tau+3-2\bar{\delta}}{3b+1}\right)^2 + \frac{4(3\bar{\delta}+2\bar{\delta}\tau-\tau^2-\bar{\delta}^2-3\tau-2)}{3b+1}}, & \text{otherwise.} \end{cases} \quad (11)$$



Combining (2), (4), (5), (6), (8), (9), and (10), we obtain

$$B = \frac{1/p + E[W] - 1}{RTT \left( \tau + 1 + \log_2(E[W]) + b \left( \frac{E[W]}{2} + 1 \right) \right)}, \quad (12)$$

where  $E[W]$  is given in (11).

According to [3], at any given time the usable window size (the amount of data that can be sent without waiting for an ACK) must be set as  $\min(cwnd, rwnd)$ . Up to now we have supposed that the send rate is only limited by  $cwnd$ , which implies that  $cwnd < rwnd$ . This assumption holds if the receiver buffer is sufficiently large and losses are frequent enough to bind evolution of  $cwnd$  below the buffer size of the receiver. The TCP window field [13] is limited to 16 bits, allowing for the maximum advertised buffer size of 65535 bytes. TCP window scale option [14] expands the definition of the TCP receiver window to 32 bits by implementing a scaling factor, which increases the maximum receiver window size from 64 KB up to 1 Gigabyte. However, due to many different reasons (e.g., incorrect TCP tuning [15], misbehaving routers [16], etc.) the send rate still can be limited by inappropriately small value of  $rwnd$  [17]. Obviously, a realistic model should take into account the impact of receiver window size on TCP performance.

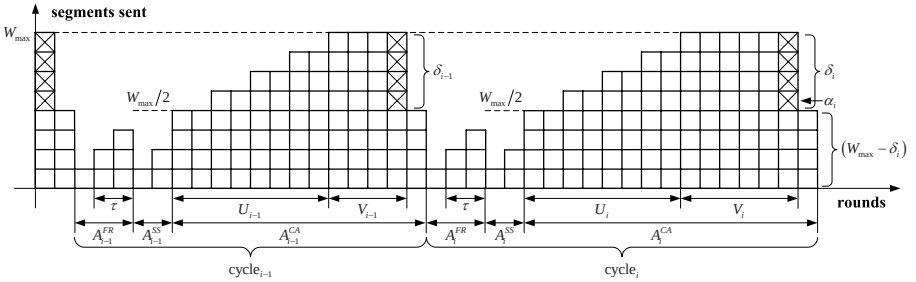


**Fig. 2.** Evolution of congestion window size limited by the receiver

Let us denote by  $W_{\max}$  the maximum receiver window size (in segments). During the loss-free period  $cwnd$  grows up to  $W_{\max}$  and then remains constant and equal to this value. Fig. 2 presents an example of evolution of congestion window size limited by the receiver window. Similarly to [11], we assume that if  $E[W] < W_{\max}$ , then the impact of receiver window size on the behavior of the sender is negligible and the steady state throughput can be found from (12). Now we need to define the steady state throughput for the case when  $E[W] \geq W_{\max}$ .

Consider the  $i$ -th cycle as in Fig. 2. The cycle starts after the loss detection via “triple-duplicate” ACKs and the sender enters fast retransmit/fast recovery phase. This phase lasts until retransmission timer expiration, then the sender moves into slow start phase with an exponential growth of  $cwnd$  from 1 up to  $W_{\max}/2$  segments. In the subsequent congestion avoidance phase, the window size increases linearly with a

slope of  $1/b$  segments per round during the first  $U_i$  rounds, after that the window size stays constant for the next  $V_i$  rounds. Taking into consideration that  $W_{\max} = W_{\max}/2 + U_i/b$ , we can define the number of segments sent during congestion avoidance phase of the  $i$ -th cycle as

$$Y_i^{CA} = \sum_{k=0}^{U_i-1} \left( \frac{W_{\max}}{2} + k \right) b + V_i W_{\max} + W_{\max} - \delta_i = \frac{U_i}{2} \left( \frac{3W_{\max}}{2} - 1 \right) + V_i W_{\max} + W_{\max} - \delta_i, \quad (13)$$

where, for simplicity, we assumed that  $W_{\max}/2$  and  $U_i/b$  are integers.

Then  $E[U] = bW_{\max}/2$  and  $E[Y^{CA}]$  is given as

$$E[Y^{CA}] = \frac{3b(W_{\max})^2}{8} - \frac{bW_{\max}}{4} + E[V]W_{\max} + W_{\max} - \bar{\delta}. \quad (14)$$

Since the number of segments sent in the  $i$ -th cycle up to the first lost segment  $\alpha_i$  (see Fig. 2) depends on value of  $p$ , rather than window limitation, based on (4), (6), (7), and (14) we can define the following system of equations:

$$\begin{cases} E[Y] = \frac{1}{p} + W_{\max} - 1 \\ E[Y] = E[Y^{FR}] + \frac{W_{\max}}{2} - 1 + \frac{3b(W_{\max})^2}{8} - \frac{bW_{\max}}{4} + E[V]W_{\max} + W_{\max} - \bar{\delta} \end{cases} \quad (15)$$

where  $E[Y^{FR}]$  can be obtained from (4) by replacing  $E[W]$  by  $W_{\max}$ .

Solving (15) for  $E[V]$ , we get the expected number of rounds when the window size remains constant:

$$E[V] = \begin{cases} \frac{2 - 2p - 3\tau p - \tau^2 p + 2\bar{\delta}\tau p + 4\bar{\delta}p}{2pW_{\max}} - \frac{\tau}{2} - 1 - \frac{3bW_{\max}}{8} + \frac{b}{4}, & \text{if } \bar{\delta} \leq \frac{W_{\max}}{2}, \\ \frac{2 - 2p - 3\tau p - \bar{\delta}^2 p + 2\bar{\delta}\tau p + 3\bar{\delta}p - \tau^2 p}{2pW_{\max}} - \frac{\tau}{2} - \frac{3}{4} - \frac{3bW_{\max}}{8} + \frac{b}{4} - \frac{W_{\max}}{8} + \frac{\bar{\delta}}{2}, & \text{otherwise.} \end{cases} \quad (16)$$

Then the expected number of rounds in a cycle can be found as

$$E[A] = E[A^{FR}] + E[A^{SS}] + E[A^{CA}] = \tau + 1 + \log_2 \left( \frac{W_{\max}}{2} \right) + E[U] + E[V] + 1. \quad (17)$$

Combining (15), (16), and (17), we obtain the steady state throughput of the Impatient variant of TCP NewReno when  $E[W] \geq W_{\max}$ :

$$B = \frac{1/p + W_{\max} - 1}{RTT \left( \tau + 1 + \log_2(W_{\max}) + \frac{bW_{\max}}{2} + E[V] \right)}. \quad (18)$$

### 3.2 The Complete Model

Now we can draw together analytical models of steady state throughputs of the Slow-but-Steady and Impatient variants of TCP NewReno. We also extended the earlier obtained model of the Slow-but-Steady variant [8] to capture the impact of window limitation:

$$B^{SBS} = \begin{cases} \frac{1/p + E[W^{SBS}] - 1}{\overline{RTT} \left( \bar{\delta} + b \left( \frac{E[W^{SBS}]}{2} + 1 \right) + 1 \right)}, & \text{if } E[W^{SBS}] < W_{\max}, \\ \frac{1/p + W_{\max} - 1}{\overline{RTT} \left( \bar{\delta} + \frac{bW_{\max}}{2} + E[V^{SBS}] + 1 \right)}, & \text{if } E[W^{SBS}] \geq W_{\max}, \end{cases} \quad (19)$$

where:

$$E[W^{SBS}] = \begin{cases} -\left( \frac{3b+2\bar{\delta}}{3b} \right) + \sqrt{\frac{8}{3bp} + \frac{4(\bar{\delta}^2 + \bar{\delta} - 2)}{3b} + \left( \frac{3b+2\bar{\delta}}{3b} \right)^2}, & \text{if } \bar{\delta} \leq \frac{E[W^{SBS}]}{2}, \\ -\left( \frac{3b-1}{3b+1} \right) + \sqrt{\frac{8(1-p)}{p(3b+1)} + \left( \frac{3b-1}{3b+1} \right)^2}, & \text{otherwise,} \end{cases} \quad (20)$$

and

$$E[V^{SBS}] = \begin{cases} \frac{1-p}{pW_{\max}} + \frac{\bar{\delta}^2 + \bar{\delta} - \bar{\delta}W_{\max}}{2W_{\max}} + \frac{b}{4} - \frac{3bW_{\max}}{8}, & \text{if } \bar{\delta} \leq \frac{W_{\max}}{2}, \\ \frac{1-p}{pW_{\max}} + \frac{2b - W_{\max}}{8} + \frac{1}{4} - \frac{3bW_{\max}}{8}, & \text{otherwise.} \end{cases} \quad (21)$$

Combining (12), (18), and (19), we get the steady state throughput of the Impatient variant of TCP NewReno as

$$B^{IMP} = \begin{cases} B^{SBS}, & \text{if } \bar{\delta} < \tau + 1 \\ \frac{1/p + E[W^{IMP}] - 1}{\overline{RTT} \left( \tau + 1 + \log_2(E[W^{IMP}]) + b \left( \frac{E[W^{IMP}]}{2} + 1 \right) \right)}, & \text{if } \bar{\delta} \geq \tau + 1 \wedge E[W^{IMP}] < W_{\max} \\ \frac{1/p + W_{\max} - 1}{\overline{RTT} \left( \tau + 1 + \log_2(W_{\max}) + \frac{bW_{\max}}{2} + E[V^{IMP}] \right)}, & \text{if } \bar{\delta} \geq \tau + 1 \wedge E[W^{IMP}] \geq W_{\max}, \end{cases} \quad (22)$$

where  $B^{SBS}$  is given (19);  $E[W^{IMP}]$  is given in (11);  $E[V^{IMP}]$  is given in (16).

## 4 Numerical Results

Armed with expressions of steady state throughputs of the Impatient and Slow-but-Steady variants, we can perform an analytical comparison of these variants over different values of  $\bar{\delta}$ ,  $p$ ,  $\tau$ , and  $W_{\max}$ . Note that the latter parameter not only captures the impact of receiver buffer on TCP NewReno performance, but also allows us to put an upper bound on the maximum value of  $\bar{\delta}$ : since the number of segments transmitted in each RTT must be no more than  $W_{\max}$  and we assume that all loss detections are exclusively due to “triple-duplicate” ACKs, then at any given time  $\delta \leq W_{\max} - 3$ . Fig. 3 presents steady state throughput of the Impatient variant as a function of  $\bar{\delta}$  and  $p$  (Fig. 3a) and difference between steady state throughputs of the Impatient and Slow-but-Steady variants (Fig. 3b). We have assumed that  $\tau = 4$  (as suggested in [18]),  $b = 2$ ,  $rwnd = 64$  KB, and  $MSS = 1460$  bytes – the most typical value in today’s Internet. In practice, instead of using a hard-coded receiver window size, TCP adjusts it to even increments of MSS negotiated during connection establishment [19]. Then we have  $W_{\max} = 44$  segments.

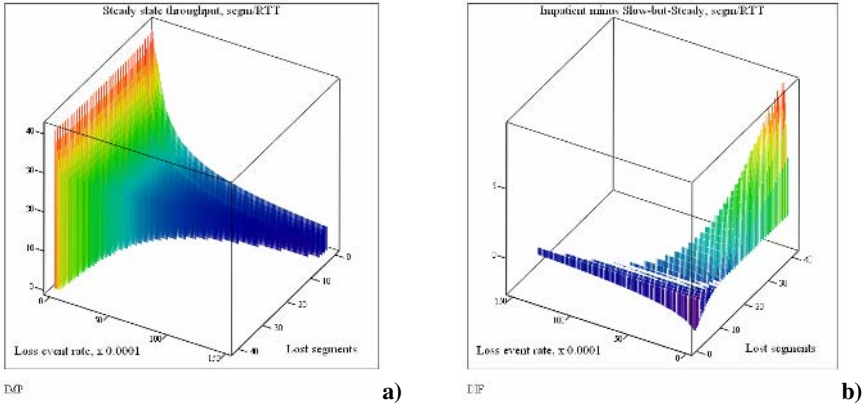


**Fig. 3.** Steady state throughput of the Impatient variant and difference between throughputs of the Impatient and Slow-but-Steady variants

As it follows from Fig. 3b, when the average loss burst length is less than 5 the throughputs of the both variants are identical as long as the sender can recover all lost segments before expiration of the retransmission timer which was reset after the first partial acknowledgement. But when the average number of lost segments per congestion event is greater than 5 the difference between the variants takes place. As it was noted in [4], neither of the two variants is optimal. When the number of lost segments is small the performance would have been better without invocation of slow start algorithm. At the same time, when the number of lost segments is sufficiently large, the Impatient variant gives a faster recovery and better performance and the gain increases with the average loss burst length. However, one can easily observe that the difference between throughputs shows a complex behavior as a function of  $\bar{\delta}$ ,  $p$ ,

and  $\tau$ . To get a clear picture of what is happening with steady state throughput of the Impatient variant, we should study its sensitivity to different protocol settings and operational conditions. On the first step we set  $\tau = 4$ ,  $b = 2$ ,  $MSS = 1460$  bytes, and vary receiver window size (assuming TCP window scale option is enabled). Fig. 4 presents difference between steady state throughputs of the Impatient and Slow-but-Steady variants normalized to steady state throughput of the Slow-but-Steady variant.

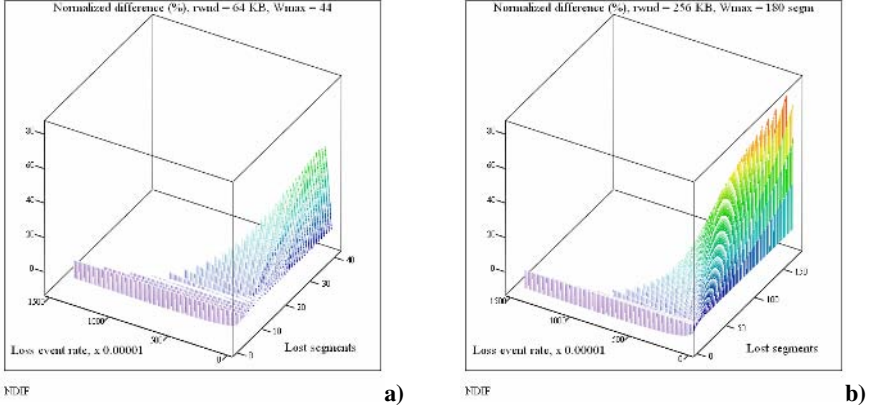
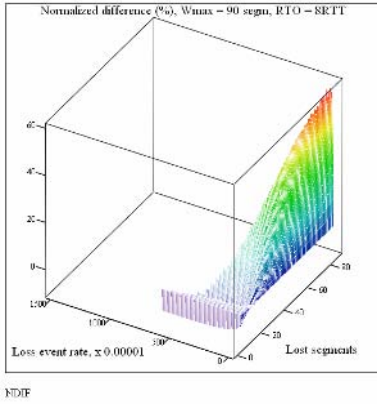


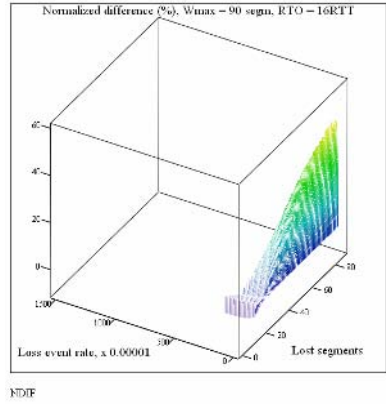
**Fig. 4.** Normalized difference when the maximum receiver window size is: a) 64 KB; b) 256 KB

Our analysis shows that for given values  $\tau$  and  $b$  the maximum gain of the Slow-but-Steady variant is constant and equals to 10.6%, while the maximum gain of the Impatient variant increases with the maximum receiver window size (e.g., 47.6% for 64 KB and 81.0% for 256 KB), as long as larger  $rwnd$  means more outstanding segments which can be theoretically dropped in case of congestion along the path. However, the Impatient variant outperforms the Slow-but-Steady one only when the average loss burst length is more than 10 segments. As was pointed out in [9], while the right tail of loss burst lengths distribution can be fairly long and reach burst length over 100 packets, the majority of losses are single packet losses. Then we can expect that in the vast majority cases the Impatient variant will behave like the Slow-but-Steady one (since all lost segments can be recovered within several rounds before RTO expiration).

On the second step we consider steady state throughput of the Impatient variant as a function of  $\tau$  value. Commonly, TCP implementations use coarse-grained retransmission timer, having granularity of 500 ms. Moreover, current standard [20] requires that whenever RTO is computed, if it is less than 1 second then the RTO should be rounded up to 1 second. At the same time some implementations use fine-grained retransmission timer and do not follow requirements of [20] by allowing, for example, the minimum limit of 200 ms [21]. Consequently, for the same path  $\tau$  can vary greatly from one TCP implementation to another. Fig. 5 presents the normalized difference between steady state throughputs of the Impatient and Slow-but-Steady variants for different values of  $\tau$  ( $rwnd$  is 128 KB).



a)



b)

**Fig. 5.** Normalized difference when RTO is equal to: a) 8 RTT; b) 16 RTT

Fig. 5 shows that the use of fine-grained retransmission timer provides higher steady state throughput in case of large windows and multiple losses, while the gain of the Impatient variant with coarse-grained retransmission timer in that case will be substantially smaller due to very lengthy fast recovery phase. Taking into account the prevalence of single segment losses [9], we also can expect that in most cases the Impatient variant with coarse-grained retransmission timer will behave like the Slow-but-Steady one.

## 5 Conclusion

In this paper, we developed an analytical model of steady state throughput of the Impatient variant of TCP NewReno. The proposed model provides the possibility to study TCP NewReno performance over the entire range of operational conditions and protocol parameter settings. We then make an analytical comparison of steady state throughputs of the Slow-but-Steady and Impatient variants. Our analysis shows that the Impatient variant provides approximately the same steady state throughput as the Slow-but-Steady one in a wide range of network conditions and significantly outperforms it in case of large windows and bursty losses. This could be extremely useful for networks with large bandwidth and long delay. However, we can expect that under normal operational conditions there will be no difference between the Impatient and Slow-but-Steady variants since in most cases all lost segments can be recovered in the Slow-but-Steady mode. Nevertheless, our recommendation is for the Impatient variant as a backup mechanism for extreme scenarios with multiple packet drops.

**Acknowledgments.** The work presented in this paper has been supported by ESF COST 290 and Nokia Foundation.

## References

1. Medina, A., Allman, M., Floyd, S.: Measuring the Evolution of Transport Protocol in the Internet. ACM SIGCOMM (2004) 336-341
2. Stevens, W.: TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms. RFC 2001 (1997)
3. Allman, M., Paxson, V., Stevens, W.: TCP Congestion Control. RFC 2581 (1999)
4. Floyd, S., Henderson, T., Gurtov, A.: The NewReno Modification to TCP's Fast Recovery Algorithm. RFC 3782 (2004)
5. Khalifa, I., Trajkovic, L.: An Overview and Comparison of Analytical TCP Models. IEEE ISCAS, Vol. 5 (2004) 469-472
6. Olsen, J.: Stochastic Modeling and Simulation of the TCP Protocol. Ph.D. Thesis, Uppsala University, Sweden (2003)
7. Parvez, N., Mahanti, A., Williamson, C.: TCP NewReno: Slow-but-Steady or Impatient? IEEE ICC (2006)
8. Dunaytsev, R., Koucheryavy, Y., Harju, J.: TCP NewReno Throughput in the Presence of Correlated Losses: The Slow-but-Steady Variant. IEEE INFOCOM Global Internet Workshop (2006) 115-120
9. Loguinov, D.: Adaptive Scalable Internet Streaming. Ph.D. Thesis, The City University of New York, the USA (2002)
10. Brandauer, C., Iannaccone, G., Diot, C., Ziegler, T., Fdida, S., May, M.: Comparison of Tail Drop and Active Queue Management Performance for Bulk-Data and Web-Like Internet Traffic. ISCC (2001) 1-22
11. Padhye, J., Firoiu, V., Towsley, D., Kurose, J.: Modeling TCP Reno Performance: A Simple Model and Its Empirical Validation. IEEE/ACM Transactions on Networking, Vol. 8, No. 2 (2000) 133-145
12. Braden, R. (ed.): Requirements for Internet Hosts. RFC 1122 (1989)
13. Postel, J. (ed.): Transmission Control Protocol. RFC 793 (1981)
14. Jacobson, V., Braden, R., Borman, D.: TCP Extensions for High Performance. RFC 1323 (1992)
15. Weigle, E., Feng, W.-C.: A Comparison of TCP Automatic Tuning Techniques for Distributed Computing. IEEE HPDC (2002) 265-272
16. Online article: TCP Window Scaling and Broken Routers. <http://lwn.net/Articles/92727/>
17. Grossman, R., Gu, Y., Hanley, D., Hong, X., Krishnaswamy, P.: Experimental Studies of Data Transport and Data Access of Earth Science Data over Networks with High Bandwidth Delay Products. Computer Networks, Vol. 46 (2004) 411-421
18. Handley, M., Floyd, S., Padhye, J., Widmer, J.: TCP Friendly Rate Control (TFRC): Protocol Specification. RFC 3448 (2003)
19. MacDonald, D., Barkley, W.: Microsoft Windows 2000 TCP/IP Implementation Details. Microsoft white paper (2000)
20. Paxson, V., Allman M.: Computing TCP's Retransmission Timer. RFC 2988 (2000)
21. Sarolahti, P., Kuznetsov, A.: Congestion Control in Linux TCP. USENIX/FREENIX Track (2002) 49-62

# SCTP Performance Issue on Path Delay Differential\*

Yuansong Qiao<sup>1,2,3</sup>, Enda Fallon<sup>1</sup>, Liam Murphy<sup>4</sup>, John Murphy<sup>4</sup>, Austin Hanley<sup>1</sup>,  
Xiaosong Zhu<sup>1</sup>, Adrian Matthews<sup>1</sup>, Eoghan Conway<sup>1</sup>, and Gregory Hayes<sup>1</sup>

<sup>1</sup> Applied Software Research Centre, Athlone Institute of Technology, Ireland

<sup>2</sup> Institute of Software, Chinese Academy of Sciences, China

<sup>3</sup> Graduate University of Chinese Academy of Sciences, China

<sup>4</sup> Performance Engineering Laboratory, University College Dublin, Ireland  
{ysqiao, efallon}@ait.ie, {Liam.Murphy, j.murphy}@ucd.ie,  
{ahanley, fzhu, amathews, econway, ghayes}@ait.ie

**Abstract.** This paper studies the effect of path delay on SCTP performance. It focuses on the SCTP fast retransmit algorithm and demonstrates that the performance in the current retransmission strategy will degrade acutely when the secondary path delay is less than the primary path delay at a certain level. The performance degradation is due to the disordered SACKs and constant congestion window size during the fast retransmit phase. Some modifications aimed at these problems are proposed and evaluated. This paper also identifies that the cause of the performance degradation in SCTP is a result of the single path configuration oriented design of the current fast retransmit algorithm. Several fast retransmission strategies are evaluated for different path delay and bandwidth configurations.

**Keywords:** SCTP, Multi-homing, Retransmission strategy, Path difference.

## 1 Introduction

Multi-homing technologies, where a host can be addressed by multiple IP addresses, are increasingly being considered by developers implementing mobile applications. An enabling factor reinforcing this adoption is the trend towards mobile devices supporting a hybrid of networking capabilities such as 802.11 and UMTS. The characteristics of mobile environments, with the possibility of frequent disconnections and fluctuating bandwidth, pose significant issues for mobile application developers and therefore the path redundancy offered by multi-homing protocols has a clear attraction.

The traditional transport layer protocols, such as TCP and UDP, only support one IP address at each endpoint in one connection. Thus there is much effort in the designing of multi-homing protocols. Stream Control Transmission Protocol (SCTP) [1] is the most mature one currently. It is a reliable transport layer protocol and employs a similar congestion control mechanism to TCP. It also introduces some

---

\* The authors wish to recognize the assistance of Enterprise Ireland through its Innovation Partnership fund in the financing of this Research programme.



attractive features which TCP does not support such as message oriented, multi-homing and multi-streaming. Two extensions for mobile environments have been proposed [2] [3] to address seamless handover for mobile clients.

This paper studies the effect of path delay on the performance of SCTP. It illustrates the performance degradation in the SCTP fast retransmit phase when the delay of the secondary path is shorter than that of the primary path. A modification for the fast retransmit algorithm is proposed to address this problem. Two burst limit algorithms are evaluated in the context of path delay difference. We also evaluated fast retransmission on the same path and fast retransmission on an alternative path with the above modifications for different path delay and bandwidth configurations.

This paper is organized as follows. Section 2 summarizes related work. Section 3 introduces the current SCTP fast retransmit algorithm. Section 4 illustrates the simulation setup. Section 5 describes the SCTP performance degradation problem in detail. Section 6 presents modifications to current fast retransmit algorithm and compares different retransmission strategies. Conclusions are presented in Section 7.

## 2 Related Work

SCTP originated as a protocol called Multi-Network Datagram Transmission Protocol (MDTP). The motivation for MDTP arose from the fact that TCP had inherent weaknesses in relation to the control of telecommunication sessions. MDTP was designed to transfer call control signalling on “carefully engineered” networks [4]. When one analyses the origins of SCTP it is interesting to note that its initial target environment was vastly different from that experienced in present day mobile networks. Given its origin as a fixed line oriented protocol, and in particular a protocol designed towards links with roughly equivalent transmission capabilities, the transition towards a mobile enabled protocol has raised a number of design issues. Many related works have raised issues in relation to the design of SCTP.

In [5] two SCTP stall scenarios are presented, the authors identify that the stalls occur as a result of SCTP coupling the logic for data acknowledgment and path monitoring. In [6] different SCTP retransmission policies are investigated for a lossy environment, a retransmission strategy which sends the fast retransmission packets on the same path and the timeout retransmission packets on an alternate path are suggested. In [7] SCTP is extended for Concurrent Multipath Transfer (CMT-SCTP) while in [8] the authors identify that a finite receiver buffer will block CMT-SCTP transmission when the quality of one path is lower than others. Several retransmission policies are studied which can alleviate receiver buffer blocking. In [9] the authors focus on making SCTP robust to packet reordering and delay spikes.

## 3 Current Fast Retransmit Algorithm

The SCTP [1] congestion algorithms are inherited from SACK TCP [10], which include slow start, congestion avoidance and fast retransmit. In [11], the authors present a detailed comparison between the congestion algorithms of SCTP and TCP. The fast retransmit algorithm concerned in this paper is based on [1] with the fast

recovery extension defined in [12], [13] and [14] which is derived from NewReno TCP [15]. The original SCTP [1] fast retransmit algorithm improperly decreases performance when multiple packets are lost in one window [13]. The rest of this section will introduce this algorithm in detail.

When the receiver receives an out of sequence data chunk, i.e. its TSN (Transmission Sequence Number) is greater than the latest Cumulative TSN, the receiver reports this situation to the sender immediately by including the TSN which has been received in the Gap Ack Block of a SACK.

The sender maintains a “potential missing reports” counter for missed data chunks. The missing report counter is increased via the HTNA (Highest TSN Newly Aacked) algorithm. When a SACK is received, the counters for the lost TSNs below the highest newly acknowledged TSN are incremented. When the “potential missing reports” counter reaches four the sender assumes that the data chunk has been lost. In this scenario the following parameter values of the primary path are set:

$$ssthresh = \max(cwnd / 2, 2 \times MTU), \quad cwnd = ssthresh$$

In the original SCTP [1], the sender reduces *cwnd* (Congestion Window) and *ssthresh* (Slow Start Threshold) for every packet loss detected from SACK even when these packets are lost in one window, which is more conservative than TCP [13]. In [12], [13] and [14], the authors introduce a fast recovery phase to improve SCTP performance. The fast recovery phase begins when the fast retransmit phase starts. When the sender enters the fast recovery phase, it saves the highest outstanding TSN via a variable *recover*. When the Cumulative TSN ACK point passes *recover*, the fast recovery phase is finished. During the fast recovery phase, only the first lost packet causes *cwnd* reduction. Afterwards, the *cwnd* is not changed until the fast recovery phase finishes.

Every time the sender finds a data chunk lost via SACKs, it will select a path to fast retransmit the data chunk immediately. In base SCTP, fast retransmission on an alternate path is recommended, whereas in [6], fast retransmission on the same path is suggested. If multiple data chunk losses are detected at the same time, the sender will only send one packet via the fast retransmit algorithm. The rest of the lost data chunks will be retransmitted when the path *cwnd* allows. After all the lost chunks have been retransmitted, the sender will send new data chunks on the primary path if the primary path *cwnd* allows. As long as the congestion window is not full, the sender can continuously send new data.

During the fast retransmit and fast recovery phase, the *cwnd* of every path is a constant, which will decrease performance when retransmitting on the secondary path. Section 5 will discuss these performance issues in detail.

## 4 Simulation Setup

The simulations focus on the situation where a mobile node has two WIFI, 3G or GPRS connections and one of two paths has various delay configurations. The path with constant delay is set to primary path in SCTP. All simulations in this paper are carried out by running a revision of Delaware University's SCTP module [16] for

NS-2 [17]. Some small bugs about transmission timer management and bursting limit in NS-2 SCTP module have been corrected.

The simulation topology is shown in Figure 1. Node S and Node R are SCTP sender and receiver respectively. Both SCTP endpoints have two addresses. R1,1, R1,2, R2,1 and R2,2 are routers. The implementation is configured with no overlap between the two paths. As only the effect of delay is considered in this paper the loss rate is set to zero. Node S begins to send 20MB ftp data to Node R at the 5th second. The MTU (Maximum Transmission Unit) of every path is 1500B. The queue lengths of bottleneck link in both paths are 50 packets. The queue lengths of other links are set to 10000 packets. The bandwidths of two bottleneck links are 10Mbps, 384Kbps or 36Kbps. The delay of the secondary path bottleneck link changes from 1ms to 1000ms. SCTP parameters are all default except those mentioned. Initially the receiver window is set to 10MB (infinite). The initial slow start threshold is set to 1MB which is large enough to ensure that the full primary path bandwidth is used. Only one SCTP stream is used and the data is delivered to the upper layer in order.

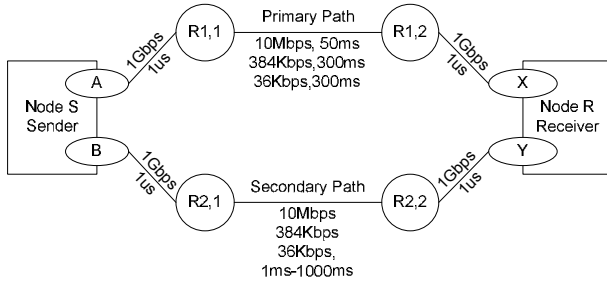


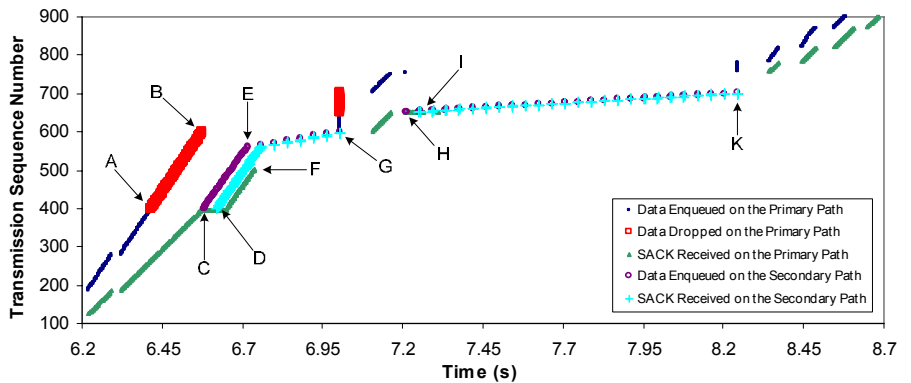
Fig. 1. Simulation network topology

## 5 Effect of Delay on Performance

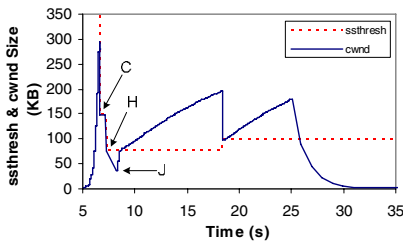
This section describes the SCTP performance issues in detail via two simulations. In both simulations, the two path bandwidths are 10Mbps, the primary path delay is 50ms. The secondary path delay is set to 20ms in the first simulation and 50ms in the secondary simulation. Node S starts to send 20MB data to Node R at the 5th second. The lost packets are retransmitted on the secondary path without burst limit, which is the default strategy defined in [1]. The data transmission time is 20.0322s in the first simulation and 18.0671s in the second simulation.

### 5.1 Test 1: The Secondary Path Delay Is 20ms

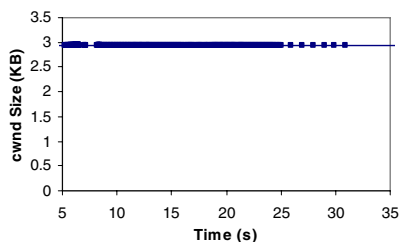
This section reviews the first simulation to illustrate the SCTP abnormal behaviour during data transmission (Figure 2). The sending process begins in slow start mode. From A to B (Figure 2a), 204 packets are sent to the primary path, and among these packets, 68 packets are dropped evenly because of congestion.



(a) Packet trace on both paths



(b) cwnd and ssthresh of the primary path



(c) cwnd of the secondary path

**Fig. 2.** The secondary path delay is 20ms. A: 6.4149s, B: 6.5757s, C: 6.5793s, D: 6.6207s, E: 6.7178s, F: 6.7378s, G: 7.0045s, H: 7.2106s, I: 7.2531s, J: 8.2094s, K: 8.2424s.

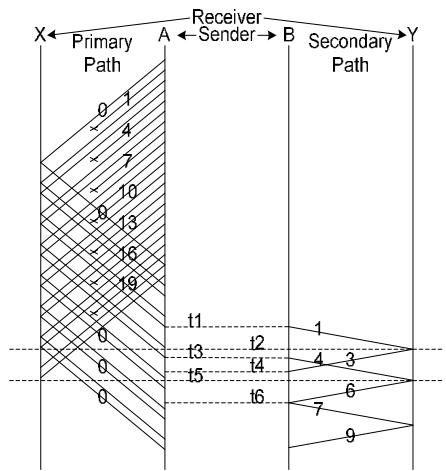


**Fig. 3.** Message sequence for fast retransmission. The numbers on the SACK lines are Cumulative TSNs, others are TSNs.

**Table 1.** Missing reports in the sender

| TSN  | 1 | 4 | 7 | 10 | 13 | 16 | 19 |
|------|---|---|---|----|----|----|----|
| Time |   |   |   |    |    |    |    |
| t1   | 4 | 2 | 0 | 0  | 0  | 0  | 0  |
| t3   | 6 | 4 | 2 | 0  | 0  | 0  | 0  |
| t4   | 7 | 5 | 3 | 1  | 1  | 1  | 0  |
| t6   | 8 | 6 | 4 | 2  | 2  | 2  | 1  |

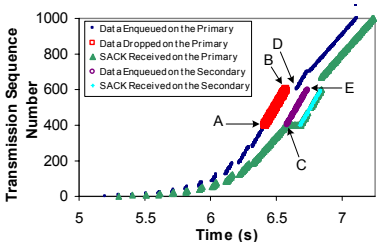


**Fig. 4.** The secondary path delay is 50ms. A: 6.4149s, B: 6.5757s, C: 6.5793s, D: 6.6586s, E: 6.7408s.

At C (Figure 2a,2b), the sender finds the packets lost by duplicate SACKs and it reduces the cwnd by half. From C to G (Figure 2a), the sender retransmits lost packets on the secondary path. Between C and E, the packets are sent by the fast retransmit algorithm. Between E and G, the packets are sent under the constraint of the secondary path cwnd.

From C to D, the fast retransmission is triggered by the SACKs received from the primary path. The lost packets are found one by one because they are not lost consecutively.

At D, the sender receives the first SACK from the secondary path. Multiple TSN gaps are found simultaneously in this SACK because the SACK arrives in advance due to the secondary path delay being less than the primary path delay. The missing reports for the corresponding TSN are set to one. When the missing reports for these lost data chunks reach four, only the first one in current retransmission buffer is fast retransmitted. The rest of the data chunk will be retransmitted when the cwnd allows. Because one SACK can only trigger one fast retransmission, the data chunks that are not fast retransmitted will be accumulated in the sender's retransmission buffer until the secondary path cwnd allows.

From D to F, all the SACKs received from the primary path are abandoned by the sender because the SACKs received from the secondary path have higher Cumulative TSN Ack value. Therefore the fast retransmission is triggered by the SACKs from the secondary path.

At E (Figure 2a), the fast retransmission finishes because all the lost packets are found and their missing reports all exceed four. There are still 13 packets left in the retransmission buffer at this moment. Because the outstanding data size of the secondary path is greater than the cwnd ( $2 \times \text{MTU}$ ) which is not changed during fast recovery (Figure 2c), the sender must wait and can not send any data out on the secondary path. From F to G, although the SACKs for all data chunks sent on the primary path are received, the sender still can not send data on the primary path because the retransmission buffer is not empty.

At G, the last lost data chunk is retransmitted and the sender begins to send new data on the primary path. Because the outstanding data of the primary path are all acknowledged or retransmitted, the sender can send a whole cwnd size data out. In this example, 101 packets are sent at the same time, and the top 51 packets are dropped by the network because the buffer on the bottleneck link is full. From H to K, the sender begins a new fast retransmit and fast recovery phase. Only one packet is fast retransmitted at H and the others are retransmitted under the constraint of the secondary path cwnd. At I, the sender receives the first SACK from the secondary path. From I to K, the sender neglects all the SACKs of the primary path. At J (between I and K), the sender decreases cwnd by half because the primary path has been idle for a RTO (Retransmission TimeOut) time. At K, all lost packets are retransmitted, and the sender sends another burst (26 packets) to the primary path as at G. After K, the transmission returns to normal.

Figure 3 presents an example to describe how the disordered SACKs result in the sender detecting multiple packet losses which are not lost consecutively. Table 1 lists the "missing report" value for lost TSNs (1~19) at different moments in the sender. At t1 (Figure 3 and Table 1), the missing report for TSN=1 reaches four. The sender fast retransmits this chunk (TSN=1) on the secondary path. At t2, the chunk arrives at the

receiver and the receiver sends a SACK to report its current receiving status. At  $t_4$ , the sender receives the SACK from the secondary path. It finds the data chunks with TSN=10,13,16 are lost and increments their missing reports. Afterwards, all SACKs received from the primary path are dropped by the sender because their Cumulative TSN Ack values are less than the sender's Cumulative TSN Ack point. The sender will use the SACKs received from the secondary path to increment missing reports.

## 5.2 Test 2: The Secondary Path Delay Is 50ms

This section explains the simulation results for the second simulation, as a comparison with the simulation in section 5.1. The packet trace is shown in Figure 4. The sender starts transmission at the 5th second in slow start mode. From A to B, 68 packets are dropped as a result of the sending speed reaching the maximum bandwidth. At C, the sender finds the packets lost by SACKs, and the congestion window of the primary path is reduced by half. From C to E, each dropped packet is fast retransmitted on the secondary path immediately when the sender receives four consecutive loss reports for it. After D, the sender begins to transmit new data on the primary path because the outstanding data size of the path is smaller than the congestion window size.

## 5.3 Summary and Analysis

The above tests show that SCTP performance decreases significantly when the secondary path delay is less than the primary path delay. The disordered SACKs caused by path delay difference make the sender detect multiple lost data chunks simultaneously which were not lost consecutively. These data chunks may block the sending of the primary path if they can not be sent out during the fast retransmission stage. At the same time, the packets sent on the primary path are acknowledged via the SACKs received from the secondary path and finally the full window of the primary path becomes empty. Therefore the sender can send out new data in a burst of a whole window size when the retransmission is finished, which may cause congestion again. Another reason for the performance degradation is that the congestion windows of all paths remain a constant during the fast recovery phase, which results in the retransmission on the secondary path ( $cwnd=2*MTU$ ) becoming ineffective.

This performance degradation comes from the SCTP design rationale. SCTP is not a load sharing protocol, so it does not send data on multiple paths simultaneously. It assumes sending data on different paths is similar to sending data on a single path with network anomalies, such as reordering or delay spikes. Consequently, it adopts the current TCP congestion control and fast retransmit algorithm without significant modifications. In single path configurations, network anomalies exist but happen randomly. In multi-homed environments, besides network anomalies, the paths differences are usually constants. Every time an alternate path is used, it will affect performance, and therefore performance degradation occurs frequently. Accordingly, path differences should be considered in the algorithm.

It can be expected that the problem described in Section 5.1 could be triggered when multiple packets are dropped in one window, especially when the packets are dropped evenly in one window. This happens in the transmission start phase because the

ssthresh value could be arbitrarily high and the sender uses the slow start algorithm to probe the available bandwidth. In the last RTT round of the slow start phase, approximately 1/3 packets are dropped. This problem is not likely happen in the congestion avoidance phase for the system with a constant bottleneck bandwidth because the cwnd is incremented every RTT round and therefore only a few packets could be dropped when the transmission speed exceeds the bottleneck bandwidth. But in the real system, the bottleneck bandwidth may change frequently. The problem can be triggered when the bottleneck bandwidth drops suddenly, such as when a new data stream joins the bottleneck, especially a UDP stream comes into the bottleneck.

## 6 Solutions

According to the analysis in the previous section, there are two factors that lead to SCTP performance degradation. The first factor is that the congestion parameters of all paths are constants during the fast recovery phase. The second factor is that the burst on the primary path after the fast recovery phase causes more congestion.

If the paths between two SCTP endpoints share the same bottleneck, increasing the congestion windows of the backup path during fast recovery may cause more severe congestion. Whereas if the paths do not share the same bottleneck, it is unnecessarily conservative to keep the congestion parameters of the backup path unchanged. Although this paper is focused on path delay difference, there is another reason to support this opinion.

Consider the situation when the primary path fails and the data is transmitting on the secondary path. If the primary path recovers from path failure, the new data will be transmitted on the primary path through the slow start algorithm. If the secondary path was in the fast recovery phase before the sender switched to the primary path, the sending speed of the primary path will be maintained at one MTU per RTT (Round Trip Time). The fast recovery phase finishes when the Cumulative TSN Ack point equals or exceeds the fast recovery exit point (*recovery*) [12]. If the fast retransmitted data is lost, the fast recovery phase will last for at least one RTO. Here we should point out that SCTP-bis [12] [14] does not define that fast recovery should exit when a transmission timeout occurs on the same path. However we adopt the rules defined for the NewReno TCP fast recovery [15]. The fast recovery phase of a path finishes when the Cumulative TSN Ack point passes the variable *recovery* or a transmission timeout of that path happens. In a lossy environment, packet loss or path failure will happen frequently. Therefore the fast recovery should only affect the path on which the fast retransmission is triggered. The congestion window of other paths should change according to their path conditions.

Consequently, the cwnd of one path will be adjusted according to the slow start algorithm or the congestion avoidance algorithm when the following conditions are true:

- (1) The received SACK has advanced the Cumulative TSN Ack point;
- (2) There are new data chunks that have been acknowledged for the path;
- (3) The path is not in the fast recovery phase.

In [14], a protocol parameter Max.Burst is employed to limit the maximum packet number that can be sent out at one time. The default value of Max.Burst is 4. Two

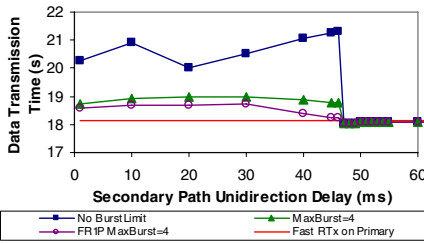




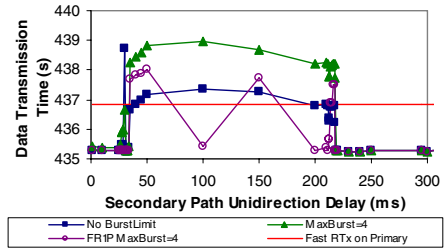
configuration, the data transmission time for the following retransmission strategies are computed:

- (1) Fast retransmission on the secondary path without burst limit;
- (2) Fast retransmission on the secondary path with maximum burst of four packets;
- (3) Fast retransmission on the secondary path with burst limit and the revised fast recovery algorithm (called FR1P in Figure 6);
- (4) Fast retransmission on the primary path.

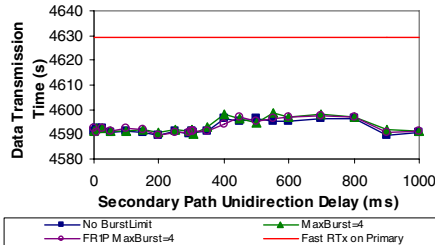
The results are presented in Figure 6. Figure 6a and 6b only show areas of major difference. No obvious changes are found outside the areas. First we discuss the strategies of retransmission on the secondary path, i.e. retransmission strategy (1), (2) and (3). The results indicate that obvious performance degradation occurs for the three retransmission strategies when the secondary path delay is lower than a certain threshold, approximately 47ms for the first simulation (10Mbps bandwidths, 50ms delay), 219ms for the second simulation (384Kbps bandwidths, 300ms delay) and 0ms for the third simulation (36Kbps bandwidths, 300ms delay). The reason has been explained in the previous section. When the secondary path delay is greater than the threshold, no significant performance degradation happens in these tests. The reason is that the SACKs received from the secondary path do not affect the packets lost pattern detected by the sender.



(a) 10Mbps bandwidths; the primary path delay is 50ms.



(b) 384Kbps bandwidths; the primary path delay is 300ms.



(c) 36Kbps bandwidths; the primary path delay is 300ms.

**Fig. 6.** Data transmission time

In test 1 (Figure 6a), the revised fast recovery algorithm (strategy 3) performs best among these three strategies. In test 2 (Figure 6b), the strategy of retransmission without burst limit performs best. The three retransmission strategies have similar performance in test 3 (Figure 6b). The reason is that higher bandwidth produces more data bursts during the fast recovery phase which will cause more packets to be lost. The revised fast recovery algorithm can avoid this congestion when the bandwidth is high; whereas when the bandwidth is low, the burst size is small and it does not cause congestion. Accordingly, retransmission without burst limit performs better when the path bandwidth is low. However transmission with burst limit is a reasonable choice because it can reduce the probability of network congestion.

The strategy of fast retransmission on the primary path can avoid the network anomalies introduced by path delay difference. It has better performance when the path bandwidth is high (10Mbps), whereas fast retransmission on the secondary path performs better when the path bandwidth is very low (384Kbps and 36Kbps), especially when the delay of the secondary path is greater than that of the primary path.

The above discussion is based on the infinite receiver buffer. If the receiver's buffer is finite, a long secondary path delay may cause receiver buffer blocking for the fast retransmission on the secondary path.

## 7 Conclusions and Future Work

This paper studies the effects of path delay on SCTP performance. It illustrates that the current SCTP fast retransmit algorithm decreases performance significantly when the secondary path delay is shorter than the primary path delay at a certain level.

The shorter secondary path delay causes the SACKs on the secondary path to arrive earlier than the SACKs on the primary path. The disordered SACKs cause the sender to detect multiple lost data chunks simultaneously which were not lost consecutively. These lost packets are marked for retransmission at the same time. SCTP can only fast retransmit one packet for each SACK. The rest of the packets will be retransmitted when the congestion window of the secondary path allows it. If these packets can not be sent out during fast retransmission, the sending on the primary path will be blocked even though the congestion window of the primary path allows transmission, which will also empty the congestion window of the primary path. When all the data chunks marked for retransmission are sent out, the sender may send a burst of packets into the primary path because of this empty window. The burst may cause network congestion once more.

Another reason for this performance degradation is that the congestion window of the secondary path is tied to the primary path congestion status. During the fast recovery phase, the congestion window of every path can not be changed. Normally, the secondary path congestion window is a small value, which means the data chunks marked for retransmission can not be sent out quickly.

For the above reasons, the fast recovery algorithm is revised so that it is only applied to the path which has detected packet loss via the fast retransmit algorithm. Two data burst control algorithms have been evaluated. It is demonstrated via simulations that limiting burst by adjusting cwnd is a safer scheme than another

scheme. This modification can also improve performance when a path handover occurs during a fast recovery phase.

This paper also indicates a problem in the current SCTP design, where it applies the fast retransmit algorithm designed for use in single path configurations to multi-homed environments. Therefore fast retransmission on the primary path is encouraged when path bandwidths are high. However, more study is needed because when the path bandwidth is low the retransmission on the secondary path can increase performance.

We plan to study the effects of path delay and bandwidth on SCTP performance with various traffic loads to find the relationship between the path bandwidth, delay and SCTP performance.

## References

- [1] R. Stewart et al: Stream Control Transmission Protocol, IETF RFC 2960, October 2000.
- [2] R. Stewart et al: Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration, IETF draft, May 2006, <http://www.ietf.org/internet-drafts/draft-ietf-tsvwg-addip-sctp-15.txt>.
- [3] M. Riegel, et al: Mobile SCTP, IETF Draft, draft-riegel-tuexen-mobile-sctp-05.txt, July 2005.
- [4] R. Stewart et al: Stream Control Transmission Protocol (SCTP), A Reference Guide, Addison-Wesley, ISBN 0-201-72186-4, January 2006.
- [5] J. Noonan et al: Stall and Path Monitoring Issues in SCTP, Proc. Of IEEE Infocom, Conference on Computer Communications, Barcelona, April 2006.
- [6] A. L. Caro Jr. et al: Retransmission Schemes for End-to-end Failover with Transport Layer Multihoming, IEEE Globecom 2004, November 2004.
- [7] J. Iyengar et al: Concurrent Multipath Transfer using SCTP Multihoming, SPECTS'04, San Jose, USA, July 2004.
- [8] J. Iyengar et al: Receive Buffer Blocking in Concurrent Multipath Transfer, IEEE Globecom 2005, St. Louis, November 2005.
- [9] S. Ladha et al: On Making SCTP Robust to Spurious Retransmissions, ACM Computer Communication Review, 34(2), April 2004.
- [10] M. Mathis et al: TCP Selective Acknowledgement Options, IETF RFC2018, October 1996.
- [11] Shaojian Fu et al: SCTP: State of the art in Research, Products, and Technical Challenges, IEEE Communications Magazine, vol. 42, no. 4, April 2004, pp. 64-76.
- [12] R. Stewart: Stream Control Transmission Protocol, IETF draft, June 2006, <http://www.ietf.org/internet-drafts/draft-ietf-tsvwg-2960bis-02.txt>.
- [13] A. L. Caro Jr. et al: SCTP and TCP Variants: Congestion Control Under Multiple Losses, Tech Report TR2003-04, CIS Dept, U of Delaware, February 2003.
- [14] R. Stewart et al: Stream Control Transmission Protocol (SCTP) Specification Errata and Issues, IETF RFC 4460, April 2006.
- [15] S. Floyd et al: The NewReno Modification to TCP's Fast Recovery Algorithm, IETF RFC2582, April 1999.
- [16] A. Caro et al: ns-2 SCTP module, Version 3.5, <http://www.armandocaro.net/software/ns2sctp/>.
- [17] UC Berkeley, LBL, USC/ISI, and Xerox Parc: ns-2 documentation and software, Version 2.29, October 2005, <http://www.isi.edu/nsnam/ns>.

# Handover for Seamless Stream Media in Mobile IPv6 Network\*

Yi Liu<sup>1,2</sup>, Mingxiu Li<sup>2</sup>, Bo Yang<sup>2</sup>, Depei Qian<sup>1,2</sup>, and Weiguo Wu<sup>2</sup>

<sup>1</sup> School of Computer, Beihang University, Beijing 100083, China

<sup>2</sup> Department of Computer, Xi'an Jiaotong University, Xi'an 710049, China  
liuyi97@263.net

**Abstract.** In mobile IPv6 network, the handoff latency for mobile node challenges service quality of real-time stream media applications. This paper proposes an improved handover scheme for stream media applications, which switch data-stream to foreign network link predictively by session-level handover between mobile and correspondent node. And to reduce the duration of dual-stream transmission in network, a time-compensation mechanism is used. The scheme works at session-level independently, and requires no modification to current standards and protocols. Simulation results show that the scheme effectively improves the handoff latency and packet-losses for stream media applications.

## 1 Introduction

Stream media applications are becoming popular in the Internet, and with the development of mobile network technologies and handheld/mobile equipments, it will be one kind of “killer” applications for future commercial mobile IPv6 networks.

The requirement of stream media services to real-time is relatively strict. In mobile IPv6 networks, transmission delay and packet losses, which are caused by roaming and handover of mobile nodes, challenge the service quality of stream media applications. Despite the handoff latency is reduced greatly by the FMIPv6 (Fast Handovers for MIPv6) [2], the results still can not satisfy the requirements of real-time stream media applications, especially interactive stream media applications such as VoIP and video conference [3,4,8].

This paper proposes an improved handover scheme for stream media applications, which switch data-stream to foreign network link predictively by session-level handover between mobile and correspondent node. And to reduce the duration of dual-stream transmission in network, a time-compensation mechanism is used. The scheme works at session-level independently, and requires no modification to current standards and protocols. Simulation results show that the scheme effectively improves the handoff latency and packet-losses for stream media applications.

---

\* This work is supported by National Science Foundation of China under grant No. 60673180, and Science Foundation of Huawei company under grant No. YJCB2006032WL.

The rest of this paper is organized as follows. Section 2 gives an overview of the FMIPv6 standard. Section 3 introduces the improved scheme, the delay-compensation mechanism and the implementation considerations of the scheme. Section 4 evaluates the scheme in NS-2, and discusses the results of experiments. Section 5 summarizes the related research works, and section 6 concludes the paper.

## 2 An Overview of FMIPv6

Fast handovers for Mobile IPv6 (FMIPv6) is an improved handover scheme based on MIPv6. According to the OSI model, the handover is composed of Layer 2(L2) and Layer 3(L3) handover. The L2 handover corresponds to a change of wireless access point (WAP) for mobile node (MN). And the L3 handover is triggered if the two WAPs in L2 handover belong to different IP subnets. When L3 handover occurs, MN needs to obtain a new address and some registration information, which commonly causes a period of communication interruption. In FMIPv6, occurrence of a handover is predicted by detecting the movement of MN, and by bringing part of network layer handover operations ahead of link layer handover, data transportation is recovered shortly after link layer handover. As the result, the handoff latency of the MN is reduced effectively.

In FMIPv6, when roaming to a new network is predicted by MN, a Router Solicitation for Proxy (RtSolPr) message is sent to Previous Access Router (PAR) to obtain a new care-of address (NCoA). Then a Fast Binding Update (FBU) message is sent to PAR by the MN, and a tunnel is established between PAR and NAR, which is used for forwarding data packets between them. As soon as the MN arrives to the new network and finishes L2 connection, it sends a Fast Neighbor Advertisement (FNA) message to NAR. On receipt of this message or L2 connection gets ready, NAR starts to deliver packets to the MN.

## 3 Improved Handover Scheme for Stream Media Applications

### 3.1 Basic Improved Scheme

For real-time stream media applications, its data transportation experiences several phases during the handover of mobile node (MN), as shown in Table 1. Among these phases, data transportation is only interrupted in Phase-II. For common applications, the transportation is recovered since the beginning of Phase-III, however, since the packets are forwarded by PAR to NAR via tunnel in this period, the resulted long latency for packets influences service quality of real-time stream media applications, such as VoIP and video conference.

To reduce the handoff latency and packet delay for stream media applications, a session-level handover between MN and CN may be done predictively, which makes CN send data packets to NAR in advance. As the result, MN can receive data packets directly from CN as soon as it connects to new foreign network.

**Table 1.** Stream Media Data Transportation Based on FMIPv6

|           | Time of Begin/end                                                   | Data Transmission                                  | Packet loss & latency |
|-----------|---------------------------------------------------------------------|----------------------------------------------------|-----------------------|
| Phase I   | Begin: MN send out RtSolPr<br>End: sometime after MN receives FBack | Transport directly between MN and CN (through PAR) | Normal                |
| Phase II  | Begin: end of Phase I<br>End: NAR receives FNA message              | Interrupted                                        | Packets lost          |
| Phase III | Begin: end of Phase II<br>End: CN receives BU message               | Forward via tunnel between PAR and NAR             | Long latency          |
| Phase IV  | Begin: end of Phase III<br>End: ---                                 | Transport directly between MN and CN (through NAR) | Normal                |

The improved scheme is: When PrRtAdv message from PAR is received by MN, the new address can be obtained from it, and then a “pre-handover” message is sent to CN by MN, to inform the new address to CN. When the “pre-handover” message is received by CN, it begins to send data packets to both old and new address simultaneously. And then, when MN connected to new network, data packets directly from CN can be received immediately. The CN stops sending data packets to old address as soon as the BU message is received from MN.

Fig.1(a) gives the handover procedure based on improved scheme. Characteristics of the improved scheme are listed below:

- By informing the new address to CN beforehand, the audio/video streams can be switched to the new network in advance, and the long latency of packets due to tunnel forwarding is avoided. As the result, the handoff latency and packet delay for stream media applications are improved.
- The new “pre-handover” message is only transported between MN and CN, that is, at end-to-end session-level, and there is no modification to FMIPv6 standard. The scheme may be implemented by software or hardware vendors in their products as an optional enhanced feature.

### 3.2 Delay-Compensation Mechanism

In the basic improved scheme mentioned above, when the CN receives the “pre-handover” message, it starts to send data packets to both new and old addresses of MN simultaneously. As the result, there will be two data streams in the network, and the situation exists until the Binding Update (BU) message is received by CN. This “dual-stream” transmission will increase the network traffic, and have no benefit to the system scalability.

In order to reduce the duration of “dual-stream” transmission, a delay-compensation mechanism is used in CN, that is, when the message “pre-handover” is received, CN delays a period of time before starting dual-stream transmission. The

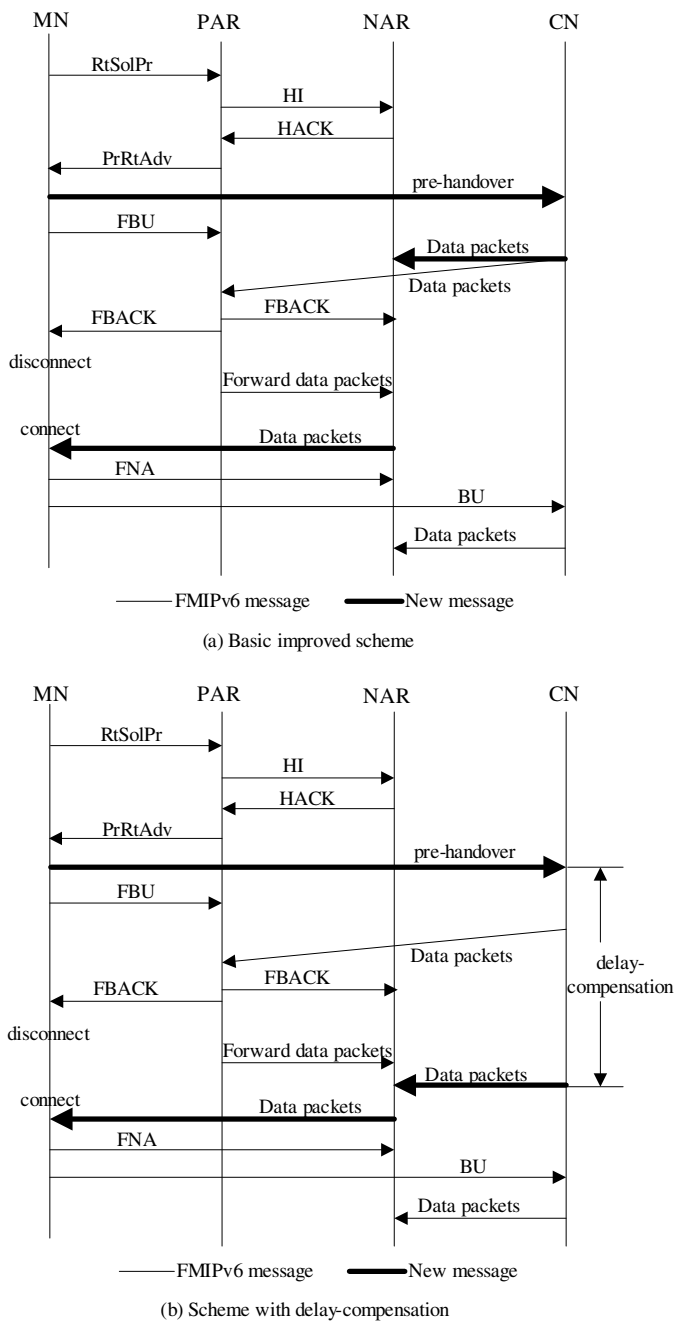


Fig. 1. Handover Procedure

ideal time to start dual-stream transmission is the time just before the MN begins to use the new address, which reduces the duration of dual-stream as more as possible, while guarantees that the MN can receive data packets as soon as it connects to the new network.

The delay-compensation value can be calculated by the MN according to the statistics of previous handovers, and is contained in the “pre-handover” message which is sent to the CN. More exactly, the delay-compensation value can be calculated according to:

$$t_{compensate} = \frac{\sum_{i=1}^n (T_{FNA}^i - T_{pre-handover}^i)}{n} \times (1 - ratio) \quad (3-1)$$

where  $t_{compensate}$  is the delay-compensation value;  $T_{FNA}$  and  $T_{pre-handover}$  are the time of sending FNA and “pre-handover” message from MN respectively;  $n$  represents the number of previous handovers used to calculate the delay-compensation; and  $ratio(0 < ratio \leq 1)$  is a factor used to adjust the scaling value of delay-compensation between 0 and maximum permitted value. The smaller the  $ratio$  is, the bigger the scale is, and the closer to the maximum value the delay-compensation is. The result is that the duration of dual-stream transmission goes shorter, while risks goes higher, that is, due to the network delay, unstable handoff latency and some other reasons, the time of starting dual-stream transmission in CN may be too late, leading to the result that on switching to the new network link, the MN cannot receive real-time data packets directly from CN. On the other side, the bigger the  $ratio$  is, the smaller the delay-compensation is. If  $ratio = 1$ , the scheme degrades to the basic scheme without delay-compensation.

To reduce the amount of calculations for implementation, the expression (3-1) may be transformed to iterative form as follows:

$$t_{compensate} = t_{previous-compensate} \times \frac{n-1}{n} + \frac{T_{FNA}^n - T_{pre-handover}^n}{n} \times (1 - ratio) \quad (3-2)$$

where  $t_{previous-compensate}$  is the delay-compensation value used in the previous handover.

### 3.3 Implementation Considerations of the Improved Scheme

In the implementation of the improved scheme, stream media application running in CN maintains a CoA list as shown in Fig.2. Each entry in the list corresponds to a session between the CN and the MN connected with it. The information contained in the entry includes home address, New CoA, Previous CoA and corresponding valid flags, etc.



When a connection request from MN is received, CN creates a new entry in the CoA list and initial the relevant fields. The delay-compensation value is initialized to 0 at this time.

According to the scheme, after the PrRtAdv message from PAR is received and new CoA is obtained, MN will send a “pre-handover” message to CN. The contents of the message include home address, new CoA and delay-compensation value.

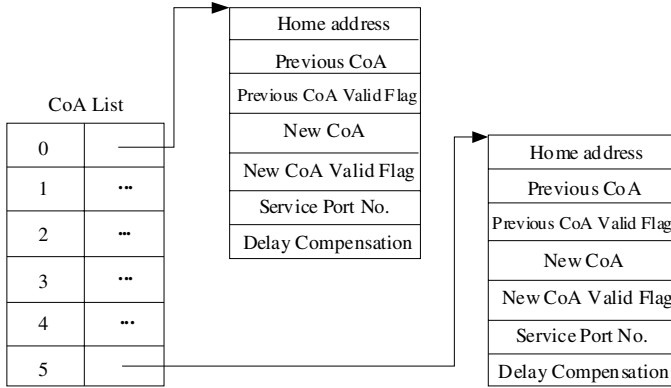


Fig. 2. Structure of CoA List

On receiving the message, CN looks up the entry in CoA list by home address of the MN, and updates the contents of the entry as: Previous CoA  $\leftarrow$  New CoA; Previous CoA Valid Flag  $\leftarrow$  New CoA Valid Flag; New CoA  $\leftarrow$  the value just received; New CoA Valid Flag  $\leftarrow$  TRUE; Delay Compensation  $\leftarrow$  value just received. Then CN send data packets to previous CoA, while counting down the “Delay Compensation”. When the timer goes to 0, data packets are started to sent to new CoA of the MN. At this time, the dual-stream transmission is started, and it will last until the BU message is received. At that time, CN stops sending data packets to previous CoA, and set its valid flag to FALSE. If the BU message is received before “pre-handover” message, or before timeout of the delay-compensation, CN will not send data packets to the two addresses, i.e. the dual-stream will not be transmitted.

On the MN side, it records the time of sending “pre-handover” and FNA message, calculates and updates delay-compensation value according to expression (3-2) by the end of each handover. The updated value will be used in next handover.

### 4 Experiments and Evaluation

Simulation is done using NS-2 with the improved scheme implemented in it. Fig.3 shows the network topology used in the experiments. The bandwidth and transmission delay of each network link are shown in the figure. In the network, the distance

between PAR and NAR is 75m, and the wireless coverage area of them is set to 40 meters. In each experiment, the MN moves from PAR to NAR in a fixed speed, while sending / receiving packets to / from CN continuously. The movement will cause a handover eventually. Handoff latency and packet losses are measured for each experiment. For comparison, each experiment is repeated for both FMIPv6 and our improved scheme.

Traffics between CN and MN are generated as CBR (Constant Bit Rate). Considering the data rate and packet size of audio and video applications, experiments are done under data rate=30kb/s with different packet size, and packet size=100bytes with different packet rate.

On calculating delay-compensation value according to expression (3-1),  $n$  is set to 5. Moreover, it is found that for the above scenarios, CN can always start dual-stream transmission in time when  $ratio \geq 0.6$ , and the following experiment results are obtained on  $ratio \geq 0.6$ .

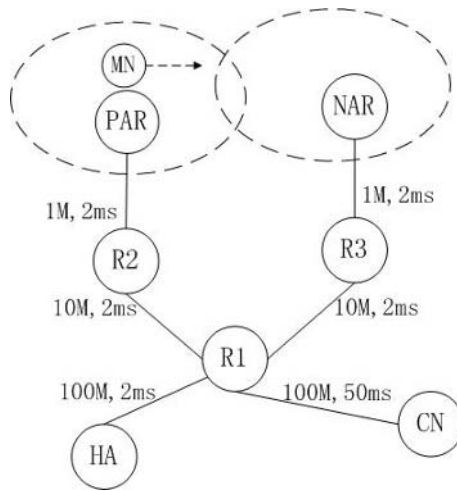
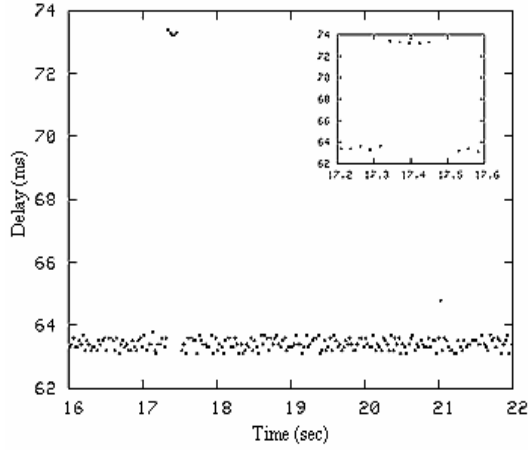


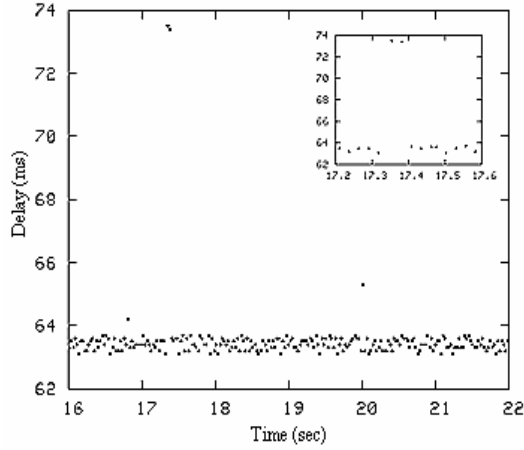
**Fig. 3. Network Topology for Simulation**

Fig.4 gives the transmission delay of data packets for FMIPv6 and the improved scheme with packet size=100bytes and data rate=30kb/s. The picture at upper-right corner is the local zoom-in for the period of handover. As Fig.4 shows, the interruption time of data transportation for the improved scheme is shorter than FMIPv6.

The handoff latency and packet losses are also measured for each experiment, and results are shown in Fig.5 and Fig.6. As these figures show, the handoff latency and number of packet losses for the improved scheme are always lower than FMIPv6.



(a) FMIPv6



(b) Improved Scheme

**Fig. 4.** Transmission Delay of Data Packets

As Fig.6 shows, the difference of packet losses between FMIPv6 and the improved scheme is not very clear, due to that the packet losses for FMIPv6 are already quite small. In fact, under FMIPv6, most of the packets transmitted during the handover period are forwarded via tunnel between PAR and NAR, that is, the transmission delay of these packets are fairly long, as shown in Fig.4. For real-time stream media applications, generally the packets arrived too late are almost useless.

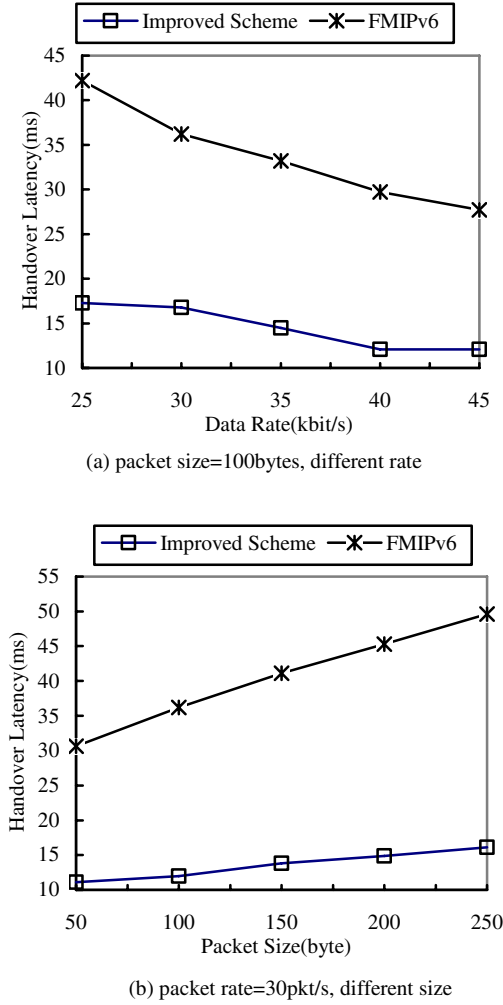


**Fig. 5.** Handoff Latency under Various Packet Size and Data Rate

To evaluate the impact of dual-stream transmission on network traffics, experiments are done under different values for *ratio* in expression (3-1), and the average network traffics in different period are listed in Table.2. As it shows, the dual-stream transmission increases network traffics momentarily. However, since the duration time is very short, with the time goes on, the average network traffic goes down very quickly. For relatively long periods, the difference between FMIPv6 and the improved scheme in network traffics is not distinct.

From the experiments, we found that, for the scenarios used in this paper, CN can always start dual-stream transmission in time on  $ratio \geq 0.6$ ; and on  $ratio < 0.6$ , the time to start dual-stream transmission will be late more or less, which influences performance of the improved scheme.

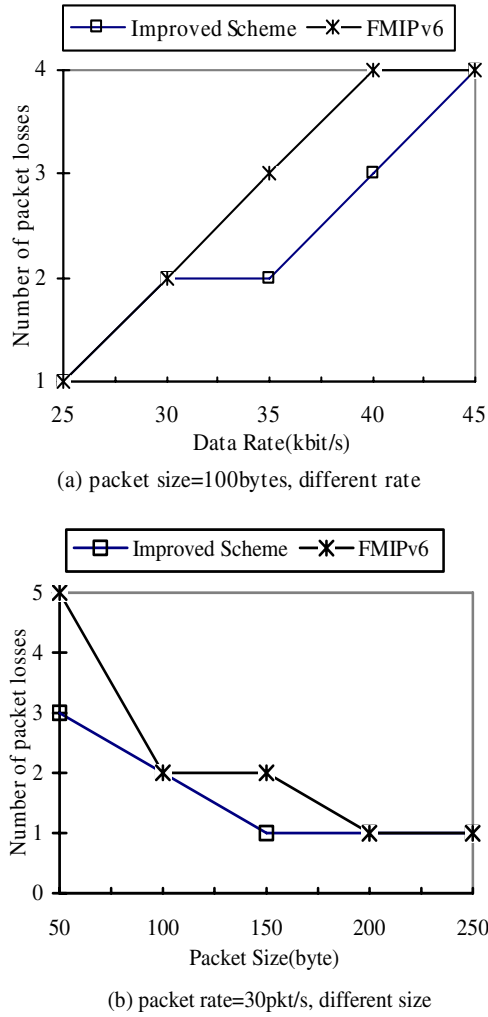


Fig. 6. Packet Losses under Different Packet Size and Data Rate

Table 2. Average Traffics in Network

| Scheme          | Time since the start of handover | Network traffics (kbit/s) |           |           |           |           |         |
|-----------------|----------------------------------|---------------------------|-----------|-----------|-----------|-----------|---------|
|                 |                                  | ratio=0                   | ratio=0.2 | ratio=0.4 | ratio=0.6 | ratio=0.8 | ratio=1 |
| FMIPv6          | All time                         | 30.00                     | 30.00     | 30.00     | 30.00     | 30.00     | 30.00   |
|                 | 50ms                             | 30.00                     | 30.00     | 30.00     | 30.00     | 30.00     | 30.00   |
|                 | 100ms                            | 30.00                     | 30.00     | 30.00     | 30.00     | 30.00     | 35.13   |
| Improved scheme | 500ms                            | 30.05                     | 31.72     | 33.40     | 35.04     | 36.75     | 38.42   |
|                 | 1s                               | 30.02                     | 30.86     | 31.70     | 32.54     | 33.37     | 34.21   |
|                 | 2s                               | 30.01                     | 30.43     | 30.85     | 31.27     | 31.69     | 32.11   |
|                 | 5s                               | 30.00                     | 30.17     | 30.34     | 30.51     | 30.67     | 30.84   |

## 5 Related Works

The main related researches about service quality of stream media applications include: [8] introduces a multicast handover scheme within HMIPv6 to improve handover performance for real-time applications such VoIP or video conferencing. [9] presents a seamless streaming framework by estimating the accurate pre-buffering level to compensate handoff latency and by adapting rate-shaping scheme to overcome the shortage of bandwidth around a handoff. The scheme can effectively improve the service quality of mobile node for uni-directional stream media applications such as VoD. However, for interactive real-time stream media applications, such as VoIP and video conference, the method of pre-buffering is unsuitable. [10] proposes an end-to-end multipath handoff scheme which reduces the rerouting and congestion packet loss during handoff 1) by establishing multiple paths to the mobile node and transmitting duplicate packets over the multiple paths to reduce the negative impact of packet loss and 2) by probing the available bandwidth in the new cell and allowing stream media applications to gradually adapt their transmission rates to the available bandwidth in a new cell. The scheme requires that the stream media sources are equipped with an adaptive multilayer encoder.

## 6 Conclusion

In mobile IPv6 network, the handoff latency and packet losses, which are caused by roaming and handover of mobile nodes, challenge service quality of real-time stream media applications. This paper proposes an improved handover scheme for stream media applications, which switch data-stream to foreign network link predictively by session-level handover between mobile and correspondent node. And to reduce the duration of dual-stream transmission in network, a time-compensation mechanism is used. The scheme works at session-level independently, and requires no modification to current standards and protocols. Simulation results show that the scheme effectively improves the handoff latency and packet-losses for stream media applications.

## References

1. D. Johnson, C. Perkins, J. Arkko. Mobility Support in IPv6. RFC3775, IETF, June 2004.
2. R. Koodli, Fast Handovers for Mobile IPv6. RFC4068, IETF, July 2005.
3. Kim YS, et al. Performance Comparison of Mobile IPv6 and Fast Handovers for Mobile IPv6 Over Wireless LANs, In the Proceedings of IEEE 62<sup>th</sup> Vehicular Technology Conference, Sept 2005.
4. Albert Cabellos-Aparicio, et al. Evaluation of the Fast Handover Implementation for Mobile IPv6 in a Real Testbed, In the Proceedings of 5th IEEE International Workshop on IP Operations and Management, IPOM 2005.
5. Douglas Howie, MIPv6 Issues in Delivering Mobile Multimedia, In Proceedings of Advanced Simulation Technologies Conference (ASTC'03), Mar 2003.
6. Steffen Sroka and Holger Karl, Performance Evaluation of QoS-Aware Handover Mechanism, In the Proceedings of the Eighth IEEE International Symposium on Computers and Communications (ISCC 2003), June 2003.

7. Chandrasekar, C. and Krishnan, A. Improving the Performance and Reliability of Wireless in Mobile Streaming Media, In Proceedings of 2005 IEEE International Conference on Personal Wireless Communications, Jan 2005.
8. Thomas C. Schmidt and Matthias Wählisch, Roaming Real-time Applications - Mobility Services in IPv6 Networks, TERENA Networking Conference, May 2003.
9. Dongwook Lee, Chul-Ho Lee, Jong Won Kim, Seamless Media Streaming over Mobile IP-enabled Wireless LAN, Second IEEE Consumer Communications and Networking Conference, Jan 2005.
10. Yi Pan, Meejeong Lee, Jaime Bae Kim, et al. An End-to-end Multipath Smooth Handoff Scheme for Stream Media, IEEE Journal on Selected Areas in Communications, May 2004.
11. W. Kellerer, E. Steinbach, P. Eisert, and B. Girod, "A Real-time Internet Streaming Media Testbed", In Proceedings of International Conference on Multimedia and Expo (ICME 2002), Aug 2002.

# A Secure Handover Protocol Design in Wireless Networks with Formal Verification

Sun-Hee Lim<sup>1</sup>, Ki-Seok Bang<sup>2</sup>, Okyeon Yi<sup>3</sup>, and Jongin Lim<sup>1</sup>

<sup>1</sup> Center for Information Security Technologies(CIST), Korea University, Seoul, Korea  
{capsunny, jilim}@korea.ac.kr

<sup>2</sup> College of Information and Electronic, Hallym University, ChunCheon, Korea  
mysaver@hallym.ac.kr

<sup>3</sup> Department of Mathematics, Kookmin University, Seoul, Korea  
oyyi@kookmin.ac.kr

**Abstract.** This paper analyses security mechanisms in wireless networks, and designs for secure and efficient protocol for a handover reusing the key materials generated before the handover phase. In order to be sure that our handover protocol does not have any security flaws, we formally validated it using automatic protocol analyzer AVIPSA.

**Keywords:** wireless, handover, security, authentication, AVISPA.

## 1 Introduction

Future mobile communication systems will encompass heterogeneous and homogeneous access such as WLAN, WiBro and UMTS. Many users will appreciate the availability of desired service anywhere, anytime, and with reasonable cost.

Security is one of the major technical challenges of handover in the wireless network. To enable the convergence of different wireless technologies, link layer security for wireless network access is also dispensable. However, each wireless network may deploy its own security mechanisms that are incompatible with others. Apart from that, seamless mobility sets time constraint on handovers.

We study the link layer security mechanisms in wireless networks and derive the similar security mechanisms such as the authentication and a common key derivation. And we propose a secure handover protocol for seamless reusing the common security contexts. Finally the proposed handover protocol is verified with an automated security analyzer AVISPA.

This paper is composed as follows: In Section 2, an analysis on security mechanisms in the wireless networks such as WLAN, WiBro and UMTS is given. Section 3 defines the architecture and security establishments for secure handover and designs the secure handover protocol and verifies the protocol using formal validation. The paper is concluded in Section 4.

## 2 Security Mechanisms in Wireless Networks

This section describes key features and authentication procedures used in the WLAN (IEEE 802.11i), the WiBro (IEEE 802.16e) and the UMTS (based on 3GPP).



2.1 WLAN

The initial WLAN security mechanism in the IEEE 802.11 standard has design flaws. Newly, IEEE 802.11i[1] is standardized to improve enhanced security of the WLAN.

The Robust Security Network Association (RSNA) defines a number of security features which are enhanced authentication mechanisms including use of IEEE 802.1x [3], data encapsulation mechanisms like CCMP and TKIP, and Cryptographic key establishment, in addition to wired equivalent privacy (WEP) and IEEE 802.11 authentication.

**1)Authentication.** The authentication procedure comprises two phases, establishing the IEEE 802.11 association and IEEE 802.1x EAP authentication as shown in the left side of Fig 1. During the first IEEE 802.11 Association phase, a STA is associated with an AP and discovers the AP's security policy. The Authentication phase uses an EAP method that supports mutual authentication of the AS and the STA. The EAP[4] is an authentication framework which supports multiple authentication methods such as AKA, TLS, TTLS, etc. EAP authentication frames pass between the STA (Supplicant) and the AS via the AP (Authenticator) and STA's Uncontrolled Ports. The IEEE 802.1x Controlled Port is blocked from passing general data traffic until an authentication procedure completes successfully over the IEEE 802.1x Uncontrolled Port. Upon successful completion of the 4-Way Handshake, the STA and AP authenticate each other and the STA and AP's IEEE 802.1x Controlled Ports are unblocked to permit general data traffic.

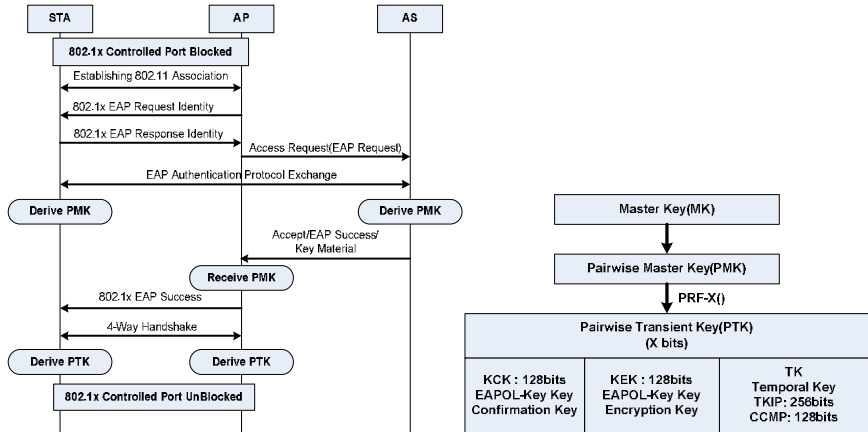


Fig. 1. IEEE 802.11i Authentication and Key Derivation

**2)Key Derivation.** IEEE 802.11i supports the hierarchical key architecture as the right side of Fig 1. The STA and AS generate a Master Key (MK) and derive a Pairwise Master Key (PMK) from the MK. The AS sends the PMK to the STA's AP over the trust channel so that the STA and AP share PMK after they complete authentication procedures. Upon successful completion of the 4-way handshake between the STA and the AP, a Pairwise Transient Key (PTK) is derived from the

PMK. The PTK is partitioned into a KCK, KEK, and TK. The Temporal Key (TK) is used to protect unicast communication between the STA and AP. The KEK and KCK are used by the EAPoL-Key frames to provide confidentiality and authenticity.

## 2.2 WiBro

The Wireless Broadband (WiBro) security architecture is based on IEEE 802.16 Privacy layer[2]. The WiBro security has two basic protocols such as an encryption protocol and a privacy key management protocol (PKM). The PKM authentication protocol comprises the following two phases.

- MS Authorization and Authorization Key (AK) exchange
- Traffic Encryption Key (TEK) exchange

### 1) Authentication

a) *PKMv1*: RSA protocol is mandatory in PKMv1. However, PKMv1 has several security weak points. That need for mutual authentication protocol, key management, explicit definition and strong data confidentiality shall be considered.

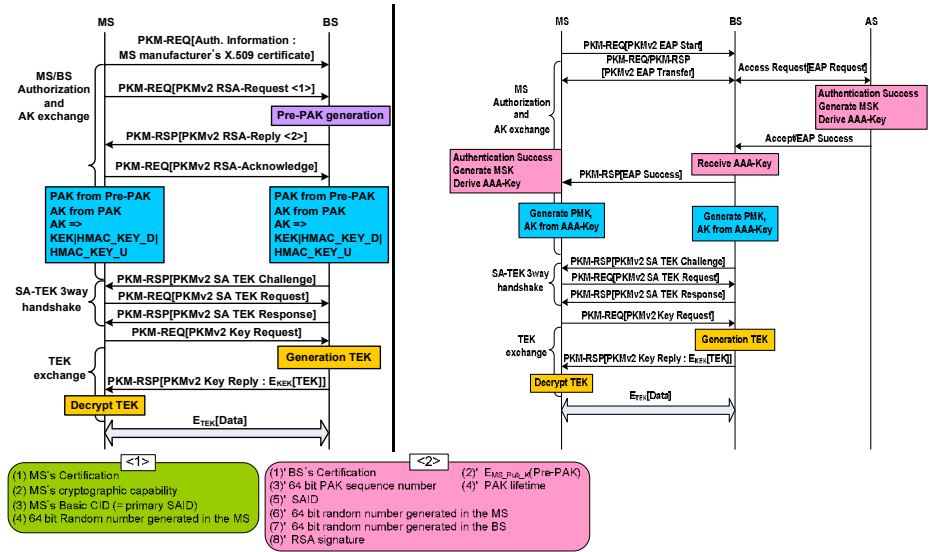


Fig. 2. RSA-based and EAP-based Authorization in PKMv2

b) *PKMv2*: It supports authentication protocol mechanisms based on RSA protocol and EAP protocol, optionally. It provides mutual authentication between the user and the network. The RSA based authorization in PKMv2 is similar to the authorization in PKMv1. However, PKMv2 RSA based authorization supports mutual authentication by verifying the MS and BS's certification and shares Pre-PAK to derive the PAK and AK between the BS and MS. An EAP based authorization in PKMv2 uses EAP protocol in conjunction with an operator-selected EAP method. The RSA based (the

left side) and EAP based (the right side) authorization detail procedures in PKMv2 are shown in Fig. 2.

**2)Key Derivation.** The IEEE 802.16e defines the PKMv2 key hierarchy. Since PKMv2 defines RSA-based and EAP-based authentication schemes, there are two primary sources of keying material. The RSA-based authorization yields the pre-Primary AK (pre-PAK) and the EAP-based authorization yields the MK. The AK will be derived by the BS and the MS from the PMK (from EAP-based) and/or the PAK (from RSA-based). The AK can be derived in one of three ways depending on the authentication scheme used RSA-based or EAP-based or both. The BS and the MS shall derive both a shared KEK to encrypt transport keys and HMAC/CMAC keys to validate the authenticity management messages before the TEK 3-Way handshake. Fig. 3 depicts the key derivation in PKMv2.

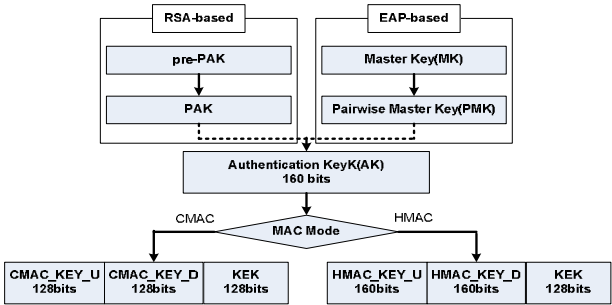


Fig. 3. Key derivation in PKMv2

### 2.3 UMTS

The Universal Mobile Telecommunication System (UMTS), which is known as the third generation (3G) cellular mobile communication system, adopts the security feature of GSM in order to interwork with GSM smoothly and adds new security features to design an authentication and key agreement protocol (AKA).

UMTS provides security features, such as mutual authentication, agreement on an integrity key between MS and SN, and freshness assurance of the agreed cipher key and integrity key.

Furthermore, standardization activities emphasize the important role of the 3G-WLAN handover[5][11]. Various interconnection mechanisms of 3G and WLAN are discussed in the literature. Extensible Authentication Protocol-Authentication and Key Agreement (EAP-AKA) is foreseen by 3GPP to be used in context of WLAN and UMTS interworking scenarios. EAP-AKA uses two roundtrips to authenticate and authorize the peer and EAP server and generate session keys using authentication vectors. The EAP server and the peer use CK and IK in key derivation. On EAP-AKA authentication, a Master Key (MK) is derived from the AKA values (CK and IK keys), and the identity. A Master Session Key (MSK) for link layer security is derived from the MK. The MSK can be used as the Pairwise Master Key (PMK) for depending on the security mechanism of the wireless networks.

### 3 Secure Handover Protocol Design with Formal Verification

The WLAN, WiBro and UMTS have their own mechanisms for authentication and link layer security. However, when the handover happens, a full mutual authentication involving MN's HN can hardly fulfill the requirements of a short delay.

In this section, we propose a secure handover protocol for a seamless handover. The security mechanisms in wireless networks have many similarities. The secure handover protocol is dependent on link layer security mechanism such as authentication and key agreement mechanisms of each wireless network. We design a secure and efficient authentication method for a seamless handover in wireless networks. In addition to, we specify and verify a secure handover protocol using Automated Validation of Internet Security Protocols and Applications (AVISPA).

#### 3.1 Handover in the Wireless Network

We briefly describe the domain model shown as the Fig. 4[9].

- Mobile Node (MN) is a user domain including the mobile terminal.
- Serving Network (SN) means the network domain that serves the MN before handover.
- Target Network (TN) means the network domain that serves the MN after handover.
- Home Network (HN) is the network domain in charge of user subscriptions and other supporting services, like billing, authorization, and authentication.

#### 3.2 Handover Security Trust Relations

The trust relations should be established beforehand for fast and secure authentication handover in wireless networks as listed in Fig.4 and Table 3[8].

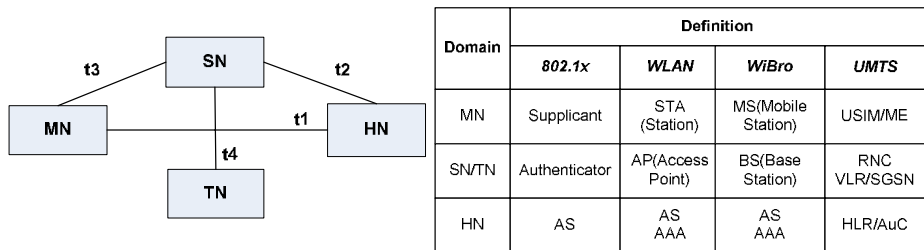


**Fig. 4.** The Trust Relation beforehand and Domains Definition involved in a Handover

#### 3.3 Secure Handover Protocol Design

We design a secure and efficient authentication method for a seamless handover without additional security materials from AS using similar security mechanisms in

**Table 1.** The Trust Relation for HO

| Trust Relation | Establishments                                                                                                                                                                                                                    |                       |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
|                | Property                                                                                                                                                                                                                          | Term                  |
| t1             | Trust between MN and HN must confirm a security key by mutual authentication via SN assuming that t2 is a trust relation.<br>Mutual authentication is established by authentication mechanisms as EAP-AKA, EAP-TLS, EAP-SIM, etc. | A long-term relation  |
| t2             | Trust between SN and HN must be assumed.                                                                                                                                                                                          | A long-term relation  |
| t3             | It is established by mutual authentication in each wireless network.<br>t3 relation derives PTK from PMK generated in t1 relation establishment or shares the protected TEK.<br>t3 generates KEK, KCK and TK.                     | A short-term relation |
| t4             | The trust is built beforehand in order to support handover.                                                                                                                                                                       | Depending on policies |

wireless networks. During the HO process, SN plays a trusted party to MN and TN, so that HN/AuC is not involved during the HO phase. That provides a faster HO for the network entities such as MN and TN. Each wireless network entity can use its own security policy and parameters, so no additional security changes for this proposed protocol are needed. We classify the handover procedures into 3 phases, before the HO phase, during the HO phase, and after the HO phase.

Fig. 5 shows the detail procedures of secure handover.

**1)Before the HO phase.** An initial authentication procedure must be performed completely. This recommends mutual authentication between a MN and a HN via a SN. After the MN and HN via a SN are authenticated successfully, the MN and the HN generate PMK from MK. The HN sends a PMK to the MN's SN. In the sequel, the t1 and t3 relations derive the KEK and the KCK from PMK.

**2)During the HO phase.** For a seamless handover, the SN should play an important role as a Trusted Third Party (TTP) as it distributes key materials to the MN and the TN. The security contexts built between the MN and SN are used to generate cryptographic keying material for the handover. A trust between the SN and the TN must be established beforehand.

#### *Secure Handover Protocol*

##### 0. Prerequisite

- t3 trust relation has a key TK for protecting the data traffic in the wireless network.
- t4 trust relation has a key ST for protecting the data between the SN and the TN.

##### 1. HO decision.

2. The SN as a TTP role distributes security information for handover to MN and TN. The SN generate  $hokek=hkey(TK,KEK)$  and  $hokck=hkey(TK,KCK)$ .

**Msg 1.**  $SN \rightarrow TN : \{TID_{MN}.ID_{MN}.hokek.hokck\}_{ST}$

**Msg 2.**  $SN \rightarrow MN : \{ID_{TN}.TID_{MN}\}_{TK}$

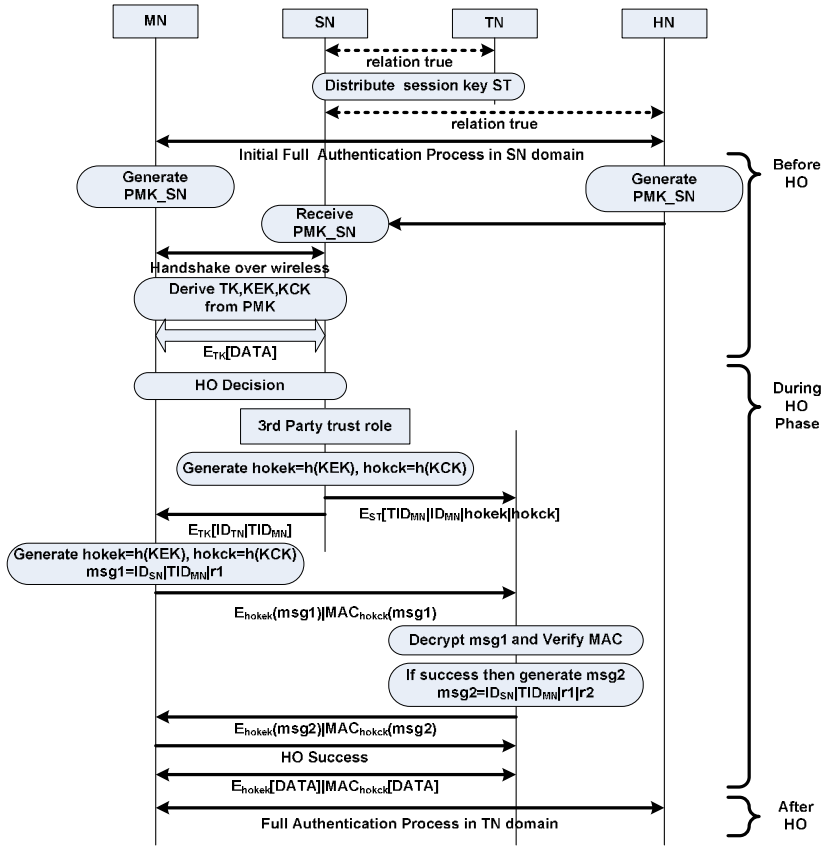


Fig. 5. Secure Handover Protocol in the wireless networks

3. The MN and the TN confirm the key agreement for a Hokek and a Hokck protecting and authenticating the data during the handover. The MN receiving Msg2 from the SN is ready for connection with the TN. The MN keeping the KEK and the KCK generate  $hokek=hkey(TK, KEK)$  and  $hokck=hkey(TK, KCK)$ .

**Msg 3.**  $MN \rightarrow TN$  :  $\{ \{ ID_{SN}.TID_{MN}.r1 \}_{hokek} \}. \{ MAC(hokck, ID_{SN}.TID_{MN}.r1) \}$

**Msg 4.**  $TN \rightarrow MN$  :  $\{ \{ ID_{SN}.TID_{MN}.r1.r2 \}_{hokek} \}. \{ MAC(hokck, ID_{SN}.TID_{MN}.r1.r2) \}$

We propose reusing KEK and KCK in the PTK derived from the PMK as the HO key material. The key materials hokek and hokck for secure handover are generated from the KEK and KCK by computing hash algorithm. The hokek and hokck have enough strong points.

- Sharing only between the MN and the SN
- Never used except in the authentication procedure
- Sufficient size for applying the HO phase (128~160 bits)
- Not compromised from other wireless networks

**Table 2.** Significant HLPSTL Syntax used in this paper

|                                                              |                                                           |
|--------------------------------------------------------------|-----------------------------------------------------------|
| MN, SN, TN:agent                                             | %principals                                               |
| TK:symmetric_key                                             | %protect data between MN and SN                           |
| ST:symmetric_key                                             | %protect data between SN and TN                           |
| KEK:symmetric_key                                            | %protect transport key between MN and SN                  |
| KCK:symmetric_key                                            | %authenticity key between MN and SN                       |
| HOKEK, HOKCK:symmetric_key                                   | %new key for handover                                     |
| MAC:hash_func                                                | %keyed Message Authentication Code                        |
|                                                              | %keyed hash function to derive HOKEK, HOKCK from KEK, KCK |
| HKEY:hash_func                                               |                                                           |
| ID <sub>MN</sub> , ID <sub>SN</sub> , ID <sub>TN</sub> :text | %Identity of each agent                                   |
| TID_MN:text                                                  | %Temporal MN's Identity                                   |
| r1, r2:protocol_id                                           | %Random Number                                            |
| SA, SB, SC, RA, RB, RC:channel(dy)                           | %Session                                                  |

- Ensure backward secrecy preventing a TN from decoding messages exchanged before the handover
- Hash computation for generating the hokek and the hokck has no effect on delay for seamless handover
- No more generate and distribute the security materials for handover with the help of the HN
- Keep the its own security mechanisms in each wireless network without additional procedure for handover

**3)After the HO phase.** After the handover is completed, if necessary, a full authentication with the help of the HN via TN could be performed.

### 3.4 Formal Specification and Validation of the Secure Handover Protocol

#### 1)AVISPA

Due to the nature and sensitive of security protocols, there has been a renewed emphasis on integrating formal validation in design and development phase. It is necessary to validate our proposed solution by automatic tools which use a formal specification language to input a protocol and backend mathematical tools to produce possible flaws in a protocol.

Automated Validation of Internet Security Protocols and Applications (AVISPA) [7] is a tool which provides a modular and expressive formal language called the High Level Protocol Specification Language (HLPSL) for specifying intended protocols and formally validating them.

We have used AVISPA in order to validate the designed secure handover protocol by the HLPSL specification.

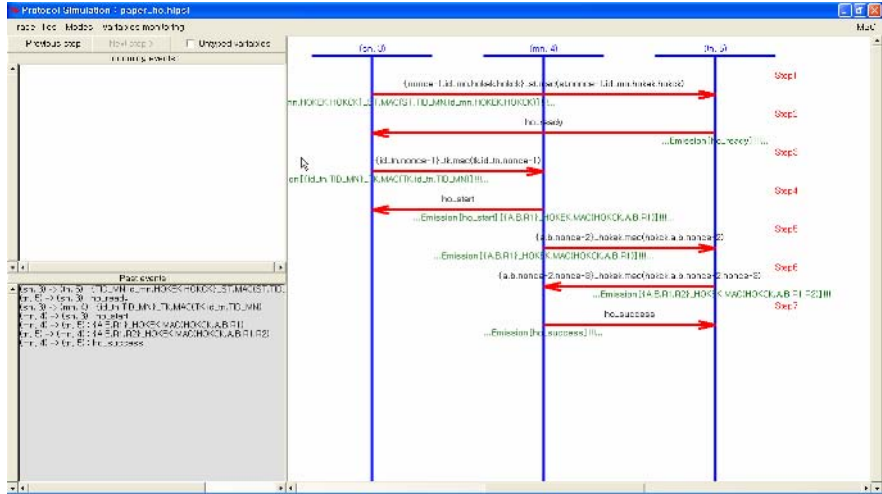


**Fig. 6.** The SPAN animator screenshot executing the proposed secure handover specification

**Table 3.** Goals of validation by AVISPA

```
goal

  % secrecy_of HOKEK, HOKCK

  secrecy_of
  sec_hokck0, sec_kck0, sec_hokek1, sec_hokck1, sec_hokek2, sec_hokck2

  secrecy_of kek0, kck0, kek1, kck1

  % MN authenticates TN on r1

  authentication_on r1

  % TN authenticates MN on r2

  authentication_on r2

end goal
```



**2)Specifying the secure handover protocol**

We specify the proposed secure handover protocol by the HLPSTL language and check the specification by the SPAN.

The role of a SPAN (Security Protocol Animator for AVISPA)[13] is to symbolically execute a HLPSTL protocol specification so as to have a better understanding of the specification, check that it is executable and that it corresponds to what is expected.

Fig.6 is the screenshot executing the proposed secure handover protocol specification by the HLPSTL.

**Table 4.** Result by AVISPA validation for secure handover protocol

|                                          |
|------------------------------------------|
| SUMMARY                                  |
| SAFE                                     |
| DETAILS                                  |
| BOUNDED_NUMBER_OF_SESSIONS               |
| TYPED_MODEL                              |
| PROTOCOL                                 |
| /home/avispa-1.1/testsuite/results/ho.if |
| GOAL                                     |
| AS Specified                             |
| BACKEND                                  |
| CL-AtSe                                  |
| STATISTICS                               |
| Analysed : 1079 states                   |
| Reachable : 215 states                   |
| Transition : 0.10 seconds                |
| Computation: 1.53 seconds                |

**3)Verifying the secure handover protocol**

We have modeled and validated the proposed secure handover protocol using Intrusion model. Validation of the secure handover protocol has goals which specify secrecy and authentication.

Suppose the intruder is playing the role MN or TN, then intruder's knowledge is defined the parameters of the corresponding instance of the role MN or TN. If the TK which is a key between the MN and the SN, and the ST which is a key between the SN and the TN, do not reveal to attacker, secure handover protocol results in secure validation by AVISPA. In the sequel, the proposed secure handover protocol has no security flaw.

## 4 Conclusion and Future Work

In this paper, we showed the security mechanisms in the wireless networks such as the WLAN, the WiBro, and the UMTS for interworking WLAN. Many discussions about a fast authentication for a seamless handover in the wireless are going actively. A fast authentication for the handover without the help of HN during the handover phase in the pervious studies[8][10] is proposed.

We analyzed the security mechanisms in the wireless network and derived the secure and efficient handover protocol by securely reusing the key generated before the handover phase. We specify and verify the proposed secure handover protocol using AVISPA. No new attack or vulnerability has been surfaced by automatic analysis. We have no significant consideration for Permanent Identity protection and use of the temporary identity.

Future works will be focused on the implementation of the protocol and performance tests. Moreover, we intend to design the protocol including the ID protection, timestamp and so on, in detail. And we will specify and verify the designed protocol approach to formal method.

**Acknowledgments.** This research was supported by the MIC(Ministry of Information and Communications), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement).

This work was supported by the 2006 Research Fund of Kookmin University and the Kookmin Research Center UICRC in Korea.

## References

1. IEEE, "Part11: Wireless LAN Medium Access Control(MAC) and Physical Layer(PHY) specifications", IEEE Std 802.11i, 2004.
2. IEEE, "Part16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems", IEEE Std.802.16e, 2006.
3. IEEE, "Port-Based Network Access Control", IEEE Std 802.1x, 2004.
4. RFC 3748, "Extensible authentication protocol(EAP)", June 2004.
5. 3GPP, "3<sup>rd</sup> Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security, Wireless Local Area Network(WLAN) interworking security", 3GPP TS 33.234, June 2005.
6. A. R. Prasad, H. Wang, "A protocol for secure seamless handover", in Proc. of International Conference on Telecommunications(ICT'04), Fortaleza, Brazil, August 1-7 2004.
7. Avispa - a tool for Automated Validation of Internet Security Protocols.  
<http://www.avispa-project.org>.

8. Hu Wang and Anand R. Prasad, "Fast authentication for inter-domain handover", in Proc. Of International Conference on Telecommunications(ICT'04), Fortaleza, Brazil, August 1-7, 2004.
9. H. Wang, A. R. Prasad, P. Schoo, "Research issues for fast authentication in inter-domain handover", in Proc. of Wireless World Research Forum(WWRF), Beijing, China, February 2004.
10. H. Wang, A. R. Prasad, "Security context transfer in vertical handover", in Proc. of PIMRC 2003, Beijing, China September , 7-10 2003.
11. K. M. Bayarou, C.Eckert, S. Rohr, A.R. Prasad,P. Schoo, H. Wang, "3G and WLAN interworking:Towards a secure solution for tight coupling", in Proc. of WPMC 2004, Italy, Padova, September 12-15, 2004.
12. M. Georgiades, H. Wang, R. Tafazolli, "Security of context transfer in future wireless communications", in Proc. of Wireless World Research Forum(WWRF), Toronto, Canada, November 4-5, 2004.
13. Span – a Security Protocol Animator for AVISPA. <http://www.irisa.fr/lande/genet/span>
14. Sun-Hee Lim, Okyeon Yi, "A study on EAP-AKA authentication architecture for WiBro wireless network", KICS2005-11-457.
15. Sun-Hee Lim, Okyeon Yi, Chang-Hoon Jung, Ki-Seok Bang, "A Fast and Efficient Authentication Protocol for a Seamless Handover between a WLAN and WiBro", Publication at IEEE COMmunication System softWAre and MiddlewaRE 2007(COMSWARE2007), Bangalore, India, Jan. 7-12, 2007.

# Seamless Handover for Multi-user Sessions with QoS and Connectivity Support

Eduardo Cerqueira<sup>1</sup>, Luis Veloso<sup>1</sup>, Paulo Mendes<sup>2</sup>, and Edmundo Monteiro<sup>1</sup>

<sup>1</sup> University of Coimbra, Pinhal de Marrocos, 3030-290 Coimbra, Portugal  
{ecoelho, lmveloso, edmundo}@dei.uc.pt

<sup>2</sup> DoCoMo Euro-Labs, Landersbergerstr, 312, 80687 Munich, Germany  
mendes@docomolab-euro.com

**Abstract.** Seamless handover over heterogeneous mobile environments is a major requirement to the success of the next generation of networks. However, seamless movement requires the control of the quality level and connectivity of communication sessions with no perceived service degradation to the users. This seamless characteristic is equally important for communication sessions encompassing only one or multiple receivers, being the latter called multi-user sessions. This paper presents a solution to allow seamless mobility for multi-user sessions over heterogeneous networks with mobile receivers and static senders. The proposed solution integrates end-to-end *Quality of Service* (QoS) mapping, QoS adaptation and connectivity control with seamless mobility support. The latter is achieved by using buffers in the mobile nodes and caches in the access-routers together with mobility prediction and context transfer schemes. Simulations present the efficiency of this proposal to setup ongoing sessions and its impact in reducing packet losses during movement.

**Keywords:** Multi-user sessions, Seamless mobility, Quality of Service, Heterogeneous Networks.

## 1 Introduction

The increasing number of wireless devices together with the offering of new multi-user services such as IPTV, video-streaming, push-media and entertainment is creating a demand for mobile group communications. This creates the need for a communication system able to simultaneously distribute content to multiple users with no perceived service degradation over networks with different connectivity schemes and capability levels. To handle such heterogeneity, scalable multi-user sessions bring the benefit of avoiding application dependent processing inside the network. Each scalable multi-user session is composed by a set of flows, with well-defined priorities and rates. This generic definition accommodates common encoders, such as H.264 and MPEG-4.

Multi-user sessions must be distributed independently of the underlying QoS model, link capacity, access and transport technologies, which may be different in each network along the communication path [1]. For instance, the *Differenced Service* (DiffServ) model and the *Protocol Independent Multicast for the Source-Specific*

*Multicast* (PIM-SSM) can be implemented inside a network to provide QoS assurance and packet distribution to multiple receivers, respectively. Between networks, packet distribution can be based on unicast communications, since the number of links between adjacent networks is not expected to be very high. Furthermore, seamless handovers must be assured to increase the satisfaction of users. The seamless characteristic should be provided by the reduction of packet loss and latency of ongoing multi-user sessions.

This paper presents a solution to allow seamless mobility across heterogeneous networks with QoS and connectivity control. This solution is achieved through the integration of *Seamless Mobility of Users for Media Distribution Services* (SEMUD) [2] and *Multi-user Session Control* (MUSC) [3] mechanisms. The former aims to assure seamless mobility based upon the cooperation between caches in access-routers and buffers in mobile devices. A higher seamless assurance is achieved by using session context transfer and mobility prediction. The MUSC mechanism controls the setup of ongoing multi-user sessions in new end-to-end paths based on the multicast remote-subscription method [4]. This control is performed by providing QoS mapping, QoS adaptation and the coordination of connectivity translations on the predicted paths. MUSC performs its control in the direction of the receivers, which may be different from the reverse-path normally used by multicast routing protocols [5]. The SEMUD and MUSC mechanisms are being investigated within the *QoS Architecture for Multi-user Mobile Multimedia* (Q3M) architecture [6], which allows seamless mobility, by controlling multi-user session, network resources and handover.

The remainder of this paper is organized as follows. Section 2 presents the related work. A brief overview of MUSC and SEMUD is described in Section 3. An illustration of the functionalities is shown in Section 4. The efficiency of MUSC to setup ongoing multi-user sessions and the benefits of the cache and buffer mechanisms in reducing packet losses are analysed in Section 5. Conclusions and future work are summarized in Section 6.

## 2 Related Work

In what concerns mobility control, IETF protocols, such as *Mobile IP* (MIP), *Hierarchical MIPv6* (HMIPv6) and *Fast Handovers for IPv6* (FMIPv6), are being developed for controlling unicast sessions. In what involves multicast, two mobility control methods can be pointed out: the bi-directional tunnelling based on MIP, and the remote-subscription technique [4]. Both lack QoS support and are dependent of specific connectivity technologies, such as multicast from source to home agent in the former, and the same multicast address realm end-to-end in the latter.

The *Session Initiation Protocol* (SIP) can also be used to control handover at the application layer. SIP has the advantage of keeping mobility support independent of the wireless and network layer elements [7]. However, this solution only allows handover in unicast environments.

The above mentioned mobility control proposals aim to provide session continuity, but not in a manner that would provide a seamless handover. An exception can be FMIP, which allows an anticipated reaction to movement followed by the transfer and buffering of packets from the old to the new access-router. However, the context

transfer provided by FMIP is specific to this protocol. The use of buffers to store packet during handover is also supported by *Low Latency Handoffs in Mobile IPv4* (LLH) [8] and *A Distributed Buffer Management Approach Supporting IPv6 Mobility* [9] proposals. However, none of these approaches are suitable for multicast sessions.

Another scheme that allows handover with faster re-establishment of unicast or multicast sessions is the IETF *Context Transfer Protocol* (CXTTP). CXTTP can achieve seamless mobility by transferring the session context to new access-routers before the handover [10] [11]. However, CXTTP alone is not sufficient to provide seamless mobility of multi-user sessions, since it does not setup ongoing sessions.

In addition to the mobility control of multi-user sessions, the heterogeneity of networks poses requirements for the mapping and adaptation of QoS, as well as for the translation among different connectivity schemes. There are several proposals to map QoS for single-user sessions, although they require proprietary modules on the end-systems [12], which reduce the system flexibility. Ruy *et al* [13] propose a centralized agent that classifies session requirements into classes of service between networks with different QoS models. However, it is focused on the QoS metrics used by its agents and does not present the cooperation between agents to control QoS mapping along the end-to-end session path. Mammeri [14] proposes a mapping scheme that provides QoS guarantees for unicast sessions across *Integrated Service* (IntServ) and DiffServ models. However, this approach is dependent of the underlying QoS model.

Furthermore, QoS adaptation solutions are used to adjust the overall quality of a session to the capability of different networks. However, most of them require the implementation of modules on end-hosts to join/leave flows of multicast sessions [15], or they need network devices to adapt the content coding (re-coding) to the available bandwidth [16]. In the latter, networks are dependent from encoders, which decreases the system flexibility.

The end-to-end connectivity control over heterogeneous networks can be accomplished by using tunnel-based [17] or translation-based [18] approaches. The former requires the same IP multicast address realm in both access-networks, which is not imposed by the latter. Existing translation solutions provide unidirectional conversion between unicast and multicast realms [19][20], but miss to control connectivity between multicast networks with different address realm, such as networks implementing *Any-Source* and *Source-Specific Multicast*.

The analysis of related work has shown that none of the approaches satisfy all requirements to support the seamless mobility of multi-user sessions over heterogeneous networks. Most of them were developed to be used in networks with specific QoS models or connectivity technologies.

### 3 Seamless Mobility over Heterogeneous Environments

Seamless mobility for multi-user sessions over heterogeneous environments is achieved through the integration of SEMUD and MUSC mechanisms. The cooperation between these two mechanisms provides seamless mobility control with end-to-end QoS mapping and adaptation, as well as connectivity support. Due to the heterogeneity of the networks, this proposal is based on the separation of multi-user session identifier and

network locator. While the session identifier has a global meaning, the network locator is only relevant for the local network. Hence, each multi-user session is described in a *Session Object* (SOBJ) identified by a session identifier, which can be composed by a set of flows. The QoS parameters of each flow are described in QSPEC object [21].

It is assumed that receivers get from the source, by any off-line or on-line means, information about the available sessions, encompassing the SOBJ and QSPECs. Each QSPEC includes the flow priority, bit-rate, tolerance to loss, delay and jitter.

The location of the MUSC and SEMUD agents is illustrated in Fig. 1, which shows that MUSC agents are implemented on network edges while SEMUD agents are implemented on access-routers and mobile devices. Agents are called *access-agents* when they encompass the SEMUD and/or MUSC functionalities. Moreover, agents can have distinct roles in different edges for different sessions: in an edge router, an agent is called an *ingress-agent* for sessions whose traffic is entering the network in that edge router, or *egress-agent* if the traffic is leaving the network.

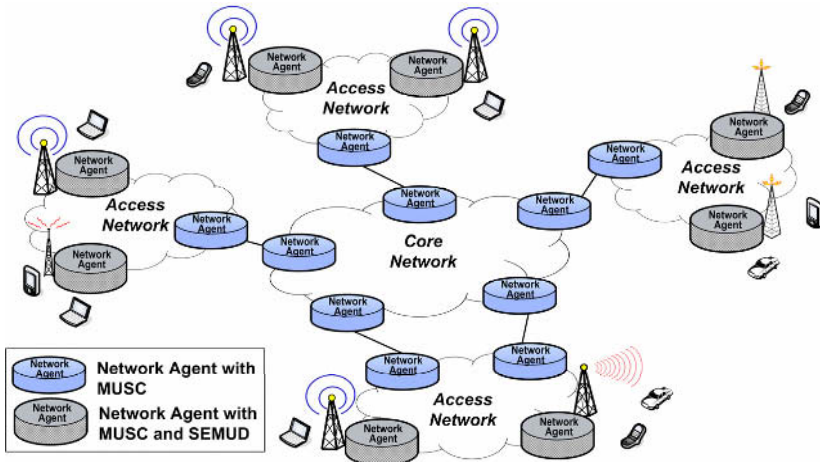


Fig. 1. MUSC and SEMUD location in a generic scenario

As an overview, the interoperation between MUSC and SEMUD mechanisms can be described as follows. After attaching to an access-agent, receivers send the session description to the MUSC agent located in the access-agent by using SIP and the *Session Description Protocol* (SDP). After the session has been configured (QoS mapping, QoS adaptation and connectivity translation) in all edge of the end-to-end path, MUSC notifies SEMUD in the access-agent about the context of the new session. During handover, SEMUD transfers the SOBJ to new access-agents, allowing the session to be configured by MUSC on the new end-to-end paths.

### 3.1 SEMUD Overview

SEMUD aims to provide seamless handover among access-agents belonging to the same or neighbour networks through the combination of context transfer, mobility

prediction and cache and buffer mechanisms. Packet losses and delay are reduced by controlling caches in access-agents and buffers in mobile devices.

The data packets received in the access-agent are stored in the cache (where the oldest packets are removed) and forwarded to the interested receivers. When those packets are received by a mobile device they are stored into the buffer and consumed by the application. In the presence of a handover, the data in the buffer of the mobile device will continue to be read in order to keep the data flow. When the handover is complete, the mobile device updates its buffer by fetching the missing packets from the cache. In the message sent to fetch the packets it will be transported information concerning the available space in the buffer, the time stamp of the last packet received in the buffer before handover and the intended multicast session represented in the PIM-SSM protocol by  $\langle \text{Source}, \text{Group} \rangle$  or  $\langle S, G \rangle$ .

The presence of the session in the next access router is guaranteed through the support of mobility prediction schemes. This is, the most probable cells to where the mobile device will move are predicted based on parameters like the moving direction, velocity, current position and historical records. Based in this forecast, the session context is transferred in advance into the predicted cells by the SEMUD-P signalling protocol. This way, the interaction between the SEMUD and the MUSC will permit to reserve the resources in advance in the predicted access routers. Additionally, information concerning the capabilities provided by the predicted access-agents is collected and conveyed by the SEMUD-P to the current access-agent. At the old access-agent, the probed information concerning the available resources in the predicted access-agents, the signal-to-noise ratio and the knowledge regarding the access technologies, gives support to the handover decision. When the handover decision is taken, the communication between SEMUD and MUSC allows the release of resources reserved on the old path, and on the new paths that the mobile device is not going to use.

### 3.2 MUSC Overview

MUSC aims to support the mobility of multi-user sessions by controlling the cooperation between all edge-agents in the end-to-end path. A receiver-driven and source-initiated protocol, called MUSC-P, is used to exchange information between MUSC agents using a soft-state approach to maintain per-session and per-flow state. It is receiver-driven because it is triggered at access-agents. It is source-initiated since MUSC starts the configuration of edge-agents at the agent nearest to the source, or at the first agent discovered with the requested session in the path towards the source. This functionality aims to build distribution trees taking into account the QoS characteristics of the path from source to receivers, which may be different from the reverse-path used by multicast protocols.

In ingress and egress-agents, the QoS mapping is based on the association between the session requirements (QSPEC object) and the available network services provided by a network resource allocation controller (e.g., *Service Level Agreement* (SLS) controller between networks). If such mapping is not optimal, an adaptation to the current network conditions is performed. The QoS adaptation mechanism can request more resources to be allocated to the selected service class, the mapping of flows into another service class, or the dropping/joining of low priority flows of the multi-user session. This adaptation allows the mobile user to keep acceptable quality level of



ongoing sessions, independently of its movement. In addition, the configuration of a connectivity translator allows services to be offered over networks with different address realms (unicast or multicast).

### 3.3 MUSC and SEMUD Interfaces

Interfaces are supported by MUSC and SEMUD to exchange information between themselves, existing solutions and standards.

The *address allocation controller interface* is used by MUSC to control the connectivity of multi-user sessions. In all downstream agents, MUSC interacts with an address allocation controller to request the allocation of channel identifiers for flows of a session. If PIM-SSM is supported, it is triggered by MUSC in edge-agents to create multicast branches associated with each flow. This interface is also used to inform an address allocation controller and PIM-SSM about flows that were removed.

The *resource allocation controller interface* is used by MUSC to query information about network classes and their available bandwidth towards the access-agent in which the receiver is attached or is moving to (the IP address of the next edge-agent is furnished by the resource controller). After the mapping process, MUSC informs which network class was selected and which bandwidth is required for each flow of multi-user sessions. In congestion situations, the QoS adaptation mechanism uses this interface to adapt flows of sessions to the current network conditions. The resource controller is also notified about flows that were release by MUSC.

The *access controller interface* allows SEMUD to keep the session context and to create a cache for the session (if the session still does not exist) when a session is accepted by MUSC. The opposite operation is triggered by MUSC when a session ends. During handover, MUSC is requested in predicted access-agent(s) to setup the session and to collect information concerning the capability and connectivity provided by the latter. At the previous access-agent, the interaction between SEMUD and MUSC allows the delete of resources associated with the session on the old path.

The *SEMUD mobility prediction interface* is used to allow the interaction with a movement prediction module. The last one is used to predict the next most probable access-agents based on the location of the base stations and on the properties of the mobile device such as location, moving direction and velocity. For instance, this interface allows the interaction with a mobility prediction scheme as the one proposed in [22].

The *multicast activate interface* allows multicast-aware receivers connected to multicast-aware access-networks to leave/join multicast channels associated with each flow of a session informed by MUSC. This is done based on the interaction of SEMUD and IGMPv3/MLDv2, where a leave message is triggered during the disconnection from the old access-agent and a join message is requested after attaching to the new access-agent.

## 4 Illustration of the Overall Functionality

Fig. 2 presents an example of MUSC and SEMUD operations in an inter-network handover scenario. It is assumed the existence of an anticipated handover scheme that

interacts with SEMUD to obtain the IP address of the candidate access-agent. Moreover, in each agent MUSC triggers a resource and an address allocation controller during the QoS (mapping and adaptation) and connectivity control operations, respectively.

Based on the interaction with the mobility prediction mechanism, SEMUD verifies that  $R1$  (step  $i$ ) is moving away from the access-agent- $A$  and that access-agent- $K$  is the candidate access-agent. Upon receiving the IP address of the predicted access-agent, SEMUD-P sends a *ResourceQuery* message to the SEMUD agent in the access-agent- $K$ . This procedure allows the session setup on the new path (by triggering MUSC and notifying it about the SOBJ) and the creation of a cache for the session in the new access-agent.

MUSC agent verifies that  $S1$  is neither locally active nor in the access-network  $N3$  and a *SessRequest* message is sent towards the source of the session. This message is stopped in agent- $F$ , since it has another branch with the same requested flows of  $S1$ . In this agent, MUSC interacts with the resource allocation controller to query information about inter-network service classes. Based on the response and QoS parameters described in the QSPEC, MUSC selects the network class and requests to the address allocation controller, the allocation of a pair of IP unicast addresses and transport ports to identify each flow of  $S1$  between  $N2$ - $N3$  (in this case, the source is the agent- $F$  and agent- $J$  is the destination). After that, the resource allocation controller is triggered to configure the required bandwidth for each flow in the selected class.

After resource and address allocation operations, MUSC in agent- $F$  starts the address translation and the packet replication of each flow of  $S1$  in the inter-network link (different flows have the same source and destination, agents- $H$  and  $F$ , but are identified also by different ports). In the control plane, the agent- $F$  sends a MUSC-P *SessResponse* message to agent- $J$  (the IP address was informed by the resource controller). The reception of this message allows MUSC in the agent- $J$  to update its state with the channel identifier allocated to the flows of  $S1$ . The MUSC interaction with the resource and address allocation controllers occurs as described before.

Since the agent- $K$  is the new access-agent of  $R1$ , PIM-SSM is triggered by MUSC to create the multicast trees for each flow of  $S1$  inside  $N3$  (rooted at ingress-agent- $J$ ). The resource controller is activated by MUSC to provide service differentiation for each flow on the wireless link. Finally, SEMUD is triggered and receives information about the request, including the new SSM channels used for each flow in  $N3$ . After activating a cache for  $S1$ , a SEMUD-P *ResourceResp* message is sent to the access-agent- $A$ .

Upon the reception of the *ResourceResp* message, SEMUD analyses the information concerning the available resources (in the predicted access-agents) and the signal-to-noise ratio, and decides to handover to the predicted access-agent- $K$ . In access-agent- $A$ , SEMUD sends a *HandoverBearer* message to inform  $R1$  about the IP address of the future access-agent and the SSM channel allocated to each flow. After the handover, SEMUD informs MUSC to adjust the number of receivers associated with  $S1$  in the previous access-agent. However, the state on the old path is not released by MUSC, because  $R2$  is still receiving data from  $S1$ .

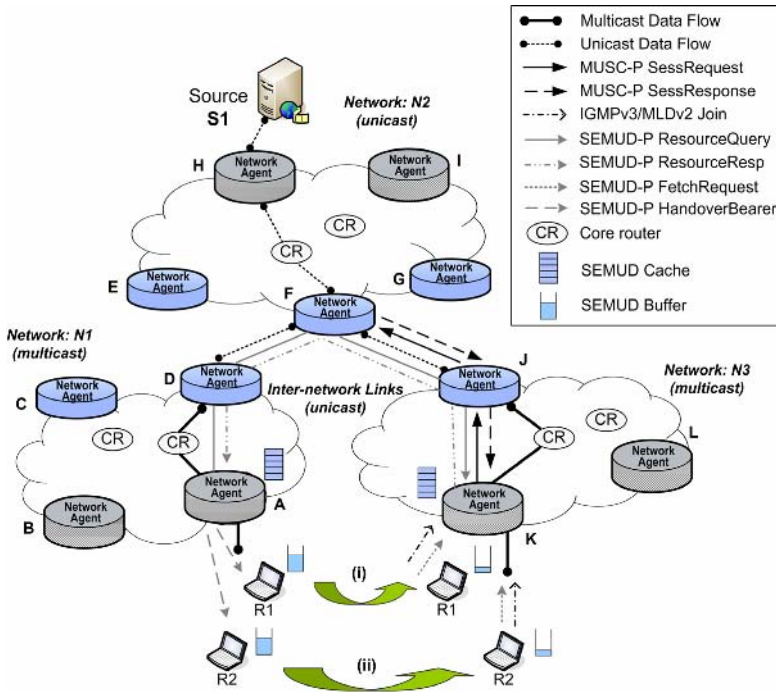


**Fig. 2.** Inter-network Handover

During handover, packets are stored in the cache of the new access-agent and after the attachment of *R1* to this agent, SEMUD-P is triggered to send a *FetchRequest* message to recovery missing packets and to sync the packet reception with the cache in agent-K. The recovered packets are sent from the cache to the buffer via unicast connections. This requires encapsulation of the multicast packets into unicast packets, which SEMUD in *R1* de-encapsulates before putting them in the session buffer. This functionality avoids packet replication to other receivers subscribed in the same multicast group and also attached to agent-K. After receiving all the fetched packets, SEMUD triggers the IGMPv3/MLDv2 to join the multicast channel allocated for each flow of the session.

After the handover of *R1*, SEMUD verifies in agent-A the next access-agent to which *R2* will move (*step ii*) and triggers MUSC to pre-set the session on the new path. Upon receiving the SOBJ transferred by the SEMUD-P *ResourceQuery* message, MUSC verifies that *S1* is already activated and increments the number of receivers receiving the requested session. After the MUSC reply, SEMUD associates *R2* with the existing cache and sends a *ResourceResp* message to the previous agent to complete the handover. The seamless handover process is accomplished as explained for *R1*. However, MUSC in agent-A releases the state associated with *S1*, triggers SEMUD to remove the cache, the resource controller to erase network resources and PIM-SSM to delete the multicast trees of each flow in *N1*.

In agent-D and agent-F, MUSC removes the *S1* state because no MUSC-P *SessRefresh* message arrives to these agents before the expiration of the MUSC

clean-up interval. This requires the interaction with resource and address allocation controllers to release their state associated with the removed flows.

## 5 Performance Evaluation

Several simulations were done in the *Network Simulator-2* (NS2) to verify the performance of MUSC and SEMUD to control the setup of ongoing sessions with seamless experience to the receivers. The convergence time of both mechanisms and the percentage of packet loss during handover with and without the cache and buffer mechanism are analysed.

Three topologies (A, B and C) were randomly generated by BRITE with the same inter-network scenario as illustrated in Fig. 2. In any topology, each of the three networks has twenty routers (four edges and sixteen cores). Two receivers are placed in the same access-agent and get one *Variable Bit Rate* flow with an average rate of 86 KB/s and packets size of 1.052 bytes. The intra and inter-network links have a bandwidth of 100 Mb/s and the wireless link capacity is of 11 Mb/s. The propagation delay inside and between networks is attributed according to the distance between the edges. A PIM-SSM agent for NS2 [23] is implemented to distribute the session packets.

Since the mobility prediction is under investigation, it is assumed that SEMUD is notified in advance about the movement of receivers to the predicted access-agent in the new access-network. This notification occurs in a period of time sufficient to allow the session setup on the new path before the disconnection from the old access-agent.

Table 1 describes the SEMUD and MUSC convergence time to setup the session before and after the attachment of each receiver to the new access-agent. Before the handover, SEMUD signalling and the cache configuration procedures are accomplished. The MUSC convergence time encompasses signalling and the configuration of the session mapping and connectivity for the first receiver. All posterior requests for the same session in the same access-agent are processed locally by MUSC as happens with R2. After the handover, SEMUD convergence time includes the fetching of the missed packets stored in the cache.

**Table 1.** MUSC and SEMUD convergence time (ms) before and after the attachment of the receivers to the new access-agent

| Topology | Receiver | Before |       | After | Total (ms) |
|----------|----------|--------|-------|-------|------------|
|          |          | SEMUD  | MUSC  | SEMUD |            |
| A        | R1       | 26.15  | 16.23 | 1.07  | 43.45      |
|          | R2       | 26.15  | -     | 1.07  | 27.22      |
| B        | R1       | 28.08  | 18.23 | 1.12  | 47.43      |
|          | R2       | 28.08  | -     | 1.12  | 29.20      |
| C        | R1       | 27.54  | 16.82 | 1.03  | 43.39      |
|          | R2       | 27.54  | -     | 1.03  | 28.57      |

The overall convergence time to setup the session for R2 is reduced in 36%. This is because the requested session is already activated. Only MUSC local procedures are done (negligible time) to configure the number of receivers, quality level and connectivity functions associated with the session and to reply SEMUD. This

functionality minimizes 48% in the overall signaling overhead, because only SEMUD-P messages are used. The latency to setup the sessions is increased in approximately 1% by MUSC and SEMUD operations. Note that the MUSC convergence time would be higher if the requested session for *RI* would be activated in an agent near the source.

The advantage of the proposal in reducing the impact of the handover on the user perceived quality was also analyzed. Since the perceived quality is directly affected by the amount of packet losses, this parameter was evaluated versus a buffer size varying between 1 and 100 KB and for a handover duration with 500 ms. The results depicted in Fig. 3 were obtained for several values of the cache size when the mechanism is enabled, and compared with the situation when it is disabled.

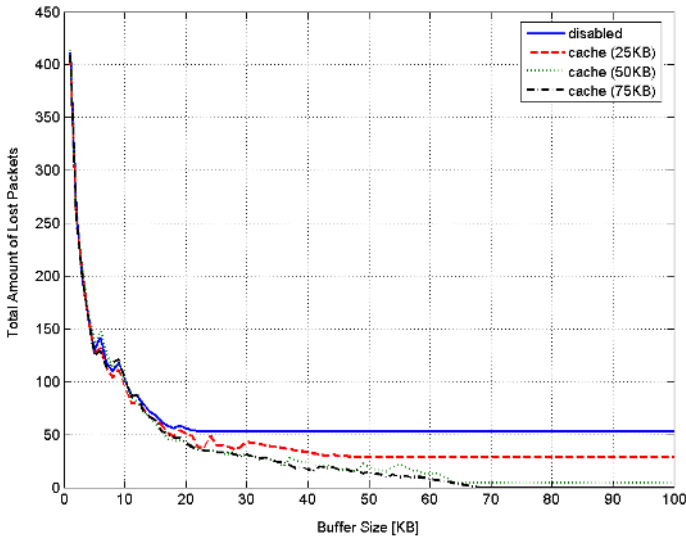


**Fig. 3.** Total amount of lost packets versus the buffer size with different cache sizes

As the cache size increases the amount of packets that is possible to recover augments, and consequently, the total number of lost packets is reduced. Similarly, as the buffer size increases it is possible to accommodate a larger number of packets coming from the cache in the recovering process. The obtained improvement in reducing packet losses when the SEMUD mechanism is enabled is notorious. Namely, the packet losses are totally reduced for a combination of a cache size with 54 KB and a buffer size with 68 KB.

## 6 Conclusions and Future Work

This paper presents a proposal to allow seamless mobility of multi-user sessions over heterogeneous networks. This is achieved through the integration of *Seamless Mobility of Users for Media Distribution Services* (SEMUD) and *Multi-user Session*

*Control* (MUSC) mechanisms. SEMUD supports seamless mobility based on the cooperation between caches placed in access-routers and buffers placed in the mobile devices, multi-user session context transfer, and interaction with mobility prediction schemes. MUSC controls mobility by assuring the control of QoS mapping, QoS adaptation and connectivity of sessions among networks.

The performance evaluation shows that MUSC has a convergence time and a signalling overhead independent of the number of receivers of the same session in the same access-network. Moreover, the expected advantages of SEMUD in reducing packet losses are confirmed by the obtained results. For example, the packets losses are reduced 75% when considering a cache size with 50KB and a buffer size with 55 KB. As future work, further evaluation will be done to confirm the performance of MUSC and SEMUD in an experimental scenario.

## References

1. F. Hartung *et al*, "Advances in Network-Supported Media Delivery in the Next-Generation Mobile Systems", *IEEE Communications Magazine*, vol. 44, issue: 8, August 2006.
2. L. Veloso *et al*, "Mobility Support of Multi-User Services in Next Generation Wireless Systems", *IEEE International Performance Computing and Communications Conference*, April 2007.
3. E. Cerqueira *et al*, "Multi-user Session Control in the Next Generation Wireless System", *ACM International Workshop on Mobility Management and Wireless Access*, October 2006.
4. Y. Min-hua *et al*, "The implementation of multicast in mobile IP", *IEEE Wireless Communication and Networking*, March 2003.
5. H. Yihua *et al*, "On routing asymmetry in the Internet", *IEEE Globecom*, November 2005.
6. E. Cerqueira *et al*, "A Unifying Architecture for Publish-Subscribe Services in the Next Generation IP Networks", *IEEE Globecom*, November 2006.
7. C. Yeh *et al*, "SIP Terminal Mobility for both IPv4 and IPv6", *International IEEE Conference on Distributed Computing Systems Workshops*, July 2006.
8. K. Malki, "Low Latency Handoffs in Mobile IPv4", *IETF Internet Draft*, October 2005.
9. Y. Liu and Y. Chen, "A Distributed Buffer Management Approach Supporting IPv6 Mobility", *10<sup>th</sup> IEEE Workshop on Future Trends of Distributed Computing Systems*, May 2004.
10. J. Hillebrand *et al*, "Quality-of-Service Management for IP-based Mobile Networks", *IEEE Wireless Communications and Networking Conference*, vol. 2, March 2005.
11. I. Miloucheva, "Context Management for Efficient Mobile Multicast Services", *International Workshop on Context in Mobile Human Computer Interfaces*, September 2005.
12. M. El-Gendy *et al*, "Paving the first mile for QoS-dependent applications and appliances", *IEEE International Workshop on*, June 2004.
13. M. Ruy *et al*, "QoS class mapping over heterogeneous networks using Application Service Map", *IEEE International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies*, April 2006.
14. Z. Mammeri, "Approach for End-to-End QoS Mapping and Handling", *Wireless and Optical Communications Networks. IFIP International Conference on*, March 2005.

15. H. Chiu and K. Yeung, "Fast-response Receiver-driven Layered Multicast with Multiple Servers", *IEEE Communications 2005 Asia-Pacific Conference on*, October 2005.
16. R. Hayder *et al*, "Scalable Video TranScaling for the Wireless Internet", *EURASIP Journal on Applied Signal Processing*, n. 2, February 2004.
17. B. Zhang *et al*, "Host Multicast: A Framework for Delivering Multicast to End Users", *Annual Joint Conference of the IEEE Computer and Communications Societies*, June 2002.
18. E. Pearce *et al*, "System and Method for Enabling Multicast Telecommunications", *Technical Report - Patent*, US7079495 B1, July 2006.
19. L. Begeja, "System and Method for Delivering Content in a Unicast/Multicast Manner", *Technical Report - Patent*, US708142 B1, July 2005.
20. Z. Jumbiao, "Multicast over Unicast in a Network", *Technical Report - Patent*, MX2006PA03857, July 2006.
21. J. Ash *et al*, "QoS NSLP QSPEC Template", *IETF Internet Draft*, October 2006.
22. M. Sricharan *et al*, "An Activity Based Mobility Prediction Strategy for Next Generation Wireless Networks", *IFIP Conference on Wireless and Optical Communications Networks*, April 2006.
23. T. Camilo, "SSM Extension for NS-2", [http://eden.dei.uc.pt/~tandre/ssm\\_extension](http://eden.dei.uc.pt/~tandre/ssm_extension), December, 2006.

# QoS and Authentication Experiences in a Residential Environment Within a Broadband Access Framework

Iván Vidal<sup>1</sup>, Francisco Valera<sup>1</sup>, Jaime García<sup>1</sup>, Arturo Azcorra<sup>1</sup>,  
Vitor Pinto<sup>2</sup>, and Vitor Ribeiro<sup>2</sup>

<sup>1</sup> Universidad Carlos III de Madrid, Avda. De la Universidad 30  
28911 Leganés, Madrid

{ividal, fvalera, jgr, azcorra}@it.uc3m.es

<sup>2</sup> Portugal Telecom Inovação, Rua Eng. José Ferreira Pinto Basto  
3810-106 Aveiro (Portugal)

{it-v-pinto, vribeiro}@ptinovacao.pt

**Abstract.** It is sometimes believed that a “broadband access” network, providing ample transmission capacity to residential environments, is enough so as to allow a flawless delivery of advanced services. However, the provisioning of a combination of multiple services with guaranteed quality up to the end-user terminal requires a carefully designed architecture incorporating the appropriated Quality of Service (QoS) concepts throughout the data path. And this path includes the Residential Gateway (RGW) as the last hop towards the home network. This paper describes the different experiences performed with the RGW prototype developed within the framework of the European IST research project MUSE. Special emphasis will be made on the QoS capabilities of the RGW as well as on authentication and auto-configuration features.

**Keywords:** RGW, triple play, broadband, QoS, 802.1X, trials.

## 1 Introduction

Nowadays, one of the most common trends mentioned in the communication network environment is the one related to ‘convergence’. Convergence from services viewpoint allowing video, audio and data to be merged on the so called triple play provisioning and convergence on networks allowing fixed and mobile scenarios (even cellular) to be combined into a single architectural model. And these convergences have been facilitated by means of the provisioning of a large amount of throughput to the final users.

Although it is very common to find that in residential environments, access lines support from 1 Mbps to 20 Mbps (ADSL, ADSL+2, etc.), it is just a question of time that users are capable of filling their access lines with multimedia or peer to peer content and all the applications will be forced to share a limited amount of resources.

In such a common resource restrictive home environment it is remarkable that bandwidth is not at all the only quality of service parameter that must be guaranteed, since there are many others parameters like latency, jitter, packet loss, etc. that may also be crucial for certain applications to run properly.



This article describes the results of the different experiences, performed with a QoS enabled RGW (RGW) within the framework of a broadband environment such as the one specified by MUSE, which is a large integrated research and development European project on broadband access, whose main objective is the specification and deployment of a future, low cost, multi-service access network.

The RGW is responsible for delivering services to the end-user terminal and so it is responsible for receiving the frames coming from the access network and transferring the quality of service devised for it towards the home network. And it is also responsible for sending the frames towards the access network tagged with the corresponding quality of service so that the network can accordingly process it. The rest of the article is structured as follows.

The second section describes the RGW, first within the MUSE research project and afterwards from a functional and architectural point of view focusing on the two main functionalities that will be trialed: authentication and QoS.

The third section explains the different test and trials performed with this RGW platform and finally the last section summarizes the most important conclusions and provides some guidelines about the possible future work that is being scheduled.

## 2 A Quality of Service Enabled Residential Gateway

### 2.1 The Residential Gateway in MUSE Project

MUSE (MultiService access Everywhere, [1]) is a large project on broadband access that belongs to the 6th Framework Programme for R&D of the European Union. MUSE aims at a consensus view of the future access and edge network achieved by the co-operative research of almost all major players in Europe (36 partners, including system and component vendors, telecom operators, SMEs and universities and research institutes). The project integrates studies in the following areas:

- Access and edge network architectures and techno-economical studies.
- First mile solutions (DSL, optical access).
- Internetworking of the access network with RGW and local networks.
- Lab trials.

Figure 1 shows an overview of MUSE general architecture where different scenarios are depicted and the important boxes are remarked. One of the most relevant entities within the whole MUSE architecture is the RGW which is located at the edge of the access network that MUSE is specifying. It means that the RGW must be compliant with all the different functionalities supported by this network in order to be able to make them compatible and to extend them towards the home network. The prototype presented in this article is included in a particular subproject in MUSE focused on a FTTH broadband access scenario (up to 1Gbps). The functionalities of this RGW prototype can be divided in three different groups:

- *Initial autoconfiguration*: the RGW software performs an automatic discovery of the hardware where it is being run, including the number of network interfaces and from them, which one is being used as WAN interface. The RGW authenticates itself towards the network provider and automatically configures the connectivity layers of the WAN and the LAN side (IP addresses, DHCP server, SIP ALG, etc.).

- *Operation*: apart from the traditional NAT and firewall functionalities available on current RGW devices, the prototype is also capable of performing some other actions such as NAT traversal for SIP, based on an application level gateway and for STUN based clients by means of an embedded STUN server, multicast delivery or QoS provisioning (both tagging upstream flows with a 802.1pq header so that the network can properly treat them and providing the corresponding QoS to downstream frames and promoting that QoS into the residential network).
- *Management*: apart from the manual configuration mechanism that is based on a Java servlet guided Web interface and allows a complete configuration of the RGW, other automatic alternatives have also been included like the DSL Forum TR-069 standard [2] mechanism or by means of the SIP signaling protocol (allowing the RGW to be integrated into an IMS/NGN architecture).

All these commented functionalities are structured in the RGW prototype architecture divided in two different layers (see Fig. 2).

The **data layer** is responsible for data processing including routing and bridging decisions, shaping/policing functions, flow classification, tagging/untagging of frames with the corresponding VLAN and corresponding p bits, queuing facilities, etc. This data layer (or kernel layer from the implementation point of view) has been implemented using the Click! platform which is a modular software router developed by the MIT, the ICSI and the UCLA for the Linux operating system [3].

The **configuration/management layer** is responsible for the management of the services and the management of the network layer, the configuration of the different parameters of the RGW like QoS parameters, NAT/ALG functionality, flow classification, etc.

In addition, this layer is also responsible for supporting applications that are capable of interpreting different signaling protocols that in turn, will also configure diverse RGW parameters like SIP, IGMP, RTCP, etc. The configuration/management layer has been implemented using Java (some specific modules have been

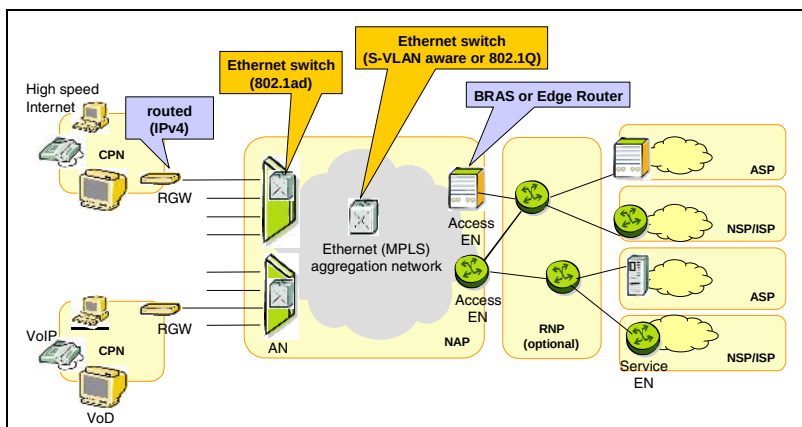


Fig. 1. MUSE overview



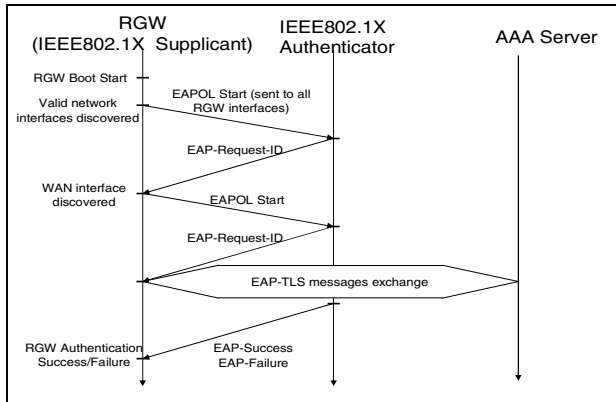


**Fig. 3.** RGW interface discovery and authentication

Although EAP-TLS [9] based authentication was used, other authentication methods are also possible (e.g. EAP-AKA, EAP-MD5, EAP-SIM, etc.). The distribution of the different authentication related entities in the MUSE network is as follows: the 802.1X supplicant is located in the RGW, the 802.1X authenticator is located in the access node and the authentication server in the aggregation network (see Figure 1).

Besides authentication, some mechanisms for auto-configuration were also added. These deal essentially with RGW network interfaces discovery and WAN interface identification, allowing the portability of the RGW software to hardware platforms that can differ on the number of network interfaces. The following paragraphs describe the boot up sequence of the RGW.

The first task performed when the RGW is powered up, is the detection of all network interfaces present in the hardware platform. Considering the discovered interfaces, several variables (e.g. physical connectivity) are taken into account to select only valid interfaces, through which the RGW can try to authenticate itself.

The next process, uses this set of valid interfaces and determines which one is the WAN interface. To achieve this, an 802.1X “EAPOL-Start” message is broadcasted through every valid RGW interfaces. The 802.1X Authenticator present in the access network, answers with an “EAP-Request-ID” message containing the string “muse.net” (in “EAP-Message” field) used to identify the WAN interface.

The next step is then to launch an EAP-TLS authentication process through the RGW WAN interface. A failed authentication process will result in re-authentication attempts and finally in the halt of the boot up process, while a successful authentication process results in the achievement of the boot up process (see Fig. 3).

### 2.3 QoS

One of the most important points to be considered when delivering services not only to business users but also to residential users is the capacity to assure a certain quality on this provisioning. This quality does not only mean a considerable amount of bandwidth, as it is usually advertised in the commercials, and should not be provided

only until the access node. End to end QoS has already reached the terminals in the mobile (cellular) world with the IP Multimedia Subsystem (IMS) and is also being promoted to the fixed networks by some entities like the ETSI TISPAN. However, no QoS schema had been specified for a device such as the RGW at the moment. The proposal in this MUSE prototype is to enable the RGW to assure QoS to transit data, and to integrate residential network resources within the whole QoS schema, since nowadays we do not have anymore the single-PC residential context and it tends to be more a set of devices connected in a LAN.

The RGW must be able to understand QoS marked packets (802.1pq tagging in MUSE) so as to prioritize their processing and to propagate that marking to the home network (mapping the QoS tagging schema used in the access network to the schemas used in home networks like the 802.11e specification, IP DSCP bits, etc.). It also means that for upstream traffic the RGW should to map the QoS that packets are bringing from the terminals or in case this action is forbidden for the terminals, mark packets by itself based on configured information.

The RGW has to consider that traffic in the home network is consuming resources that are not usually taken into account (i.e. upstream traffic and downstream traffic may be transmitted over a shared medium together with local traffic, etc.).

Some of the functionalities implemented in the RGW to allow this QoS support include scheduling and policing, traffic shaping, call admission control, per flow classification or frame tagging. These possibilities are typically included in other network nodes, but here it is proposed their inclusion in the RGW because otherwise the home network would be left out of the overall QoS schema. These functionalities combined with the flexibility offered by the Session Initialization Protocol (SIP) so as to automatically set up services, will allow the development during the second phase of MUSE project of an IMS/NGN compatible RGW and would also facilitate new QoS enabled scenarios that may involve the RGW into P2P overlay networks, community networks, mobility, roaming, etc.

### 3 RGW TRIALS

#### 3.1 Authentication Trials

The trials were divided in three phases, each one intended to test a specific part of the RGW boot up process.

- **Discovery of the number of Ethernet interfaces on the RGW:** the objective of this trial was to check if all Ethernet compliant interfaces currently available on the RGW were correctly detected, considering as invalid interfaces with no physical connectivity. Tests were performed for different sets of interfaces with different physical connectivity availability (all available, none available, some available).
- **Discovery of the WAN interface:** the purpose of the test was to verify that among all valid interfaces, the WAN interface was correctly identified. Trials were performed for the following test cases:
  - Changing the interface of the RGW that is connected to the access network.
  - Changing “EAP-Message” field of the “EAP-Request-ID” message issued by the authenticator to other string than “muse.net”.

- **Authentication process:** the goal of this trial was to check the correct operation of the 802.1X supplicant implemented on the RGW. For this purpose both 802.1X supplicant and AAA Server (RADIUS server) were configured to perform an EAP-TLS authentication. Trials were performed for the below test cases:
  - Credentials sent by the supplicant contain an invalid certificate because the Certification Authority who issued the certificate is not known to the authentication server (the RGW should receive an “EAP-Failure” message).
  - In the same conditions than the previous point, check that after a failed authentication, re-authentication is tried again after 120 seconds.
  - Credentials sent by the supplicant are correct (the RGW should receive an “EAP-Success” message).

### 3.2 Operation Trials

A triple play scenario with applications like voice over IP, video streaming and bulk data transfer, is defined to test the Queue and Scheduling Functional Blocks inside the RGW, since these are the principal blocks to be tested in order to assure a complete end to end QoS. The main RGW operation characteristics to be tested in these trials are the following:

- **Queues functionality:** the RGW implements four queues (one per CoS) per interface. This functionality will be tested in several scenarios where different upstream/downstream flows (associated with video streaming, data bulk traffic and VoIP applications) will be processed at the corresponding queue.
- **Signaling functionality:** how the RGW processes the signaling flows, the overhead of the special treatment of these predefined signaling flows and their marking with a specific CoS different from the data flows.

The selection of applications used for the trials covers the four CoS defined in MUSE project (low latency, real time, elastic and best effort) and is based on the premise of providing a “full service” testing with a limited number of applications:

- **Voice over IP** as an example of the low latency CoS application for signaling flows and real time for data flows with a strict delay requirements. The main requirements for this application are the very low delay and jitter.
- **Video streaming** as an example of real time CoS application and a possible killer application for broadband networks. In addition, this application might generate a considerable amount of traffic and we have divided the tests in two cases, low quality video and high quality video. Within these RGW trials we will test video quality for both unicast and multicast traffic. The bandwidth and packet-loss are the main parameters that can affect the quality of this application. The results of the tests with the video streaming application can be classified in qualitative (subjective) and quantitative ones (objective).
- **Bulk data** as an example of elastic or best effort CoS application. It is related with Internet browsing or peer to peer communication and will be simulated with the Iperf application [10]. They can be considered commodity applications and can be provided with guaranteed QoS or with the lowest quality.

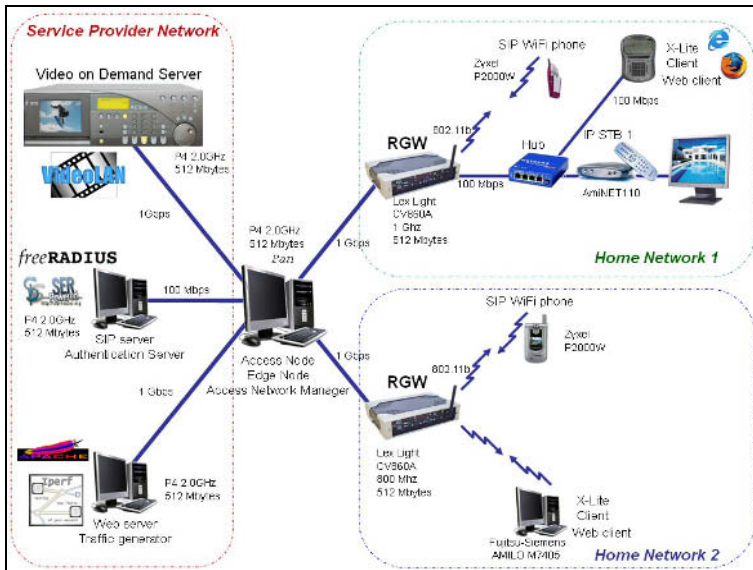


**Fig. 4.** Network trial testbed

An association between these applications and end to end network requirements (throughput, delay, packet loss and jitter) has to be defined in the trials. The results of the trials with respect to QoS will consider how to score and measure the perceived QoS at each application and/or scenario: objective measurements, subjective measurements and the mapping from network quality to perceived quality.

In order to test all the different concepts and characteristics previously described the considered scenario will be the one depicted in Figure 4. This scenario shows the residential environment on the right part of the picture with two different home networks connected to the Network Access Provider network through different RGWs, that may be connected to the same or different access nodes.

The access node will be 802.1pq aware (like the 802.1ad Ethernet switch in Figure 1) so that it will understand the VLAN encapsulation coming from the RGW with the corresponding p-bits and it will also be able to reformat the frame according to the VLAN schema used within the Network Access Provider network. At the other end (left hand side in the picture), the traffic will be received by the different servers that will provide the requested demand.

Since the development done by our working team in MUSE is focused on the RGW itself, the rest of the network is out of the scope of our workpackage. However, in order to properly test the RGW it was mandatory to emulate the whole network so that the RGW could in fact be involved into a real triple play scenario with real autoconfiguration on startup performed towards the Access Network, real authentication phase towards the Access Network, real signaling messages exchanged towards the corresponding counter part in the network and real services received from the service provider domain.

The triple play services will be provided to two different residential environments that will also interact between them (through a VoIP scenario based on SIP). All this

traffic interchange will be performed within a certain QoS framework that will guarantee the proper treatment of the different flows in the different QoS aware entities so that clients will receive the service without degradation.

### 3.2.1 Qualitative Trials

For these trials, three different kinds of flows were used to test the triple play scenario where video and audio applications are present in the video streaming (using the VLC application [11] for both server and client sides) and VoIP (using SER as the SIP server [12] and X-Lite as user clients [13]). To simulate constant user data (intensive Web browsing, FTP or peer to peer data for example) Iperf is used to generate raw frames in the Server Provider side and collect statistics in the client side.

Although video and data applications were configured using different rates, VoIP was tested using just one codec generating traffic at 120 kbps (high quality audio). To test the video scenario, two different sources were used with different video and audio codecs: low quality where both video and audio are transmitted at 2 Mbps (DivX for video and MP3 for audio) and high quality using 5.2 Mbps (MPEG2 for video and AC3 for audio).

It is important to notice that all experiments were executed during 30 seconds reinstalling all the devices at the beginning and gathering the results at the end. Due to this fact, the relevance of the queue sizes (10000 frames for each queue) is not so important because for a very long experiment the results could differ for a given value. The aim of these tests is to probe the feasibility and performance of the QoS system standardized in MUSE and developed for this prototype and not to obtain the best values for the queues length for a given performance.

Iperf was executed from different servers depending on the required QoS. It is always invoked to generate 100 Mbps.

In the first and simpler test, two different types of flows were generated: the SIP signaling, treated as low latency (the best quality) and the RTP media transfer (the voice) treated as real time. The registration process of both the SIP phone and the X-Lite SIP software is always almost instantaneous and the delay could be considered negligible. In the data (voice) transmission, no packets were dropped in the RGWs and no delay was appreciated.

In the second test, the goal is to observe the performance of the VoIP communication in a high load scenario when both signaling and data voice are marked with the highest priority. The results of this test were clear: both registration and data (voice) transmissions are performed with no delay even when the high load traffic was marked with the best quality of service. Only when there is a continuous audio stream being transmitted it can be appreciated that very small cuts appear and that the sound is perceived with a metallic tune. These effects are not appreciated in a normal VoIP conversation.

The third test tries to represent a complete triple-play scenario (with voice, low quality video and Iperf) where voice is marked with the highest priority, video uses the next one and Iperf simulates a high load traffic using variable priority (the lowest one in the first scenario, the same one than the video in the second scenario and the highest in the last one). With these three different scenarios the behavior of both the voice and the video transmission depending on the high load priority traffic is compared. The results confirm that the voice has too a low rate to be affected (even although a high rate codec was selected precisely because of this reason) by other traffic and video with low quality is unaffected too due to this fact.



The final scenario repeats the last tests using a high quality video. This time, when the high load is marked as low latency, the video reception is too bad (neither the video nor the audio are received during the high load transmission). As soon as the Iperf transmission ends, the video restarts its reception in the client side (it needs 2 or 3 seconds to resynchronize).

### 3.2.2 Quantitative Trials

The purpose of these final tests is to determine the real quality of the different flows focusing on the bandwidth and the jitter, so that the subjective results obtained in previous tests can be measured.

In the first scenario, three different Iperf flows are sent towards the same end user device in order to test the efficiency of the QoS procedure implemented in the RGW. Table 1 gathers the information about the test parameters including the bandwidth and the instant when the Iperf flows are started and stopped and the results are presented in Figure 5. As it can be seen, during the first ten seconds, the first flow shows the input rate at the output due to the RGW LAN interface whose performance is just 95 Mbps due to hardware limitations. As soon as the medium priority flow starts, the low one decreases its rate while the former is totally served. In the 20-40 seconds range, the three flows share the LAN interface but, as the RGW prioritizes the frames following the marks, the low priority flow is not served any more and its frames are lost because the queues have just capacity for 1000 frames. When the high priority flow ends, the low priority one gets some bandwidth and is totally served again when the medium priority ends. Regarding the jitter, the most important area is the range between 20-40 seconds where there are huge jitter variations for the medium priority frames and high jitter variations for the low priority one. The reason for this is that almost all low priority frames go lost so there are no frames to estimate the jitter. The jitter for high priority frames is almost inappreciable.

The aim of the second test is to show how low priority applications always use the available bandwidth if there is any. The results are as expected: between 20-40 seconds, the three flows share the medium with the two highest priority flows completely served and the best effort one using the available bandwidth. High priority frames do not suffer any jitter (or it is not perceptible), medium priority ones have less jitter than in the previous test because this time all frames are served but, as the higher priority are served first, there exists some jitter. Best effort frames are also served but after higher ones so a big jitter is introduced.

**Table 1.** Flow parameters for the first, second and third tests

| Priority           | Rate[Mbps]      |                 |                 | Start [s]       |                 |                 | End [s]         |                 |                 |
|--------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
|                    | 1 <sup>st</sup> | 2 <sup>nd</sup> | 3 <sup>rd</sup> | 1 <sup>st</sup> | 2 <sup>nd</sup> | 3 <sup>rd</sup> | 1 <sup>st</sup> | 2 <sup>nd</sup> | 3 <sup>rd</sup> |
| Best Effort (Low)  | 100             | 100             | 40              | 0               | 0               | 0               | 60              | 60              | 40              |
| Real Time (Medium) | 30              | 20              | 40              | 10              | 10              | 0               | 50              | 50              | 40              |
| Low Latency (High) | 80              | 60              | 60              | 20              | 20              | 10              | 40              | 40              | 30              |

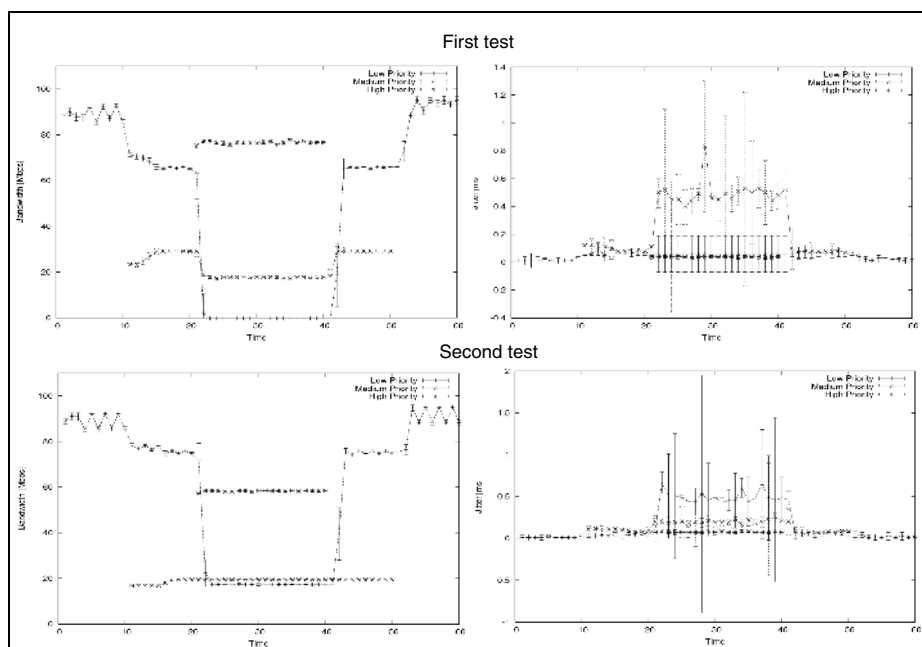


**Fig. 5.** Bandwidth and jitter value for the first and the second test

In the last test it is intended to see how two flows with the same priority share the remaining bandwidth in a Round Robin fashion. The result is the expected one: in the 10-30 seconds range, the highest priority flow gets the best service, while the two other flows reduce the output to 20 Mbps.

## 4 Conclusions

This article has presented the RGW prototype that has been developed and trialed within the framework of an Ethernet access network in a FTTH broadband scenario such as the one specified in the MUSE project.

The RGW prototype is prepared so as to be integrated in a QoS environment like the one specified by the MUSE project and the most important characteristics included in the prototype, are the following ones:

- The RGW prototype is capable of autoconfiguring itself independently of the hardware and of the network environment where it is deployed.
- The authentication procedure based on IEEE 802.1X is very flexible and allows many specific authentication methods to be applied.
- The RGW offers a very flexible configuration/management API making it possible to access it by means of a Web based interface, through the DSL Forum TR-069 standard, or through SIP (these are the implemented possibilities although any other could be integrated).

- The QoS capabilities included in the RGW allows it to expand the IEEE 802.1pq tagging schema used in MUSE access network towards the home network and it is also capable of treating the different flows accordingly to their marked QoS.
- The RGW is also capable of marking the different flows in the upstream direction with the required QoS.
- The RGW prototype incorporates an ALG in order to overcome NAT traversal problems caused by the SIP signaling messages.

These characteristics have been trialed and the different results have been shown in this article both from a qualitative and from a quantitative viewpoint.

For the second phase of the project (MUSE will finish at the end of 2007), different enhancements are being studied: integration of prototype into a TISPAN-NGN scenario, users and service roaming, fixed mobile convergence scenarios, value added services provided within the RGW (video server) or managed by the RGW (e-care), etc.

## Acknowledgements

This article has been partially granted by the European Commission through the MUSE (IST-026442) project.

## References

1. MUSE. Multimedia Access Everywhere. European Union 6th Framework Programme for Research and Technological Development. <http://www.ist-muse.org>
2. DSL Forum TR-069: CPE WAN Management Protocol, May 2004
3. The Click! modular router project. <http://www.read.cs.ucla.edu/click/>
4. QoS Management in Fixed Broadband RGWs. C. Guerrero, J. Garcia, F. Valera, and A. Azcorra. IFIP/IEEE International Conference on Management of Multimedia Networks and Services. MMNS 2005 (Oct 2005), Barcelona (Spain)
5. Designing a broadband RGW using Click! modular router. H. Gascón, D. Díez, J. García, F. Valera, C. Guerrero and A. Azcorra. IFIP EUNICE (Networked applications) 2005 (Jul 2005), Madrid (Spain).
6. Demo of Triple Play Services with QoS in a Broadband Access RGW. F. Valera, J. García, C. Guerrero, V. Pinto, V. Ribeiro. IEEE Infocom 2006. Barcelona (Spain)
7. IEEE Std 802.1X™- 2004, 802.1 IEEE Standard for Local and metropolitan area networks
8. IETF RFC 3748, June 2004. Extensible Authentication Protocol (EAP)
9. IETF RFC 2716, October 1999. PPP EAP-TLS Authentication Protocol
10. Iperf: The TCP/UDP Bandwidth Measurement Tool. <http://dast.nlanr.net/Projects/Iperf/>
11. VLC: VideoLAN Client. <http://www.videolan.org/vlc/>
12. SER: SIP Express Router. <http://www.iptel.org/ser/>
13. X-Lite. SIP Software Phone. <http://www.xten.com>

# Security and Service Quality Analysis for Cluster-Based Wireless Sensor Networks

Emrah Tomur and Y. Murat Erten

Department of Information Systems, Middle East Technical University, Ankara, Turkey

etomur@bddk.org.tr

Department of Computer Engineering, TOBB University of Economics & Technology,  
Ankara, Turkey

erten@etu.edu.tr

**Abstract.** In this study, we analyze security and quality of service (QoS) issues in cluster-based wireless sensor networks (WSN). Taking spatial resolution as the main QoS metric and limiting the security definition to data integrity and authentication, we present a control strategy to maintain desired QoS and security levels during the entire operation of a cluster-based sensor network. Besides, our proposed strategy considers some other WSN QoS attributes such as coverage, packet collision and system lifetime. It provides sufficient coverage by statistical means and minimizes packet loss due to collisions by employment of a slotted MAC (media access control) scheme. It also tries to maximize the operational lifetime of the sensor network by a power conserving scheme which make all sensors participate equally. In this study, we also determine the best tradeoff between security and spatial resolution.

**Keywords:** Sensor networks, security, QoS, spatial resolution, coverage.

## 1 Introduction

There has been a considerable amount of research on WSN [1][2][3] and majority of these research studies focus on conventional data communications where main concern is energy-efficiency. Nonetheless, there has not been much research regarding the quality of service issues in wireless sensor networks.

One of the recent works that introduce QoS concept for sensor networks is [4] where authors equate service quality to *spatial resolution* referring to the number of sensors which are active in sending data toward the information sinks so that necessary information required for system functionality can be extracted from collected raw data. There are several other sensor network QoS definitions existing in the literature. As surveyed in [5] and [6], these WSN QoS definitions include both network QoS attributes such as latency, jitter, throughput and packet loss, and application level QoS attributes such as spatial resolution, coverage, exposure and system lifetime. The QoS perspective that we will use throughout this study covers four of the sensor network service quality attributes mentioned above, namely, spatial resolution, coverage, packet collision and network lifetime. In fact, we will build a QoS control strategy

which mainly concentrates on the spatial resolution attribute as defined in [4]. Yet, at the end, we will show that the proposed strategy also takes care of other three QoS attributes.

For envisioned sensor network applications of near future, another requirement, which is also as important as QoS, is an effective security mechanism. Since sensor networks may be in interaction with sensitive data or operate in hostile unattended environments like battlefields, protection of sensor data from adversaries is an inevitable requirement. There are several studies which propose security solutions tailored for sensor networks such as [7] and [8].

In the previous two paragraphs, we mentioned research studies which consider QoS for sensor networks ([4], [5], [6]) and security for sensor networks ([7], [8]). To the best of our knowledge, there are hardly any studies in the literature considering both QoS and security for sensor networks together. Only a few articles on WSN such as [9], [10] and [11] deal with QoS and security at the same time but all from a constrained viewpoint which only analyze the effect of applied security mechanisms on the performance of the sensor networks. In fact, again to the best of our knowledge, there is only a single work [12] which tries to simultaneously control security and QoS levels of a sensor network.

In this paper, we present a control strategy inspired from *ACK strategy* of [13] for satisfying the time-varying spatial resolution and security requirements of a wireless sensor network during its entire operation. We take energy constraints of sensor networks into account and propose an energy-efficient method to maximize network lifetime. Our proposed strategy also takes care of coverage and packet collisions, which are important QoS parameters for WSN. In addition, we formulate an optimization problem to determine optimal tradeoffs between security and spatial resolution, which is our main QoS metric.

The remainder of this paper is organized as follows: Section 2 presents the scope of our work describing the assumptions, system model and problem formulation. Section 3 is where we present our proposed control strategy and also the determination of best security-spatial resolution tradeoffs. After presenting our simulation results in Section 4, we finally conclude in Section 5 by summarizing intended future extensions to this study.

## 2 Scope of Our Work

### 2.1 Communication Model and Topology Assumptions

We assume a clustered sensor network topology similar to the one used in the LEACH architecture [14]. In this topology, overall network is divided into non-overlapping clusters. In each cluster, there is a cluster head located in the communication range of all sensors in this cluster. All sensors can send their data directly (in one hop) to their corresponding cluster head. In this paper, we consider only one single cluster of such a network and try to control the security and spatial resolution levels for just this single cluster.

In order to take the advantage of the existence of a central entity (cluster head) to reduce packet collisions, we prefer to use a centralized MAC scheme rather than

purely contention-based distributed schemes such as ALOHA or CSMA. Since fixed-assignment based MAC strategies like pure TDMA may cause channel inefficiency due to empty slots assigned to non-transmitting sensors, our assumed MAC scheme is a reservation-based one. There are several reservation-based MAC schemes proposed for sensor networks such as DR-TDMA [15] and TRACE [16]. In this paper, we use a version of TRACE that we have modified to suit our specific needs.

We assume that total channel capacity of the cluster under question is limited and this limit is known in advance as bits per second. Sensors send their data in their assigned slot of the MAC frame. Each sensor is assigned one and only one data slot for each frame and in each data slot, a sensor transmits only one single packet. We assume TinyOS type packets composed of a data part and an overhead part. Data part has constant length. Overhead part has variable length due to the security overhead which increases as security level increases. This varying length security overhead causes the overall packet to be of varying length. Therefore, the length of the data slot assigned for a sensor's packet should also have non-constant length to accommodate packets of different security levels. However, total frame time and total data transmission period in each frame has constant duration in accordance with the upper bound of the channel capacity. This means that number of data slots that can be accommodated in a single frame is upper bounded. This upper bound is equal to the duration of data transmission period in a frame divided by duration of a single data slot. Therefore, number of active sensors sending data to the cluster head in one frame duration has also the same upper bound. Because the duration of a data slot varies with security level, this mentioned limit in number of active sensors is correlated with security level. The correlation between spatial resolution and security resulting from the capacity limits of underlying communication channel is an important point taken into consideration in our control strategy.

## 2.2 QoS Assumptions

We have previously stated that one of the main attributes of our QoS perception is spatial resolution. Though it is true that spatial resolution taken as number of active sensors is a measure of service quality, it does not by itself represent the overall sensor network QoS as assumed in [4] since geographic locations of individual sensors really matter. In fact, a high level of spatial resolution does not guarantee a full coverage of the network, especially if active sensors are accumulated in a particular region of the cluster under consideration.

Therefore, we consider spatial resolution and coverage together and also include other service quality attributes such as packet collision rate and network lifetime. Our focus is on spatial resolution in the sense that we propose a control strategy to maintain spatial resolution and security levels in a cluster-based sensor network. Yet, we design our control strategy to take also care of other three QoS attributes.

Regarding spatial resolution, we assume that there are several spatial resolution levels to meet different requirements. In fact, spatial resolution  $N$  of the sensor network cluster can take any positive integer values between  $N_{min}$  and  $N_{max}$  which represent minimum and maximum defined spatial resolution levels respectively.

### 2.3 Security Assumptions

In our study, we consider only the security of sensor to cluster head communication and assume communications from cluster head to sensors or to the sink are secured by other means. Our security definition includes only integrity and authentication of data packets sent by sensor nodes and does not include confidentiality. We assume that message authentication/integrity codes are used for this purpose.

Another assumption on security is that there are multiple security levels defined for our sensor network. Each security level is associated with a different length message integrity code (MIC). We represent security level with  $S$  and  $S=0$  corresponds to lowest security level where no MIC is used and  $S=S_{max}$  corresponds to highest security level where longest MIC is used.  $S$  can take any positive integer values between 0 and  $S_{max}$ .

### 2.4 Problem Description

In this study, we consider a wireless sensor network application with simultaneous security and spatial resolution requirements. Under the assumptions and constraints given in sections 2.1-2.3, the problem to which a solution is presented in the next section is the following: To control the sensor network in such a way that time-varying security and spatial resolutions requirements are fulfilled during the entire operation and at the same time the operational lifetime of the network is maximized. So, we have three main objectives: (1) to keep enough number of sensor nodes active (ON) to attain desired spatial resolution, (2) to have these active sensors communicate at the required security and (3) to maximize network lifetime by having active sensors periodically power down and inactive ones power up for balanced energy dissipation. Besides, we aim to provide sufficient coverage and limit packet loss due to collisions.

The problem which comprises of only part (1) and (3) of problem above, i.e. controlling spatial resolution and maximizing network life time has already been solved in [4] and [13]. Our main contribution is to append security (2) as an additional parameter to this problem and also allow both desired spatial resolution and security requirements change in time as needed. Moreover, we extend the QoS concepts used in [4] and [13] to include coverage and collision rate. So, in this paper, we seek to find a strategy for wireless sensor networks to control three parameters, which are security, spatial resolution and energy usage. In the end, we want this control strategy to also provide some other QoS attributes such as coverage and minimal collisions.

## 3 Proposed Spatial Resolution and Security Control Strategy

The solution of the problem described in previous section involves several challenges. The main challenge is to find a control strategy to keep the spatial resolution and security levels of the sensor network at the required values. Another challenge is to find a way to check whether the required security and spatial values can be supported under the defined channel capacity. If they are not, the supported values which are closest to required values should be found and the optimal values among the supported ones should be chosen. So, as for the third challenge, we need to compute the optimal supported (security, spatial resolution) tuple yielding best tradeoff for cases when required security and spatial resolution levels exceed channel capacity.

3.1 Formulation of the Correlation Between Security and Spatial Resolution

A TinyOS packet has a total length of 36 bytes. A security method suitable for our security assumption is the authentication only (TinySec-Auth) option of TinySec [9] WSN security protocol. Total length of a TinySec-Auth packet with 4-byte MIC appended is 37 bytes. As proposed in *security suite* feature of IEEE 802.15.4 specification [17], we take a multi-level security approach and assume that we have four security levels one of which is no security and others include 4, 8 and 16 byte MICs. Knowing that a TinyOS packet with no MIC is 36 bytes and a TinySec-Auth packet with 4-byte MIC is 37 bytes, the relationship between security level  $S$  and corresponding packet length  $P_s$  is as given in Table 1.

Table 1. Packet lengths corresponding to different security levels

| Security level $S$ | Description  | Packet length $P_s$ |
|--------------------|--------------|---------------------|
| 0                  | No security  | 36 bytes            |
| 1                  | 4 bytes MIC  | 37 bytes            |
| 2                  | 8 bytes MIC  | 41 bytes            |
| 3                  | 16 bytes MIC | 49 bytes            |

After determining the packet lengths corresponding to each security level, we should now find the slot durations required for these packet lengths and how many data slots can fit into one frame for each security level.

As we have previously stated, we use a modified version of the reservation-based dynamic TDMA protocol named TRACE [16]. The symbolic representation for the frame format of our MAC scheme is given in Figure 1 for two frames.

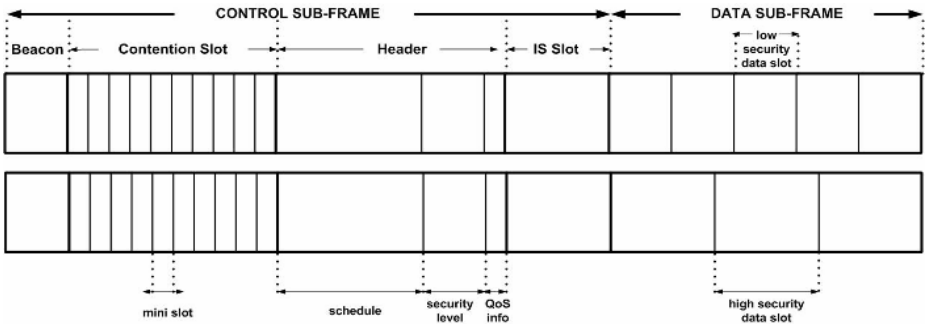


Fig. 1. The frame format of our MAC scheme (2 frames are shown)

Each frame consists of two sub-frames: a control sub-frame (reservation period) and a data sub-frame (data transmission period). Nodes that have data to send randomly choose one of the contention mini slots in control sub-frame to transmit their request. If the contention is successful, the contending sensor node is granted a data slot in the data sub-frame and can transmit its packet without any collision risk in the data transmission period. The controller, i.e., cluster head, then transmits the header,



which includes the data transmission schedule for the current frame. Unlike the original TRACE scheme, the header also includes two more pieces of information, which are the information on the current security level and information regarding the comparison of current and required spatial resolution levels.

The other difference from the original TRACE protocol is that data sub-frame is broken into variable length data slots. These data slots have variable lengths to accommodate different length packets of different security levels. Actually, the security level of all nodes during a frame duration is assumed to be the same, so all data slot lengths of the same frame are equal. However, the security levels of different frames may not be the same and therefore, the data slot lengths of different frames may vary (see Figure 1). We represent the data slot length required to accommodate a packet at security level  $S$  with  $D_s$ . This  $D_s$  value is not the same as the packet length  $P_s$  because of the overheads required at each data slot. As in the original TRACE protocol, we take 6 bytes overhead for each data slot (4 bytes for header, 2 bytes for IFS). So, our  $D_s$  values corresponding to the  $P_s$  values of Table 1 are  $D_0=42$  bytes,  $D_1=43$  bytes,  $D_2=47$  bytes and  $D_3=55$  bytes.

Now, representing the constant data sub-frame length with  $DSF$  and maximum number of data slots that can fit into a frame at security level  $S$  with  $N_{s,max}$ , the following inequality should hold not to exceed the channel capacity:  $N_{s,max} \leq DSF/D_s$ . Since we know the constant value  $DSF$  and have computed all  $D_s$  values, we are able to determine maximum spatial resolution that can be supported at security level  $S$ , which is  $N_{s,max}$ . And, the inequality  $N_{s,max} \leq DSF/D_s$  is the relationship between security and spatial resolution that we were seeking. Representing security level requirement by  $S^*$  and spatial resolution requirement by  $N^*$ , we can easily check whether a required security-spatial resolution pair  $(S^*, N^*)$  is supported by substituting these values into above inequality. If  $N^* \leq DSF/D_{S^*}$ , required levels are supported, otherwise they are not.

If we use our example network parameters together with a  $DSF$  value of 1050 bytes (coming from an assumption of 25 data slots can fit in a frame at no security, i.e.,  $25 \times 42 = 1050$ ), we can compute that  $N_{1,max} = 24$ ,  $N_{2,max} = 22$ ,  $N_{3,max} = 19$  and we already know that  $N_{0,max} = 25$ . Then, for example,  $(S^*, N^*) = (1, 23)$ ,  $(3, 15)$  and  $(2, 20)$  are supported whereas  $(1, 25)$ ,  $(3, 20)$  and  $(2, 23)$  are not. Next subsection explains how we deal with such unsupported  $(S^*, N^*)$  requirements.

### 3.2 Determination of Optimal Security and Spatial Resolution Values

We have just shown that there might be cases when required spatial resolution and security values cannot be attained. For such cases, we have to determine the supported values that are closest to the requirements. Yet, this is not a simple task since there usually exist more than one supported security-spatial resolution pair. Take the example case for a requirement of  $(S^*, N^*) = (3, 25)$  which cannot be supported. In this case, should we sacrifice security choosing supported pair of  $(0, 25)$ , or sacrifice spatial resolution and choose  $(3, 19)$ , or sacrifice from both sides and choose  $(2, 22)$ .

In fact, determination of the best tradeoff for unsupported  $(S^*, N^*)$  requirements is a resource allocation problem where the scarce resource is channel capacity and competing factors are security and spatial resolution. Such resource allocation problems are optimization problems which are studied in several works in the literature. One of

such studies is [18] whose problem modeling fits into our setting. So, we will utilize their main approach which is based on finding the values which maximize an aggregate utility function. The aggregate utility function is a weighted sum of individual utility functions which reflect the marginal benefits of each factor competing for the scarce resource.

For our case, we have two individual utility functions for security and spatial resolution represented as  $Us(S)$  and  $UN(N)$ , respectively and our overall utility function to be maximized is the weighted sum of those and equal to  $U = Ws.Us(S) + WN.UN(N)$ .

As a result, in order to determine optimal supported security and spatial resolution values when requirements cannot be satisfied, we should solve the optimization problem given in Formula 1 for  $S$  and  $N$ .

$$\begin{aligned}
 &\text{Maximize} && U = Ws.Us(S) + WN.UN(N) \\
 &\text{Subject to} && N \leq DSF/Ds, \\
 & && Nmin \leq N \leq N^*, \\
 & && Smin \leq S \leq S^*
 \end{aligned} \tag{1}$$

This optimization problem can be solved by brute force or dynamic programming.

### 3.3 Spatial Resolution and Security Control Strategy

We shall now present our proposed control strategy to keep the sensor network on required or supported security and spatial resolution levels. We assume that time is divided into discrete epochs synchronized with MAC frames. During each epoch, the following events occur.

1. Cluster head (CH) starts transmitting the beacon message.
2. CH checks whether there is a change in the required security and spatial resolution levels ( $S^*, N^*$ ) which are announced by the control center of the sensor network. If there is a change in either  $S^*$  or  $N^*$  with respect to previous epoch, CH proceeds to step 3, otherwise it goes to step 6.
3. CH checks whether new security and spatial resolution requirements are supported by using the method given in section 3.1. If they are supported it goes to step 6. If required levels ( $S^*, N^*$ ) are not supported, it computes the optimal supported levels ( $S', N'$ ) by the method of section 3.2 and then proceeds to step 6.
4. Before the beacon period ends, each and every nodes decides whether to transmit or not during the current epoch. Nodes make this decision in the same way as in ACK strategy, i.e., by comparing their locally generated random number to the transmit probability of their current state on a finite state automaton (Each state  $i$  corresponds to a different transmit probability  $T_i$  such that  $T_i > T_j$  for  $i > j$ ). Nodes which decide to transmit open their radio, synchronize with the beacon and proceed to step 5. Others shutdown their radio.
5. After the beacon period ends, nodes deciding to transmit in the previous step contend for a mini slot in the contention slot.
6. Before the transmission of Header packet, CH should have finished the calculation of optimal security and spatial resolution values ( $S', N'$ ). Also, in this step, CH determines the number of sensors that want to be active for the current epoch by counting the number of accessed contention mini slots. This number  $N_t$

- represents the current (expected) level of the spatial resolution of the current epoch  $t$  and will be compared to desired value of  $N^*$  as done in ACK strategy.
7. During the Header period, CH first unicasts the schedule of data transmissions for the current frame. This schedule is an ordered list of sensor nodes prepared according to the order in contention mini slot access and it also includes the slot duration  $D_s$  corresponding to desired security level of current epoch. Before announcing the schedule, CH should check that number of sensors desiring to transmit  $N_t$  do not exceed the maximum supported number of active sensors  $N_{s,max}$ . If  $N_t$  exceeds  $N_{s,max}$ , CH chooses only  $N_{s,max}$  of sensor nodes randomly and include only those sensors in the announced schedule.
  8. After the schedule is announced, CH informs nodes that want to transmit about two more issues during the Header period. The first one is the desired security level of current epoch ( $S^*$  or  $S'$ ) and the second one is the information on whether current spatial resolution  $N_t$  is above or below the desired level  $N^*$  (or  $N'$ ).
  9. All of the  $N_t$  nodes receiving the 1-bit information regarding the comparison of current and desired spatial resolution levels change their state. If this value is 1, they reward themselves by jumping into a higher state, otherwise, they punish.
  10. Of  $N_t$  nodes which changed state in previous step, the ones which are not listed in the announced transmission schedule shut down their radio. Only the nodes which find their name in the schedule transmit their packets at the security level announced and in the data slot assigned to them.
  11. After the data sub-frame ends, all sensors return to step 4 and CH goes to step 1.

## 4 Simulations and Analysis

We performed several simulations using our own code written in MATLAB starting with an initial deployment of 100 sensors.

In the first simulation whose results are illustrated in Figure 2, we set  $S^*=0$  and  $N^*=35$  for whole simulation duration to benchmark our proposed strategy against the ACK strategy of [13] which aim to maintain 35 active sensors throughout network operation. As seen from Figure 2, our proposed control strategy is able to keep spatial resolution level at around 35 till most of initially deployed 100 sensors die at around time epoch 130. This result is very similar to the simulation outputs presented in [13] where it is shown that ACK strategy outperforms the Gur Strategy of [4] regarding overall network life since it dies before time epoch 30. Therefore, the results of this first simulation show us that our proposed control method performs well in both maintaining spatial resolution at desired level and also in maximizing WSN lifetime.

In order to see the performance of the proposed strategy in controlling both security and spatial resolution levels simultaneously, we performed another simulation. This time we had time-varying security and spatial resolution requirements as shown in the top subplots of Figure 3 and Figure 4 respectively. Illustrated in those top subplots are also the supported levels  $S'$  and  $N'$  which are computed using the optimization problem for cases when  $S^*$  and  $N^*$  are not attainable due to limited channel capacity. In down subplots of Figure 3 and Figure 4, the actual attained security and spatial resolution values  $S$  and  $N$  that our proposed method produce are shown against the

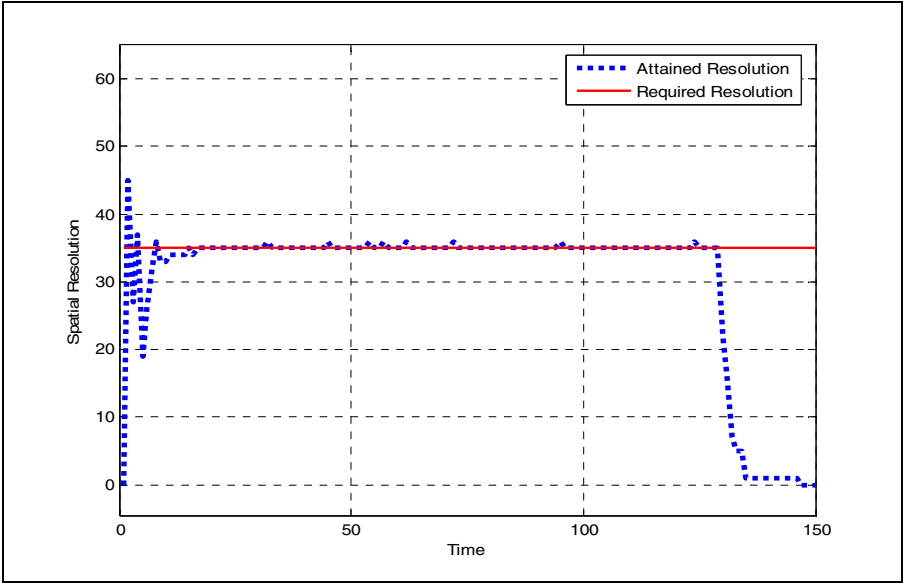


Fig. 2. Spatial resolution vs time for zero level security

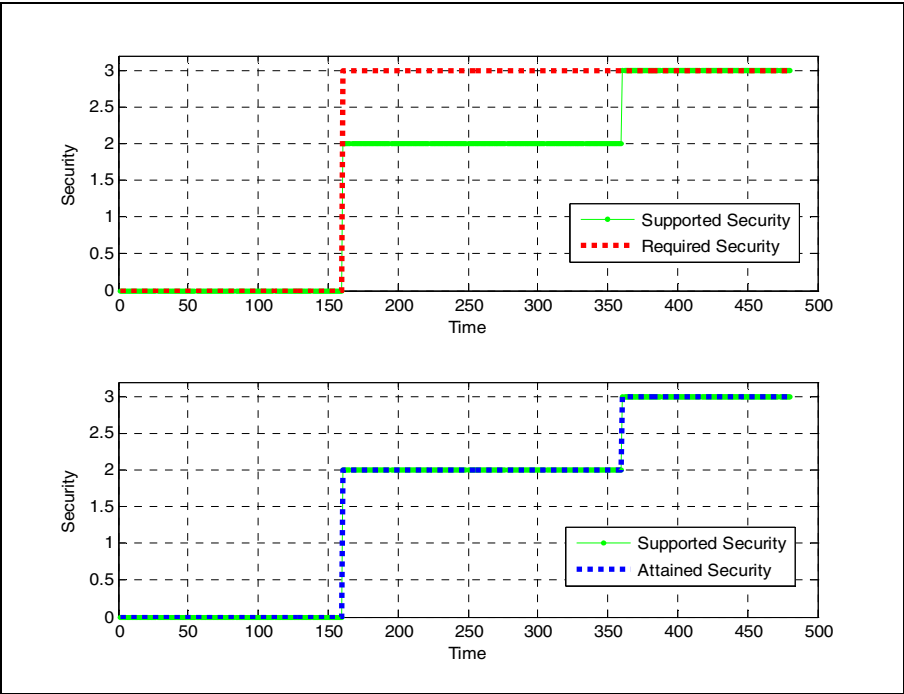


Fig. 3. Supported & required (top subplot) and supported & attained (down subplot) security levels

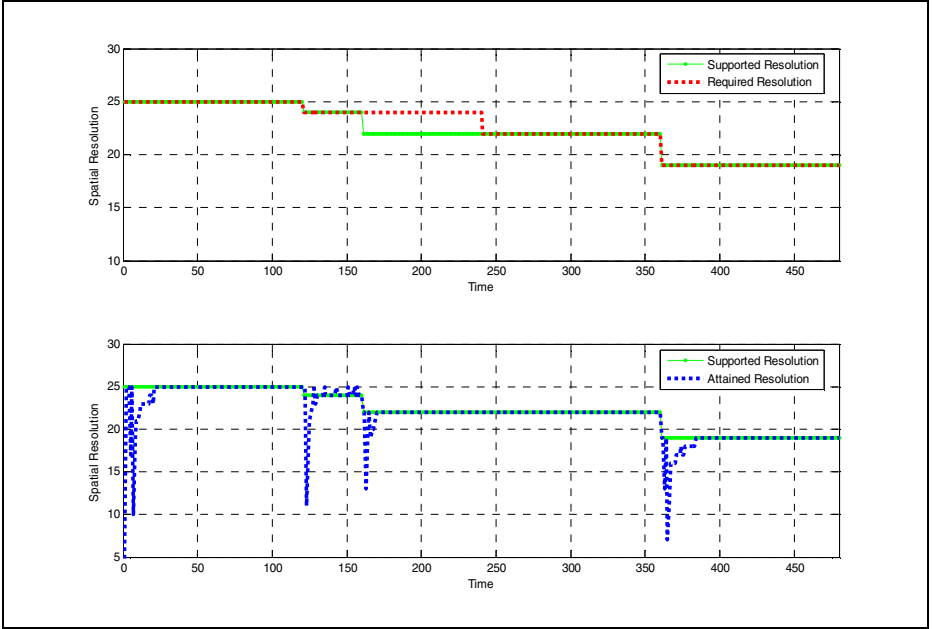


**Fig. 4.** Supported & required (top subplot) and supported & attained (down subplot) resolution level

supported values. As can be seen, the security level  $S$  that we are able to attain exactly traces the supported security value  $S'$  since our strategy forces all active sensors to transmit at the required or supported security level. Similarly, except the transient times when spatial resolution requirement  $N^*$  changes, the attained level  $N$  is able to track the supported value  $N'$ .

As the final simulation output, in Figure 5, we present the results regarding the coverage performance of our proposed strategy. In this case, we divide our sensor network cluster into four geographic sub-regions over which sensors are initially deployed in a random but uniform way. Then, we simulate this setup with the same parameters/requirements of previous case and observe the geographic distribution of active sensors contributing to spatial resolution over those four sub-regions. As illustrated in Figure 5, active sensors are quite evenly distributed and in each sub-region there is more than one active sensor almost at all times. This is an indication of good coverage in the sensor network cluster since there are active sensors taking measurements in all of the geographic regions. Though it cannot guarantee that full coverage is ensured at all times, our method provides a probabilistic assurance on the coverage provided for the sensor network under consideration.

So far in this section, we have shown the proposed strategy of this study performs well in maintaining security and spatial resolution levels (Fig.3 and 4), extending network lifetime (Fig.2) and finally providing coverage (Fig.5). Regarding the packet collision rate, we have not performed any simulations. Yet, in our MAC protocol based on TRACE [16], the probability of contention in the data slots is zero because

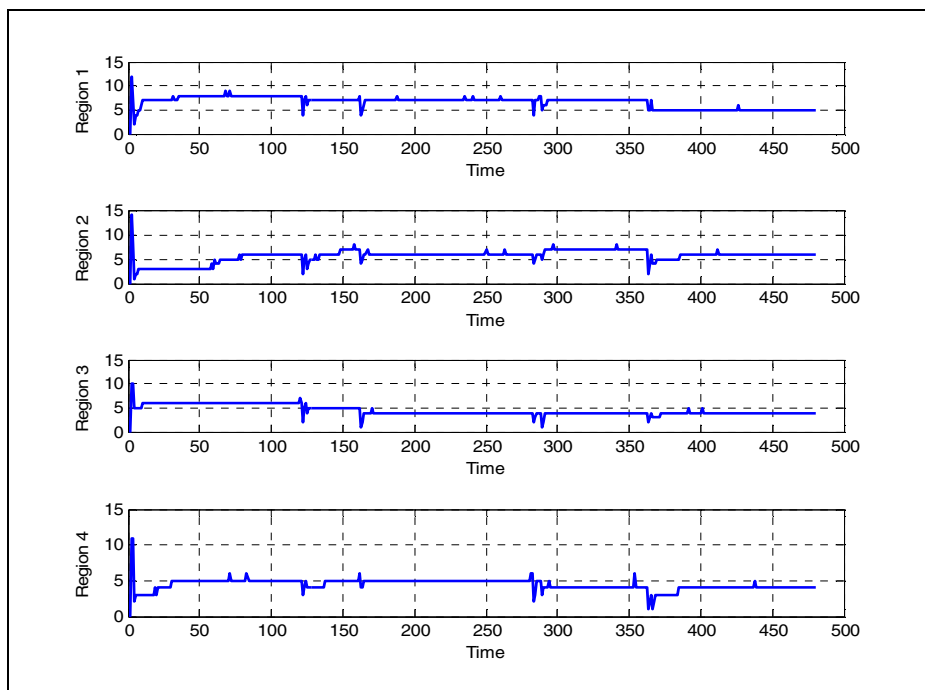


**Fig. 5.** Distribution of active sensors over sub-regions

data slots are dedicated to successfully contending nodes of the control sub-frame. Also, the number of contention slots is higher than the number of data slots and this further reduces the collisions that can occur during control sub-frame.

## 5 Conclusion and Further Work

In this paper, we have presented a control strategy to maintain required spatial resolution and security levels during the operation of a cluster-based sensor network. In addition to the spatial resolution, our proposed control strategy also takes care of some other WSN QoS attributes such as coverage, network life time and packet collision rate. In this work, we have also analyzed the correlation between security and spatial resolution. As an extension of this study, we plan to build a novel spatial resolution control strategy that will outperform the ACK method of [13].

## References

1. Akyildiz, I. F., Vuran, M. C., Akan, O. B., Su, W.: Wireless Sensor Networks: A Survey REVISITED. In: to appear in Elsevier's Computer Networks Journal (2006)
2. Shah, R., Rabaey, J.: Energy Aware Routing for Low Energy Ad Hoc Sensor Networks. In: Proceedings of IEEE Wireless Communications and Networking Conference, Orlando, FL (March 2002 )

3. Younis, M., Youssef, M., Arisha, K.: Energy-Aware Routing in Cluster-Based Sensor Networks. In: Proceedings of the 10th IEEE/ACM Sym. on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'02), Fort Worth, TX (October 2002)
4. Iyer, R. and Kleinrock, L.: QoS Control for Sensor Networks. In: Proceedings of IEEE International Communication Conference (ICC 2003), Anchorage, AK (May 2003)
5. Wang, Y., Liu, X., Yin, J.: Requirements of Quality of Service in Wireless Sensor Networks. In: Proceedings of the International Conference on Networking, Systems, Mobile Communications and Learning technologies (ICNICONSMCL06), (2006)
6. Chen, D., Varshney, P. K.: QoS Support in Wireless Sensor Networks: A Survey. In: Proceedings of International Conference on Wireless Networks (2004)
7. Slijepcevic, S., Potkonjak, M., Tsiatsis, V., Zimbeck, S., Srivastava, M. B.: On Communication Security in Wireless Ad-Hoc Sensor Networks. In: Proceedings of International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WET ICE 2002), Pittsburgh, PA (2002)
8. Zhu, S., Setia, S., Jajodia, S.: Leap: efficient security mechanisms for large-scale distributed sensor networks. In: Proceedings of the 10th ACM conference on Computer and communications security, ACM Press (2003)
9. Karlof, C., Sastry, N., Wagner, D.: TinySec: A Link Layer Security Architecture for Wireless Sensor Networks. In: Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (November 2004)
10. Guimarães, G., Souto, E., Kelner, J., Sadok, D.: Evaluation of Security Mechanisms in Wireless Sensor Networks. In: Proceedings of International Conference on Sensor Networks (August 2005)
11. Deng, J., Han, R., Mishra, S.: A Performance Evaluation of Intrusion-Tolerant Routing in Wireless Sensor Networks. In: Proceedings of 2nd IEEE International Workshop on Information Processing in Sensor Networks (2003)
12. Chigan, C., Ye, Y., Li, L.: Balancing Security Against Performance in Wireless Ad Hoc and Sensor Networks. In: Proceedings of IEEE Vehicular Technology Conference (2005)
13. Kay, J., Frolik, J.: Quality of Service Analysis and Control for Wireless Sensor Networks. In: Proceedings of 1<sup>st</sup> International Conference on Mobile Ad-Hoc and Sensor Systems, Ft. Lauderdale, FL. (Oct. 25-27, 2004)
14. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: 33rd Annual Hawaii International Conference on System Sciences (2000)
15. Frigon, J.-F., Chan, H.C.B., Leung, V.C.M.: Dynamic reservation TDMA protocol for wireless ATM networks. In: IEEE JSAC, vol. 19, pp. 370-383 (Feb. 2001)
16. Tavli, B., Heinzelman, W.: TRACE: Time Reservation Using Adaptive Control For Energy Efficiency. In: IEEE Journal on Selected Areas in Communications, Vol. 21 (2003)
17. Wireless medium access control and physical layer specifications for low-rate wireless personal area networks. IEEE Standard, 802.15.4-2003, May 2003. ISBN 0-7381-3677-5.
18. Lee, C., Lehoczy, J., Rajkumar, R., Siewiorek, D.: On Quality of Service Optimization with Discrete QoS Options. In: Proceedings of the Fifth IEEE Real-Time Technology and Applications Symposium (1999)

# Admission Control for Inter-domain Real-Time Traffic Originating from Differentiated Services Stub Domains

Stylianos Georgoulas<sup>1</sup>, George Pavlou<sup>1</sup>, Panos Trimintzios<sup>2</sup>, and Kin-Hon Ho<sup>1</sup>

<sup>1</sup> Centre for Communication Systems Research, University of Surrey  
Guildford, Surrey, GU2 7XH, United Kingdom

<sup>2</sup> ENISA, EU, PO Box 1309, 71001, Heraklion, Crete, Greece

**Abstract.** Differentiated Services (DiffServ) are seen as the technology to support Quality of Service (QoS) in IP networks in a scalable manner by allowing traffic aggregation within the engineered traffic classes. In DiffServ domains, admission control additionally needs to be employed in order to control the amount of traffic into the engineered traffic classes so as to prevent overloads that can lead to QoS violations. In this paper we present an admission control scheme for inter-domain real-time traffic originating from DiffServ stub domains; that is real-time traffic originating from end-users connected to a DiffServ stub domain towards destinations outside the geographical scope of that domain. By means of simulations we show that our scheme performs well and that it compares favorably against other schemes found in the literature.

**Keywords:** Admission Control, Real-time Traffic, Differentiated Services.

## 1 Introduction

DiffServ offers a scalable approach towards QoS in the Internet by grouping traffic with similar QoS requirements into one of the engineered traffic classes and forwarding it in an aggregate fashion. To provide QoS guarantees, DiffServ domains must additionally deploy admission control in order to control the amount of traffic injected into the traffic classes so as to prevent overloads that can lead to QoS violations.

The various admission control schemes can be classified into three categories: end-point admission control (EAC), traffic descriptor-based admission control (TDAC), and measurement-based admission control (MBAC). EAC is based on metrics applied to probing packets sent along the transmission path before the flow is established [1]. The probing packets can be sent either at the same priority as flow packets (in-band probing) or at a lower priority (out-of-band probing). One problem of EAC schemes is that simultaneous probing by many sources can lead to a situation known as thrashing [1]. That is, even though the number of admitted flows is small, the cumulative level of probing packets prevents further admissions. TDAC is based on the assumption that traffic descriptors are provided for each flow prior to its establishment. This approach achieves high utilization when the traffic descriptors used by the scheme are appropriate. Nevertheless, in practice, it suffers from several problems [2]. One is the inability to come up with appropriate traffic descriptors before establishing the flow. MBAC tries to avoid this problem by shifting the task of traffic characterization to the network [2].



That means that the network attempts to “learn” the characteristics of existing flows through real-time measurements. This approach has certain advantages. For example, a conservative specification does not result in overallocation of resources for the entire duration of the service session. Also, when traffic from different flows is multiplexed, the QoS experienced depends on their aggregate behavior, the statistics of which are easier to estimate. However, relying on measured quantities raises issues, such as estimation errors and memory related issues [2].

The various admission control schemes can also be classified according to the location where the admission control decision is made; at a centralized server or at various possible points in a network in a distributed manner. The idea of centralized schemes is simple. Signaling messages are exchanged between the sender of the flow and the centralized entity and between routers in the network and the centralized entity. These messages include the requirements of the flow and the resources state at each router, therefore admission control is performed by an entity that has complete and up-to-date knowledge of the network topology and resources, which is an ideal situation. However, in practice, centralized schemes have certain disadvantages. One is that a centralized entity constitutes a single point of failure. Another is the scalability problems that a centralized scheme raises [3]. Distributed schemes avoid these problems, but the existence of multiple admission control decision points means that concurrent admission control decisions may be made by distinct decision points for flows competing for the same resources; this can lead to QoS violations. In order for concurrency to be handled there exist some proposals in the literature [4], such as employing some safety margins to absorb the negative effects of concurrency.

Most of the schemes, to be applicable in practice, explicitly or implicitly make the assumption that the traffic is intra-domain; that is it originates and terminates within the same domain. The schemes that do not make this assumption, in many cases, e.g. see [5], require the cooperation of adjacent domains along the end-to-end paths on a per-flow basis as well as the existence of a commonly understandable signaling protocol end-to-end in order to perform admission control in each domain and propagate downstream the admission control decision and/or the QoS received so far.

Contrary to these schemes, in this paper we present a measurement-based admission control scheme for inter-domain real-time traffic originating from DiffServ stub domains, which when deployed in the context of a cascaded QoS peering model, does not require the cooperation and signaling among adjacent domains on a per-flow basis. In the rest we will first present the assumptions and conditions needed for this scheme to provide end-to-end QoS (section 2). We will then describe in detail our scheme (section 3) and we will evaluate and compare its performance against other schemes found in the literature (section 4) before concluding the paper in section 5.

## 2 Assumptions and Conditions

### 2.1 Existence of a Cascaded QoS Peering Model

The main assumption in our scheme is that a cascaded QoS peering model, similar to the one of the MESCAL project [6], is employed in the Internet. Each network provider or Autonomous System (AS) establishes provider service level agreements

(pSLAs) with the directly interconnected network providers. This type of peering agreement is used to provide QoS connectivity from a customer to reachable destinations several domains away. Fig. 1 gives an overview of the operations in this model.

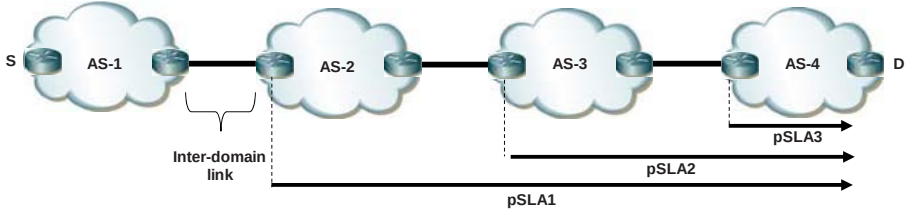


Fig. 1. A cascaded QoS peering model

AS-3 negotiates and establishes a peering agreement with AS-4 ( $pSLA_3$ ) that will allow customers of AS-3 to reach destinations in AS-4 with specific QoS guarantees as long as the total aggregate demand from AS-3 does not exceed the negotiated and agreed bandwidth value in  $pSLA_3$ . AS-2, in turn, can negotiate with AS-3 a peering agreement ( $pSLA_2$ ) in order to reach destinations in AS-4 with specific QoS guarantees. These guarantees are derived by combining the guarantees specified in  $pSLA_3$  and the local QoS capabilities of AS-3. In a similar way, AS-1, which is the DiffServ stub domain, can establish a peering agreement  $pSLA_1$  with AS-2 which defines the QoS guarantees that the traffic exiting AS-1 will receive from the ingress nodes of AS-2 till the end-customers connected to AS-4 as long as the aggregate demand from AS-1 does not exceed the negotiated and agreed in  $pSLA_1$  bandwidth value.

Since pSLAs are established for aggregate demands, each network provider typically only has to manage a limited number for pSLAs, making the cascaded model scalable. By assuming that such a cascaded QoS peering model exists, DiffServ domain AS-1 does not need to cooperate and signal any of the downstream domains on a per-flow basis for traffic destined to remote destinations. It only needs to ensure that its inter-domain traffic does not exceed the negotiated bandwidth value in the corresponding pSLAs and that the QoS received by this traffic inside AS-1, when combined with the QoS values specified in the pSLAs is adequate to meet the end-to-end QoS requirements. The domains in a QoS chain just need to ensure they enforce the local QoS and that the traffic exiting them towards the next domain in this chain, is mapped to the appropriate class of the downstream domain according to the relevant pSLAs.

In this paper we focus on how the DiffServ stub domain AS-1 will ensure that the inter-domain real-time traffic originating from end-users of AS-1 will receive the required 'local' QoS treatment so that when combined with the QoS specified in the corresponding pSLAs it will still meet the end-to-end QoS requirements.

In the rest and for the sake of simplicity we will assume that towards the destinations of interest in a remote domain, AS-1 has one pSLA in place with AS-2 which specifies a bandwidth value  $C_{pSLA}$  and the associated packet loss rate  $PLR_{pSLA}$ , delay  $D_{pSLA}$  and jitter  $J_{pSLA}$  guarantees that will be met as long as the real-time traffic demand does not exceed the negotiated and agreed bandwidth value  $C_{pSLA}$ .

## 2.2 Local QoS Versus End-to-End QoS

Given that the delay and jitter parameters are additive and that, for low values, packet loss is also additive [5], and by knowing the end-to-end requirements regarding packet loss  $PLR_{end-to-end}$ , delay  $D_{end-to-end}$  and jitter  $J_{end-to-end}$  of the real-time traffic and also the relevant values agreed in the pSLA, it is straightforward to deduce the local QoS values that need to be enforced in the DiffServ domain AS-1. If we denote as  $PLR_{local}$  the local PLR requirement, as  $D_{local}$  the local delay requirement and as  $J_{local}$  the local jitter requirement, then these are given by:

$$\begin{aligned} PLR_{local} &\leq PLR_{end-to-end} - PLR_{pSLA} \\ D_{local} &\leq D_{end-to-end} - D_{pSLA} \\ J_{local} &\leq J_{end-to-end} - J_{pSLA} \end{aligned} \tag{1}$$

## 2.3 Enforcing Local QoS for Inter-domain Real-Time Traffic

We define as real-time traffic, sources that have strict *delay* and *jitter* requirements and a bounded *packet loss rate* (PLR) requirement. Regarding low delay and jitter, both requirements are likely to be met in a high-speed network core [7]. Furthermore, certain off-line traffic engineering actions can be taken so that delay and jitter are kept within low bounds. For example, the delay requirement can be taken into account by: a) configuring appropriately small queues for the real-time traffic in order to keep the per-hop delay small, and b) controlling routing to choose paths with a constrained number of hops. Jitter can remain controlled as long as the real-time traffic flows are shaped to their nominal peak rate at the network ingress [8]. Also, the deployment of non-work conserving scheduling can be beneficial for controlling jitter [9].

Given that a certain small amount of packet loss can be acceptable [7] without significant quality degradation and that delay and jitter can be controlled by taking the above actions, in this paper we employ the PLR as the QoS metric that needs to be controlled by the admission control scheme employed in the DiffServ stub domain and we focus on keeping it in values lower than the local PLR requirement.

## 2.4 Measurement/Enforcement Points

A measurement based admission control scheme needs to ensure that it controls the flow of traffic across all possible congestion points (bottlenecks).

As stated in [10], the edge links are currently considered as the most probable congestion points of a domain, whereas backbone links are overprovisioned. Therefore we assume that packets are lost at the DiffServ domain's ingress nodes, whereas in the core of the DiffServ domain, real-time traffic aggregates from different ingress nodes are treated in a peak rate manner. This means that the core is transparent to the real-time traffic sources with respect to packet loss. By assuming that the interior of the DiffServ domain has been engineered in this way and by taking into account the routing behavior, at each ingress node we can have an estimate of the bandwidth available for the inter-domain real-time traffic aggregate from that ingress (to be more precise, from that ingress node output interfaces) to each of the corresponding egress

nodes specified in the corresponding pSLA. For inter-domain traffic, however, one also needs to take into account that peering links at the border routers between domains are also bottlenecks [11], therefore they cannot be considered overprovisioned.

Taking the above into account, the proposed scheme applies actions at these bottleneck points (output interfaces of ingress nodes and output interfaces of egress nodes) and aims to ensure that the total PLR incurred at these points is less than the local PLR requirement for the inter-domain real-time traffic. This means that for each pair of ingress-egress nodes output interfaces the following condition is met:

$$\begin{aligned} &\forall(l(i), m(e)) \text{ with } i \in I, e \in E, l(i) \in L_i, m(e) \in M_e \\ &\text{and } f(l(i), m(e)) = 1 : PLR_{l(i)} + PLR_{m(e)} \leq PLR_{local} \end{aligned} \quad (2)$$

where  $l(i)$  is the output interface  $l$  of ingress node  $i$ ,  $m(e)$  is the output interface  $m$  of egress node  $e$ ,  $I$  is the set of ingress nodes with end-customers generating real-time traffic towards the destinations in the pSLA,  $E$  is the set of egress nodes that are specified in the pSLA as exit points for inter-domain real-time traffic from the DiffServ domain towards the destinations in the pSLA,  $L_i$  is the set of output interfaces of ingress node  $i$ ,  $M_e$  is the set of output interfaces of egress node  $e$ ,  $PLR_{l(i)}$  is the incurred PLR at the output interface  $l$  of ingress node  $i$ ,  $PLR_{m(e)}$  is the incurred PLR at the output interface  $m$  of egress node  $e$  and  $f(l(i), m(e)) = 1$  indicates that the output interface  $l$  of ingress node  $i$  uses the output interface  $m$  of egress node  $e$  as the exit point towards the destinations in the pSLA.

We assume that as a result of the provisioning phase, these sets of ingress-egress pairs, as well as the output interfaces pairing and the bandwidth allocated within the domain are already known. We will also denote as  $C_{l(i) \rightarrow m(e)}$  the available bandwidth for the inter-domain real-time traffic from the output interface  $l$  of ingress node  $i$  until the output interface  $m$  of egress node  $e$ , as  $C_{m(e), pSLA}$  the available bandwidth from the output interface  $m$  of egress node  $e$  till the destinations specified in the pSLA, and finally as  $C_{e, pSLA}$  the available bandwidth for the inter-domain real-time traffic from egress node  $e$  till the destinations specified in the pSLA in place.

We assume that it holds:

$$C_{m(e), pSLA} = UF_{m(e)} \times \sum_{i \in I: f(l(i), m(e))=1} C_{l(i) \rightarrow m(e)}, \forall e \in E \text{ and } UF_{m(e)} < 1 \quad (3)$$

where  $UF_{m(e)}$  is the underprovisioning factor, which indicates the extent to which the inter-domain links are underprovisioned with respect to the aggregate bandwidth reservations at the output interfaces of the ingress nodes.  $UF_{m(e)}$  needs to be a number with value less than 1, otherwise the inter-domain links would not be bottlenecks.

We also assume that it holds:

$$\sum_{m \in M_e} C_{m(e), pSLA} = C_{e, pSLA} \quad \text{and} \quad \sum_{e \in E} C_{e, pSLA} = C_{pSLA} \quad (4)$$

In the next section we will present our scheme, which is distributed and does not require any cooperation between ingress nodes or any per ingress-egress operations or monitoring. It requires per-flow signaling only from the end-users till the ingress node of the DiffServ stub domain they are connected to but not further downstream and it tries to ensure that the local PLR requirement is met by regulating the admission of new flows but not by penalizing or terminating prematurely existing flows.

### 3 Admission Control Scheme

As stated in [12], in order for an admission control scheme to be successful in practice, it has to fulfill the following requirements.

- *Robustness*: A scheme must ensure that the requested QoS is provided. This is not trivial; for measurement-based schemes, measurement inevitably has some uncertainty, potentially leading to admission errors. The QoS should also be robust to traffic heterogeneity, long-range dependency, and to heavy offered loads.
- *Resource utilization*: The secondary goal for admission control is to maximize resource utilization, subject to the QoS constraints for the admitted flows.
- *Implementation*: The cost of deploying a scheme must be smaller than its benefits. In addition, the traffic characteristics required by the scheme should be easily obtained and the scheme should scale well with the number of flows.

#### 3.1 Admission Control Logic

Our scheme consists of two modules, one module running at each ingress node  $i$  serving inter-domain real-time traffic from that node till each one of the egress nodes and one module running at each egress node  $e$ . The modules running at the ingress nodes make admission control decisions independently from each other, aiming to regulate the admission of new flows, based on feedback from the egress nodes modules.

The egress nodes modules continuously monitor the state of the egress output interfaces (to be more precise, the status of each of the output queues configured with bandwidth limit  $C_{m(e),pSLA}$ ) and based on their status, at intervals of duration  $S$  they communicate PLR information to the ingress nodes that use these egress output interfaces as exit points for their inter-domain real-time traffic. This PLR information is used by the ingress nodes modules to calculate new PLR values to be used -if needed- for the admission of new flows. This means that each egress node only communicates with the ingress nodes that actually use them as exit points for their inter-domain real-time traffic. We need to clarify here that the communicated information relates to the PLR of the aggregate traffic using this egress output interface and not to the PLR of traffic originating from distinct ingresses, therefore the egress nodes do not need to keep any per-ingress state or perform any ingress-specific operations.

#### 3.2 The Ingress Node Module

The functionality of the ingress node module is very similar to the functionality of the module in case of intra-domain traffic described in detail in [13].

We assume that every time an inter-domain real-time flow wants to be established, it signals this to the ingress node  $i$ . Then the module, based on a target PLR level  $PLR_{l(i),target}$ , decides to accept the flow establishment if the bandwidth  $C_{l(i) \rightarrow m(e)}$  from that output interface  $l$  of ingress node  $i$  till the egress node  $e$  is enough in order to accommodate the existing flows and the new flow requesting admission, while at the same time satisfying this  $PLR_{l(i),target}$  value. Since, as stated above, each egress node does not keep any per-ingress state and only communicates one PLR information per egress output interface, all  $PLR_{l(i),target}$  values for all ingress nodes that use the same output interface of egress node  $e$  as exit point, should be the same. For the rest we will denote the PLR target at interface  $l$  of ingress node  $i$ , associated with the interface  $m$  of egress node  $e$ , as  $PLR_{l(i),target}^{m(e)}$ . This target  $PLR_{l(i),target}^{m(e)}$  level is not fixed but is adjusted based on the feedback. Also, in order for the scheme to be able to recover the total locally incurred PLR to values less than the local PLR requirement without having to penalize or terminate existing flows, this  $PLR_{l(i),target}^{m(e)}$  should be less than the local PLR requirement, that is:

$$PLR_{l(i),target}^{m(e)} \leq PLR_{local} \times OMF_{l(i)}^{m(e)} \text{ with } OMF_{l(i)}^{m(e)} \in (0,1) \quad (5)$$

where  $OMF_{l(i)}^{m(e)}$  is an Operational Margin Factor, defining the operational area within which  $PLR_{l(i),target}^{m(e)}$  can range.  $OMF_{l(i)}^{m(e)}$  should not be given a value close to one and the reason for this is that if, for example,  $PLR_{l(i),target}^{m(e)}$  is allowed to get close or become equal to  $PLR_{local}$  and an overload situation occurs at the egress node output interface with bandwidth limit  $C_{m(e),pSLA}$ , then it may not be possible to recover the total locally incurred PLR to values less than the local PLR requirement just by regulating the admission of new flows, because the overload is caused by the existing flows and it will persist until some of the existing flows are terminated. In a similar manner,  $OMF_{l(i)}^{m(e)}$  should not be set to very low values, because then the range  $[0, PLR_{local} \times OMF_{l(i)}^{m(e)}]$  within which  $PLR_{l(i),target}^{m(e)}$  can range will be very limited, which will reduce the ability of the ingress nodes modules to react and regulate the admission of new flows, regardless of the feedback information.

### 3.3 The Egress Node Module

The egress node module passively monitors the output interfaces with bandwidth limit  $C_{m(e),pSLA}$  (for the sake of simplicity, we will focus on one egress output interface and refer to it simply as output queue) and every  $S$  seconds (we will refer to  $S$  as the reporting period) it calculates the packet loss during the past interval of  $T$  seconds and depending on its value it reports back to the ingress nodes, which then adjust the target  $PLR_{l(i),target}^{m(e)}$  level accordingly.

### 3.3.1 Egress Node Module Functionality

The desired functionality for the egress node module is to be able to react not abruptly but smoothly (still in a timely fashion) and provide feedback to the ingress nodes modules to regulate the admission of new flows. In order to achieve this smooth but timely operation, the egress node module when first senses a possible congestion situation, it initially tries to correct it by applying a set of ‘mild’ actions and if this situation is not resolved then it adopts more drastic ‘emergency’ measures.

In order to achieve this progressive operation, we define two threshold PLR values, named *soft threshold* and *hard threshold* respectively, against which  $PLR_{m(e),T}$  is compared and depending on whether it crosses them (upwards or downwards) a specific set of actions is taken. The former threshold is denoted as *soft*, because it is allowed to be crossed upwards and still the status of the inter-domain link can be considered as not imminently close to becoming congested, whereas the latter is denoted as *hard*, because when it is crossed upwards, it means that the inter-domain link is imminently close to becoming congested. Since by employing the Operational Margin Factor, we have defined an upper value for the PLR allowed at the ingress nodes, both these thresholds should belong to the range  $[0, PLR_{local} - PLR_{local} \times OMF_{l(i)}^{m(e)}]$ .

### 3.3.2 Soft and Hard Threshold

The *soft threshold*  $PLR_{m(e)}^{soft}$  is a PLR value, which, as long as it is not crossed upwards by  $PLR_{m(e),T}$ , no action is taken by the ingress node modules and no communication packets are sent. The range  $[0, PLR_{m(e)}^{soft}]$  for  $PLR_{m(e),T}$ , therefore, corresponds to a ‘normal operations’ range. While in this range, the ingress nodes modules perform admission control using  $PLR_{local} \times OMF_{l(i),target}^{m(e)}$  as the  $PLR_{l(i),target}^{m(e)}$  level.

The *hard threshold*  $PLR_{m(e)}^{hard}$  is a PLR value that defines a range  $(PLR_{m(e)}^{soft}, PLR_{m(e)}^{hard}]$ , which indicates that a potential congestion situation may arise. While the measured  $PLR_{m(e),T}$  is in this range, the egress node sends back to the ingress nodes communication packets that contain as information the difference between  $PLR_{m(e),T}$  and  $PLR_{m(e)}^{soft}$ ; that is the  $PLR_{m(e),T} - PLR_{m(e)}^{soft}$  value. The ingress nodes receiving this value react to the potential congestion situation by adjusting the  $PLR_{l(i),target}^{m(e)}$  level. In order for the ingress node modules to perform more conservative admission control as  $PLR_{m(e),T}$  increases, we set the  $PLR_{l(i),target}^{m(e)}$  level to be:

$$PLR_{l(i),target}^{m(e)} = PLR_{local} \times OMF_{l(i)}^{m(e)} - (PLR_{m(e),T} - PLR_{m(e)}^{soft}) \quad (6)$$

That is, the more the measured  $PLR_{m(e),T}$  deviates from the soft threshold and approaches the hard threshold, the more conservative the admission control becomes. In practice, the ingress node modules attempt to compensate for these deviations by

decreasing by the same amount the  $PLR_{l(i),target}^{m(e)}$  value. If, however, despite the regulation of the admission of new flows,  $PLR_{m(e),T}$  continues to increase and crosses upwards the hard threshold, then the ingress node modules completely block all incoming admission requests until  $PLR_{m(e),T}$  returns to a value lower than the hard threshold. If  $PLR_{m(e),T}$  keeps decreasing and becomes lower than the soft threshold, then the  $PLR_{l(i),target}^{m(e)}$  level is set equal to  $PLR_{local} \times OMF_{l(i)}^{m(e)}$  and the egress node stops sending communication packets till the soft threshold is crossed upwards again.

This approach minimizes the control overhead, since communication packets are only sent when needed. However, if these packets cannot be guaranteed loss-free delivery, then the ingress node modules may erroneously translate the non-delivery of a communication packet as a recovery to the ‘normal operations’ range. In such cases, one alternative would be to have communication packets sent continuously every  $S$  seconds so that the ingress nodes can detect the loss of a packet.

### 3.4 On the Selection of the Parameter Values

#### 3.4.1 The Reporting Period $S$

The reporting period  $S$  defines how up-to-date with the current egress node output queue status, the ingress node modules are.

The lower the value of  $S$  the more up-to-date the information the ingress node modules use when making admission control decisions. However, the lower the value of  $S$ , the higher the control overhead. Furthermore, a very low value of  $S$  will not allow the traffic contribution of the recently admitted flows to be depicted properly in the measured  $PLR_{m(e),T}$  and, therefore, in the reported  $PLR_{m(e),T} - PLR_{m(e)}^{soft}$  value.

On the other hand, when an ingress node module performs admission control for flows arriving within two reporting periods, it is not aware of the actual effect that each of these flows will have at the egress nodes output interfaces. Therefore, within a longer  $S$  seconds period, the higher the number of arriving flows requesting admission and as a consequence the higher the possibility of making erroneous admission control decisions. In order for this phenomenon to be minimized, egress routers should explicitly perform admission control on a per-flow basis.

Moreover, since the ingress node modules do not cooperate with each other, they may make concurrent admission control decisions. This means that every ingress node is not aware of the traffic contribution from the other ingress nodes towards the same egress node output interface during an  $S$  seconds period. And the longer this  $S$  seconds period, the higher the number of arriving flows, and, therefore, the higher the possibility for each ingress node to make erroneous admission control decisions. In order for concurrency to be accounted for in our scheme, where competition between ingress nodes takes place only for resources on the inter-domain links, we employ some safety margins when setting the soft and hard threshold values.

As a result of the above discussion we conclude that the value for reporting period  $S$  should be a compromise between the above mentioned contradicting requirements.



### 3.4.2 The Measurement Window $T$

A small value of  $T$  will have as an effect the egress node modules to react abruptly to bursts. Moreover, for low values of PLR, a small value of  $T$  will mean that the measured  $PLR_{m(e),T}$  may not be representative of the real output queue congestion status. On the other hand, a high value of  $T$  will reduce the ability of the scheme to react to non-stationarities and will also introduce correlation between successive admission control decisions [2]. Therefore, the value of  $T$  should be a compromise between these contradicting requirements.

### 3.4.3 The Soft and Hard Thresholds

The soft and hard threshold values define three operation ranges, which are:

- $[0, PLR_{m(e)}^{soft}]$ , normal operation
- $(PLR_{m(e)}^{soft}, PLR_{m(e)}^{hard}]$ , potential congestion
- $(PLR_{m(e)}^{hard}, PLR_{local} - PLR_{local} \times OMF_{l(i)}^{m(e)})$ , immediate congestion

Therefore, the value  $PLR_{m(e)}^{soft}$  determines when the scheme will start reacting to increases in the measured  $PLR_{m(e),T}$ . The value  $PLR_{m(e)}^{hard}$  determines when the scheme will start taking ‘emergency actions’ to heal immediately impending congestion situations and the difference  $PLR_{m(e)}^{hard} - PLR_{m(e)}^{soft}$  determines for how long the scheme will try to recover the system by applying ‘mild’ actions.

The  $PLR_{m(e)}^{soft}$  value setting should take into account the  $PLR_{local} \times OMF_{l(i)}^{m(e)}$  value, e.g. to guarantee that eq. 6 does not become negative before  $PLR_{m(e),T}$  reaches the  $PLR_{m(e)}^{hard}$  value. Also, the  $PLR_{m(e)}^{hard}$  value, even though it could go up to  $PLR_{local} - PLR_{local} \times OMF_{l(i)}^{m(e)}$ , it should be set to lower values than that so as:

- To compensate for the effect of measurement errors.
- To compensate for concurrency-related issues.
- To allow the ingress node modules to react fast enough so that the local PLR requirement is met without having to penalize or terminate existing flows.
- To compensate for the fact that the exact effect of newly admitted flows on the status of the egress node output interfaces cannot be known beforehand. This is especially true, since the egress node modules are not aware of the traffic characteristics of individual flows.

To compensate for all the above, the practical solution we adopt is to set  $PLR_{m(e)}^{soft}$  to a relatively low value and leave a margin between  $PLR_{m(e)}^{hard}$  and  $PLR_{local} - PLR_{local} \times OMF_{l(i)}^{m(e)}$ .

## 4 Performance Evaluation

In order to evaluate the performance of our admission control scheme, we run simulations using the network simulator *ns-2* [14], with the topology of Fig. 2.

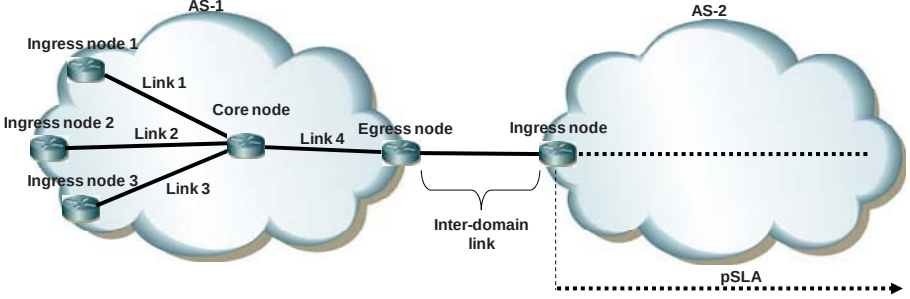


Fig. 2. Simulation topology

We use scenarios with the target local bound on PLR ( $PLR_{local}$ ) for the inter-domain real-time traffic equal to 0.001. Since the value 0.01 defines a typically acceptable upper value of PLR for the VoIP service and for real-time applications in general [15], this implicitly means that the pSLA has to provide low, but not zero loss guarantees, to keep the end-to-end PLR below 0.01. We set the Operational Margin Factor for the ingress links 1-3 equal to 0.5, which means that the upper value that the target PLR at the ingress nodes output interfaces is allowed to get is equal to the half of the target local PLR, that is 0.0005.

We set the capacities allocated at links 1-3 for the inter-domain real-time traffic ( $C_{l(i) \rightarrow m(e)}$ ) equal to 3.56Mbps. Since we assume that real-time traffic aggregates from different ingress node output interfaces are treated in the core in a peak rate manner, this means that the capacity allocated for the inter-domain real time traffic at link 4 is 10.68Mbps. We assume that the underprovisioning factor ( $UF_{m(e)}$ ) is equal to 0.8, which means that the capacity allocated at the inter-domain link is 8.544Mbps. We also configure the queues at all links for the aggregate inter-domain real-time traffic to hold a maximum of 500bytes and we set the propagation delays at all links to be 5msec. For the sake of simplicity, we do not simulate the communication traffic, we do, however, consider the propagation delays from the instant it is generated at the egress node till the moment it can be used for admission control at the ingress nodes.

Regarding the algorithm's parameters, the employed values are:  $S = 1\text{sec}$ ,  $T = 3\text{sec}$ , and we set the soft and hard thresholds equal to 40% and 60% of the  $PLR_{local} - PLR_{local} \times OMF_{l(i)}^{m(e)}$  margin, which means that the employed value for the soft and hard threshold pair is (0.0002, 0.0003), meaning that 40% of the range  $[0, PLR_{local} - PLR_{local} \times OMF_{l(i)}^{m(e)}]$  is left as safety margin.

In order to test the *robustness* of the scheme with respect to traffic *heterogeneity* and *long-range dependency*, we use a scenario with mixed VoIP and Videoconference

traffic sources, the same as in [13]. In order to test the *robustness* of the scheme with respect to *offered load*, as in [13] we test varying loading conditions ranging from 0.5 to 5, where the value 1 (*reference load*) corresponds to the average load that would be incurred by a source activation rate equal to 1000 VoIP sources/hour.

In order to compare the performance of our scheme, which we call inter-MBAC, against other schemes, we implement the EAC scheme described by Karlsson et al in [16]. Since this scheme (we call it EAC-KAR) is an out-of-band probing scheme, we implement a lower priority queue for the probing packets that can store, as in [17], a single probe packet. As in [16], we set the probing rate equal to the peak rate of the source requesting admission and we consider probe durations of 0.5sec up to 5sec. Since the path that needs to be probed includes the inter-domain link, we assume that the probing takes place between the ingress nodes 1-3 of AS-1 and the ingress node of AS-2, which after the end of the probing process signals back to the ingress nodes of AS-1 the PLR that the probing packets experienced. We do not simulate these signaling flows, we do, though, for fairness reasons consider the propagation delays.

As stated in [17], any admission control scheme must address the trade-off between *packet loss* and *utilization*. Therefore, for performance evaluation we use as metrics the locally incurred PLR and the utilization of the inter-domain link, which is the main bottleneck, together with the average blocking rate. For most loading conditions, EAC-KAR is not able to keep the total locally incurred PLR below the 0.001 local PLR target. The results shown are for 5 seconds of probe duration, which gives the lower violation of the local target PLR.

#### 4.1 Simulation Results

Inter-MBAC satisfies the target local PLR for all loading conditions. We observe an increase in the incurred PLR for higher loading conditions, which is anticipated because it relies on measurements, so every new admission request has the potential of being a wrong decision [2]. Furthermore, this is due to concurrency related issues; the higher the load, the more flows arrive within every reporting period  $S$ .

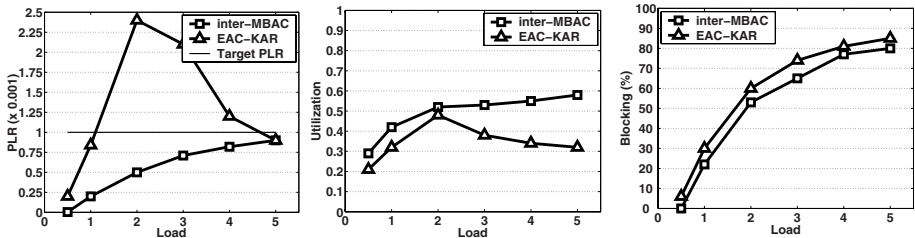


Fig. 3. Incurred PLR (left), inter-domain link utilization (centre) and blocking rate (right)

EAC-KAR violates the target local PLR for loading conditions more than one time the *reference load*. The trend of the incurred PLR for EAC-KAR indicates that it enters very early the thrashing region (for load more than two times the *reference load*) and despite the much higher (compared to inter-MBAC) incurred PLR, the achieved utilization is much lower and the incurred blocking is also higher. This behavior

seems to be a consequence of concurrency related issues which exaggerate the thrashing effect and create an oscillation effect. Flows are initially admitted, then because of the amount of probing packets, subsequent flows are rejected, the real-time traffic class is emptied, then a batch of flows is erroneously admitted (which justifies the violations of PLR), then the subsequent flows are rejected (which justifies the high blocking and the low utilization) and so on.

## 4.2 Further Discussion of the Simulation Results

The simulation results show that inter-MBAC can satisfy the target PLR for all tested loading conditions without requiring reconfiguration of its parameters for individual loading conditions. EAC-KAR fails to satisfy the local target PLR for most loading conditions despite reconfiguring its probe duration. The local target PLR is satisfied for very high load conditions but this is actually due to the thrashing effect.

Regarding the control overhead, it is not straightforward to compare the two schemes using an absolute metric since we have not implemented the communication process or the signaling control process for EAC-KAR. However, we can state that since for EAC-KAR the control overhead is dependent on the number of flows, whereas for inter-MBAC the control overhead is dependent not on the number of flows but on the number of edge nodes, the control overhead of inter-MBAC is expected to be less than that of EAC-KAR in real network situations.

Moreover, for our simulation setup, for inter-MBAC and for low loading conditions (less than load equal to the *reference load*) the simulations show that no communication packets need to be sent back to the ingress nodes because the soft threshold value is not violated at any time. Therefore, there is no control overhead associated with inter-MBAC at very low loading conditions. For higher loading conditions, the control overhead increases and for loading conditions more than one time the *reference load* it stabilizes, since its frequency is determined by the reporting period  $S$  and not by the flow arrival dynamics. For EAC-KAR, there is control overhead at all loading conditions and it increases proportionally with the load.

## 5 Conclusions

In this paper we presented a measurement based admission control for inter-domain real-time traffic originating from DiffServ stub domains.

We showed through simulations that the scheme is *robust* to traffic heterogeneity, time-scale fluctuations and heavy offered loads. The scheme can meet the QoS objectives for a variety of loading conditions without requiring any reconfiguration of its parameters and without incurring significant control overhead. Furthermore, the scheme achieves satisfactory *utilization* and compares well against existing admission control approaches for the same simulation setup.

Our scheme is also easy to *implement*. It is distributed and does not require any co-operation between ingress nodes. Per-flow operations are only performed at the ingress nodes, and egress nodes do not need to keep any per-flow state or perform any per-flow or ingress-specific operations. The scheme requires per-flow signaling only from the end-users till the ingress node of the DiffServ stub domain they are

connected to. Also since it makes the assumption that a hop-by-hop cascaded QoS peering model between adjacent domains exists, it does not require any cooperation of adjacent domains along the end-to-end paths on a per-flow basis or the existence of a commonly understandable signaling protocol end-to-end.

**Acknowledgments.** This work was undertaken in the context of the IST ENTHRONE phase 2 and IST EMANICS projects, which are partially funded by the Commission of the European Union.

## References

1. L. Breslau et al. "Endpoint Admission Control: Architectural Issues and Performance", SIGCOMM 2000.
2. M. Grossglauser et al. "A Framework for Robust Measurement-Based Admission Control", IEEE/ACM Transactions on Networking, June 1999.
3. C. Chuah et al. "Resource Provisioning using a Clearing House Architecture", IEEE IW-QoS 2000.
4. S. Lima et al. "Distributed Admission Control in Multiservice IP Networks: Concurrency issues", Journal of Communications, June 2006.
5. S. Lima et al. "Distributed Admission Control for QoS and SLS Management", Journal of Network and Systems Management, September 2004.
6. M. Howarth et al. "Provisioning for Interdomain Quality of Service: the MESCAL Approach", IEEE Communications Magazine, June 2005.
7. G. Schollmeier et al. "Providing Sustainable QoS in Next-Generation Networks", IEEE Communications Magazine, June 2004.
8. T. Bonald et al. "Statistical Performance Guarantees for Streaming Flows using Expedited Forwarding", IEEE INFOCOM 2001.
9. M. Mowbray et al. "Capacity Reservation for Multimedia Traffics", Distr. Syst. Eng., 1998.
10. V. Padmanabhan et al. "Server-based inference of Internet Link Lossiness", IEEE INFOCOM 2003.
11. T. Bressoud et al. "Optimal Configuration for BGP Route Selection", IEEE INFOCOM 2003.
12. M. Grossglauser et al. "A Time-Scale Decomposition Approach to Measurement-Based Admission Control", IEEE/ACM Transactions on Networking, August 2003.
13. S. Georgoulas et al. "Heterogeneous Real-time Traffic Admission Control in Differentiated Services Domains", IEEE GLOBECOM 2005.
14. K. Fall et al. "The ns manual" ([www.isi.edu/nsnam/ns/ns\\_doc.pdf](http://www.isi.edu/nsnam/ns/ns_doc.pdf)).
15. T. Chaded, "IP QoS Parameters", TF-NGN, November 2000.
16. V. Elek et al. "Admission Control based on End-to End Measurements", IEEE INFOCOM 2000.
17. R. Gibbens et al. "Measurement-based connection admission control", 15<sup>th</sup> International Teletraffic Congress, June 1997.

# Fault Tolerant Scalable Support for Network Portability and Traffic Engineering

Marcelo Bagnulo<sup>1</sup>, Alberto García-Martínez<sup>2</sup>, and Arturo Azcorra<sup>2</sup>

<sup>1</sup> Huawei Labs at UC3M

<sup>2</sup> U. Carlos III de Madrid

Avda de la Universidad, 30, Leganés, 28911 Madrid

{marcelo, alberto, azcorra}@it.uc3m.es

**Abstract.** The P-SHIM6 architecture provides ISP independence to IPv6 sites without compromising scalability. This architecture is based on a middle-box, the P-SHIM6, which manages the SHIM6 protocol exchange on behalf of the nodes of a site, which are configured with provider independent addresses. Incoming and outgoing packets are processed by the P-SHIM6 box, which can assign different locators to a given communication, either when it is started, or dynamically after the communication has been established. As a consequence, changes required for provider portability are minimized, and fine-grained Traffic Engineering can be enforced at the P-SHIM6 box, in addition to the fault tolerance support provided by SHIM6.

## 1 Introduction<sup>1</sup>

The SHIM6 architecture [1] provides scalable support for IPv6 end site multihoming. As opposed to the BGP-style of multihoming, where the multihomed site injects its own prefix through the different providers, in the SHIM6 approach a multihomed site obtains a Provider Aggregatable (PA) prefix from each of its providers' address blocks. This fosters prefix aggregation in the global routing table, since the multihomed site prefixes do not need to be announced independently in the global routing table and only PA prefixes corresponding to the ISPs are announced. From the multihomed site perspective, this configuration results in the presence of multiple prefixes in the site (one per provider) and multiple global addresses configured in the hosts (again, one per provider).

The goal of the SHIM6 architecture is to preserve established communications through outages in the paths to a multihomed site with multiple addresses. The SHIM6 protocol [2] is an end-to-end protocol that is used between the peers of a communication to securely create SHIM6 contexts that contain the different addresses available for the communication. The SHIM6 architecture defines a SHIM6 sublayer located between the IP endpoint sublayer and the IP forwarding sublayer. This sublayer uses the SHIM6 context state to map the addresses used by the upper layers

---

<sup>1</sup> This work has been supported by the RiNG project IST-2005-035167 and by the IMPROVISA project TSI2005-07384-C03-02.

(known as *Upper Layer Identifiers*, *ULID*) and the actual addresses used for packet forwarding (called *locators*). In case that a failure is detected in the communication path, any of the alternative addresses stored in the SHIM6 context can be used as a new locator, while ULIDs are presented unchanged to the upper layers.

However, the SHIM6 protocol fails to provide some key features of the current BGP-based approach to multihoming. In particular, SHIM6 fails to provide the portability of the address block that is used by the multihomed site. This basically means that when a multihomed end-site changes one of its providers, the addresses, that were associated with this ISP, need to be changed in a process known as *renumbering*. Renumbering may be a costly and painful process, so imposing it when changing providers does increase provider lock-in. Moreover, another capability missing in the SHIM6 architecture is traffic engineering policy enforcement. In the BGP-based multihoming framework site administrators can deeply influence the links through which ingress and egress traffic is exchanged. In this way, objectives such as balancing the traffic proportionally to the capacities of the links with the neighbouring sites, or diverting the desired amount of traffic through the cheapest provider, can be fulfilled. While SHIM6 supports some forms of traffic engineering at the end nodes, because of its end-to-end nature it is hard to enforce traffic engineering policies at a site level. To end with missing features, it may be worthy to be able to off-load the SHIM6 context management from the end nodes to specialised middle boxes to ease the deployment in domains in which end-hosts cannot be upgraded to the SHIM6 protocol, and to distribute the performance penalty imposed by SHIM6 operation when required.

In this paper we present an architecture based on the functionality provided by a SHIM6 proxy (P-SHIM6) to achieve the following capabilities:

- Provide Upper Layer Identifier portability, in order to ease renumbering
- Provide Traffic Engineering policy enforcement
- Enable legacy IPv6 nodes located in the multihomed site to obtain full SHIM6 multihoming support, without the modification of the end nodes
- Off-load of the SHIM6 context management from the actual peers of the communication

The rest of the paper is structured as follows: First we introduce the SHIM6 protocol. Then we describe the P-SHIM6 architecture, first as an overview, and then detailing the configuration and data exchange phases. After this, support for multiple P-SHIM6 boxes in order to increase fault tolerance is discussed. Finally, we analyse related work, and draw the conclusions.

## 2 SHIM6 Overview

To provide fault tolerance to established communications, the SHIM6 architecture enables diverting a packet of a communication to an alternative address of the host, which may be delegated by an alternative ISP. Since current transport layers identify

the endpoints of a communication through the IP addresses of the nodes involved, translation among ULIDs and locators must be performed in a transparent fashion with respect to transport and application layers. The SHIM6 architecture relies on the SHIM6 protocol to allow both ends to exchange their alternative locators, and on a security framework based on addresses with cryptographic properties to ensure that only legitimate locators can be exchanged. Additionally, the *REAchability Protocol* (REAP) is used to detect communication failures and to explore new paths when required. The next paragraphs detail these components.

The security architecture proposed for the multihoming protocol is based in the use of cryptographic addresses such as CGA (Cryptographically Generated Addresses, [4]). CGA incorporate into the 64-bit interface identifier ( $\Pi_{CGA}$ ) a cryptographic one-way hash of a public key ( $K_{public\_key}$ ), a prefix owned by the node ( $P_{CGA}$ ), and a Modifier, creating a binding between the public key and the resulting address. The Modifier is defined to enhance privacy by adding randomness to the resulting address.

$$\Pi_{CGA} = \text{hash}_{64}(K_{public\_key} \parallel P_{CGA} \parallel \text{Modifier}).$$

The CGA is built by appending the resulting CGA interface identifier to the CGA network prefix:  $P_{CGA}::\Pi_{CGA}$ . The private key corresponding to the  $K_{public\_key}$  can sign the alternative locators that are conveyed in the SHIM6 protocol exchange described later. The trust chain is as follows: the ULID used for the communication, that is a CGA, is securely bound to the key pair, because it contains the hash of the public key, and any alternative locator is bound to the public key through the signature.

The SHIM6 protocol [2] defines a 4-way handshake to create and manage the SHIM6 context associated with the communication between two end-points, so that data packets can be exchanged using different locators while preserving the established communication. After this handshake, the validity of the CGAs of both end-points are checked, along with the validity of the signature of the locators.

As it has been commented above, the SHIM6 layer performs the translation between the ULIDs and the locators used for a given communication. While a locator change is not required, the address included in the data packet assumes both identifier and locator roles, as it occurs in normal IP operation. However, if the locators are changed for an established communication, because of an outage, or the result of the application of a TE policy, the initial ULIDs have to be preserved when interfacing to upper layers. In this case, additional information is carried in the packets as a *context tag*, a number that is unique for each communication at the receiver. The context tag is conveyed into a SHIM6 Payload Extension Header in the packets for which the locators differ from the identifiers.

SHIM6 provide the means for recovering SHIM6 contexts that have been lost in one of the communication peers. This is achieved by repeating part of the initial 4-way handshake, using the context tag of a packet received in the end point that lost the context as a hint for the peer that still maintains the context. This may be needed, for example, in a heavily loaded server that uses aggressive strategies for releasing context state.

Additional protocols, as defined in [3], are used to detect failures affecting the currently used path, and to explore alternative paths and select among them the most appropriate one to divert the communication to.



### 3 P-SHIM6 Operation Overview

In the P-SHIM6 architecture (see fig. 1), a multihomed site obtains, apart from a PA prefix from each of its providers, non-routable globally unique addresses, i.e. independent of the location of the network to which they are assigned, that are permanently allocated to the end site. These addresses can be obtained from a central registry, such as it is specified in the Centrally Managed Unique Local Address (CMULA) [5] specification.

Consider that the hosts within the multihomed site are IPv6 hosts without SHIM6 support. These hosts are configured only with a single address with the CMULA prefix, so the hosts in the multihomed site do not depend on the ISPs, and a change in the ISP would not imply a renumbering of the hosts of the multihomed site. The multihomed site is served by one or more Proxy-SHIM6 (*P-SHIM6*) boxes, which execute the SHIM6 protocol functions on behalf of the hosts of the multihomed site.

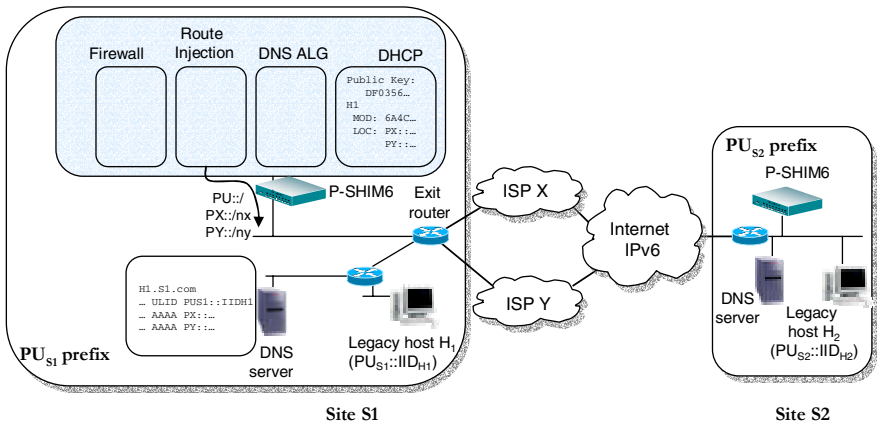


Fig. 1. P-SHIM6 architecture overview

An external communication can be established with a host located in another multihomed site or in a single-homed site. However, in order to enable the SHIM6 support for the communication, either the peer has to be SHIM6-capable or it has to be behind another P-SHIM6 that executed the SHIM6 protocol on its behalf. In the latter case, the result is that the SHIM6 protocol is executed between the P-SHIM6s that are serving each of the peers of the communication, as it is the case for figure 1.

For DNS operation when a P-SHIM6 is used, PA addresses are made public to the Internet in AAAA Resource Records (RR), while CMULAs are published in a newly defined *ULID RR*. The use of a new register prevents that an external non-SHIM6 aware node could try to use the CMULAs as a regular routable address.

When a (non-SHIM6 capable) host H1 located within the multihomed site S1 initiates a communication with a peer host H2, H1 normally performs a DNS query searching for H2. `foo.com` requesting an AAAA RR. Since the P-SHIM6 at S1 is configured as the DNS server for the hosts of the site, the query is sent to the

P-SHIM6. The P-SHIM6 behaves as a DNS ALG, and transforms the original request into a query for both AAAA and ULID records to the DNS of site S2. The reply from the DNS of S2 is processed by the DNS ALG of S1, so that only the CMULA is returned to the legacy host H1 in an AAAA RR. In addition to that, the P-SHIM6 at S1 stores the PA address information returned in the original DNS reply in the AAAA RR associated with the CMULA identifier obtained.

When H1 sends the first packet addressed to the CMULA of H2, the packet is intercepted and processed by the P-SHIM6 of the multihomed site S1. After forwarding the packet, the P-SHIM6 initiates the 4-way exchange to create a SHIM6 context with the P-SHIM6 of the peer network S2. This exchange conveys the PA addresses as locators and the CMULAs as ULIDs. Once the SHIM6 context is established between the local P-SHIM6 and the remote P-SHIM6, the local one can forward the first and subsequent data packets with a SHIM6 payload header referring to the established SHIM6 context. From now on, all packets belonging to the communication are intercepted by the P-SHIM6 and are processed so that the locators associated with the established context are included in the address fields of the packet and the negotiated context tag is included in all packets. Note this process only involves network-layer operations, as opposed to the application level rewriting that can be required by regular NAT operation, since in our case the applications of the communicating peers see the same identifiers at both sides.

The communication is now protected against failures by the SHIM6 protocol, in the sense that the reachability detection mechanisms of the REAP protocol will monitor the path availability of the communication. In case a failure is detected, alternative locator pairs are explored and the communication is diverted to an available path. Once the communication stops, heuristics are used at the P-SHIM6s to discard the associated SHIM6 state.

For intra-site communications, hosts can use CMULAs, which can be routed inside a domain in the same way as a regular address. So, in this case, the DNS should return the CMULAs of the internal hosts in AAAA records for internal queries. Again, the DNS ALG is responsible for processing the DNS reply of the actual DNS at S1, so that the CMULAs are returned in AAAA records.

The reverse tree of the DNS is used to store the locator set associated with the CMULAs in case that the communication does not start with a DNS query or there is no cached locator information available in the DNS ALG. This requires proper population of the reverse DNS tree of the CMULAs. Then, when a reverse DNS lookup is performed, the FQDN is returned and the locator information can be included in the Additional Information section of the DNS query.

Once a general overview of the mechanism has been presented, we next detail the configuration and the data exchange phases of the P-SHIM6 operation.

### 3.1 Detailed Configuration Phase

Consider the P-SHIM6 architecture depicted in figure 1, in which a multihomed site S1 is served by ISP X and ISP Y. Each of the ISPs delegates a Provider Aggregatable address block to the multihomed sites, with prefixes PX and PY respectively. Since these addresses are PA, the address block delegated by ISP X can only be reached through ISP X, and the address block delegated by ISP Y can only be reached through ISP Y. Besides,

we assume that ISPs are performing ingress filtering, meaning that packets containing source addresses belonging to the address block delegated by a given ISP X can only exit through the same ISP. In addition, a CMULA block (prefix  $PU_{S1}$ ) is assigned to the site so that CMULAs can be used as ULIDs for SHIM6 communications. So, each host within the multihomed site has conceptually three addresses: a CMULA from prefix PU and one address per PA prefix available in the site, prefixes PX and PY.

To enable SHIM6 operation, CMULAs have to be configured as cryptographic addresses, such as CGAs. Since the SHIM6 processing will be performed by the P-SHIM6, the CGA Parameter Data Structure and the associated private key must reside in the P-SHIM6 and not in the end host itself. Then, a DHCP component is required to generate CMULA CGAs on behalf of the hosts that are located behind the proxy, to store the associated parameters (CGA parameter Data structure and private key), and to assign the corresponding CMULA to each host when requested. As the hosts themselves are not involved in the SHIM6 protocol, the end hosts do not need to be aware that the address assigned is a CGA, neither they need to know the associated parameters. Note that all the different CMULA CGAs of the site can be generated using the same key pair, by only changing the Modifier field of the CGA Parameter Data Structure. This allows the P-SHIM6 to just maintain a single key pair for all its SHIM6 contexts.

In addition to the CMULA CGA, the P-SHIM6 internally assigns one address from each PA prefix available in the multihomed site to each host, although these addresses are not configured in the host itself. These addresses play the role of locators, and are permanently mapped in the P-SHIM6 to each corresponding host to allow external hosts to initiate a communication.

Regarding DNS configuration, the hosts inside S1 need to be configured to point to the P-SHIM6 as their DNS server, in order to assure that the DNS ALG is used. DHCP can be used to perform this configuration.

Finally, some configuration is required to assure that packets going from internal hosts to external ones, and vice versa are processed by the P-SHIM6. To do this, the P-SHIM6 injects an announce in the IGP (or either static routes are configured) to the root CMULA prefix, so that any packet generated from an internal host address to CMULA prefixes different from the ones assigned to the site are directed to the P-SHIM6. On the other hand, the P-SHIM6 announces internally reachability to PX and PY, so that the exit router(s) delivers to the P-SHIM6 any packet addressed to the locators assigned to the site.

Because of ingress filters, it may be necessary to route packets containing a given prefix in the source address through the ISP that has delegated this prefix. This can be achieved using tunnels from the P-SHIM6 to the exit routers, if many exist, and allowing the P-SHIM6 to route packets containing a given prefix in the source address through the corresponding ISP.

### 3.2 Data Exchange Phase

With the setup presented above, the behaviour of the P-SHIM6 architecture is the following:

1. A host H1 behind the P-SHIM6 at site S1 wants to initiate a communication with a host H2 located at site S2 with FQDN `H2.foo.com`. For that purpose, H1 performs a DNS query to its DNS server (the P-SHIM6) for `H2.foo.com`.

2. The P-SHIM6 performs a DNS query for `H2 . f00 . com`. If the query returns a ULID RR and one or more AAAA/A records, then the P-SHIM6 stores the information about the ULID and the associated locators and returns a single AAAA RR in the reply containing the CMULA ( $PU_{S2}:H2$ ). At this point, the P-SHIM6 assumes that the host H1 will start sending data packets to the destination and it initiates the 4-way handshake defined in the SHIM6 protocol to establish a SHIM6 context.
3. When the host H1 receives the DNS reply containing a CMULA  $PU_{S2}:H2$  in the AAAA record, it starts sending packets addressed to  $PU_{S2}:H2$ . Because of the longest prefix match of the address selection algorithm defined in RFC3484 [6], host H will choose the CMULA  $PU_{S1}:H1$  as source address.
4. The intra-site routing will forward packets containing an external CMULA as destination address to the P-SHIM6. When a packet containing a CMULA as a destination address arrives, the P-SHIM6 performs the following processing:
  - If a SHIM6 context exists with the addresses contained in the packet as ULID pair, then it uses the existing SHIM6 context to process the packet (the context may be already in use, or may be just created when the DNS reply was received).
  - If no SHIM6 context exists, but there is locator information associated with the CMULA contained in the destination address (cached from the DNS reply), it uses that locator information to initiate the 4-way handshake to create a SHIM6 context for that ULID pair. Once the SHIM6 context is established, it is used to process the packet.
  - If no SHIM6 context exists and there is no locator information associated with the destination CMULA cached (for example, because the application used directly IP addresses to identify the peer, instead of a FQDN), the P-SHIM6 performs a DNS reverse lookup on the CMULA contained in the destination address field, and it obtains the locator set associated with the CMULA. Once the locator information is obtained, the 4-way handshake used to establish the SHIM6 context is performed. When the SHIM6 context is established, it is used to process the packet.
5. The packets addressed to any of the locators of site S2 are forwarded to the corresponding provider of site S2, then to S2, and finally to the P-SHIM6 at S2, since it internally propagates a route to those prefixes. Then this P-SHIM6 at S2,
  - If the packet is the first packet of the SHIM6 protocol exchange, it continues with the 4-way handshake for the establishment of the SHIM6 context.
  - If the packet is a payload packet and the P-SHIM6 has an existent context associated with it, it processes the data packet and replaces the locators by the associated identifiers, and forwards the packet to the final destination.
  - If the packet is a payload packet and the P-SHIM6 does not have an associated SHIM6 context, it initiates the SHIM6 Context Recovery Procedure, sending a *R1bis* packet [2] back, to the locator carried in the packet as source address, so that context can be restored.
6. After the SHIM6 context is established, the communication continues and both P-SHIM6s perform the translation between ULIDs and locator pairs as needed. In addition the REAP protocol for failure detection and alternative path exploration is used when needed, as defined in the SHIM6 protocol.
7. When the communication is finished, the P-SHIM6s use some heuristics to discard the SHIM6 context.



structure of each CMULA address assigned to a host within the site. Note that this state is per node, not per communication, so the overhead incurred in this replication may not be very high.

Outgoing data packets must be forwarded through the primary P-SHIM6 as long as it is working. This is achieved by configuring the primary P-SHIM6 to announce a route towards the generic CMULA prefix with a high priority, and configuring the backup P-SHIM6 to announce a route to the generic CMULA prefix with a low priority. In case of a failure of the primary P-SHIM6, the associated route would disappear and the alternative routes associated with the backup P-SHIM6 would be used. Similar considerations can be applied to incoming packets, so the primary P-SHIM6 will be configured to announce routes towards the prefixes assigned by the providers that are used to allocate the locators for end-hosts with high priority, while the backup P-SHIM6 announces the same routes with lower priority.

## 4.2 Data Exchange Phase in Case of Failures

In case the primary P-SHIM6 fails, the ongoing communications that have been established by the P-SHIM6 need to be preserved. This can be done by diverting the packets towards the secondary P-SHIM6 and allowing it to recover the SHIM6 context associated with the ongoing communication.

We assume that when a P-SHIM6 fails, the associated routes are no longer announced. This implies that the routes to the secondary P-SHIM6 will become the preferred ones. So, after the primary P-SHIM6 has failed, the following packets that belong to an ongoing communication can reach the secondary P-SHIM6:

- An incoming packet including a Payload Header with a context tag, that can be a data packet or a probe packet from the REAP protocol. The secondary P-SHIM6 will receive the packet and it will find that there is no existent context for that packet. Then the secondary P-SHIM6 will activate the recovery mechanism of the SHIM6 protocol by replying with a *Ribis* packet, and the remote P-SHIM6 or SHIM6 node will provide the missing context (identifiers being used, alternative locators for the remote node, context tag to use, etc.)
- An outgoing packet coming from one of the internal hosts is received. The secondary P-SHIM6 will unsuccessfully look for an existent SHIM6 context or for cached locator information retrieved from the DNS query. Since there is no locator information associated with the destination identifier, it will perform a reverse DNS query using the CMULA included as destination in the packet, and it will obtain the locator information. At this point it will perform the 4-way handshake and the SHIM6 context will be re-established.

## 5 Related Work

IPv4 NATs are the reference middle-box architecture for IP networks. Compared to a P-SHIM6, both devices intercept packets exchanged between the site and the rest of the Internet, and process the IP header modifying the addresses using a per-communication state. Additionally, the P-SHIM6 model requires the middle-box to perform a 4-way handshake with external SHIM6-aware peers. However, P-SHIM6

provides many advantages compared to the deployment of NATs. Some derive from the fact that the identifiers are preserved in both end-points, avoiding the requirement for application inspection and processing at the middle-box, and allowing fully end-to-end operation, such as the one required by IPsec. Another advantage comes from the fact that IPv6 provides enough addresses so that stable mappings from PA addresses and CMULAs are possible, enabling externally initiated communications. Additionally, NATs are not able to preserve communications in case of failure, as P-SHIM6s do, even in case of failures in the P-SHIM6 itself. Protection against DoS attacks is provided by the use of the SHIM6 mechanism. The performance impact of deploying P-SHIM6 in a site is similar to deploying a NAT box in a site, since in both cases per-packet address rewriting and per-connection state maintenance are required, although for the P-SHIM6 case application processing is avoided.

An extended NAT architecture for IPv4 is proposed in IPnl [7]. Although this architecture may provide many benefits similar to P-SHIM6, such as network portability and fault tolerance against failures in data paths, it requires major changes not only in proxies but in hosts. More experience should be gained to determine the whole set of implications resulting from the deployment of this model.

GSE (Global, Site, End system) [8], is an IPng proposal in which a middle-box is used to rewrite addresses to gain provider independence, fault tolerance support, etc. However, this proposal raises some security vulnerabilities, such as the ones derived from the lack of tools to bind locators to an identifier.

In [9], a HIP proxy for 3G environments is described. The Host Identity Protocol (HIP) architecture [10] presents some commonalities with SHIM6, such as relying in an IP sublayer for performing a mapping between identifiers and locators. The fundamental difference between these two approaches is that a strict separation between locators and identifiers is proposed for HIP, so the identifiers are no longer valid locators, making difficult to manage application referrals and call-backs. Moreover, because of the non-hierarchical nature of the identifier name space, it is hard to deploy a directory service that stores the information about identifier to locator mapping. Apart from this, the proxy presented in [9] is specifically tailored to 3G environments, so 3G signaling is used to trigger state creation, and no hints for deployment in a full IP environment are described. In addition to this, the HIP approach imposes an extensive usage of public key cryptography, which is expensive in nature, and could be overkill for a proxy serving an IP site. Finally, proxy replication has not been considered for improving fault tolerance.

## 6 Conclusions

In this paper we have presented an architecture that relies on the configuration of provider independent addresses within a site and the deployment of SHIM6 proxies (P-SHIM6s) that intercept and process incoming and outgoing packets. In this way, non-SHIM6 aware hosts can benefit from SHIM6 when communicating with external SHIM6 hosts or hosts behind other P-SHIM6 proxies. The proxy uses the SHIM6 protocol to securely exchange the locators available for a communication, to detect communication failures, and to divert packets through an alternative path, in a transparent fashion to applications. The mechanism heavily relies on DNS to store the

mapping between CMULA and the actual locators assigned by each of the actual providers of the site, and in some cases requires proper configuration of reverse DNS. The addressing and DNS specificities of this P-SHIM6 architecture affecting to legacy hosts are managed by the P-SHIM6 by means of a DHCP and a DNS-ALG component. Therefore, the P-SHIM6 architecture allows off-loading the SHIM6 protocol operation from the hosts inside the site, easing SHIM6 deployment since legacy hosts are not required to be migrated, and SHIM6 performance costs are not charged against existing nodes.

The resulting architecture enhances the SHIM6 multihoming model in several ways:

First, it enables multihomed sites to benefit from portability of address blocks when changing providers, freeing medium and small sites from the costs of a renumbering procedure, which *de facto* results in provider lock-in. In case a provider is changed, most of the configuration to be updated resides on the P-SHIM6, along with the DNS (both direct and reverse) and the site exit routers connected to the provider.

Next, P-SHIM6 determines the egress and ingress path for the packets of a given communication as a result of the selection of the locators. Therefore, Traffic Engineering policies can be easily enforced by properly configuring the selection of the locators. Since SHIM6 can enforce different ingress and egress paths for communications with different destinations, fine-grained Traffic Engineering can be achieved. Note that if the number of communications is high, the match with a target traffic profile can be achieved with very small deviations. If required, on-going communications could be reassigned to different locators to comply with Traffic Engineering objectives.

It should be highlighted that current BGP-based solution does not scale when applied to medium to small sites that require ISP independence and site traffic engineering capabilities when medium to small sites are involved.

Regarding to fault tolerance, the SHIM6 protocol executed between the P-SHIM6s uses the REAP protocol to detect failures along the communication path and to explore alternative paths. Once a failure is detected and an alternative path is discovered, the P-SHIM6 can divert the context affected by the failure through the new path, using the corresponding locator pair. It is also possible to feed the P-SHIM6 with additional information that can be used for failure detection. In particular, the P-SHIM6 can be fed with BGP information from the different ISPs – note that the site does not inject any information into BGP. In this case, the P-SHIM6 would have access to routing information and could divert the communication through an alternative ISP in case of a failure without requiring the use of REAP (or limiting it). Considering that multihoming, and therefore SHIM6, is aimed to enhance fault tolerance capabilities, special care has been devoted to describe configurations that preserve established communications in the case that the P-SHIM6 fails.

Communication with external legacy hosts that are not served by a P-SHIM6 is achieved by making the P-SHIM6 behave as a NATv6. In this case, the P-SHIM6 would simply translate the CMULA to one of the globally routable addresses. Of course this configuration presents some of the limitations of NATs in IPv4, including that the address of the host behind the P-SHIM6 is not restored end-to-end, so if addresses are included as application layer information, they will not match with the



address actually contained in the header. However, since it is possible to perform a stateless one to many mapping between the CMULA and the global addresses, some of the limitations of NATs in IPv4, such as difficulties in allowing externally initiated communications, are lifted.

Finally, it should be noted that the architecture proposed does not require any modification neither in the hosts nor in the routers of the site.

## References

1. G. Huston, "Architectural Commentary on Site Multi-homing using a Level 3 Shim", draft-ietf-shim6-arch-00 (work in progress), July 2005.
2. M. Bagnulo, E. Nordmark, "Level 3 multihoming shim protocol", draft-ietf-shim6-protocol-07 (work in progress), November 2006.
3. J. Arkko, I. Beijnum, "Failure Detection and Locator Pair Exploration Protocol for IPv6 Multihoming", draft-ietf-shim6-failure-detection-07 (work in progress), December 2006.
4. T. Aura, "Cryptographically Generated Addresses (CGA)", RFC 3972, March 2005.
5. R. Hinden, B. Haberman, "Centrally Assigned Unique Local IPv6 Unicast Addresses", draft-ietf-ipv6-ula-central-01 (work in progress), February 2005.
6. R. Draves. "Default Address Selection for Internet Protocol version 6 (IPv6)". Feb. 2003.
7. P. Francis, R. Gummadi. "IPNL: A NAT-extended Internet architecture", Computer Communications Review 31 (4), pages 69-80, October 2001.
8. M. O'Dell, "GSE - An Alternate Addressing Architecture for IPv6", draft-ietf-ipngwg-gseaddr-00.txt, February 1997.
9. P. Salmela. "Host Identity Protocol proxy in a 3G system", Master Thesis, Helsinki University of Technology. February 2005.
10. R. Moskowitz, P. Nikander, P. Jokela, T. Henderson, "Host Identity Protocol", draft-ietf-hip-base-07.txt, Internet Draft (work in progress), February 2007.

# Class-Based OSPF Traffic Engineering Inspired on Evolutionary Computation

Pedro Sousa<sup>1</sup>, Miguel Rocha<sup>1</sup>, Miguel Rio<sup>2</sup>, and Paulo Cortez<sup>3</sup>

<sup>1</sup> Department of Informatics/CCTC, University of Minho, Portugal  
{pns,mrocha}@di.uminho.pt

<sup>2</sup> Department of Electronic and Electrical Engineering, University College London, UK  
m.rio@ee.ucl.ac.uk

<sup>3</sup> Department of Information Systems, University of Minho, 4800-058 Guimarães, Portugal  
pcortez@dsi.uminho.pt

**Abstract.** This paper proposes a novel traffic engineering framework able to automatically provide near-optimal OSPF routing configurations for QoS constrained scenarios. Within this purpose, this work defines a mathematical model able to measure the QoS compliance in a class-based networking domain. Based on such model, the NP-hard optimization problem of OSPF weight setting is faced resorting to Evolutionary Algorithms. The presented results show that, independently of other QoS aware mechanisms that might be in place, the proposed framework is able to improve the QoS level of a given domain only taking into account the direct influence of the routing component of the network. The devised optimization tool is able to optimize OSPF weight configurations in scenarios either considering a single level of link weights or using multiple levels of weights (one for each class) in multi-topology routing scenarios.

## 1 Introduction

The integration of new types of applications in TCP/IP based networks has fostered the development of several solutions to provide QoS (Quality of Service) [1] support to end-users and corresponding applications in place. In this perspective, ISPs (Internet Service Providers) have Service Level Agreements (SLAs) [2] with end-users and with other peered ISPs that should be obeyed. However, there is not an unique solution to create a QoS aware networking domain and, in general, any solution requires a number of components working together. Independently of specific QoS solutions adopted in a given network domain, there are a set of components which, by their nature, have a major influence in the QoS performance of the network. One example of such components is the routing mechanism that is used in a given domain.

The research efforts presented in this paper focus on the most commonly used intra-domain routing protocol, the Open Shortest Path First (OSPF) [3,4], trying to devise a traffic engineering framework able to provide network managers with near-optimal OSPF link weight configurations. To accomplish this goals, this work will follow the traffic engineering perspective of previous works (e.g. [5]) assuming the existence of a demand matrix associated with the network (there are several alternatives to estimate such matrices, see [6] [7]). In practice, this matrix represents an estimation of the traffic demands between each source/destination router pair of the network domain (e.g. an

ingress/egress node pair). Based on such information, the aim is to devise optimal OSPF link weight configurations which optimize a given objective function, a process which is usually viewed as a NP-hard optimization problem. As result, the main objective of the current proposal is to devise a traffic engineering framework able to automatically provide near-optimal OSPF configurations to network administrators. This main objective is supported by several innovative aspects of the proposed framework which are effective contributions when compared with previous work in the area (e.g. [5]): (i) extend previous models to tackle multiconstrained QoS optimization; (ii) develop an optimization framework in order to support multiservice networks based on the Class of Service paradigm; (iii) allow for versatile multiconstrained optimization based on an unique level of OSPF weights or resorting to scenarios using multiple levels of OSPF weights (as proposed by IETF in [8]) and (iv) achieve near-optimal OSPF configuration resorting to the field of Evolutionary Computation to improve the network performance.

In this context, the framework proposed in this paper should be viewed as a network management tool which, while focusing only at the OSPF routing level, aims at optimizing the overall QoS performance of a given domain. This does not hinder that other complementary QoS aware mechanisms might be used by network administrators, either to improve the network performance or to provide more strict QoS guarantees. However, the key point is that, based on our experiments, class-based networks using the proposed optimization framework are able to clearly outperform the QoS performance obtained by networks using common OSPF weight setting heuristics.

## 2 Problem Description

In the example of Figure 1 several network nodes are interconnected by links with distinct capacities and propagation delays representing a given ISP networking domain. Lets assume that the ISP resorts to specific techniques in order to have an estimate of the clients overall demands. In a network traffic engineering perspective such information is usually modeled and viewed as a *demand matrix* which summarizes for each source/destination router pair a given amount of resources required to be supported by the ISP. To obtain such information the ISP may resort to techniques having distinct

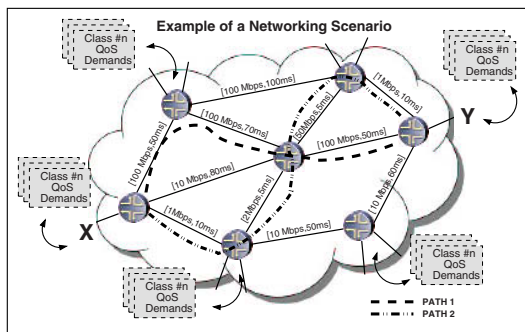


**Fig. 1.** Example of a network scenario with distinct end-to-end paths between nodes X and Y

levels of accuracy and requiring different computational efforts [6]. Based on such information, and taking into account that the routing process might have an high influence in the QoS performance of a given domain, it is possible to optimize the OSPF weight setting process in order to obtain more efficient configurations of the network. To illustrate such concepts let's assume the ISP network domain example depicted in Figure 1. Let's also assume that after studying the network behaviour, the ISP can have an estimate of the client demands and is able to map such values (demands) to matrices summarizing for each source/destination router pair (e.g. an ingress/egress router pair) a given amount of bandwidth and/or end-to-end delay required to be support by the ISP. Based on that, the optimization methods of OSPF weight setting will try to find configurations aiming at maximizing the overall QoS performance of the network.

Figure 1 shows a very simple scenario involving an individual demand between two network nodes (X and Y). Assuming that this demand is expressed as a given bandwidth requirement (e.g. 90Mbps), then the optimization methods would try to minimize the network congestion and, consequently, assign OSPF weights to force a data path inducing the lowest level of losses in the traffic (PATH 1 in the case of the scenario presented in Figure 1). In opposition, if the demand is mainly expressed in terms of a delay target<sup>1</sup>, then the ISP, in the absence of other traffic, should be able to compute OSPF weights that will result in a data path with the minimum propagation delays between X and Y (see PATH 2 in Figure 1). Moreover, in Figure 1 if a given demand has simultaneously bandwidth and delay constraints, it is expected that the OSPF weights set by the optimization algorithms are chosen in order to find a data path representing a tradeoff between the bandwidth and delay metrics (i.e. in a multiobjective perspective). Also note that if the network domain represented in Figure 1 is also viewed as a multiservice domain, e.g. supported by a class-based IP infrastructure, then each router pair of a given ISP might have also specific per-class bandwidth and delay demands. It is easy now to understand the NP-hard nature of the problem and how difficult it is to correctly set OSPF weights using simple heuristics. The proposed optimization framework assumes that the OSPF routing scheme is able to operate with one level of OSPF weights (i.e. one weight per link), which is the currently most common scenario, but is also able to provide near-optimal solutions when multiple levels of OSPF weights are used in the network. By this way, an additional feature of the proposed optimization model is the ability to assess the QoS improvements obtained when the network domain migrates to a multi-topology routing perspective, allowing for class-based QoS routing.

### 3 Mathematical Model

The mathematical model used in this work represents routers and links by a set of nodes ( $N$ ) and arcs ( $A$ ) in a directed graph  $G = (N, A)$  [9]. In this model,  $c_a$  represents the capacity of each link  $a \in A$ . A demand matrix  $D_c$  is available for each class  $c$  ( $c \in C$ ), where each element  $d_{st}^c$  represents the demand of traffic from class  $c$ , between nodes  $s$  and  $t$ . For each arc  $a$ ,  $f_{st,a}^c$  represents how much of the traffic demand from class  $c$  between  $s$  and  $t$  travels over arc  $a$ . The total load on each arc  $a$  for class  $c$

<sup>1</sup> e.g. if a large part of the traffic crossing the domain through nodes X and Y is high delay sensitive.

( $l_a^c$ ) can be defined as in Eq. (1) and the total load in arc  $a$  ( $l_a$ ) is therefore given by:  $l_a = \sum_{c \in C} l_a^c$ . The link utilization rate  $u_a$  is given by:  $u_a = \frac{l_a}{c_a}$ . It is then possible to define a congestion measure for each link ( $\Phi_a = p(u_a)$ ), using a penalty function  $p$  that has small values near 0, but as the values approach the unity it becomes more expensive and exponentially penalizes values above 1 [5]. To obtain the penalty related to each class, this value is weighted and  $\Phi_a^c$  can be obtained as in Eq. (2).

$$l_a^c = \sum_{(s,t) \in N \times N} f_{st,a}^c \quad (1) \quad \Phi_a^c = \Phi_a \frac{l_a^c}{l_a} \quad (2) \quad \gamma_c(w) = \sum_{(s,t) \in N \times N} \gamma_{st}^c(w) \quad (3)$$

In *OSPF*, all arcs have an integer weight and every node uses these weights in the Dijkstra algorithm [10] to calculate the shortest paths to all other nodes in the network. All the traffic from a given source to a destination travels along the shortest path. If there are two or more paths with the same length, traffic is evenly divided among the arcs in these paths (load balancing) [11]. Let us assume a given solution, a weight assignment ( $w$ ), and the corresponding loads and utilization rates on each arc. In this case, the total routing cost for each class is expressed by  $\Phi_c(w) = \sum_{a \in A} \Phi_a^c(w)$  for the loads and corresponding penalties calculated based on the given *OSPF* weights  $w$ . The congestion measures can be normalized ( $\Phi_c^*$ ) over distinct topology scenarios and its value is in the range [1,5000]. It is important to note that when  $\Phi_c^*$  equals 1, all loads are below  $1/3$  of the link capacity; in the case when all arcs are exactly full, the value of  $\Phi_c^*$  is  $10\frac{2}{3}$ . This value will be considered as a threshold that bounds the acceptable working region of the network. As explained, it is also useful to include delay constraints in this model. Delay requirements were modeled as a matrix  $DR_c$  (one per each class  $c$ ), that for each pair of nodes  $(s, t) \in N \times N$  gives the delay target for traffic of class  $c$  between  $s$  and  $t$  (denoted by  $DR_{st}^c$ ). In a way similar to the model presented before, a cost function was developed to evaluate the delay compliance for a solution, that takes into account the average delay of the class  $c$  traffic between the two nodes ( $Del_{st}^c$ ), a value calculated by considering all paths between  $s$  and  $t$  with minimum cost and averaging the delays in each. The delay in each path is the sum of the propagation delays in its arcs ( $Del_{st,p}^c$ ) and queuing delays in the nodes along the path ( $Del_{st,q}^c$ ). Note that in some network scenarios the latter component might be neglected (e.g. if the propagation delay component has an higher order of magnitude than queuing delays). However, if required, the  $Del_{st,q}^c$  component might be approximated, resorting to queuing theory [12], taking into account the following parameters at each node: the capacity of the corresponding output link ( $c_a$ ), the classes link utilization rates ( $l_a^c$ ) and more specific technical information such as the type of scheduling mechanisms used in the network nodes and corresponding parameter and queue size configurations.

Given this framework, the *delay compliance ratio* for a given pair  $(s, t) \in N \times N$  and class  $c$  is, therefore, defined as  $dc_{st}^c = \frac{Del_{st}^c}{DR_{st}^c}$ . A penalty for delay compliance can be calculated using function  $p$ . The  $\gamma_{st}^c$  function is defined according to  $\gamma_{st}^c = p(dc_{st}^c)$ . This allows the definition of a delay cost function, given a set of *OSPF* weights ( $w$ ), where the  $\gamma_{st}^c(w)$  values represent the delay penalties for each end-to-end path, given the routes determined by the *OSPF* weight set  $w$  (see Eq. (3)). This function can be normalized dividing the values by the sum of all minimum end-to-end delays to reach the value of  $\gamma_c^*(w)$  (for each pair of nodes the minimum end-to-end delay is calculated as

the delay of the path with minimum possible overall delay). It is now possible to define the optimization problem addressed in this work. Indeed, given a network represented by a graph  $G$ , the demand matrices  $D_c$  and the delay requirements matrix's  $DR_c$ , the aim is to find the set of *OSPF* weights  $w$  that simultaneously minimize the functions  $\Phi_c^*(w)$  and  $\gamma_c^*(w)$ , for  $c \in C$ . This is a multi-objective optimization problem and a quite simple scheme was devised to define an overall cost function, where the cost of the solution is given by Eq. (4) where  $\sum_{c \in C} (\alpha_c + \beta_c) = 1$ . This scheme, although simple, can be effective since all cost functions are normalized in the same range. The previous problem formulation considers a single *OSPF* weight set that is applied to all the traffic. An alternative is to consider that each class has its own distinct weight set (represented by  $w_c$ ), as in Eq. (5).

$$f(w) = \sum_{c \in C} (\alpha_c \Phi_c^*(w) + \beta_c \gamma_c^*(w)) \quad (4) \quad f(w) = \sum_{c \in C} (\alpha_c \Phi_c^*(w_c) + \beta_c \gamma_c^*(w_c)) \quad (5)$$

## 4 Evolutionary Algorithms and Heuristics for OSPF Setting

In order to improve the quality of the *OSPF* configurations this work resorts to *Evolutionary Algorithms (EAs)*. In the proposed *EA*, each individual encodes a solution as a vector of integer values, where each value (gene) corresponds to the weight of an arc in the network (the values range from 1 to  $w_{max}$ ). Therefore, the size of the vector equals the number of arcs in the graph (links in the network). In the case of multiple sets of weights (one per each class), the size of the solution is given by the number of classes multiplied by the number of links. The sets of weights are, in this case, still encoded in a single linear vector, that represents the concatenation of the individual sets of weights. Therefore, the *EA* is similar in both situations varying only in the decoding process. The individuals in the initial population are randomly generated, with the arc weights taken from a uniform distribution in the allowed range. In order to create new solutions, several reproduction operators were used, more specifically two mutation and two crossover operators: *Random Mutation*, replaces a given gene by a new randomly generated value, within the allowed range  $[1, w_{max}]$ ; *Incremental/decremental Mutation*, replaces a given gene by the next or by the previous value (with equal probabilities) and constrained to respect the range of allowed values; *Uniform crossover* and *Two-point crossover*, two standard crossover operators, applied in the traditional way [13]. All operators have equal probabilities in generating new solutions. The selection procedure is done by converting the fitness value into a linear ranking in the population, and then applying a roulette wheel scheme. In each generation, 50% of the individuals are kept from the previous generation, and 50% are bred by the application of the genetic operators.

In order to assess the order of magnitude of the improvements obtained by the proposed framework a number of traditional weight setting heuristic methods was also implemented [5], to provide a comparison<sup>2</sup> with the results obtained by the *EA*, namely:

<sup>2</sup> The results of the heuristics are only compared with the EAs results for scenarios with a single level of weights. For multiple levels of weights only the EAs results will be plotted to assess the improvement obtained in the network QoS.

*Unit*, sets all arc weights to 1 (one); *InvCap*, sets arc weights to a value inversely proportional to the capacity of the link; *L2*, sets arc weights to a value proportional to the physical Euclidean distance (L2 norm) of the link.

## 5 Experimental Framework and Results

The experimental platform that was used in this work is presented in Figure 2. In order to evaluate the effectiveness of the proposed *EAs*, a number of experiments was conducted. For this purpose, a set of 12 networks was generated by using the Brite topology generator [14], varying the number of nodes ( $N = 30, 50, 80, 100$ ) and the average degree of each node ( $m = 2, 3, 4$ ). This resulted in 12 networks ranging from 57 to 390 links (graph edges). The link bandwidth (capacity) was generated by an uniform distribution between 1 and 10 Gbits/s. The network was generated using the Barabasi-Albert model, using a heavy-tail distribution and an incremental grow type (parameters *HS* and *LS* were set to 1000 and 100, respectively). In the generated examples, the propagation delays were assumed as the major component of the end-to-end delay of the networks paths. Thus, the network queuing delays at each network node were not considered (i.e.  $Del_{st,q}^c = 0$ ).

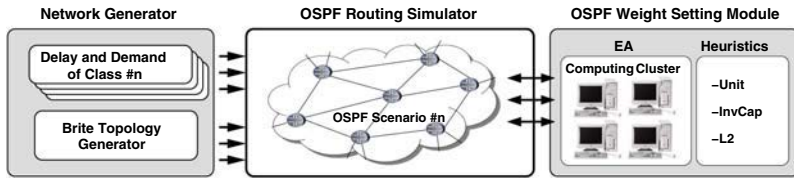


Fig. 2. Experimental platform for OSPF performance evaluation

Next, for each network, the overall demand matrices ( $D$ ) were generated. For each of the 12 instances a set of three distinct instances of  $D$  were created. A parameter ( $D_p$ ) was considered which determined the expected mean of the congestion in each link ( $u_a$ ) (values for  $D_p$  in the experiments were 0.1, 0.2 and 0.3). In the experiments a scenario with two classes was considered. Class 1 was defined as a class with an average of 75% of the overall traffic and class 2 with 25%. In each origin/destination pair, the demand from  $D$  was split between the two classes. The proportion of traffic assigned to class 1 ( $d_{st}^1$ ) was generated, for each case, from a uniform distribution within the range  $P_1(1 \pm h)$ , where  $P_1$  is the average proportion of class 1 and  $h$  is a parameter that defines traffic heterogeneity between the different origin/destination nodes ( $h$  is set to 20% in this work). In each case, the traffic demand of class 2 is the remaining from the original traffic ( $d_{st}^2 = d_{st} - d_{st}^1$ ). Using this method the matrices  $D_1$  and  $D_2$  were created for each problem instance. For the  $DR$  matrices, the strategy was to calculate the average of the minimum possible delays, over all pairs of nodes. A parameter ( $DR_p$ ) was considered, representing a multiplier applied to the previous value to get the matrices  $DR_c$ .



(values for  $DR_p$  in the experiments were 3, 4 and  $5^3$ ). This method was used to create the  $DR_2$  matrices, since the class 1 was considered not to impose delay constraints<sup>4</sup>.

In the experiments the following weights were set to each optimization aim:  $\alpha_1 = 0.5$ ,  $\beta_1 = 0$ ,  $\alpha_2 = 0.25$  and  $\beta_2 = 0.25$ . In this way, both classes have a total weight of 50%, and in the case of class 2 both aims are taken to be of equal importance. Due to the fact that both class have a similar overall contribution for the optimization aim (50%), it is expected that the optimization process using the EA will give similar importance to the objectives of each class. Overall, a set of  $12 \times 3 \times 3 = 108$  instances of the optimization problem were considered. The proposed EA, the heuristics and the OSPF routing simulator were implemented by the authors using the *Java* programming language. The EA was run for a number of generations ranging from 1000 to 6000, a value that was incremented proportionally to the number of variables optimized by the EA. The EA's population size was kept in 100 and the  $w_{max}$  was set to 20. The running times varied from a few minutes in the small networks, to a few hours in the larger ones. So, in order to perform all the tests, a computing cluster with 46 dual Xeon nodes was used. For all the optimization instances several results were collected, to allow to assess the effectiveness of the EA and of the heuristics used for comparison. Since the number of performed experiments is quite high, it was decided present aggregate results to draw conclusions. In this way, in the next sections the results obtained in all of the 108 optimization instances are averaged by  $D_p$  and  $DR_p$  (to understand the quality of the obtained solutions for distinct difficulty levels of the optimization instances) and by the number of edges considered in the experiments (to study the scalability issues of the solutions). In all figures presented in the following sections the data was plotted in a logarithmic scale, given the exponential nature of the penalty function adopted. In the figures the white area represents the acceptable working region whereas a gray area is used to identify regions with increasing levels of QoS degradation.

## 5.1 One Level of OSPF Weights

Figures 3 and 4 plot the QoS results obtained for class 1. As previously explained, this class is only constrained by the congestion performance (function  $\Phi_1^*$ ). The comparison between the methods in both figures shows an impressive superiority of the EA when compared to the heuristic methods. In fact, the EA achieves solutions which manage a very reasonable behavior in all scenarios (both for results averaged by  $D_p$  and by  $DR_p$ ), while the other heuristics manage very poorly. Regarding the results averaged by  $D_p$ , even *InvCap*, an heuristic quite used in practice, gets poor results when  $D_p$  is 0.2 or 0.3, which means that the optimization with the EAs assures good network behavior in scenarios where demands are at least 200% larger than the ones where *InvCap* would assure similar acceptable levels of congestion (i.e. results within the white area of the figures). For the results averaged by  $DR_p$  (Figure 4) the superiority of the EAs results is clearly visible for all the scenarios considered. Figure 5, on the other hand, represents

<sup>3</sup> Note that in this case lower values of  $DR_p$  represent harder optimization problems.

<sup>4</sup> In this specific experimental scenario, the class 1 was considered as only having bandwidth constraints while class 2 imposes both delay and bandwidth constraints. A practical example of this scenario might be obtained if one considers that class 1 is used to support elastic traffic (e.g. generated by TCP sources) while class 2 traffic is used to support delay sensitive traffic.



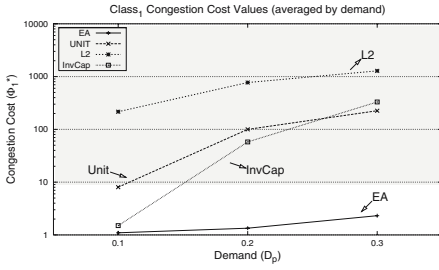


**Fig. 3.** Class 1 congestion results averaged by  $D_p$  (1 level of weights)

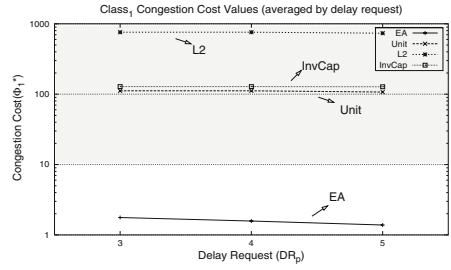


**Fig. 4.** Class 1 congestion results averaged by  $DR_p$  (1 level of weights)

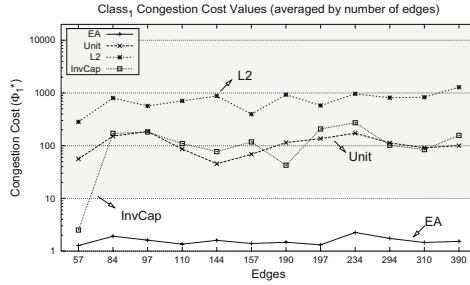


**Fig. 5.** Class 1 congestion results averaged by edges (1 level of weights)

the congestion values obtained by class 1, but aggregated by the number of arcs (links). It is clear that the results obtained by the EAs are quite scalable, since the quality levels are not affected by the number of nodes or edges in the network graph.

To analyse the performance of class 2 a distinct graphical representation is used. As explained before, class 2 was considered as a multiconstrained QoS class, both in a congestion and in a delay perspective. In that way, the graphical representation of the results plotted by Figures 6 and 7 ( $D_p$  and  $DR_p$  averaged values, respectively) have the values of the two penalty measures in each axis (x axis for congestion and y axis for delay). In these graphs, the good overall network behavior of the solutions provided by the EA is clearly visible, both in absolute terms, regarding class 2 QoS behavior in terms of congestion and delays, and when compared to all other alternative methods. In fact, it is easy to see that no single heuristic is capable of acceptable results in both aims simultaneously. L2 behaves well in the delay minimization but fails completely in congestion; InvCap class 2 congestion results are acceptable only for  $D_p = 0.1$  but fail completely in the delays. EAs, on the other hand, are capable of a good compromise between both optimization targets. As observed, EA solutions are well within the white area of the figures, which means that on average the congestion and delay demands of class 2 are satisfied by the networking domain using the solutions provided by the proposed framework using an unique level of OSPF weights.

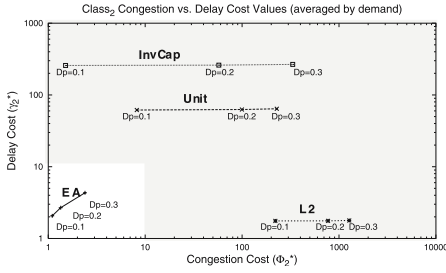


**Fig. 6.** Class 2 congestion vs delay results averaged by  $D_p$  (1 level of weights)

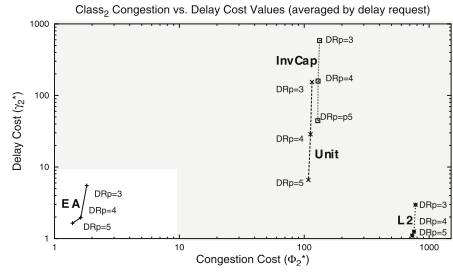


**Fig. 7.** Class 2 congestion vs delay results averaged by  $DR_p$  (1 level of weights)

## 5.2 Two Levels of OSPF Weights

The results discussed in the previous section showed that EAs are able to obtain near-optimal OSPF weight settings satisfying the QoS demands of the traffic classes. This section will now focus in the task of using several levels of OSPF weights to improve the network performance. Thus, the objective is to verify if the good results obtained by the EAs in scenarios with one level of weights can even be improved if two levels of weights are considered. For this purpose, in the optimization process of the overall cost function, two distinct levels of OSPF weights are now considered. One of the levels is used to compute the network paths for traffic belonging to class 1, while a distinct set of weights is used to route class 2 traffic. Based on this assumption, novel EA solutions were obtained for all instances of the optimization problem previously described (108 instances). The figures included in this section provide a comparison between the results obtained assuming one level of OSPF weights and the ones obtained using two levels of weights<sup>5</sup>.

The results presented in Figures 8 and 9 show the congestion cost values associated with class 1 (cost function  $\Phi_1^*$ ). In this case, the congestion cost values obtained in all optimization instances are averaged by demand levels,  $D_p$ , and delay requirements,  $DR_p$  ( see Figures 8 and 9, respectively). As observed, in both scenarios the improvements of using two levels of weights are visible for all values of  $D_p$  and  $DR_p$ . As expected, higher improvements are obtained in scenarios assuming harder QoS requirements, i.e. for  $D_p = 0.3$  and  $DR_p = 3$ . In these scenarios, the cost function  $\Phi_1^*$  achieves an improvement close to 21%. As expected, the results also show that as harder the QoS requirements get in a given network domain, the higher are the improvements expected to be obtained from the use of multiple levels of OSPF weights.

A different view of the congestion results of class 1 is presented in Figure 10. In this case, the values are averaged by the number of edges of the optimization instances. As before, the improvements obtained using two levels of weights are clear. In fact, for all values of number of edges, the  $\Phi_1^*$  function cost values decrease in scenarios assuming two levels of weights (the higher improvement is obtained for the scenario with 234 edges where  $\Phi_1^*$  is reduced in a value close to 36%). As regards to the

<sup>5</sup> In this case, due to the quality of the solutions obtained by the EAs, the figures only include the acceptable working region, i.e. the white area of the figures presented in the previous section.

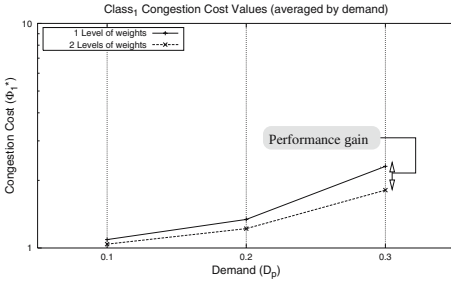


**Fig. 8.** Class 1 congestion results averaged by  $D_p$  (1 vs 2 levels of weights)

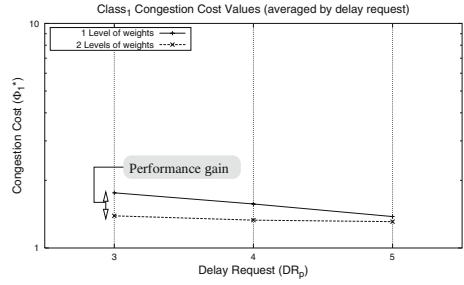


**Fig. 9.** Class 1 congestion results averaged by  $DR_p$  (1 vs 2 levels of weights)

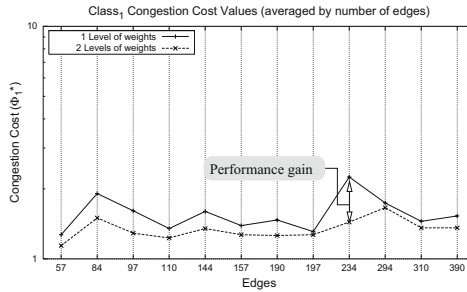


**Fig. 10.** Class 1 congestion results averaged by edges (1 vs 2 levels of weights)

performance of class 2, Figures 11 and 12 present the results of the improvements obtained in the congestion ( $\Phi_2^*$ ) and delay ( $\gamma_2^*$ ) cost functions averaged by the number of edges used in the experimental scenarios. The improvements on each function are visible, since in all scenarios there is a congestion and delay performance gain. As example, there is a maximum congestion cost improvement close 30% (scenario with 84 edges) and a delay cost improvement with reaches a value of 60% (scenario with 234 edges).

As for the case of class 1 it is important to study the improvements observed in the class 2 performance when harder optimization problems are considered. In this context, the congestion and delay cost values of class 2 are also averaged according with the  $D_p$  and  $DR_p$  values. This information is provided by Figures 13 and 14 where the performance of class 2 is analysed for distinct values of the  $D_p$  and  $DR_p$ . As observed in Figure 13, for similar values of  $D_p$  there is a congestion and delay performance gain in class 2 when two levels of weights are used. The same reasoning is valid for the case of Figure 14 showing the values of the congestion and delay cost functions averaged by the  $DR_p$  parameters. In fact, it can be observed in both graphics that as the number of weight levels increases, the plots are shifted toward to the lower left corner of the graphs, which confirms the achievement of solutions having simultaneously a better delay and congestion performance.

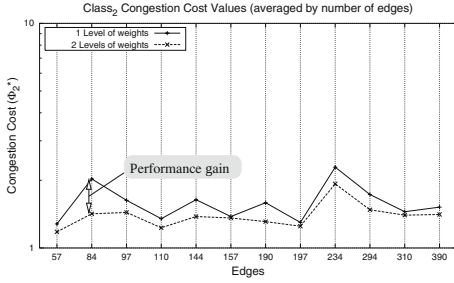


**Fig. 11.** Class 2 congestion results averaged by edges (1 vs 2 levels of weights)

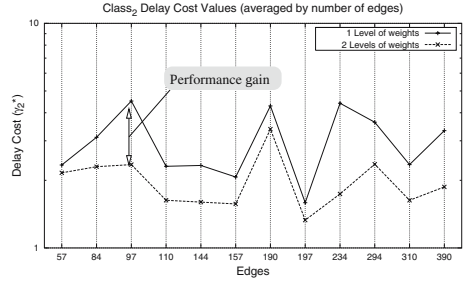


**Fig. 12.** Class 2 delay results averaged by edges (1 vs 2 levels of weights)

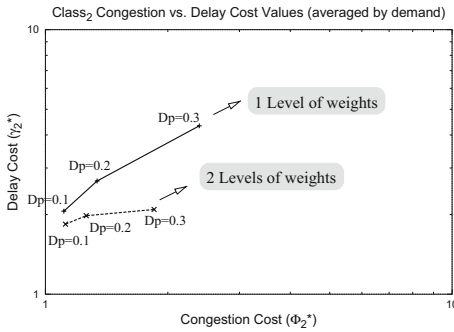


**Fig. 13.** Class 2 congestion vs delay results averaged by  $D_p$  (1 vs 2 levels of weights)

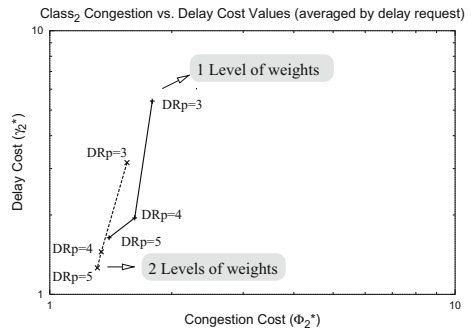


**Fig. 14.** Class 2 congestion vs delay results averaged by  $DR_p$  (1 vs 2 levels of weights)

## 6 Conclusions and Further Work

This work presented a novel traffic engineering optimization framework able to provide near-optimal OSPF weight configurations to network administrators. Resorting to a large number of QoS constrained scenarios, it was shown that high quality network configurations can be obtained using techniques inspired on the Evolutionary Computation field. The proposed framework is able to deal with single or multiple levels of OSPF weights. The results showed that, independently of other QoS aware mechanisms that might be in place, the proposed framework is able to improve the QoS level of a given class-based domain only taking into account the direct influence of the routing component of the network. In the future, the consideration of more specific EAs to handle this class of multi-objective problems [15][16] will also be taken into account. In a similar way, it is also possible to study the impact of link failures in the solutions devised by the proposed framework. Another important future research topic is the study of the sensibility of the achieved EA solutions to changes in the demand matrices. This specific research topic will allow for the use of this type of management tools in scenarios where the demands matrices are obtained in a more finer-grain temporal perspective.

## References

1. Zheng Wang. *Internet QoS: Architectures and Mechanisms for Quality of Service*. Morgan Kaufmann Publishers, 2001.
2. D. Verma. *Supporting Service Level Agreement on IP Networks*. McMillan Publishing, 1999.
3. J. Moy. RFC 2328: OSPF version 2, April 1998.
4. T.M. ThomasII. *OSPF Network Design Solutions*. Cisco Press, 1998.
5. B. Fortz and M. Thorup. Internet Traffic Engineering by Optimizing OSPF Weights. In *Proceedings of IEEE INFOCOM*, pages 519–528, 2000.
6. A. Medina et al. Traffic matrix estimation: Existing techniques and new directions. *Computer Communication Review*, 32(4):161–176, 2002.
7. Alan Davy, Dmitri Botvich, and Brendan Jennings. An efficient process for estimation of network demand for qos-aware ip networking planning. In Gerard Parr, David Malone, and Mícheál Ó Foghlú, editors, *6<sup>th</sup> IEEE International Workshop on IP Operations and Management, IPOM 2006, LNCS 4268*, pages 120–131. Springer-Verlag, 2006.
8. P. Psenak et al. Multi-topology (mt) routing in ospf (internet draft), November 2006.
9. Ravindra et al. *Network Flows*. Prentice Hall, 1993.
10. E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(269-271), 1959.
11. J. Moy. *OSPF, Anatomy of an Internet Routing Protocol*. Addison Wesley, 1998.
12. G. Bolch et al. *Queueing Networks and Markov Chains - Modeling and Performance Evaluation with Computer Science Applications*. Jhon Wiley and Sons INC., 1998.
13. Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, USA, third edition, 1996.
14. A. Medina et al. BRITE: Universal Topology Generation from a User's Perspective. Technical Report 2001-003, January 2001.
15. C.M. Fonseca and P.J. Fleming. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3(1):1–16, 1995.
16. C.A. Coello Coello. *Recent Trends in Evolutionary Multiobjective Optimization*, pages 7–32. Springer-Verlag, London, 2005.

# An Experimental Investigation of the Congestion Control Used by Skype VoIP\*

Luca De Cicco, Saverio Mascolo, and Vittorio Palmisano

Dipartimento di Elettrotecnica ed Elettronica, Politecnico di Bari,  
Via Orabona 4, Italy  
{ldecicco, mascolo, vpalmisano}@poliba.it

**Abstract.** The explosive growth of VoIP traffic poses a potential challenge to the stability of the Internet that, up to now, has been guaranteed by the TCP congestion control. In this paper, we investigate how Skype behaves in the presence of time-varying available bandwidth in order to discover if some sort of congestion control mechanism is implemented at the application layer to match the network available bandwidth and cope with congestion. We have found that Skype flows are somewhat elastic, i.e. they employ some sort of congestion control when sharing the bandwidth with unresponsive flows, but are inelastic in the presence of classic TCP responsive flows, which provokes extreme unfair use of the available bandwidth in this case. Finally, we have found that when more Skype calls are established on the same link, they are not able to adapt their sending rate to correctly match the available bandwidth, which would confirm the risk of network congestion collapse.

## 1 Introduction

Skype is by far the most used VoIP application, with an ever growing user-base which today counts more than 8 million users. This explosive growth poses challenges to telecom operators and ISPs both from the point of view of business model and network stability. Regarding network stability, the issue here is to check if the growth of unresponsive non-TCP flows, i.e. flows without end-to-end congestion control, would impact the stability of the best-effort Internet that everyone knows.

Other important issues that have been addressed in recent literature on Skype are the employed peer-to-peer protocol and the evaluation of the QoS of VoIP calls placed using Skype [1,3].

The goal of this paper is to investigate how Skype reacts to network congestion, that is, how Skype manages to adapt its sending packet rate to match the network available bandwidth when competing with TCP flows or with multiple Skype calls placed over the same link.

---

\* This work was partially supported by the MIUR-PRIN project no. 2005093971 "FAMOUS Fluid Analytical Models Of aUtonomic Systems".

The results we have obtained are twofold: i) Skype flows are somewhat elastic and they employ some sort of congestion control when coexisting with unresponsive flows; ii) Skype flows exhibit unresponsive behaviour when sharing the bandwidth with responsive TCP flows.

The paper is organized as follows: in Section 2 we present a brief analysis of the state of the art related to Skype; in Section 3 we describe the experimental testbed that has been set up in order to carry out our investigations; Section 4 describes the considered scenarios and presents the obtained results. Finally Section 5 concludes the paper.

## 2 Related Work

The efficient transport of multimedia flows is an open issue and it is currently an hot topic as long as multimedia services are rapidly increasing their importance. In this area the voice over IP applications are taking an ever increasing relevance as it is shown by the success of Skype application for end users and by the large deployment of SIP-based networks.

In spite of this explosive growth it is not clear what will be the impact of VoIP traffic on the stability of the Internet when a very large amount of VoIP flows will populate the network. The main driver of Internet stability is the congestion control algorithm developed by V. Jacobson for the TCP [7]. For this reason many researchers have conjectured a congestion collapse in case VoIP flows don't employ a responsive congestion control algorithm [4].

As a consequence, several efforts have been carried out to design multimedia congestion control protocols that are TCP friendly, where friendliness here means that the audio flows will share the network bandwidth with TCP flows fairly.

The TCP Friendly Rate Control (TFRC) protocol is currently being discussed within the IETF as a possible congestion control algorithm for multimedia flows [6]. In particular the Small Packet version of TFRC has been proposed in order to be employed for VoIP applications [5]. In spite of this effort, as a matter of fact, all commercial audio/video applications run over UDP and we conjecture that they implement some congestion control algorithm at the application level. Other proposals for multimedia traffic are RAP [9] and TEAR [10].

A recent investigation [2] concludes that VoIP traffic does not harm network stability because of the user behaviour that would drop a call if an unacceptable quality is perceived due to congestion. In other terms, the user behaviour would provide an intrinsic stability mechanism for congestion avoidance. In particular, by carrying out simulations, authors infer that, by taking into account the user back-off behaviour, VoIP flows consume much less bandwidth than TCP flows and respond to congestion when network is overloaded. Differently from [2], in this paper we will investigate if the Skype VoIP application implements its own congestion control algorithm to match the available bandwidth. As it will be shown in this paper the answer is affirmative.

### 3 Experimental Testbed

In order to investigate how Skype adapts to variations in the available bandwidth we have set up a local testbed using a measurement tool we have developed. In each host in Figure 1 we have routed all packets generated from Skype application to the ingress queues  $q_1$  and  $q_2$ . The measurement tool allows delays, available bandwidth and buffer size of each queue be set by the user. It is worth noticing that the connection implemented is strictly equivalent to a connection made by using a Dummynet-like router [11], the only difference being that in this way we can use two hosts instead of three.

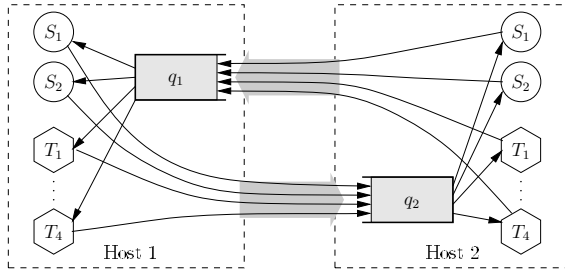


Fig. 1. Experimental testbed

On each host we have installed Skype ( $S_1$  and  $S_2$ ) and `iperf` ( $T_1, \dots, T_4$ ) [14] in order to generate TCP flows and we have collected logfiles by tracing the per-flow data arriving to and departing from the queue. By comparing data at the input of the queue and at its output, we have been able to compute packet drop rates and goodputs for Skype and TCP flows. Goodput, throughput and loss rate are defined as follows:

$$\text{goodput} = \frac{\Delta_{sent} - \Delta_{loss}}{\Delta T}; \text{throughput} = \frac{\Delta_{sent}}{\Delta T}; \text{loss rate} = \frac{\Delta_{loss}}{\Delta T}$$

where  $\Delta_{sent}$  is the number of bits sent in the period  $\Delta T$ ,  $\Delta_{loss}$  is the number of bits lost in the same period. We have considered  $\Delta T = 0.4$  s in our measurements.

Finally, it is worth noticing that Skype flows are generated using always the same audio sequence by hijacking audio I/O by using [15]. From now on, the  $RTT$  of the connection is set at 100 ms and the queue size is set equal to the bandwidth delay product unless otherwise specified.

### 4 Investigating the Skype Congestion Control

In order to understand how Skype behaves in the presence of congestion we start by considering a step-like time-varying available bandwidth. Considering a step-like input is common practice in control theory when testing the dynamic



behaviour of a system [8]. In particular we start by considering square-wave available bandwidths characterized by different periods in order to test not only the Skype capability to match the available bandwidth but also the transient time required for the matching.

Before starting to report our results, it is worth noticing that Skype employs the adaptive codecs iSAC and iLBC both developed by Global IP Sound [12,13] to provide sending rate adaptation capability.

#### 4.1 Case 1: One Skype Flow over a Square Wave Form Available Bandwidth

This scenario aims at investigating how Skype sending rate reacts to sudden changes of available bandwidth in order to infer if it employs some sort of congestion control. In order to do this, we have used a technique that is often employed in system identification, i.e. we have used an available bandwidth that varies as a square wave with maximum value  $A_M = 160$  kb/s and minimum value  $A_m = 16$  kb/s (see Figure 2).

We have considered two periods for the square wave form in order to identify how fast is the Skype response to bandwidth changes.

We have run the first experiment by setting the period of the square wave equal to 200s, which happened to be large enough to show all the transient dynamics.

Figure 2 shows that Skype decreases the sending rate when the link capacity drops from the high value  $A_M$  to the low value  $A_m$ . It is worth noticing that the Skype flow takes approximately 40s to track the available bandwidth during which it experiences a significant loss rate. From Figure 2 we also argue that the Skype reaction is triggered by the high loss rate.

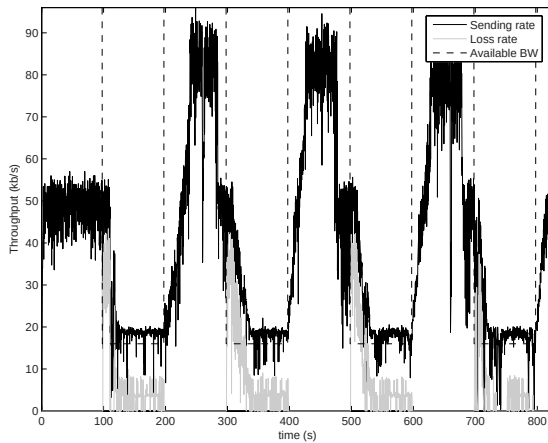


**Fig. 2.** Sending rate and Loss rate in the presence of square wave available bandwidth of 200 s period

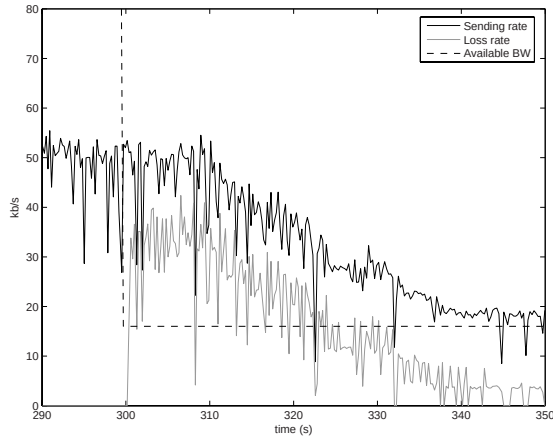


**Fig. 3.** Zoom of the Figure 2 around the bandwidth drop at  $t = 300$  s

To provide a further insight, Figure 3 shows a zoom of Figure 2 in the time interval  $[290, 350]$  in order to look at what happens when the link bandwidth drops from 120 kb/s to 16 kb/s at  $t = 300$  s. It can be viewed that when the available bandwidth drops, the loss rate increases to a peak value of 35 kb/s whereas the sending rate reduces to less than 20 kb/s in 40 s.

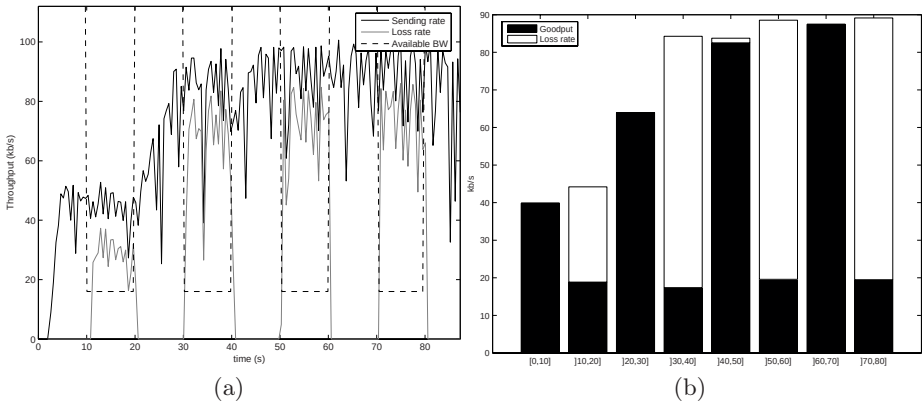


**Fig. 4.** (a) Sending rate and Loss rate with a square wave available bandwidth of 20 s period; (b) Goodput and Loss rate during time intervals at constant available bandwidth

By observing this behaviour it may be conjectured that Skype implements a form of congestion control algorithm that reduces the sending rate when a high packet loss rate is measured.

When the link capacity increases (i.e. at  $t = 400$  s) the input rate goes up to 90 kb/s again in 40 s. Therefore it seems that Skype reacts to bandwidth variations with a transient dynamics that lasts 40 s. In order to validate this finding we consider the same square wave with a period of 20 s. Figure 4 (a) shows that in this case Skype is not able to match the available bandwidth thus provoking congestion loss rates up to 80 kb/s.

From this experiment it results that Skype is able to match the available bandwidth within a transient time of 40 s. We conjecture that this somewhat slow response to bandwidth variation is due to the fact that a sudden variation in the encoding bitrate would have negative effects on user perceived quality. However, such a slow response to congestion episodes is likely to cause unfriendliness when TCP flows share the available bandwidth with Skype flows (see Section 4.3 for details). Moreover, the high packet loss rate experienced in this scenario and shown in Figure 4 (b) may not guarantee perceived quality either.

## 4.2 Case 2: One Skype Flow in the Presence of Variable Bandwidth

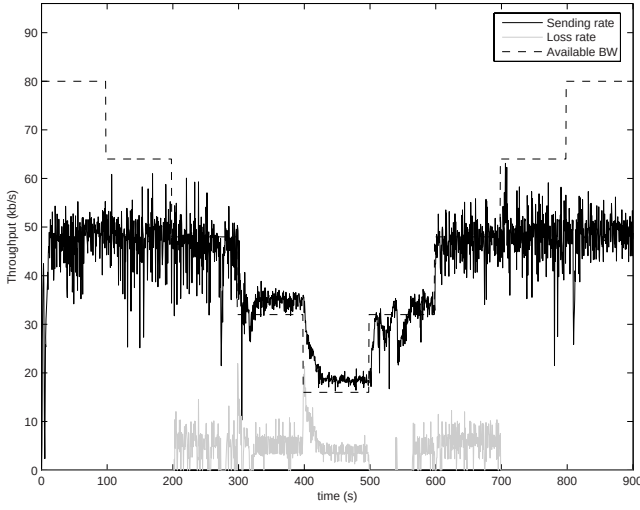
This scenario is aimed at investigating how a Skype's sending rate reacts to small step-like increases and decreases of available bandwidth. To the purpose we allow the available bandwidth to vary in the range  $[16, 80]$  kb/s, which are the minimum and maximum Skype encoder bitrates that we have measured in our experiments.

By using the knowledge about transient times that we have gathered in Section 4.1, we set bandwidth variations to occur every 100 s in order to let sending rates extinguish transients. In particular, the available bandwidth is set as follows: in the first half of the experiment, the flow experiences a 16 kb/s bandwidth drop every 100 s, whereas, in the second half, a bandwidth increase of 16 kb/s occurs every 100 s. This bandwidth variation pattern is particularly suited to test how the Skype sending rate adapts to a sequence of drops and increases.

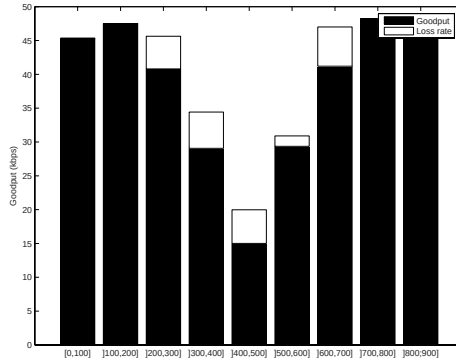
Figure 5 (a) clearly shows how Skype reacts to sudden drops in available bandwidth: when the loss rate is under a given threshold (such as in the time interval  $[200, 300]$ ) the sending rate is kept unchanged until a burst of losses is detected, which triggers a sending rate reduction.

During the bandwidth increasing phase (for  $t > 500$  s), the sending rate is able to match the available bandwidth without congesting the link. It is worthy to focus on the time interval  $[600, 700]$ : we conjecture that, similarly to what happens in the time interval  $[200, 300]$ , the sending rate is kept constant since measured loss rate is not considered harmful for the user perceived quality by the codecs employed by Skype [12,13].

Finally Figure 5 (b) shows goodputs and loss rates measured in each interval during which the bandwidth is constant.



(a)



(b)

**Fig. 5.** (a) Sending rate and loss rate in kb/s of a Skype connection over a time-varying available bandwidth; (b) Goodput and loss rate measured during time intervals at constant available bandwidth

### 4.3 Case 3: One Skype Flow with One Concurrent TCP Connection

The Transmission Control Protocol is by far the most used transmission protocol in the Internet, so it is very important to evaluate how Skype behaves when it shares the network with TCP flows.

#### One Skype Flow and One TCP Flow

In this test we consider a link with a capacity of 56 kb/s. We first start a TCP connection at  $t = 0$  and then a Skype call at  $t = 70$  s that lasts 200 s.

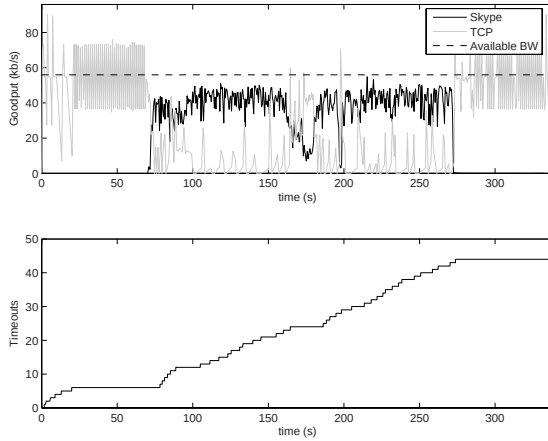


**Fig. 6.** Goodput of Skype and TCP flows; Number of timeouts of the TCP flow

Figure 6 shows that when the Skype flow enters the bottleneck, the TCP connection is not able to get a bandwidth share anymore. In particular, Figure 6 shows that when the Skype flow is ON, the TCP flow experiences around 40 timeouts! This result seems to contradict results obtained before when we have shown that Skype matches the available bandwidth.

The reason is that the TCP congestion control reacts to loss events by halving its congestion window whereas, on the other hand, Skype flows adapt to the available bandwidth slowly. Therefore, even though the TCP congestion control continuously probes for the link bandwidth using its additive increase phase, it is not able to get any significant bandwidth share.

### One Skype Flow and One TCP Flow over a Square Wave Available Bandwidth

In this scenario we consider a square wave available bandwidth with 50% duty cycle, 200 s of period, the maximum value  $A_M = 160$  kb/s and minimum value  $A_m = 40$  kb/s. Moreover, the TCP flow starts at  $t = 0$ , whereas the Skype call is placed after 50 s.

Figure 7 (a) shows that when the available bandwidth is high, both TCP and Skype are able to use the link because the TCP is allowed to take the left over capacity. On the other hand, when the available bandwidth is low, the TCP flow is not able to get any share, and its goodput is close to zero (see Figure 7 (b)).

## 4.4 Skype with 4 Concurrent TCP Flows

This scenario aims at investigating how Skype behaves when multiple TCP flows share the link with one Skype flow. In this experiment, the available bandwidth is set at 120 kb/s and the four TCP flows will join the link following the timing pattern depicted in Figure 9 (a).

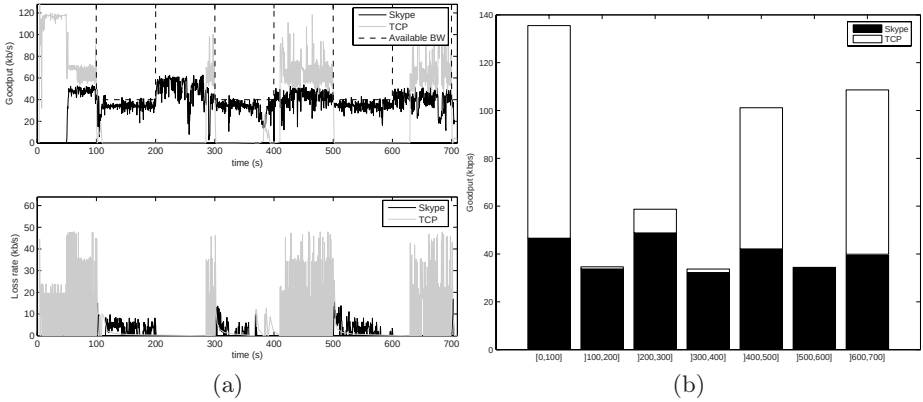


**Fig. 7.** (a) Goodput and loss rate for the Skype and the TCP flows in the presence of a square wave available bandwidth; (b) Goodput of Skype and TCP flows in time intervals where the available bandwidth is kept constant

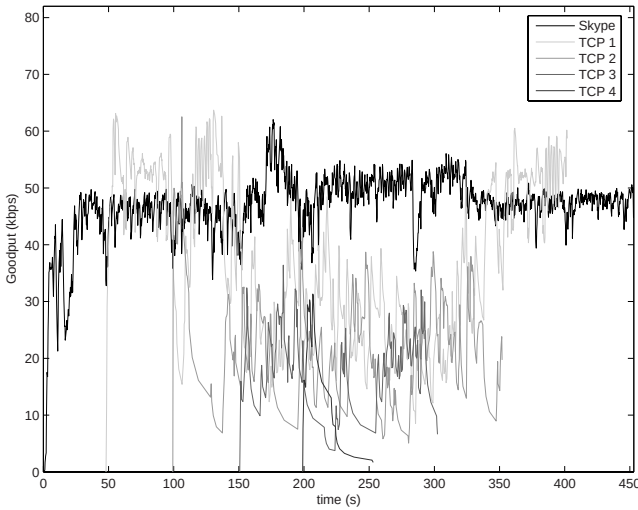


**Fig. 8.** Goodput of the flows

The Figure 8 depicts goodput as a function of time for the Skype flow and for the four TCP flows. It can be noticed that Skype doesn't adapt its sending rate when a new TCP flow joins the bottleneck; on the other hand, TCP flows adapt their rate in order to avoid congestion on the link as it is expected. Figure 9 (b) clearly shows that Skype's goodput is kept unchanged during all the time, while TCP flows share the left available bandwidth.

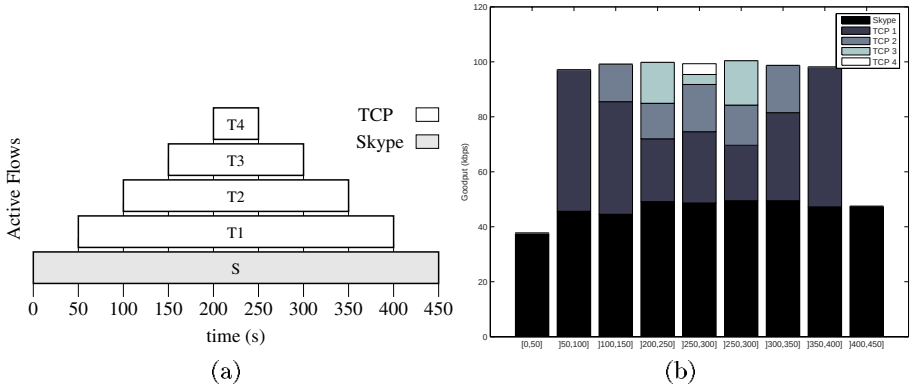


**Fig. 9.** (a) Time intervals during which Skype(s) and TCP flows ( $T_i$ ) are active; (b) Goodput of the flows during each time interval

#### 4.5 Two Skype Connections over a Square Wave Available Bandwidth

In this scenario we consider a square wave available bandwidth with 50% duty cycle, 200 s of period, the maximum value  $A_M = 144$  kb/s and minimum value  $A_m = 64$  kb/s. We start two Skype calls over the same link in order to investigate how Skype clients share the available bandwidth. The first Skype call starts at  $t = 0$ , whereas the second one at  $t = 25$  s. We have run several experiments and we report two of them showing different behaviours.

*Experiment 1.* Figure 10 (a) depicts sending and loss rates for both Skype flows. It can be noticed that except for the first period, i.e. for  $t > 200$  s, the first flow increases its sending rate when the available bandwidth is high, whereas the second flow maintains its sending rate at 30 kb/s throughout all the experiment. When the bandwidth drop is experienced the first flow decreases its sending rate in about 40 s, as we have seen before.

This experiments seems to indicate that the two Skype flows behave differently even if they share the same link. Moreover it can be seen that, when the bandwidth is high, the goodput achieved by the first flow is much higher than the one achieved by the second flow.

*Experiment 2.* Figure 10 (b) shows sending and loss rate for the second experiment we have run. In this case, differently from what has happened in the first experiment, the two flows experience similar sending rates, but congesting the link as it is shown by the high loss rates during the interval  $t \in [100, 500]$ . In particular, by focusing on the bandwidth drop that occurs at  $t = 300$  s it can be shown that both the Skype flows reduce their sending rate and after  $\sim 30$  s they maintain a sending rate of 50 kb/s, which provokes congestion on the link, i.e. high value of the loss rate.

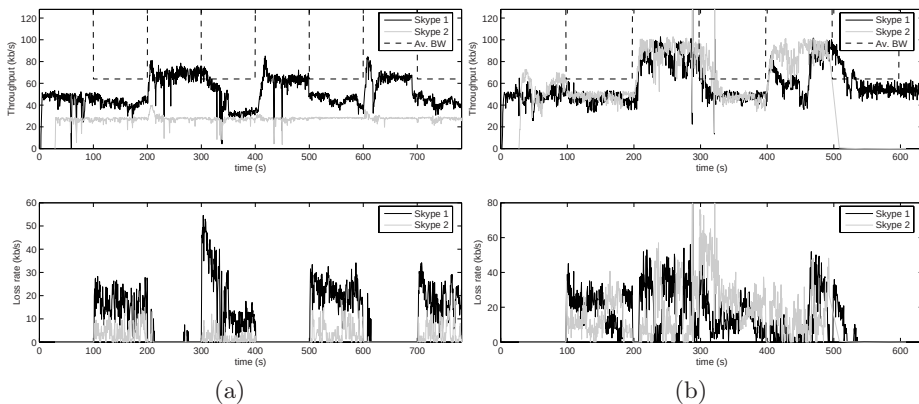


**Fig. 10.** Sending and loss rates for two Skype flows sharing the bottleneck. (a) Experiment 1; (b) Experiment 2.

## 5 Conclusions

We have carried out an experimental investigation of Skype VoIP in a controlled environment in order to find out if and how Skype implements congestion control to match network available bandwidth.

By examining results of our experiments we have found that Skype implements some sort of congestion control algorithm. However, the reaction speed of this algorithm revealed to be very slow. For this reason Skype has shown two main drawbacks: i) Large packet drops rates during the transients following a bandwidth change; ii) unresponsive behaviour when coexisting with responsive flows such as TCP that provoke extreme unfair use of the limited available bandwidth.

Finally, we have also found that when more Skype calls are established on the same link, they are not able to adapt their sending rate to match correctly the available bandwidth, which would confirm the risk of network congestion collapse.

## References

1. S. A. Baset and H. Schulzrinne. "An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol". In *Proceedings of IEEE Infocom '06*, Barcelona, Spain, Apr. 2006.
2. T. Bu, Y. Liu, D. Towsley, "On the TCP-Friendliness of VoIP Traffic", In *Proceedings of IEEE INFOCOM '06*, Barcelona, Spain, Apr. 2006.
3. K. Chen, C. Huang, P. Huang, C. Lei, "Quantifying Skype User Satisfaction", in *Proceedings of SIGCOMM '06*, Pisa, Italy, Sep. 2006.
4. S. Floyd, K. Fall, "Promoting the use of end-to-end congestion control in the Internet", *IEEE/ACM Transaction on Networking*, vol.7, no. 4, pp. 458-472, 1999.



5. S. Floyd, E. Kohler. "TCP Friendly Rate Control (TFRC): the Small-Packet (SP) Variant". IETF draft, 20 Nov. 2006.
6. M. Handley, S. Floyd, J. Padhye, and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification". RFC 3448, Proposed Standard, January 2003.
7. V. Jacobson, "Congestion avoidance and control", ACM SIGCOMM Computer Communication Review, 1995.
8. S. Mascolo, "Congestion control in high-speed communication networks using the Smith principle", *Automatica*, vol. 35, no. 12, Dec. 1999, pp. 1921-1935. Special Issue on "Control methods for communication networks".
9. R. Rejaie, M. Handley, D. Estrin, "RAP: An end-to-end rate-based congestion control mechanism for realtime streams in the Internet", In *Proceedings of IEEE INFOCOM '99*, vol. 3, pp. 1337-1345, 1999.
10. I. Rhee, V. Ozdemir, and Y. Yi, "TEAR: TCP Emulation at Receivers - Flow Control for Multimedia Streaming", Dept. of Comp. Sci., NCSU, Tech. rep., Apr. 2000.
11. L. Rizzo, "Dummysnet: a simple approach to the evaluation of network protocols", ACM SIGCOMM Computer Communication Review, 1997
12. Global IP Sound. "iSAC codec datasheet". [Online] Available: <http://www.globalipsound.com/datasheets/iSAC.pdf>
13. Global IP Sound. "iLBC codec datasheet". [Online] Available: <http://www.globalipsound.com/datasheets/iLBC.pdf>
14. **iperf** [Online]. Available: <http://dast.nlanr.net/Projects/Iperf/>
15. Skype DSP hijacker [Online]. Available: <http://195.38.3.142:6502/skype/>

# A Quality Adaptation Scheme for Internet Video Streams

Panagiotis Papadimitriou and Vassilis Tsaoussidis

Demokritos University, Electrical & Computer Engineering Department,  
12 Vas. Sofias Street, Xanthi, 67100, Greece  
{ppapadim, vtsaousi}@ee.duth.gr

**Abstract.** We propose a layered quality adaptation scheme for video streams to smooth the short-term oscillations induced by *Additive Increase Multiplicative Decrease (AIMD)* mechanisms, and eventually refine the perceptual video quality. The layered scheme utilizes receiver buffering, adapting the video quality along with long-term variations in the available bandwidth. The allocation of a new layer is based on explicit criteria that consider the available bandwidth, as well as the amount of buffering at the receiver. Consequently, the adaptation mechanism prevents wasteful layer changes that have an adverse effect on user-perceived quality. In the sequel, we concentrate on the interactions of the layered approach with *Scalable Streaming Video Protocol (SSVP)*. Exploiting performance measures related to the perceived quality of rate-adaptive video streams, we quantify the combination of SSVP rate control and receiver-buffered layered adaptation.

## 1 Introduction

An increasing demand for multimedia data delivery coupled with reliance on best-effort networks, such as the Internet, has spurred interest in rate-adaptive multimedia streams. Video streaming, in particular, is comparatively intolerant to delay and variations of throughput and delay. Unlike bulk-data transfers, video delivery requires a minimum and continuous bandwidth guarantee. Rate adaptive video streams offer the clients the benefit of being resilient to changing network conditions and allow for a large number of streams to concurrently share network resources. Video streams can be adaptive, since user-perceived *Quality of Service (QoS)* is often satisfactory over a range of stream compression levels. Although this adaptivity is limited (i.e. multimedia streams have minimum subscription levels, below which service quality is unacceptable), they have the capability of adjusting their subscription levels in response to congestion, much as elastic flows do.

Today's Internet is governed by the rules of *Additive Increase Multiplicative Decrease (AIMD)* [2], which effectively contribute to its stability. Essentially, the goal of such algorithms is to prevent applications from either overloading or under-utilizing the available network resources. Although *Transmission Control Protocol (TCP)* provides reliable and efficient services for bulk-data transfers, several design issues render the protocol a less attractive solution for multimedia applications. More precisely, the process of probing for bandwidth and reacting to the observed congestion causes oscillations to the achievable transmission rate. Furthermore, TCP occasionally introduces arbitrary delays, since it enforces reliability and in-order delivery. In response to

standard TCP's limitations, several TCP protocol extensions [1, 4, 12] have emerged, providing more effective bandwidth utilization and sophisticated mechanisms for congestion control. *TCP-friendly* protocols [4, 11, 12] achieve smoother window adjustments, while they manage to compete fairly with TCP flows. In order to achieve smoothness, they use gentle backward adjustments upon congestion. However, they compromise responsiveness through moderated upward adjustments [9].

Considering TCP's limitations and the impending threat of unresponsive UDP traffic, rate-based congestion control has become an attractive alternative [4, 6, 8]. Avoiding the burstiness occasionally induced by the window-based mechanisms, rate-based protocols generate a smooth data flow by spreading the data transmission across a time interval. Therefore, rate-based mechanisms compose plausible candidates for media-streaming applications.

In addition to congestion control, a streaming video server should be able to control the quality of the video stream depending on the prevailing network conditions. *Simulcast* and *layered adaptation* compose two widely-used quality adaptation techniques. Simulcast uses multiple versions of the stream, encoded at different bitrates. The server transmits all the alternate streams and the client switches to the stream version that best matches its capacity. Layered adaptation has been proposed as a solution to the bandwidth redundancy introduced by simulcast. This approach is based on information decomposition. That is, the video stream is encoded at a *base layer* and one or more *enhancement layers*, which can be combined to render the stream at high quality. Layered adaptation is performed by adding or dropping enhancement layers depending on current conditions. In general, a large number of video layers results in greater bandwidth resolution. However, depending on the coding method, there is a trade-off between the number of layers and the video coding efficiency.

We are particularly interested in the interactions between layered adaptation and AIMD-based congestion control. In this context, we propose a receiver-buffered layered scheme to adapt video quality along with long-term variations in the available bandwidth. Receiver buffering reduces jitter, and depending on the amount of buffered data, the receiver is enabled to sustain temporary drops in the sending rate. In order to prevent wasteful layer changes which impair the perceived video quality, the proposed mechanism allocates additional layers based on certain criteria that consider the available bandwidth, as well the amount of buffering at the receiver. The layered scheme is designed to effectively interact with AIMD mechanisms: layered encoding allows video quality adjustments over long periods of time, whereas AIMD congestion control adjusts the transmission rate rapidly over short-time intervals. In particular, we study the interactions of the layered approach with *Scalable Streaming Video Protocol (SSVP)* [6], adjusting the quality of congestion-controlled video on-the-fly. SSVP is an AIMD-oriented rate control scheme optimized for video streaming applications. Quantifying the interactions of SSVP with the specific adaptation scheme, we identify that layered adaptation in conjunction with receiver buffering smoothes the short-term oscillations induced by AIMD mechanisms, and eventually refines the perceptual video quality.

The remainder of the paper is organized as follows. The following section summarizes related work. In Section 3, we analyze the parameters of the proposed layered scheme and study its interactions with SSVP. Section 4 includes conclusive performance studies based on simulations. Finally, Section 5 concludes the paper.

## 2 Related Work

Numerous studies for adaptive video delivery appear in [3, 5, 7, 10]. An overview of existing solutions for video adaptation is presented in [5]. Authors in [3] analyze the impact of selected congestion control algorithms on the performance of streaming video delivery. They concentrate on binomial congestion control [1] and especially on SQRT, which responds to packet drops by reducing the congestion window size proportional to the square root of its value instead of halving it. [7] proposes a layered mechanism to adapt the quality of congestion-controlled video. The mechanism is able to control the level of smoothing in order to improve the quality of the delivered video stream. Authors in [10] provide a system-level analysis of performance and design issues surrounding rate adaptive networks.

The literature includes numerous studies and proposals towards efficient rate/congestion control for multimedia applications in the Internet. *Rate Adaptation Protocol (RAP)* [8] is a rate-based protocol which employs an AIMD algorithm for the transmission of real-time streams. The sending rate is continuously adjusted by RAP in a TCP-friendly fashion, using feedback from the receiver. However, since RAP employs TCP's congestion control parameters (i.e. 1, 0.5), it causes short-term rate oscillations, primarily due to the multiplicative decrease. Furthermore, RAP occasionally does not result in inter-protocol fairness.

TFRC [4] is a representative equation-based protocol, which adjusts its transmission rate in response to the level of congestion, as estimated based on the calculated loss rate. Multiple packet drops in the same *Round Trip Time (RTT)* are considered as a single loss event by TFRC and hence, the protocol follows a more gentle congestion control strategy. More precisely, the TFRC sender uses the following response function:

$$T(p, RTT, RTO) = \frac{1}{RTT \sqrt{\frac{2p}{3}} + RTO (3 \sqrt{\frac{3p}{8}}) p (1 + 32p^2)} \quad (1)$$

where  $p$  is the steady-state loss event rate and  $RTO$  is the retransmission timeout value. Equation (1) enforces an upper bound on the sending rate  $T$ . However, the throughput model is quite sensitive to parameters (e.g.  $p$ ,  $RTT$ ), which are often difficult to measure efficiently and to predict accurately. Also, the long-term TCP throughput equation does not capture the transit and short-lived TCP behaviours, and it is less responsive to short-term network and session dynamics. TFRC eventually achieves the smoothing of the transmission gaps and is suitable for applications requiring a smooth sending rate. However, this smoothness has a negative impact, as the protocol becomes less responsive to bandwidth availability [9].

*GAIMD* [12] is a TCP-friendly protocol that generalizes AIMD congestion control by parameterizing the additive increase rate  $\alpha$  and multiplicative decrease ratio  $\beta$ . For the family of AIMD protocols, authors in [12] derive a simple relationship between  $\alpha$  and  $\beta$  in order to be friendly to standard TCP (i.e.  $\alpha = 4 (1 - \beta^2) / 3$ ). Based on experiments, they propose an adjustment of  $\beta = 0.875$  as an appropriate smooth decrease ratio, and a moderated increase value  $\alpha = 0.31$  to achieve TCP friendliness.

### 3 Quality Adaptation

The rationale of a quality adaptation scheme mainly rests on the assumption that a user's perception is sensitive to the changes in video quality, as well as to potential interruptions in the stream playback. Despite the degradation in visual quality, we consider smooth video of reduced bitrate more preferable than inconsistent and jerky video at highest quality.

#### 3.1 Receiver-Buffered Layered Adaptation

We propose a quality adaptation mechanism in order to sustain smooth video delivery in a wide range of network dynamics. We specifically adopt the layered approach, where the streaming server coarsely adjusts video quality on-the-fly, without the need to implement transcoding. The server encodes raw video into  $n$  cumulative layers using a layered coder: layer 1 is the base layer and layer  $n$  is the least important enhancement layer. The layer rates are given by  $r_i$ ,  $i = 1, 2, \dots, n$  and  $l_j$  denotes the

cumulative layer up to layer  $j$ , i.e.  $l_j = \sum_{i=1}^j r_i$ ,  $j = 1, 2, \dots, n$ . The receiver obtains a certain number of layers depending on bandwidth availability. All active layers  $k$  ( $1 \leq k \leq n$ ) are typically multiplexed in a single AIMD flow with rate  $R = \sum_{i=1}^k r_i = k\bar{r}$ ,

where  $\bar{r}$  denotes the average rate among the  $k$  layers.

Quality adaptation should not be simply supported by initial buffering at the receiver, because long-lived mismatch between the available bandwidth and the playback quality results in either buffer overflow or underflow. Alternatively, we rely on a receiver-buffered layered scheme, where each layer supplies a corresponding buffer, as depicted in Fig. 1. We assume that each buffer is drained with a constant rate  $c_i$ ,  $i = 1, 2, \dots, k$ . Consequently, the total consumption rate at the receiver buffers is  $C = \sum_{i=1}^k c_i = k\bar{c}$ , where  $\bar{c}$  denotes the average consumption rate

of the buffers that correspond to the  $k$  active layers. The efficiency of a layered mechanism is significantly affected by the frequency of layer changes. Since we do not know in advance how long we will be able to sustain a specific layer, minimal rate variations should not directly trigger video quality adjustments. Generally, switching layers unduly usually induces perceptible video quality variations with frustrating consequences to the end-user. A simplified approach for the allocation of additional layers could be based exclusively on the instantaneous available bandwidth. Subsequently, the server would send video at the highest possible layer at any given time. In this case, the frequency and magnitude of oscillations in the congestion control algorithm would eventually govern the corresponding frequency and magnitude of layer changes. This would inevitably distract the end-user, considering that the Internet operates in the transient than in the stationary regime with occasional and rapid changes in the available bandwidth.

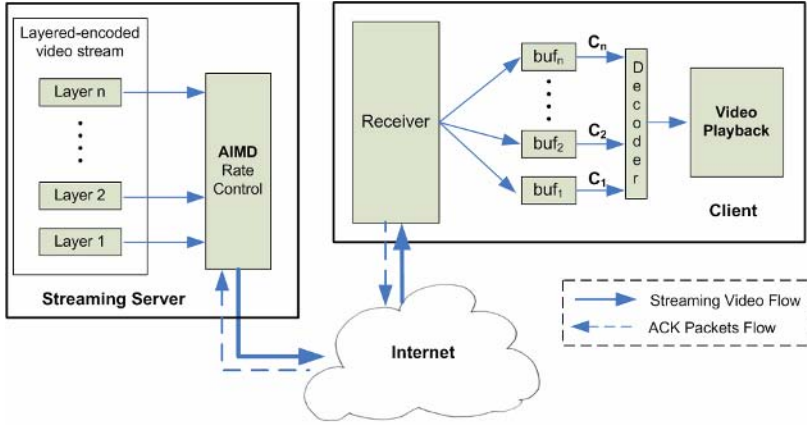


Fig. 1. Receiver-buffered layered adaptation

Following [7], certain conditions in the instantaneous available bandwidth and the amount of buffering at the receiver can be applied in order to reduce the number of layer changes. Initially, we concentrate on defining a priori whether a new layer should be allocated under the properties of AIMD rate control. Instead of delivering the maximum number of layers that can be accommodated by the network resources, we add a new layer, as soon as the current transmission rate  $R$  exceeds the total consumption rate of all currently active layers plus a new one:

$$R > (k + 1) \bar{c} \quad (2)$$

In this case, the new layer can be decoded immediately. However, relying on such rule does not eventually prevent layer changes; oscillations in the congestion control algorithm may still result in adding or dropping layers. Only sufficient buffering at the receiver can smooth out the variations in the available bandwidth and sustain a relatively constant number of active layers throughout the connection.

Along these lines, we attempt to derive a second rule associated with the amount of buffering required at the receiver. Fig. 2 depicts the behavior of an adaptive video

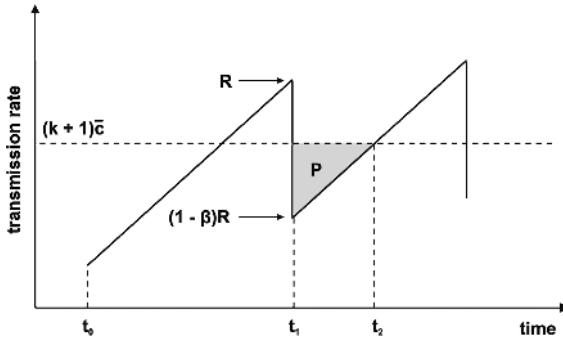


Fig. 2. Layered adaptation with AIMD rate control

flow under generalized AIMD ( $\alpha$ ,  $\beta$ ) congestion control. After one backoff at time  $t_1$ , we observe that at time  $t_2$ , condition (2) holds; in addition, the buffering requirements are satisfied when:

$$\sum_{i=1}^k \text{buf}_i \geq P \quad (3)$$

where  $\text{buf}_i$  is the amount of video-data buffered for the  $i^{\text{th}}$  layer and  $P$  is the area of the shaded portion in Fig. 2.  $P$  represents the area of a triangle and can be derived by:

$$P = \frac{1}{2} (t_2 - t_1) [(k+1) \bar{c} - (1-\beta) R] \quad (4)$$

Assuming that the AIMD flow is in the congestion avoidance phase, the transmission rate  $R$  follows a line with slope  $\alpha$  (Fig. 2). Subsequently, we obtain:

$$(k+1) \bar{c} = (1-\beta) R + \alpha (t_2 - t_1) \Rightarrow t_2 - t_1 = \frac{(k+1) \bar{c} - (1-\beta) R}{\alpha} \quad (5)$$

Combining equations (4) and (5),  $P$  is given by:

$$P = \frac{[(k+1) \bar{c} - (1-\beta) R]^2}{2\alpha} \quad (6)$$

Finally, we derive the second necessary and sufficient condition in order to allocate a new layer:

$$\sum_{i=1}^k \text{buf}_i \geq \frac{[(k+1) \bar{c} - (1-\beta) R]^2}{2\alpha} \quad (7)$$

Employing both rules for the allocation of additional layers enables the adaptation mechanism to trade short-term improvements for long-term quality smoothing, preventing buffer overflows and eventually rapid fluctuations in quality that frustrate the end-user.

### 3.2 Receiver-Buffered Layered Adaptation with SSVP Rate Control

We exploit the proposed receiver-buffered scheme to abolish the negative impact of short-term oscillations induced by AIMD mechanisms on video delivery. We hereby concentrate on the interactions between the layered approach and SSVP [6] rate control. SSVP, in a complementary role, operates on top of UDP, relying on sender and receiver interaction. The recipient uses control packets to send feedback of reception statistics to the sender. In accordance with the relaxed packet loss requirements of streaming video and considering the delays induced by retransmitted packets, SSVP does not integrate reliability into UDP datagrams. Hence, control packets do not trigger retransmissions. However, they are effectively used in order to determine bandwidth and RTT estimates, and properly adjust the rate of the transmitted video stream. The recipient uses packet drops or re-ordering as congestion indicator. SSVP employs an AIMD-oriented congestion control mechanism with  $\alpha = 0.2$  and  $\beta = 0.875$ . The transmission rate is controlled by properly adjusting the inter-packet-gap (IPG). If no congestion is sensed, IPG is reduced additively; otherwise, it is increased multiplicatively. Further details in the operation of SSVP can be found in [6].

Considering SSVp's rate control parameters ( $\alpha = 0.2$ ,  $\beta = 0.875$ ), we rewrite equation (7), as:

$$\sum_{i=1}^k \text{buf}_i \geq \frac{5}{2} \left[ (k+1) \bar{c} - \frac{R}{8} \right]^2 \quad (8)$$

SSVP enables a closed-loop control between server and client, where the receiver detects the state of congestion, determines the proper transmission rate and eventually opts for the number of layers that should be delivered. As a receiver-centric protocol, SSVp has first knowledge on the amount of buffering at the receiver and can augment the layered scheme to determine the optimal number of layers according to the prevailing conditions. Delegating on the client the selection of the number of layers to send composes a more tractable approach, since the sender may not be always able to accurately predict the amount of buffered data at the receiver. The allocation of a new layer is enabled only when both conditions (2) and (8) are satisfied. Feedback is provided to the server via the SSVp control packets.

### 3.3 QoS Assessment for Rate-Adaptive Video Streams

Consider that changes in stream subscription level have an adverse effect on user-perceived QoS, due to the distraction caused by changing video resolution. The term *subscription level* corresponds to the average rate of a stream for all possible encoding and compression levels for the given stream. In the case of a server utilizing layered encoding, a subscription level  $S$  infers that the client receives the base layer plus  $S-1$  enhancement layers. We are interested in the assessment of dynamic rate adaptation, where a client changes its subscription level throughout stream playback in response to bandwidth availability along its route. Following [10], we use two QoS measures for rate-adaptive video streams: (i) the normalized average subscription level  $\bar{S}$  within  $(0, 1]$ , and (ii) the frequency of rate adaptation observed by an end-user, defined as:

$$F = \frac{1}{d} \sum |S(t^+) - S(t^-)|, \quad 0 \leq t \leq d \quad (9)$$

where  $S(t)$  denotes the instantaneous subscription level at each time  $t$ . For example,  $F = 1$  corresponds to one change in the stream subscription level per second on average. A combination of both QoS measures provides a useful insight to the user-perceived performance for rate-adaptive video streams.

## 4 Performance Evaluation

### 4.1 Experimental Environment

The evaluation plan was implemented on the NS-2 network simulator. Simulations were conducted on a single-bottleneck *dumbbell* topology (Fig. 3) with a round-trip link delay of 64 ms. The bottleneck link is shared by competing MPEG and FTP connections and its capacity is configured depending on the experiment. The capacity of all access links to the source and sink nodes is set to 1 Mbps. The routers are drop-tail with buffer size adjusted in accordance with the *bandwidth-delay* product. We set the



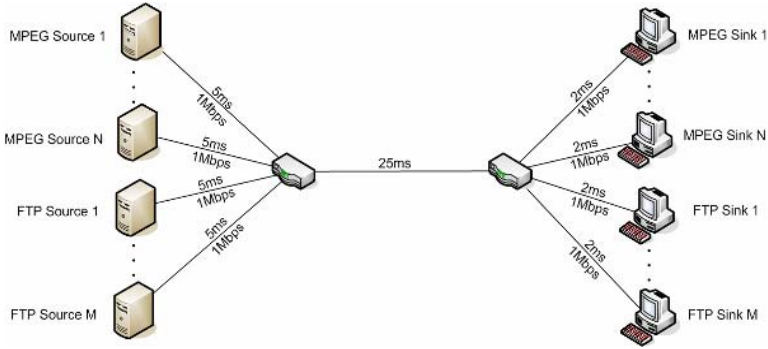


Fig. 3. Simulation topology

packet size to 1000 bytes for all system flows and the maximum congestion window to 64 KB for all TCP connections. The duration of each simulation is 60 sec.

In order to simulate MPEG traffic, we developed an *MPEG-4 Traffic Generator*. The traffic generated closely matches the statistical characteristics of an original MPEG-4 video trace. The model developed is based on *Transform Expand Sample (TES)*. We used three separate *TES* models for modeling I, P and B frames, respectively. The resulting MPEG-4 stream is generated by interleaving data obtained by the three models.

We hereby refer to the performance metrics supported by our simulation model. *Goodput* is used to measure the efficiency in bandwidth utilization. Following the metric in [11], we use *Coefficient of Variation (CoV)* in order to gauge the throughput smoothness experienced by flow  $i$ :

$$\text{CoV}_i = \frac{\sqrt{E_t\{\text{throughput}_i^2(t)\} - E_t\{\text{throughput}_i(t)\}^2}}{E_t\{\text{throughput}_i(t)\}}$$

where  $E_t\{\}$  denotes the computation of the mean along time. Throughput rates are sampled every 150 ms. Long-term fairness is measured by the *Fairness Index*, derived

from the formula given in [2], and defined as  $(\sum_{i=1}^n \text{Throughput}_i)^2 / (n \sum_{i=1}^n \text{Throughput}_i^2)$ ,

where  $\text{Throughput}_i$  is the throughput of the  $i^{\text{th}}$  flow and  $n$  is the total number of flows.

In order to quantify the performance on video delivery, we monitor packet inter-arrival times and eventually distinguish the packets that can be effectively used by the client application from delayed packets according to a configurable packet inter-arrival threshold. The proportion of delayed packets is denoted as *Delayed Packets Rate*. In accordance with video streaming requirements, we adjusted the packet inter-arrival threshold at 75 ms. Since MPEG traffic is sensitive to packet loss, we additionally define *Packet Loss Rate*, as the ratio of the number of lost packets over the number of packets sent by the application.

## 4.2 Results and Discussion

In the sequel, we demonstrate conclusive performance studies based on selected simulation results. Initially, we simulated one MPEG flow over SSVP (i) without adaptation, (ii) with *instantaneous* adaptation (SSVP-IA), and (iii) with the proposed receiver-buffered layered adaptation scheme (SSVP-LA). In the case of instantaneous adaptation, the server adapts the number of layers based on the instantaneous available bandwidth. The simulations were conducted on the dumbbell topology with bottleneck capacity of 1 Mbps. The system also includes two FTP flows over TCP Reno in order to enforce contention with the interfering MPEG flow. Fig. 4 illustrates the sending rate from the MPEG transfer in each case. On the occurrence of layered adaptation (i.e. Figs. 4b, 4c), the corresponding layer changes are also demonstrated. Table 1 includes useful statistics to assess the QoS perceived by the end-user.

According to Fig. 4c and the frequency of rate adaptation (i.e.  $F = 0.11667$ ), the receiver-buffered scheme results in a minimal number of layer changes, adapting video quality along with long-term variations in the available bandwidth. Receiver buffering is able to sustain temporary drops in the sending rate and prevents wasteful dropping of layers. A new layer is allocated based on the available bandwidth, as well the amount of buffering at the receiver. The normalized average subscription level for the MPEG stream in the case of SSVP-LA is 0.75758. On the contrary, instantaneous adaptation causes frequent layer changes in response to the variations in the available bandwidth. The frequency of rate adaptation (i.e.  $F = 1.05$ ) is significantly higher in comparison with the receiver-buffered approach, and has an adverse effect on user-perceiver quality. Despite a slightly higher  $\bar{S} = 0.84413$  for the instantaneously adapted MPEG stream, the frequency of oscillations in the congestion control algorithm eventually governs the corresponding frequency of layer switching. Subsequently, the end-user is often distracted by the changes in video quality.

We are also interested in the interactions between the receiver-buffered adaptation scheme and the underlying AIMD protocol. We exploit CoV (Table 1) to quantify the magnitude of variation in the transmission rate. Note that a lower CoV expresses a lower variation in the sending rate, and consequently higher smoothness. Apparently, the receiver-buffered scheme reduces SSVP's short-term oscillations and the protocol delivers a smoothed video flow, which remains relatively immunized from the disturbances caused by the interfering FTP connections. The gains in smoothness (compared to SSVP) are less perceptible in the case of instantaneous adaptation; the measured CoV is still lower than SSVP alone, but notably higher than SSVP-LA. We point out that SSVP is a protocol that anticipates smoothness and delivers a smoothed flow, even without the support of a quality adaptation mechanism. As a result, the protocol does not leave much room for improvement in terms of smoothness. Therefore, we expect more significant gains, when the receiver-buffered mechanism augments other AIMD protocols that exhibit lower levels of smoothness.

We also carried out a series of simulations in order to assess the performance of receiver-buffered adaptation with SSVP-LA and multiple flows. In this context, we simulated a wide range of MPEG flows (1-50) over (i) SSVP-LA, (ii) TFRC, and (iii) GAIMD, competing with 5 FTP connections over TCP Reno, successively. GAIMD

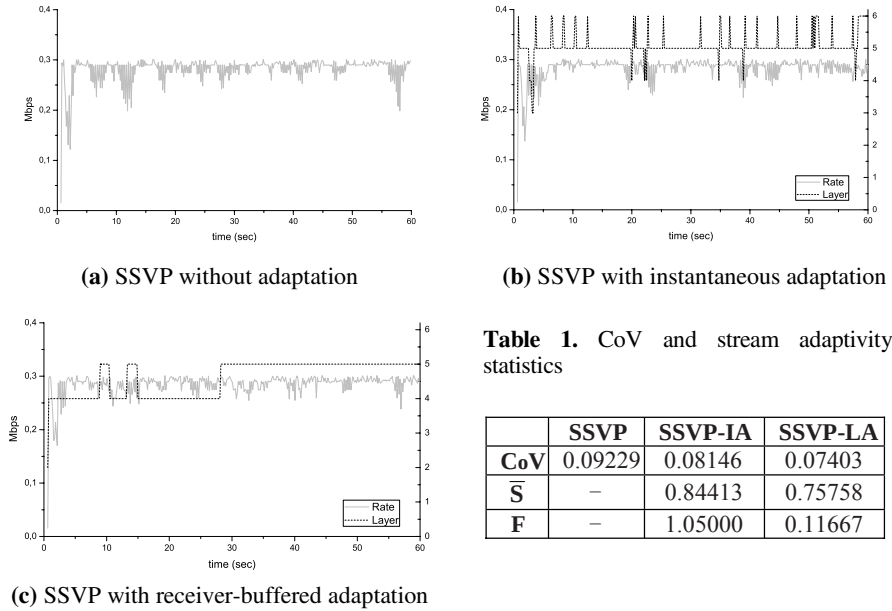


**Table 1.** CoV and stream adaptivity statistics

|           | SSVP    | SSVP-IA | SSVP-LA |
|-----------|---------|---------|---------|
| CoV       | 0.09229 | 0.08146 | 0.07403 |
| $\bar{S}$ | —       | 0.84413 | 0.75758 |
| F         | —       | 1.05000 | 0.11667 |

**Fig. 4.** Transmission rate and layer variation

and especially TFRC are implied to achieve remarkable efficiency on media delivery over a wide range of network and session dynamics. When the MPEG-4 server transmits over TFRC or GAIMD, the video is streamed at the optimal quality (i.e. layered adaptation is disabled). As a result, we evaluate the transmission rate control performed by each protocol at the transport layer. The experiments were conducted on the dumb-bell topology with a bottleneck capacity of 10 Mbps. Since today's multimedia applications are expected to run in physically heterogeneous environments composed of both wired and wireless components, we included *NS-2* error models into the access links to the MPEG sink nodes. We used the Bernoulli model in order to simulate the link errors with packet error rate adjusted at 0.01. We measured *Goodput*, *Fairness Index*, and we additionally demonstrate statistics from delayed and lost packets, since both compose influencing factors for perceived video quality (Figs. 5-7).

According to Fig. 5, SSVP-LA flows utilize a high fraction of the available bandwidth, despite the delivery of lower bitrate video (the difference in rate/quality is subject to the prevailing network conditions) in comparison with TFRC and GAIMD. We observe that with high link-multiplexing, SSVP-LA outperforms the rest of the protocols in terms of bandwidth utilization, since the interaction of layered encoding with AIMD congestion control enables the effective adaptation of the video rate to the current network dynamics. GAIMD, in particular, yields inadequate bandwidth utilization, since it composes a *blind* window mechanism that relies on specific events triggered by violated thresholds.

Fig. 6 illustrates that SSVP-LA and GAIMD achieve high levels of fairness. The AIMD-based responses during congestion enforce competing flows to converge to the fairness point for both protocols. Furthermore, the receiver-buffered adaptation

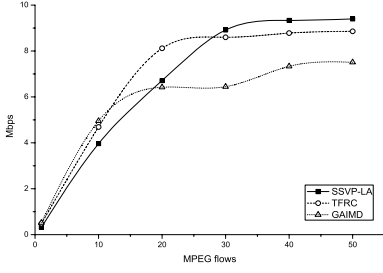


Fig. 5. Goodput of MPEG flows

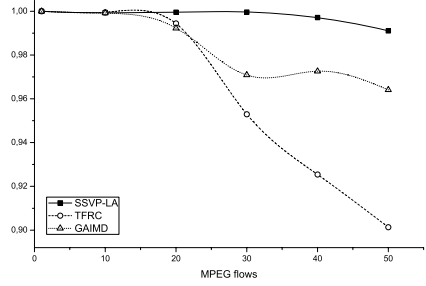
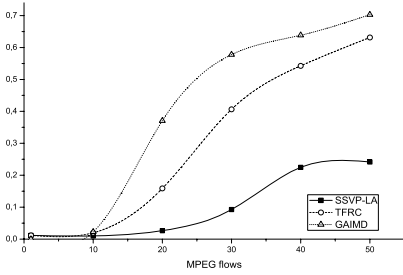
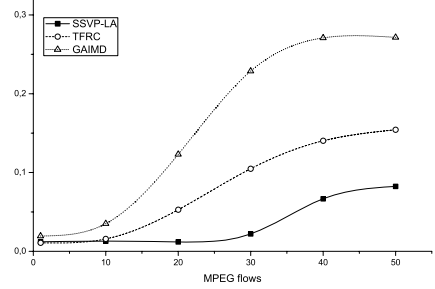


Fig. 6. Fairness Index



(a) Packet Loss Rate



(b) Delayed Packets Rate

Fig. 7. Performance on video delivery

mechanism confines packet loss (Fig. 7a) and therefore reduces the number of congestion cycles (i.e. the period between two consecutive loss indications), which results in fairness gains for SSVP-LA. On the other hand, we observe that the *Fairness Index* for TFRC degrades abruptly, reflecting a throughput imbalance between the connections. Apparently, TFRC's equation-based responses to packet loss undermine long-term fairness, along with contention increase.

According to Fig. 7, SSVP-LA achieves the timely delivery of most packets maintaining an uninterrupted and smooth video flow that is slightly affected by contention. The layered mechanism eventually bridges the gap between the short-term adjustments in the transmission rate caused by congestion control with the need for stable quality in streaming applications. Therefore, the flexibility of the receiver-buffered layered approach contributes in the delivery of smooth video in a wide range of network dynamics. SSVP-LA eventually enforces an upper bound to the magnitude of delay variation (Fig. 7b), providing a possible guarantee for streaming applications that can efficiently operate within this QoS provision. On the other hand, TFRC and GAIMD induce considerable variations in the receiving rate, with the effect of jitter becoming evident to the end-user. Furthermore, Fig. 7a depicts considerable packet loss for TFRC, which inevitably deteriorates the perceived video quality. The protocol occasionally fails to obtain accurate estimates of the loss event rate, invoking an inappropriate equation-based recovery, since TFRC's throughput model is sensitive to packet loss. We also observe a high packet loss rate for GAIMD, due to its gentle window decrease ratio at periods of congestion.

## 5 Conclusions

We have proposed a quality adaptation scheme that takes advantage of receiver buffering to alleviate most of the impairments caused by the oscillatory behavior of AIMD congestion control algorithms, improving the perceptual video quality. Our performance studies have validated the efficiency of the specific mechanism, uncovering significant performance gains, especially in the presence of limited bandwidth. Essentially, the proposed adaptation scheme provides a strong incentive for streaming applications to rely on AIMD rate control for efficient and smooth video delivery.

## References

1. D. Bansal and H. Balakrishnan, Binomial Congestion Control Algorithms, in Proc. IEEE INFOCOM 2001, Anchorage, Alaska, USA, (2001)
2. D. Chiu and R. Jain, Analysis of the increase/decrease algorithms for congestion avoidance in computer networks, *Journal of Computer Networks and ISDN*, Vol. 17, (1989) 1-14
3. N. Feamster, D. Bansal, and H. Balakrishnan, On the Interactions Between Layered Quality Adaptation and Congestion Control for Streaming Video, in Proc. 11<sup>th</sup> Int/nal Packet Video Workshop, Korea, (2001)
4. S. Floyd, M. Handley, J. Padhye, and J. Widmer, Equation-Based Congestion Control for Unicast Applications, in Proc. ACM SIGCOMM 2000, Stockholm, Sweden, (2000)
5. J. Liu, B. Li and Y. Zhang, Adaptive Video Multicast over the Internet, *IEEE Multimedia*, Vol. 10, (2003) 22-33
6. P. Papadimitriou and V. Tsaoussidis, End-to-end Congestion Management for Real-Time Streaming Video over the Internet, in Proc. 49<sup>th</sup> IEEE GLOBECOM, San Francisco, USA, (2006)
7. R. Rejaie, M. Handley, and D. Estrin, "Layered Quality Adaptation for Internet Video Streaming", *IEEE Journal on Selected Areas in Communications (JSAC)*, Vol. 18, (2000) 2530-2544
8. R. Rejaie, M. Handley, and D. Estrin, RAP: An End-to-end Rate-based Congestion Control Mechanism for Realtime Streams in the Internet, in Proc. IEEE INFOCOM 1999, New York, USA, (1999)
9. V. Tsaoussidis and C. Zhang, The Dynamics of Responsiveness and Smoothness in Heterogeneous networks, *IEEE Journal on Selected Areas in Communications (JSAC)*, Vol. 23, (2005) 1178-1189
10. S. Weber and G. Veciana, Network Design for Rate Adaptive Media Streams, in Proc. IEEE INFOCOM 2003, San Francisco, USA, (2003)
11. Y. R. Yang, M. S. Kim, and S. S. Lam, Transient Behaviors of TCP-friendly Congestion Control Protocols, in Proc. IEEE INFOCOM 2001, Anchorage, Alaska, USA, (2001)
12. Y. R. Yang and S. S. Lam, General AIMD Congestion Control, in Proc. 8<sup>th</sup> IEEE Int/nal Conference on Network Protocols (ICNP), Osaka, Japan, (2000)

# Performance Analysis of VoIP over HSDPA in a Multi-cell Environment

Irene de Bruin<sup>1</sup>, Frank Brouwer<sup>1</sup>, Neill Whillans<sup>1</sup>,  
Yusun Fu<sup>2</sup>, and Youqian Xiao<sup>2</sup>

<sup>1</sup> Twente Institute for Wireless and Mobile Communications,  
Institutenweg 30, 7521 PK, Enschede, The Netherlands  
[Irene.de.Bruin@ti-wmc.nl](mailto:Irene.de.Bruin@ti-wmc.nl)  
<http://www.ti-wmc.nl>

<sup>2</sup> Huawei Technologies Co., Ltd, No. 98 Lane 91, E'shan Road,  
Pudong New District, 200127 Shanghai, China

**Abstract.** This paper describes network level simulation results for VoIP and Best Effort traffic running over HSDPA. The main focus involves an analysis of how the system is able to deal with user mobility, and in particular handover. Both a handover threshold level as well as a handover setup delay have been included in the simulation model, and a discard timer is used to remove outdated packets from the Node B queue.

Link adaptation and H-ARQ (Hybrid-Automatic Repeat reQuest) functionalities are also included in the simulator. An infinite multi-cell environment has been modeled and the users are assumed to move at vehicular speeds. A performance analysis will show the end-to-end QoS. The handover threshold, delay and discard times are also varied in order to investigate their impact on the performance.

## 1 Introduction

It is envisioned that upcoming services will be multimedia in nature. While the initial scope of HSDPA (High Speed Downlink Packet Access) was on Best Effort services, the interest in using HSDPA also for real-time applications is growing. An important reason for this interest is the increasing role of VoIP (Voice over IP) in both fixed and wireless networks. Although UMTS and HSDPA are already being rolled out, network simulations provide an add-on value since these allow for the execution of controlled experiments, addressing bottlenecks that are not feasible yet in real networks or testbeds [1,9,11,12].

VoIP, being a real-time service, has significantly different requirements on the connection than Best Effort services. Delay is the most critical issue for VoIP, whereas loss of some packets can be tolerated. At the same time, Best Effort services require guaranteed delivery of the data, but do accept additional delay. Under high mobility situations, handovers become an important issue for VoIP and other real-time services. As the soft and softer handover do not apply to the HS-DSCH (High-Speed Downlink Shared Channel, the data channel of HSDPA), the handover is a hard handover, with 'break before make'. This process is depicted in Figure 1. After sending the last packet over the Iub to the

serving cell, some time needs to be included to empty the buffer of the serving cell (*i.e.* from *switch\_Iub* to *switch\_Uu*). In the mean time, no transmission should start from the new cell yet. The delay between these two switch moments is referred to as the 'handover delay'.

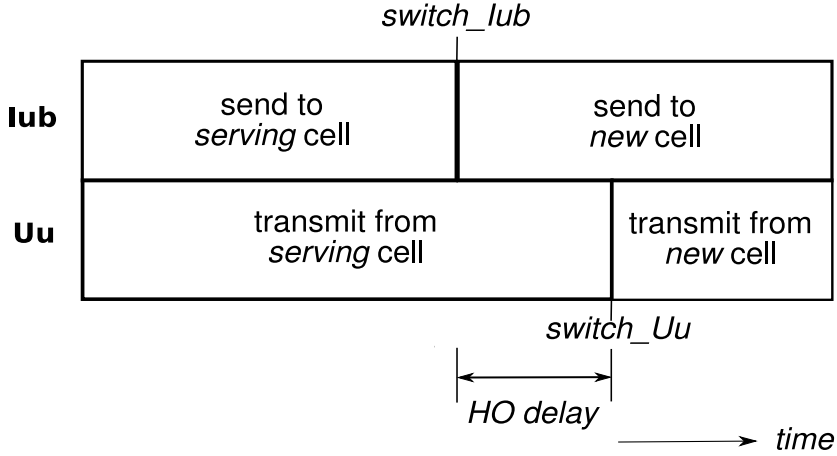


**Fig. 1.** Time evolution of the handover process, including the two switch moments that have *Handover (HO) delay* in between them

As the VoIP packets are time-critical and the AMR (Adaptive Multi Rate) codec allows some packet loss, the RLC (Radio Link Control) transmits the packets in UM (Unacknowledged Mode). Limiting the packet loss in UM requires that the time instances of stopping and starting should be such that the serving (old) cell has sufficient time to complete transmission of packets already sent over the Iub while avoiding a time gap in the transmission. Too short a handover delay results in packets that are still waiting in the queue of the old Node B at the time of switching. These packets are, by definition, lost. At the same time, any additional waiting time causes extra delay to packets waiting in the new cell. This paper investigates this trade-off, in particular with respect to handover aspects of an HSDPA system carrying VoIP services. In order to come closer to a realistic HSDPA scenario, Best Effort traffic is also included.

In the remainder of this paper, Section 2 discusses the simulation model this analysis uses, including the description of the scenario and the propagation modelling. Section 3 describes some simulation results for the default scenario. These will be used as a reference case to discuss the relation between handovers and the performance of VoIP over HSDPA in Section 4. Finally, Section 5 concludes this paper.

## 2 Scenario and Model Description

All results described in this study are gathered with the ns-2 based network level simulator EURANE (Enhanced UMTS Radio Access Networks Extension) [10].

The path loss model for EURANE consists of three parts: distance loss, shadowing and multi-path. The model treats these three components independently. The simulator includes many details, in particular the MAC-hs (Medium Access Control) and RLC (Radio Link Control) functionality, as well as a more abstract modelling of the PHY, MAC-d and Iub flow control, of which high-speed versions of these protocols are implemented for HSDPA according to 3GPP standards. The H-ARQ model assumes Chase Combining, which utilizes retransmissions to obtain a higher likelihood of packet reception. For the analysis described in this paper it is assumed that the UEs (User Equipment) are of categories 1 to 6. The simulator selects the TBS (Transport Block Size) from table 7A of 3GPP specification 25.214 [2].

We consider a Macro cell with the COST 231-Hata distance loss model. The site-to-site distance is 1000 m. The cell layout has a hexagonal structure, where each Node-B covers three cells. The users are moving at 120 km/h in a Vehicular A environment. The RBS (Radio Base Station) maximum power is set to 43 dBm per cell. The antenna pattern has a front-to-back ratio of 30 dB, a 3 dB opening angle of  $66^\circ$ , a gain of 17 dBi and a 3 dB feeder loss. It should be stressed that the interpretation of some results may change for different scenario settings. Also a different overall load and/or traffic mix may give different conclusions, in particular *e.g.* the optimization of the discard timer settings.

One aim of the multi cell propagation model is to create a realistic pattern of handovers. However, this should be done with as little cells as possible in order to keep the simulation time of the complex network simulator within reasonable time limits. The current study uses a model with only three 3-sector sites, resulting in nine cells. Through the small size, the simulations run considerably fast, creating more results in the same time. It has been verified that this model is large enough to create a realistic pattern of handovers.

Previous works carried out in this area have used link-level simulator results to analytically map a given SNR (Signal Noise Ratio) and CQI (Channel Quality Indicator) to a block error probability in an Additive White Gaussian Noise channel (see [5,6]). Overall, a delay of 3 TTIs (Transmission Time Interval, 2 ms) is assumed between the CQI measurement at the user and the use of this control information at the Node B.

Position updates take place every 90 ms. With a typical handover execution time of round 100 ms, setting the update interval much smaller makes little sense. On a position update, the model uses the new position to calculate the distance loss. The pre-calculated map contains distance loss values for the nine cells, including an antenna gain. Next, the shadowing adds to the distance loss. The model assumes a single shadowing value per 3-sector site. The shadowing models as a distance correlated stochastic process with a lognormal distribution, assuming a standard deviation of 5 dB and a correlation distance of 40 m.

In order to feed the actual physical layer model, the model adds multi-path to the distance loss and shadowing. The multi-path fading is in line with standard UMTS models. The model is a correlated stochastic process for which the correlation depends on the UE speed. The multi-path component uses a tapped



delay line model, including inter-symbol interference. The number of calculations in the model prevents in-line calculation of the multi-path model. Instead, the value is obtained from a trace. The trace is sufficiently long: at least 2000 seconds, corresponding to 1 million samples. Each UE cell combination has a different starting point in the trace.

In the mixed traffic scenario, described in this study, both voice and data traffic are roughly one half of the downlink traffic expressed in offered data rate. The modelled voice traffic is 12.2 kbps AMR. Taking into account robust header compressing, silence detection, an activity factor of 50% and a packet interarrival time of 20 ms, the VoIP traffic is modeled with a PDU (Packet Data Unit) size of 37 bytes by means of a continuous UDP (User Datagram Protocol) stream. The BLER requirement for AMR is set to  $7 \cdot 10^{-3}$ . In order to avoid specific choices in the TCP protocol that may bias the analysis, the model offers the Best Effort traffic as a UDP/IP stream. Randomly divided over the nine cells, the system assumes 300 VoIP users and 20 Best Effort users. Each Best Effort user carries a UDP stream of 118 kbps, which is 15 times the traffic load of a VoIP user.

Table 1 collects some remaining parameter settings. The fixed 70 ms delay from transmitter to the Node B assumes a voice connection with a UE at the far end, where the far-end UE uses a CS voice connection. Scheduling at the Node-B depends on many factors, including the scheduling algorithm, the channel condition, the traffic load, HS-DSCH power etc. When many VoIP packets are waiting, the availability of HS-SCCHs (High-Speed Shared Control Channels) also becomes important. In order to prevent severely delayed packets from consuming capacity, a discard timer is set. The Iub flow control takes place at intervals of 10 ms.

**Table 1.** System parameter settings

| Parameter                                 | Value               |
|-------------------------------------------|---------------------|
| MAC-d PDU size                            | 296 bits (37 bytes) |
| Fixed delay from transmitter to Node B    | 70 ms               |
| Maximum number of H-ARQ (re)transmissions | 3                   |
| Number of HS-SCCHs                        | 4                   |
| Number of parallel H-ARQ processes        | 8                   |
| Number of HS-PDSCHs (codes)               | 8                   |
| Discard time of PDUs in MAC-hs            | 200 ms              |
| Flow control interval                     | 10 ms               |

### 3 Analysis of the Default Scenario

In order to examine the impact of the system parameters that are varied in Section 4, some results from the default parameter settings are collected in this section. All results described in this study are based on simulations of 50 seconds. This corresponds to 225000 nearly independent TTI-cell instances at which

the scheduler can allocate traffic to 4 HS-SCCHs, resulting in almost 1000000 scheduling opportunities in total. This figure guarantees sufficient convergence of statistical results to draw reliable conclusions.

Since the main focus of this study is on VoIP traffic, we have used a Round Robin scheduler. Each TTI the scheduler starts with the transmission queue whose packet at the head of the queue has been waiting longest. The scheduler allocates the maximum number of packets that can be served based on the channel condition of this link and the remaining resources at the Node B. The same procedure is repeated for the next queue selected. In this scheduling, no priority is given to VoIP packets. The time-based character of the Round-Robin scheduler already implies that the VoIP packets have a higher chance to be selected for transmission than the Best Effort data streams because of the fifteen times higher load. The results from the current study may also be compared to findings from earlier EURANE studies on the performance of scheduling algorithms in scenarios with only Best Effort traffic [8,7].

As each VoIP stream only contains one or a few PDUs, the scheduler splits the system resources over multiple users, resulting in the use of multiple HS-SCCHs, see Figure 2.a. The fact that the number of HS-SCCHs in use can be less than four is due to at least two reasons. Since the load is not constant, there are situations where not enough different users have PDUs waiting at the Node-B. Moreover, there is a realistic probability that the available power runs out.

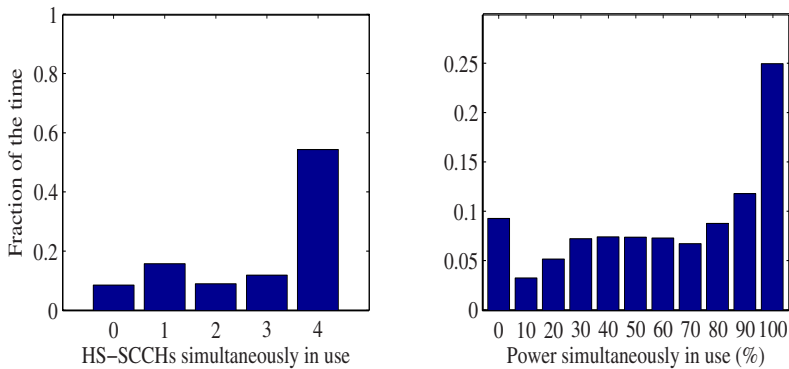


**Fig. 2.** Resource consumption in the default scenario (per TTI; per cell): a) number of HS-SCCHs; b) percentage of HS-DSCH power

As each individual power allocation is a result of propagation conditions, queue size and available power, the result displayed in Figure 2.b is not that straightforward. One fourth of the time (nearly) all HS-DSCH power is allocated. This peak at 100% represents cases where the scheduler can efficiently distribute the available power to queues with (many) packets waiting. The peak that is slightly visible around 30-40% of the power is due to the cases where the HS-SCCHs are occupied with VoIP users. Concluding, most PDUs only require a

fraction of the total available power. It is clear from both plots that during 8% of the time, nothing is scheduled. This number implies that the overall traffic load imposed on the system is good, taking into account that the users are moving and are not always divided equally over all cells.

In Table 2 the BLER values for the H-ARQ process have been collected. At the high user velocity considered in this paper, the CQI reported from the UE to the Node B will be very outdated. The user meanwhile has moved about 20 cm, which is more than the fast fading correlation distance. As a result, the BLER will definitely exceed the target value of 10%. This explains the high value for transmission 1. Due to the rapidly changing channel conditions, the second and third H-ARQ transmissions have a much higher chance of success. Finally, note that the residual BLER for the H-ARQ process, as a whole, is 0.17%. This is well below the AMR target of 0.7% for VoIP packets. However, this is not the case until after the third transmission. So, H-ARQ adds a considerable delay of up to 24 ms (2 periods of 6 TTIs each, resulting in  $2 \cdot 6 \cdot 2 \text{ ms} = 24 \text{ ms}$ ), as only after the third transmission the residual BLER is below the required packet loss ratio.

**Table 2.** BLER values for the H-ARQ process

| Transmission | Conditional BLER | Residual BLER | Extra delay |
|--------------|------------------|---------------|-------------|
| 1            | 29.0%            | 29.0%         | 0 ms        |
| 2            | 8.0%             | 2.3%          | 6 ms        |
| 3            | 7.3%             | 0.17%         | 12 ms       |

## 4 Optimization of Handover Parameters

This section analyzes the effect of the handover threshold, the handover delay and the discard timer, see Table 3. These parameters are illustrated in the following definition of the handover process (see also Figure 1):

### Definition (Handover Process)

*if  $signal_{target}(t) > signal_{source}(t) + HOT$ , then trigger the handover procedure:*  
 release the old Iub link, and create the new Iub link at time  $t$ ,  
 release the old Uu link, and create the new Uu link at time  $t + HOD$ .

**Table 3.** Handover parameters that are varied during this study

| Parameter          | Acronym | Default value | Variation            |
|--------------------|---------|---------------|----------------------|
| Handover Threshold | HOT     | 6 dB          | 5, 6, 7 dB           |
| Handover Delay     | HOD     | 100 ms        | 70, 100, 130, 160 ms |
| Discard Timer      | DT      | 200 ms        | 75, 100, 150, 200 ms |

### 4.1 Handover Threshold

The handover algorithm measures the SNR of the common pilot channel of each cell. When the difference between the SNR (Signal to Noise Ratio) of the current cell and the best cell exceeds the threshold, the RNC (Radio Network Controller) initiates the handover of the HSDPA connection [3], in 3GPP referred to as re-pointing.

The packets that arrive to the RNC during a handover are already forwarded to the new cell and therefore experience extra delay while waiting in the new cell. Higher handover intensities increase the chance of a user being in handover, and therefore create higher probabilities for packets to arrive during the handover. The analysis varies the handover threshold between 5 and 7 dB.

Table 4 shows the resulting handover intensity expressed as the number of handovers per UE per hour. The table splits the handovers into inter- and intra-Node B. The intra-Node B cases allow for a redirection of the scheduling queue at the Node B to the new cell, if the Node B implementation can handle it. As a reference, the bottom line shows the intensity of the border crossing of geographical hexagonal cells. This is the intensity when no fading would exist. As the model assumes that the shadowing for all cells of a single Node B is the same, the intensity for the intra Node B is close to the theoretical value. The reason that it is even lower is that some intra Node B handovers are replaced by a combination of two inter Node B handovers.

**Table 4.** Handover intensity related to handover threshold

| Handover<br>threshold (dB)     | Intensity (# handovers/UE/hour) |              |       | time ratio of being<br>in a handover |
|--------------------------------|---------------------------------|--------------|-------|--------------------------------------|
|                                | Inter Node B                    | Intra Node B | Total |                                      |
| 5                              | 891                             | 71           | 962   | 2.7%                                 |
| 6                              | 741                             | 70           | 811   | 2.3%                                 |
| 7                              | 626                             | 71           | 697   | 1.9%                                 |
| Geographic<br>border crossings | 176                             | 88           | 265   | 0.7%                                 |

Furthermore, Table 4 shows that for realistic handover thresholds the fraction of packets that arrive during the handover delay of 100 ms is round 2.5%. The performance of the VoIP connection depends on the delay that at most 0.7% of the packets do not reach. This includes lost packets. So the delay of the packets arriving during the handover plays a measurable role in the overall performance.

Next we consider the transmission quality for the different handover thresholds in Table 5. Due to the RR scheduler, the quality for VoIP is significantly different from the performance of the Best Effort traffic. An other scheduler will give a different balance. For both types of traffic the non-scheduled fraction of the MAC-d PDUs increases with the handover threshold, while the residual BLER of the H-ARQ remains the same. The difference in residual BLER between both types of traffic is due to the difference in TBS that is used. As the Best Effort traffic produces more traffic per time instance, the traffic is packed into larger

TBSs. Larger TBSs have a larger residual BLER after  $N$  transmissions. While for VoIP the residual BLER is the dominant contributor to the end-to-end Packet Error Rate (PER), the number of non-scheduled PDUs dominates for the Best Effort traffic.

**Table 5.** Transmission quality related to handover threshold for the VoIP and Best Effort traffic

| Handover threshold (dB) | VoIP               |                     |                     | Best Effort        |                     |                |
|-------------------------|--------------------|---------------------|---------------------|--------------------|---------------------|----------------|
|                         | PDUs not scheduled | H-ARQ residual BLER | End-to-end PER      | PDUs not scheduled | H-ARQ residual BLER | End-to-end PER |
| 5                       | $1 \cdot 10^{-5}$  | $1.5 \cdot 10^{-3}$ | $1.5 \cdot 10^{-3}$ | 6.4 %              | $3.9 \cdot 10^{-3}$ | 6.8 %          |
| 6                       | $3 \cdot 10^{-5}$  | $1.5 \cdot 10^{-3}$ | $1.5 \cdot 10^{-3}$ | 7.5 %              | $3.7 \cdot 10^{-3}$ | 7.9 %          |
| 7                       | $12 \cdot 10^{-5}$ | $1.6 \cdot 10^{-3}$ | $1.7 \cdot 10^{-3}$ | 9.1 %              | $3.8 \cdot 10^{-3}$ | 9.4 %          |

For VoIP traffic, the delay decreases for an increasing handover threshold, while the opposite is true for the Best Effort traffic. This shows the balance between two forces in the system:

1. An increase in handover threshold reduces the chance of an unnecessary handover pair triggered by a temporally extreme shadow fading contribution (while the geographic border crossing has not been reached yet)
2. An increase in handover threshold causes power to become scarcer, resulting in a larger probability that the lack of this resource introduces a delay.

In the system under study, the first force dominates for VoIP, while the second force dominates for the Best Effort traffic. Overall, executing a handover to save power tends to prevail over avoiding handovers in order to limit the probability a packet arrives during a handover.

## 4.2 Handover Delay

The handover delay, as discussed here, is the time between the moment the RNC stops transmission over the Iub to the old cell, until the first PDU can be scheduled from the new cell, which is a model of the handover process described in 3GPP [4]. There are two opposite effects in the handover delay. A longer delay allows for more PDUs to be scheduled, emptying the queues of the old cell. A shorter delay decreases the probability that a packet arrives during this time (see Table 6). In addition, a shorter delay will decrease the average time such a packet has to wait until the new cell can start scheduling it.

Figure 3 shows the complementary CDF of the end-to-end delay for various handover delays. The VOIP curves for all values of the handover delay display distinct kinks, indicating a significant increase in end-to-end delay. This percentage level is strongly correlated to the amount of VOIP packets that encounter a handover process, as shown in Table 5. It should also be noted that the horizontal distances between all four VoIP curves at the lower right hand side of

**Table 6.** Probability that packets arrive during the handover process

| Handover Delay                               | 70 ms | 100 ms | 130 ms | 160 ms |
|----------------------------------------------|-------|--------|--------|--------|
| Packets arriving during the handover process | 1.6 % | 2.3 %  | 3.0 %  | 3.7 %  |

the figure are separated by 30 ms, corresponding to the difference in subsequent values of the handover delay *i.e.*, the delay of these packets is large, purely due to the handover process. The bending of all curves for Best Effort traffic takes place at a end-to-end delay of around 120 ms. This is mainly due to the fact that the scheduler sorts the transmissions queues, according to the time the scheduler served each queue last, while each Best Effort user carries 15 times the traffic of a VoIP user. Finally it can be noted that the minimum of all end-to-end delay curves indeed is the expected fixed delay of 70 ms, see Table 1.

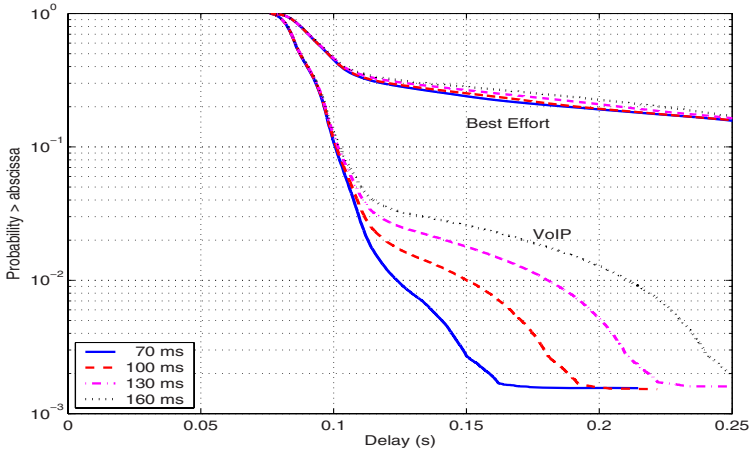


**Fig. 3.** Complementary CDF of the end-to-end delay for the mixed traffic scenario for varying handover delays

Figure 4 shows the complementary CDF of the scheduling delay, *i.e.*, the time a packet has been waiting at the Node B (case of a 70 ms handover delay). The figure makes a distinction between PDUs arriving at the RNC during a handover, and packets arriving when no handover takes place. The packets arriving during the handover delay have to wait at least until the handover process completes. The packets that, during the handover (HO) process, have been redirected to the *new* cell, and experience an extra delay since the Uu connection has not been set up yet, are marked with 'In HO', while all others are marked with 'Out of HO'. The current setting of the handover delay (70 ms) results in an average extra delay of at least 35 ms. Directly after the handover, the packets are sent as soon as possible, resulting in a steep curve, in particular for VoIP.

Figure 4 shows an additional effect for the Best Effort traffic. Directly *after* a handover, the probability of a large delay is significantly lower than during

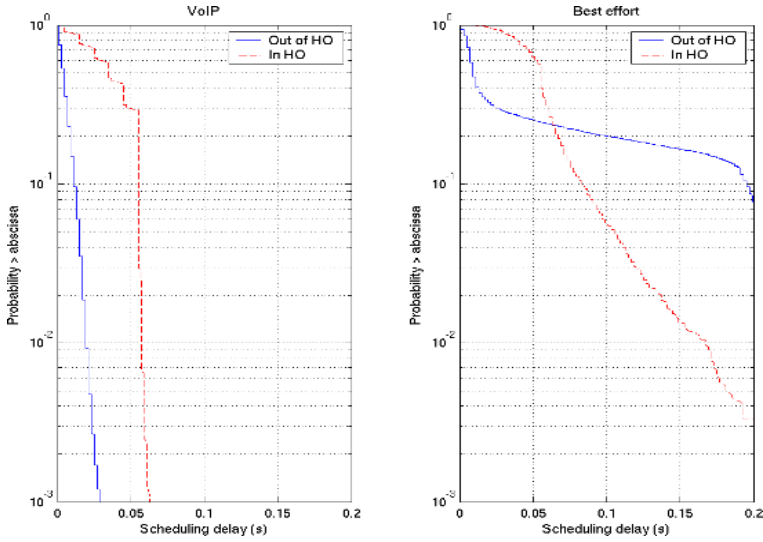


**Fig. 4.** Complementary CDF of the scheduling delay for the mixed traffic scenario for a handover delay of 70 ms a) VoIP traffic; b) Best Effort traffic

the *average* situation. During the average situation, the queues at the Node B can fill-up due to lack of resources. The amount of packets that are generated during a handover process is rather small, so the queue build-up at the new cell is limited and much smaller than the queues of a cell that has already been up and running for a while.

Considering the intersection between the two curves in Figure 4.b, about 25% of *all* Best Effort packets experience a scheduling delay of more than 70 ms. So, this value of the handover delay (70 ms) is considered too short to empty the source cell, and will therefore result in a significant loss of Best Effort packets. As the RLC will typically transmit Best Effort packets in AM (Acknowledged Mode), packets arriving at the UE via the new cell will trigger the RLC to retransmit the packets discarded from the Node B, causing a limited additional delay. Figure 3 shows that the shorter handover delay, similarly as for VoIP, is also better for the Best Effort traffic with its long delays. The handover delay is preferably set in the order of 50 - 100 ms.

### 4.3 Discard Timer

The Node-B contains a discard timer that discards packets waiting too long. The residual BLER of the H-ARQ process is likely to be in the order of  $1 \cdot 10^{-3} - 2 \cdot 10^{-3}$ . Moreover, AMR requires a PER of less than  $7 \cdot 10^{-3}$  for the A field. Combining these two values, up to  $5 \cdot 10^{-3}$  of the VoIP packets could be discarded in the Node-B without violating this requirement.

For Best Effort traffic, the situation is different, as it is persistent traffic. The RLC will retransmit packets that have been discarded by the Node-B. In general,

discarding packets does not change the traffic load. It however removes the peaks, making room for other traffic. Figure 5 shows the complementary CDF for the end-to-end delay for varying values of the discard timer. Considering the curves for VoIP, the only effect of decreasing the discard timer is an increase of the error floor. Discarding late packets does not provide a gain, only a risk when setting it too tight. Decreasing the discard timer also increases the error floor for the Best Effort traffic. It however also decreases the delay for the majority of the packets. The discard timer relates to the handover delay, and the typical scheduling delay. Here the handover delay is set to the default value of 100 ms. The majority of the packets have a delay of at most 100 ms. So a reasonable setting for the discard timer would be 100 ms.

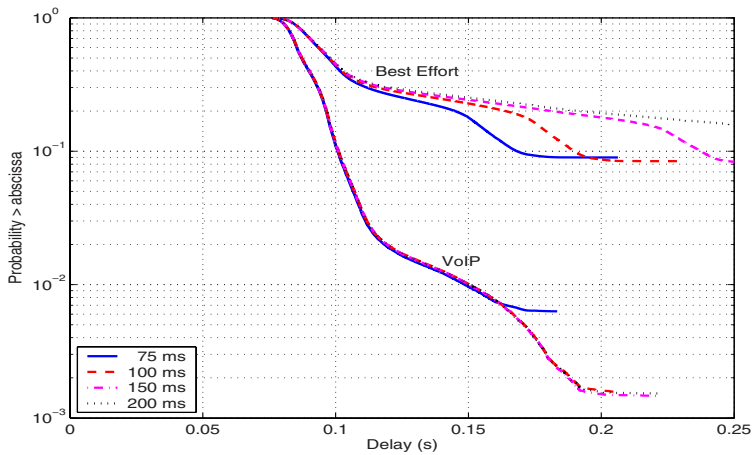


**Fig. 5.** Complementary CDF of the end-to-end delay for the mixed traffic scenario for varying discard timers

## 5 Conclusions

In this paper, we have analysed the performance of VoIP and Best Effort traffic over HSDPA in a multi-cell scenario with users moving at high velocity. As VoIP is a real-time service the analysis considers delay of packets as the key performance indicator. It turns out that VoIP can be carried by the HS-DSCH effectively. H-ARQ adds a considerable delay (24 ms), as only after the third transmission the residual BLER is below the required packet loss ratio. Of all system resources, power is the main bottleneck when the Best Effort data is scheduled, while the multiplexing of VoIP packets is confined by the four HS-SCCH channels restriction. The prescribed AMR of 0.7% for VoIP is achievable as long as the handover delay is chosen carefully. Due to the high velocity of the users, the scenario described in this paper is considered a worst case. Users moving at lower speed will encounter less handovers, which implies a lower packet loss.



**Acknowledgment.** The foundation of the EURANE simulator has been laid in the IST project IST-2001-34900 SEACORN (Simulation of Enhanced UMTS Access and Core Networks). The simulator is used worldwide throughout academia and industry and has regularly been extended with many HSDPA features. Part of the results presented here reflect the implementation work performed in the international ITEA Easy wireless IP 03008 project.

## References

1. Simulation Results on VoIP Performance over HSDPA, *3GPP TSG-RAN WG2*, R2-052832, November 2005.
2. Technical Specification Group Radio Access Network; Physical layer procedures (FDD), 3GPP TS 25.214 V6.6.0, June 2005.
3. Support of RT services over HSDPA, 3GPP TSG-RAN RP-050106, March 2005.
4. Improved HSDPA Re-pointing Procedure, 3GPP TSG-RAN WG2 R2-061298, 2006.
5. F. Brouwer e.a., Usage of Link Level Performance indicators for HSDPA Network-Level Simulations in E-UMTS, *ISSSTA*, Sydney, 2004.
6. I. de Bruin e.a., Fair Channel-Dependent Scheduling in CDMA Systems, *12th IST Summit on Mobile and Wireless Communications*, pp. 737-741, Aveiro, Portugal, June 2003.
7. I.C.C. de Bruin, Network-level Simulation Results of Fair Channel-Dependent Scheduling in Enhanced UMTS, *IFIP-TC6 Ninth International Conference on Personal Wireless Communications*, Delft, 2004.
8. I. de Bruin e.a., Performance Analysis of Hybrid-ARQ Characteristics in HSDPA, to appear in *Wireless Personal Communications*, 2007.
9. M. Ericson et al., Providing Reliable and Efficient VoIP over WCDMA, *Ericsson Review*, No. 2, 2005.
10. EURANE (Enhanced UMTS Radio Access Networks Extensions for ns-2), <http://www.ti-wmc.nl/eurane>.
11. H. Holma & A. Toskala, *HSDPA/HSUPA for UMTS*, John Wiley & Sons, 2006.
12. K.I. Pedersen, P.E. Mogensen, T.E. Kolding, Overview of QoS Options for HSDPA, *IEEE Communications Magazine*, Vol. 44, No. 7, pp. 100-105, July 2006.

# Feasibility of Supporting Real-Time Traffic in DiffServ Architecture

Jinoo Joung

Sangmyung University, Seoul, Korea  
jjoung@smu.ac.kr

**Abstract.** Since Integrated Services architecture is not scalable, it seems the only solutions for Quality of Service (QoS) architecture in the Internet are Differentiated Services (DiffServ) or its variations. It is generally understood that networks with DiffServ architectures can guarantee the end-to-end delay for packets of the highest priority class, only in lightly utilized networks. We show that, in networks without loops, the delay bounds for highest priority packets exist regardless of the level of network utilization with DiffServ architecture. These bounds are quadratically proportional to the maximum hop counts in heavily utilized networks; and are linearly proportional to the maximum hop counts in lightly utilized networks. We argue that, based on the analysis of these delay bounds in realistic situations, DiffServ architecture is able to support real time applications even in large networks. Considering that loop-free networks, especially the Ethernet networks, are being adopted more than ever for access networks and for provider networks as well, this conclusion is quite encouraging. Throughout the paper we use Latency-Rate ( $\mathcal{LR}$ ) server model, with which it has been proved that FIFO and Strict Priority schedulers are  $\mathcal{LR}$  servers to each *flows* in certain conditions.

## 1 Introduction

IETF has defined two services on IP networks which are collectively called Integrated services: the Controlled Load service and the Guaranteed Rate service [1,2]. The Controlled Load service defines a service that approximates the behavior of best-effort service under lightly utilized networks. The Guaranteed Rate service, which we will refer as IntServ in this paper, guarantees the end-to-end QoS by means of reserving, allocating and providing an amount of predefined resource to each data traffic unit, which often is called a flow or a session, in every server. Let alone the resource reservation, managing flows in a network node means a lot of works. This complexity inhibits IntServ-type QoS architectures from being adopted in real networks. DiffServ [3] is another approach that has been proposed to solve the scalability problem of IntServ. It classifies packets, or the flows they belong, into a number of traffic classes. The packets are marked accordingly at the edge of a network. Therefore the hard works are necessary only at the edge nodes. Classes may be assigned with strict priorities, or a certain

amount of bandwidth is provisioned for each class, as was the case with a flow in IntServ. With the support from a proper signaling scheme, DiffServ is an overly simplified version of IntServ, where many flows are aggregated into a single *class*, which is another name for an aggregated flow, and treated as a whole. We consider networks with DiffServ architecture, especially the QoS characteristics of the highest priority class traffic with the strict priority scheduling scheme. We focus on the queueing and scheduling behaviors of the flows or aggregated flows, and investigate the delay characteristics of them.

QoS characteristics of the network with IntServ architecture have been well studied and understood by numerous researches in the past decade. Providing the allocated bandwidths, or service rates, or simply rates of an output link to multiple sharing flows plays a key role in this approach. A myriad of scheduling algorithms have been proposed. The Packetized Generalized Processor Sharing (PGPS) [4] and Deficit Round Robin (DRR) [5], and many other rate-providing servers are proved to be a Latency-Rate server [6], or simply  $\mathcal{LR}$  server. All the work-conserving servers that guarantee rates can be modeled as  $\mathcal{LR}$  servers. The behavior of an  $\mathcal{LR}$  server is determined by two parameters, the latency and the allocated rate. The latency of an  $\mathcal{LR}$  server may be considered as the worst-case delay seen by the first packet of the busy period of a flow. It was shown that the maximum end-to-end delay experienced by a packet in a network of  $\mathcal{LR}$  servers can be calculated from only the latencies of the individual servers on the path of the flow, and the traffic parameters of the flow that generated the packet. More specifically for a leaky-bucket constrained flow,

$$D_i \leq \frac{\sigma_i - L_i}{\rho_i} + \sum_{j=1}^N \Theta_i^{S_j}, \quad (1)$$

where  $D_i$  is the delay of flow  $i$  within a network,  $\sigma_i$  and  $\rho_i$  are the well known leaky bucket parameters, the maximum burst size and the average rate, respectively,  $L_i$  is the maximum packet length and  $\Theta_i^{S_j}$  is the latency of flow  $i$  at the server  $S_j$ . In networks with DiffServ architecture, packets in a same class are enqueued to a single queue and scheduled in a FIFO manner within the class. The higher priority class may be served with a strict priority over the lower classes, or a certain amount of bandwidth may be assigned to each class. It is proved that FIFO schedulers and strict priority schedulers used in this network is also an  $\mathcal{LR}$  server to the individual flows within a class, under conditions that every flow conforms to leaky-bucket model and the aggregated rate is less than or equal to the link capacity [7]. We will apply this result with the  $\mathcal{LR}$  server model to DiffServ networks for further analysis.

The current belief on such networks is that only with the low enough network utilization, one can guarantee delay bounds. Otherwise it is not certain that the bounds exist, or if they do they *explode* to infinity with just sufficient number of hops or the size of the network [8,9]. Based on this argument, there have been a trend of aborting DiffServ for delay sensitive real-time applications such as Voice over IP (VoIP) [10,11,12,13,14,15,16]. One notable research direction is to aggregate flows selectively [10,11] so that effective compromises are achieved in the

area between the extreme points with the completely flow-state aware IntServ and the unaware DiffServ. A series of implementation practices in this regard are being proposed, including flow aggregation using Multi-Protocol Label Switching (MPLS) framework [12,13]. Another direction is aligned with the argument that achieving absolute performance guarantee is hard with conventional DiffServ so that only a *relative* differentiation is meaningful [14]. Some even go further by arguing that in a core network, traditional approach of requesting, reserving, and allocating a certain rate to a flow or the flow aggregates (e.g. a class in DiffServ) is too burdensome and inefficient so that the flows should not explicitly request for a service, but rather they should be implicitly detected by the network and treated accordingly [15,16]. The exploding nature of the delay bounds that has stimulated such diverse research activities, however, is the conclusion with general network topology. If we can somehow avoid the burst size accumulation due to the loops formed in a network, which is suspected as the major reason for the explosion, the delay bounds may be still useful with the simple DiffServ architecture. We investigate this possibility throughout the paper. We focus on the tree networks where the loops are avoided. The tree networks can be found in many types of Local Area Network (LAN), especially Ethernet network. Ethernet networks are gaining momentum to be extensively used more and more in access networks as well as provider networks [17]. In a large scale, these networks together form a tree topology, by means of spanning tree algorithms or manual configurations [18,19].

In the next section, a brief review on delay bounds with DiffServ architecture is given. In the third section, we focus on tree networks, and obtain the closed formula for delay bounds. We discuss our result in section 4. The conclusion is given in the final section.

## 2 Iterative Calculation for Delay Bounds in Networks with DiffServ Architecture

Under a condition that there is no flow that violates the leaky bucket constraint specified for itself, we can guarantee the delay upper bound for each flows, even with relatively simple scheduling strategies. If there is only a single class of flows, which specifies for their required bandwidths then First-in-first-out (FIFO) servers can guarantee the delay bounds. If in the network there is best-effort traffic that does not specify its bandwidth, the minimum protection against such best-effort traffic is necessary. In this regard the Strict Priority (SP) scheduling can guarantee delay bounds for flows of the real-time traffic. A Strict Priority (SP) server is a server that maintains at least two queues. The queue with highest priority, which is for the flows with real-time constraints, transmits packets whenever it has one, right after the completion of the current packet that is being served.

In this environment, the sum of all flows that want to pass through a server is compared with the link capacity, and if it's less than the link capacity then

delay upper bounds will be prescribed, as it can be calculated with the method explained in this section. We consider a network of packet level servers.

**Theorem 1.** *A FIFO server or an SP server, under conditions that all the input flows are leaky bucket constrained and the sum of average rates is less than the link capacity, is an  $\mathcal{LR}$  server for individual flows with latency given as the following:*

$$\Theta_i^S = \frac{\sigma^S - \sigma_i^S}{r^S} + \Theta^S, \quad (2)$$

where  $\sigma^S$  is the sum of all the  $\sigma_i^S$  within the server  $\mathcal{S}$ ,  $r^S$  is the link capacity of  $\mathcal{S}$ , and

$$\Theta^S = \begin{cases} L/r^S & \text{when } \mathcal{S} \text{ is FIFO} \\ 2L/r^S & \text{when } \mathcal{S} \text{ is SP.} \end{cases}$$

*Proof.* See the proof in section 3 of [7]. □

**Corollary 1.** *The output traffic of flow  $i$  from a FIFO server or an SP server  $\mathcal{S}$  conforms to the leaky bucket model with parameters  $(\sigma_i^S + \rho_i \Theta_i^S, \rho_i)$ , where  $\sigma_i^S$  is the maximum burst size of flow  $i$  into the server  $\mathcal{S}$ .*

The end-to-end delay of a network with DiffServ architecture with FIFO servers and/or Strict Priority servers can be obtained by the following sets of equations:

$$\begin{aligned} D_i &\leq \frac{\sigma_i - L_i}{\rho_i} + \sum_{n=1}^N \Theta_i^{I_n}, \\ \Theta_i^{I_n} &= \frac{\sigma^{I_n} - \sigma_i^{I_n}}{r^{I_n}} + \Theta^{I_n}, \\ \sigma_i^{I_{n+1}} &= \sigma_i^{I_n} + \rho_i \Theta_i^{I_n}, \text{ for } n \geq 1, \end{aligned} \quad (3)$$

where  $I_n$  is the  $n$ th server of the network for  $i$ ,  $N$  is the number of servers that  $i$  traverses in the network; and  $\sigma_i^{I_1} = \sigma_i$ .

### 3 Closed-Form Delay Bound in Tree Network

While (3) gives a tight bound through iterative computation of the latencies and the maximum burst sizes of each server, we still have to make assumptions about the burst sizes of other flows. With a reasonable restriction on the network topology, the other flows' burst sizes can be inferred, and the delay bound for a whole network can be obtained. In this section we consider delay bounds in tree networks, which are defined to be acyclic connected graphs. Tree networks appear in a broad range of networks. For example in Ethernet, both in LAN and in wide area networks, the logical tree-based network topology is achieved by running Spanning Tree Protocol (STP) or static configuration (e.g. configuring Virtual LANs) [18,19].

Let us define *hop* by the server and the accompanying link through which packets are queued and serviced and then transmitted. First we start by observing an important property of tree networks. Let  $i$  be the flow under observation.

**Lemma 1.** *Consider a tree network with the flow under observation,  $i$ . Assume that the flow  $i$  traverses the path of the maximum possible hop counts. Then at any server in the  $i$ 's path, other flows that are confronted by  $i$  have traversed less number of hops than  $i$  just have.*

*Proof.* Let us denote the  $i$ 's path by  $(I_1, I_2, \dots, I_H)$ , where  $I_n$  is the  $n$ th server in the  $i$ 's path and  $H$  is the maximum number of hops possible in the given tree network. Similarly let us denote another flow  $j$ 's path by  $(J_1, J_2, \dots, J_m)$ , where  $m \leq H$ . Let  $I_k = J_l$ , for some  $k$  and  $l$  where  $1 \leq k \leq H$  and  $1 \leq l \leq m$ , that is the flow  $i$  and  $j$  confront each other at a server in the path. A path is defined to be a sequence of nodes, with no repeated nodes, in which each adjacent node pair is linked. We will show that  $k \geq l$  for any case, by contradiction. Assume  $k < l$ .

**Case 1:**  $\{J_1, J_2, \dots, J_{l-1}\}$  is disjoint with  $\{I_{k+1}, \dots, I_H\}$ . This is to say that there is no server that takes part of both the remaining path of  $i$  and the traveled path of  $j$ . Then the path  $(J_1, J_2, \dots, J_{l-1}, I_k, I_{k+1}, \dots, I_H)$  exists that has more hops than  $H$ . This contradicts to the assumption that  $H$  is the maximum possible hop counts.

**Case 2:** There is at least one server in the remaining path of  $i$ ,  $(I_{k+1}, \dots, I_H)$ , that is also a server in the traveled path of  $j$ ,  $(J_1, \dots, J_{l-1})$ . Let us call this server  $J_p$ . Then  $(J_p, J_{p+1}, \dots, J_{l-1}, I_k, I_{k+1}, \dots, J_p)$  forms a cycle, which contradicts the assumption that the network is a tree.

In both cases the statement  $k < l$  contradicts the assumption. Therefore  $k \geq l$  for any case and the lemma follows.  $\square$

Lemma 1 lets us infer about the maximum burst sizes of the confronted flows in the path. Therefore the end-to-end delay bound can be obtained from a few network parameters.

Now consider a server  $I_n$  in the path of  $i$ . Let the set of flows  $\mathcal{F}_{I_n}$ , including  $i$ , are competing for service in  $I_n$ . For any flow  $j$ ,  $j \in \mathcal{F}_{I_n}$ , which has traveled  $(n-m-1)$  hops until reaching  $I_n$ , imagine a corresponding flow  $j'$  with  $m$  more hops from the starting node of the flow  $j$ . Moreover  $j'$  has entered the network with the same parameter with  $j$ . Further imagine that for additional hops for each flows, not the existing nodes but new nodes are attached to the starting nodes of each flows, so that the numbers of flows in the upstream nodes of  $I_n$  are intact. See figure 1. Now we have constructed an imaginary network, in which at  $I_n$ , the flows in  $\mathcal{F}_{I_n}$  all have traveled exactly  $(n-1)$  hops until reaching  $I_n$ . We claim the following.

**Lemma 2.** *The maximum burst size of any flow at the entrance of  $I_n$  in the constructed network is always greater than or equal to that of the original network. That is,*

$$\sigma_j^{I_n} \leq \sigma_{j'}^{I_n}, \quad (4)$$

for all  $j \in \mathcal{F}_{I_n}$ .

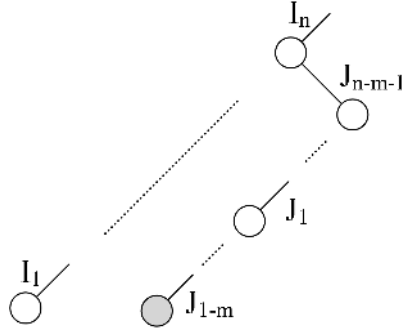


**Fig. 1.** Construction of an imaginary network. The shaded nodes are added ones.

*Proof.* It is enough to show that  $\sigma_j^{J1} \leq \sigma_{j'}^{J1}$  for any  $j, j \in \mathcal{F}_{In}$ . Since  $\sigma_j^{Jn}$  is a nondecreasing function of  $n$ ,

$$\sigma_{j'}^{J1} \geq \sigma_{j'}^{J(1-m)} = \sigma_j = \sigma_j^{J1}. \quad (5)$$

The lemma follows.  $\square$

We argue the main result of the paper as the following.

**Theorem 2.** *The end-to-end delay of a tree network with DiffServ architecture with FIFO servers is bounded by*

$$D_i \leq \frac{\sigma_i - L_i}{\rho_i} + \tau \frac{(1 + \alpha)^H - 1}{\alpha}, \quad (6)$$

where  $\tau$  and  $\alpha$  are defined as

$$\sum_{j \in \mathcal{F}_S} \sigma_j \leq \tau r^S, \quad \sum_{j \in \mathcal{F}_S} \rho_j \leq \alpha r^S, \quad (7)$$

for any server  $S$  in the network, in which there is a set of flows,  $\mathcal{F}_S$ .

The parameters  $\tau$  and  $\alpha$  is similarly defined in [8]. We will call  $\tau$  the *burst allowance level* measured in time for their transmission, and  $\alpha$  the *network utilization*. Note that  $0 < \alpha < 1$  in our network configurations.

*Proof.* First, we will show that the following inequalities hold for any  $I_n$ ,  $1 \leq n \leq H$ .

$$\begin{aligned} \Theta_i^{In} &\leq \tau(1 + \alpha)^{n-1}, \\ \sigma_i^{In} &\leq \sigma_i + \tau \rho_i \frac{(1 + \alpha)^{n-1} - 1}{\alpha}. \end{aligned} \quad (8)$$

Let us first assume that (8) is true for  $I_n$ . We will show that it holds for  $I_{n+1}$  as well, and then it holds for  $I_1$ , therefore it holds for any  $I_n$ .

If (8) is true for  $I_n$ , from (3) we get

$$\begin{aligned}\sigma_i^{In+1} &= \sigma_i^{In} + \rho_i \Theta_i^{In} \\ &\leq \sigma_i + \tau \rho_i ((1 + \alpha)^n - 1) / \alpha.\end{aligned}\quad (9)$$

From (9), and from lemma 2, the sum of the maximum burst sizes of all the incoming priority flows in  $I_{n+1}$ ,

$$\begin{aligned}\sigma^{In+1} &= \sum_{j \in \mathcal{F}_{In}} \sigma_j^{In+1} \leq \sum_{j' \in \mathcal{F}_{In}} \sigma_{j'}^{In+1} \\ &\leq \sum_{j'} \{ \sigma_j + \tau \rho_j ((1 + \alpha)^n - 1) / \alpha \},\end{aligned}\quad (10)$$

since any  $j, j \in \mathcal{F}_{In}$ , has traveled  $(n-1)$  hops, therefore  $\sigma_j^{In+1} = \sigma_j + \tau \rho_j ((1 + \alpha)^n - 1) / \alpha$  by the assumption at the beginning of the proof. We obtain

$$\sigma^{In+1} \leq \tau r^{In+1} (1 + \alpha)^n \quad (11)$$

from (7). Equation (3) yields

$$\Theta_i^{In+1} = \frac{\sigma^{In+1} - \sigma_i^{In+1}}{r^{In+1}} + \Theta^{In+1}. \quad (12)$$

Note that  $\Theta^{In+1}$  is  $L/r^{In+1}$  for a FIFO server. The maximum burst size of a flow, by definition, is always greater than or equal to the maximum packet length, that is  $\sigma_i^{In+1} \geq L$ . Therefore, and from (11),

$$\Theta_i^{In+1} \leq \frac{\sigma^{In+1}}{r^{In+1}} \leq \tau (1 + \alpha)^n. \quad (13)$$

With (9) and (13), we have shown that (8) holds for  $I_{n+1}$ . Now we'll consider the case for  $I_1$ . For  $I_1$ ,  $\sigma^{I1} = \sum_{j \in \mathcal{F}_{I1}} \sigma_j \leq \tau r^{I1}$ , and

$$\Theta_i^{I1} \leq \tau - \frac{\sigma_i}{r^{I1}} + \frac{L}{r^{I1}} \leq \tau, \quad (14)$$

which shows that (8) holds for  $I_1$  as well. From (3),

$$\begin{aligned}D_i &\leq \frac{\sigma_i - L_i}{\rho_i} + \sum_{n=1}^H \tau (1 + \alpha)^{n-1} \\ &\leq \frac{\sigma_i - L_i}{\rho_i} + \tau \sum_{n=0}^{H-1} (1 + \alpha)^n \\ &\leq \frac{\sigma_i - L_i}{\rho_i} + \tau \frac{(1 + \alpha)^H - 1}{\alpha},\end{aligned}\quad (15)$$

for  $H \geq 1$ . □

Similar conclusion can be claimed for the network with SP servers.



**Theorem 3.** *The end-to-end delay of a tree network with DiffServ architecture with SP servers is bounded by*

$$D_i \leq \frac{\sigma_i - L_i}{\rho_i} + \tau' \frac{(1 + \alpha)^H - 1}{\alpha}, \quad (16)$$

where  $\tau'$  and  $\alpha$  are defined as

$$L + \sum_{j \in \mathcal{F}_S} \sigma_j \leq \tau' r^S, \quad \sum_{j \in \mathcal{F}_S} \rho_j \leq \alpha r^S, \quad (17)$$

for any server  $S$  in the network, in which there is a set of flows,  $\mathcal{F}_S$ .

*Proof.* The proof of this theorem is exactly the same with that of the previous theorem except that  $\Theta^{In+1}$  in (12) is  $2L/r^{In+1}$  for an SP server, therefore (13) is still valid with the modified  $\tau'$ . We omit the details.  $\square$

## 4 Discussion

We first examine the extreme cases; in which  $\alpha \rightarrow 0$  or  $\alpha \rightarrow 1$ . When  $\alpha \rightarrow 0$ , the delay bound becomes  $(\sigma_i/\rho_i + H\tau')$ . For the worst case delay we can say that it is  $(\max_j(\sigma_j/\rho_j) + H\tau')$ . When  $\alpha \rightarrow 1$ , the delay bounds becomes  $(\max_j(\sigma_j/\rho_j) + (2^H - 1)\tau')$ . The delay bounds increases linearly as hop count increases, when the utilization is low. When the utilization is high, however, the delay bounds increases quadratically with hop counts.

The delay bounds in a general topology network with DiffServ architecture have been obtained in the literatures [8,9]. In [8], it was concluded that unless the link utilizations are kept under a certain level, the end-to-end delay explodes to infinity. The delay bound obtained in [8] is, only under condition that  $\alpha < 1/(H-1)$ ,

$$D \leq \frac{H}{1 - (H-1)\alpha} \tau', \quad (18)$$

for a case with infinite incoming links' capacity, which is also the case considered in this paper. Equation (18) becomes, as  $\alpha \rightarrow 0$ ,  $D \leq \tau' H$ . As it was already noted in [9], however, (18) has apparently not taken the non-preemptive nature of SP servers into consideration, therefore was later corrected in [9] to, under the same condition,

$$D \leq \frac{H}{1 - (H-1)\alpha} (\tau' + \max_j (L / \sum_j \rho_j)). \quad (19)$$

Equation (19) becomes, as  $\alpha \rightarrow 0$ ,  $D \leq (\tau' + \max_j (L / \sum_j \rho_j)) H$ . Table 1 summarizes the delay bounds obtained from this paper and from two previous works, with the suggested network parameters both in [8,9]. The reason that the bounds obtained in this paper is less than the one from [9] is twofold. The first and obvious reason is that our network is strictly confined in a tree topology. This

**Table 1.** The bounds from this paper and from the related works, in milliseconds:  $H=10$ ,  $\sigma_j=100$  bytes for all flows,  $\rho_i=32$ kbps for all flows,  $L=1500$  bytes,  $r^S = 149.760$ Mbps for all  $\mathcal{S}$

| $\alpha$      | 0.04  | 0.08   |
|---------------|-------|--------|
| Bound by (16) | 12.97 | 30.13  |
| Bound in [8]  | 16.88 | 74.29  |
| Bound in [9]  | 48.18 | 110.06 |

restriction inhibits the burst accumulation in loop, so that the bound still exists even in high network utilization. The second reason is that, even in the low network utilization cases, the bounds in previous works are simply the summation of nodal delays. Our bound, on the other hand, is from considering the whole network as a single virtual node, so that the correlation among consecutive nodes are taken into account. This characteristic is the primary virtue of the analysis based on the  $\mathcal{LR}$  server [6]. In the following table, we examine the bounds with varying  $H$ , the maximum number of hops, and with varying  $\alpha$ , the network utilization. As table 2 suggests, even in a tree network, delay bounds with moderate to high network utilization seems quite unacceptable. Since in numerous standardization bodies it is suggested that the end-to-end delay bound for voice to be less than 400ms [20], only with  $\alpha$  much less than 0.1 can only meet those standards, if network parameters in table 2 are to be used.

**Table 2.** The bounds obtained from (16), in seconds, with varying  $H$  and  $\alpha$ :  $\sigma_j = L = 1500$  bytes for all flows,  $\rho_i=32$ kbps for all flows,  $r^S = 149.760$ Mbps for all  $\mathcal{S}$

| $\alpha$ | 0.1   | 0.3   | 0.5    | 0.7     |
|----------|-------|-------|--------|---------|
| $H = 6$  | 0.290 | 1.436 | 3.898  | 8.679   |
| $H = 8$  | 0.430 | 2.686 | 9.240  | 25.792  |
| $H = 10$ | 0.599 | 4.798 | 21.258 | 75.245  |
| $H = 12$ | 0.804 | 8.368 | 48.301 | 218.175 |

Consider now where the maximum burst size, one of primary contributor to delay bound increment, and the maximum packet length can be controlled to be much less than the ordinary IP packet length. This assumption on the maximum packet length, therefore on the maximum burst size, is not extravagant since if we are to transmit the MPEG-2 Transport Streams (TS) data whose lengths are fixed at 188 bytes [21], with 12 bytes RTP fixed header [22], 4 bytes RTP video-specific header [23], 8 bytes UDP header, 20 bytes IP header and finally 26 bytes Ethernet header and trailer including preamble, then the maximum packet length in this case becomes 258 bytes. Considering the extended headers fields and Ethernet inter-frame gap, the maximum packet length will be about

**Table 3.** The bounds obtained from (16), in seconds, with varying  $H$  and  $\alpha$ :  $\sigma_j = L = 300$  bytes for all flows,  $\rho_i = 32$  kbps for all flows,  $r^S = 149.760$  Mbps for all  $S$

| $\alpha$ | 0.05   | 0.1    | 0.2   | 0.3   |
|----------|--------|--------|-------|-------|
| $H = 6$  | 0.0256 | 0.0580 | 0.149 | 0.287 |
| $H = 8$  | 0.0360 | 0.0859 | 0.248 | 0.537 |
| $H = 10$ | 0.0474 | 0.120  | 0.390 | 0.960 |
| $H = 12$ | 0.0600 | 0.161  | 0.594 | 1.674 |
| $H = 14$ | 0.0738 | 0.210  | 0.889 | 2.880 |
| $H = 16$ | 0.0891 | 0.270  | 1.313 | 4.919 |

300 bytes. In such a network of limited packet length, the 400ms requirement can be reasonably met as table 3 suggests. For example, if we manage to restrict the maximum packet length to be 300 bytes and send one packet at a time to the network (that is to restrict the maximum burst size at the entrance of the network), then at the 10% network utilization even a very large network with 16 hop counts can successfully support the voice applications. Considering that the applications requesting Premium Service in DiffServ are usually a small portion of the total traffic, table 3 suggests the DiffServ may indeed be useful in some cases.

## 5 Conclusion

In this paper we have investigated the end-to-end delay bounds of FIFO servers and Strict Priority servers in DiffServ architecture, especially in tree networks. Based on the observation that the FIFO and SP servers are  $\mathcal{LR}$  servers for each *micro flows* under the condition that all the flows in the high priority class conform to the arranged leaky bucket parameters and the sum of allocated rates is less than the link capacity for all the links, and on the established analysis technique with  $\mathcal{LR}$  servers, we suggested the iterative computation method to calculate the tight end-to-end delay bounds with FIFO or SP servers, i.e. in Expedited Forwarding DiffServ architecture. This iterative computation sequence, however, requires an assumption on the burst size increase on other flows. Concurrent iteration on every flow in the network may be possible. Instead, however, we focused on tree networks, a special type but widely used networks, and derived a closed formula for the delay bound.

Contrary to the traditional belief that without a very small network utilization delay bound goes to infinity, we have shown that in tree networks delay bounds always exist. We have shown also that this bound is linearly proportional to the number of hop counts when the utilization is small; and is quadratically proportional to the number of hop counts when the utilization is large. It may be argued, however, that the existing bound is not acceptable for moderate to large sized networks with moderate to large network utilization, even in tree networks. On the other hand, with a manipulation to the network configuration

such as maximum packet length restriction, we have shown that the DiffServ architecture can support the real-time application even in large networks with moderate network utilization.

Although the result we have obtained in this paper is applicable only to tree networks, in which loops are strictly avoided, the insight we gained that a tighter bound can be derived by taking into consideration the topology characteristics of a network, is still valid. Hence the future work may consist of the delay bound investigation on a broader range of network topology, for example *extended tree networks* that will include the shortest path tree which is the outcome of well-known link-state routing algorithms. These networks certainly have loops, but partially acyclic in the sense that each node maintains its own tree network to decide the shortest path to other nodes.

## References

1. R. Braden, D. Clark and S. Shenker, "Integrated Services in the Internet Architecture: an Overview," IETF RFC 1633, 1994.
2. P. P. White, "RSVP and Integrated services in the Internet: A tutorial," *IEEE Communications Mag.*, vol. 35, pp. 100–106, May 1997.
3. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for Differentiated Services," IETF RFC 2475, 1998.
4. 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, no. 3, June 1993.
5. M. Shreedhar and G. Varghese, "Efficient fair queueing using deficit round-robin," *IEEE/ACM Trans. Networking*, vol. 4, no. 3, pp. 375–385, June 1996.
6. D. Stiliadis and A. Varma, "Latency-Rate servers: A general model for analysis of traffic scheduling algorithms," *IEEE/ACM Trans. Networking*, vol. 6, no. 5, Oct. 1998.
7. J. Joung, B-S. Choe, H. Jeong, and H. Ryu: Effect of Flow Aggregation on the Maximum End-to-End Delay. Lecture Notes in Computer Science, Vol. 4208. Springer-Verlag (2006) 426–435
8. A. Chanry and J.-Y. LeBoudec, "Delay bounds in a network with aggregate scheduling," in *Proc. First International Workshop of Quality of Future Internet Services (QOFIS2000)*, 2000.
9. Y. Jiang, "Delay Bounds for a Network of Guaranteed Rate Servers with FIFO," *Computer Networks*, Elsevier Science, Vol. 40, No. 6, pp. 683–694, 2002.
10. J. A. Cobb, "Preserving quality of service guarantees in spite of flow aggregation," *IEEE/ACM Trans. on Networking*, vol. 10, no. 1, pp. 43–53, Feb. 2002.
11. W. Sun and K. G. Shin, "End-to-End Delay Bounds for Traffic Aggregates Under Guaranteed-Rate Scheduling Algorithms," *IEEE/ACM Trans. on Networking*, Vol. 13, No. 5, Oct. 2005.
12. F. Le Faucheur et al., "Multi-Protocol Label Switching (MPLS) Support of Differentiated Services," IETF RFC 3270, May 2002.
13. B. Rong, et al., "Modeling and Simulation of Traffic Aggregation Based SIP over MPLS Network Architecture," in *Proc. IEEE 38th Annual Simulation Symposium (ANSS05)*, 2005.

14. C. Dovrolis, D. Stiliadis, and P. Ramanathan, "Proportional Differentiated Services: Delay Differentiation and Packet Scheduling," *IEEE/ACM Trans. on Networking*, Vol. 10, No. 1, Feb. 2002.
15. A. Kortebe, S. Oueslati, and J. Roberts, "Cross-protect: implicit service differentiation and admission control," in *Proc. IEEE HPSR*, April 2004.
16. S. Oueslati and J. Roberts, "A new direction for quality of service: Flow-aware networking," in *Proc. Conference on Next Generation Internet Networks (NGI)*, April 2005.
17. "Layered Network Architecture and Implementation for Ethernet Services," White Paper, Fujitsu Network Communications, Inc., 2004.
18. "802.1ad Virtual Bridged Local Area Networks – Amendment 4: Provider Bridges," IEEE Higher Layer LAN Protocols Working Group (IEEE 802.1), Amendment to IEEE 802.1Q, May 2006.
19. "Metro Ethernet Service – A Technical Overview," White Paper, R. Santitoro, Metro Ethernet Forum, 2003. [Online]. Available: <http://www.metroethernetforum.org/metro-ethernet-services.pdf>
20. "Network performance objectives for IP-based services," International Telecommunication Union – Telecommunication Standardization Sector (ITU-T), Recommendation Y.1541, Feb. 2006.
21. "Information Technology – Generic Coding of Moving Pictures and Associated Audio Information Part 1: Systems," ISO/IEC International Standard IS 13818, November 1994.
22. H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," IETF RFC 3550, July 2003.
23. Hoffman, D., Fernando, G., Goyal, V. and M. Civanlar, "RTP Payload Format for MPEG1/MPEG2 Video," IETF RFC 2250, January 1998.

# Multi-rate Relaying for Performance Improvement in IEEE 802.11 WLANs

Laura Marie Feeney<sup>1</sup>, Bilge Cetin<sup>1</sup>, Daniel Hollos<sup>2</sup>, Martin Kubisch<sup>2</sup>,  
Seble Mengesha<sup>2</sup>, and Holger Karl<sup>3</sup>

<sup>1</sup> Swedish Institute of Computer Science, Kista, Sweden  
`{lmfeeney,bilge}@sics.se`

<sup>2</sup> Technical University Berlin, Germany  
`{hollos,kubisch,mengesha}@tkn.tu-berlin.de`

<sup>3</sup> University of Paderborn, Germany  
`holger.karl@uni-paderborn.de`

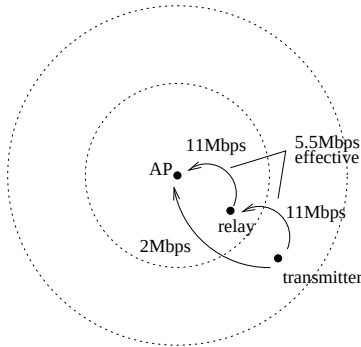
**Abstract.** It is well known that the presence of nodes using a low data transmit rate has a disproportionate impact on the performance of an IEEE 802.11 WLAN. ORP is an opportunistic relay protocol that allows nodes to increase their effective transmit rate by replacing a low data rate transmission with a two-hop sequence of shorter range, higher data rate transmissions, using an intermediate node as a relay. ORP differs from existing protocols in discovering relays experimentally, by optimistically making frames available for relaying. Relays identify themselves as suitable relays by forwarding these frames. This approach has several advantages compared with previously proposed relay protocols: Most importantly, ORP does not rely on observations of received signal strength to infer the availability of relay nodes and transmit rates. We present analytic and simulation results showing that ORP improves the throughput by up to 40% in a saturated IEEE 802.11b network.

**Keywords:** IEEE 802.11; cooperative communication protocols; rate adaptation; multi-hop wireless networks; wireless LAN.

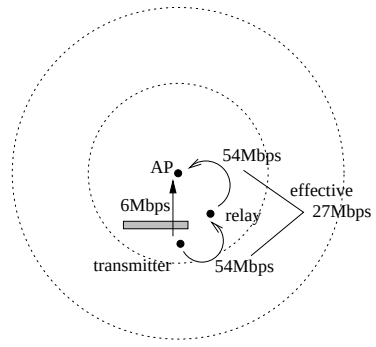
## 1 Introduction

The so-called IEEE 802.11 “performance anomaly” [1] implies that the presence of a node using a low data transmit rate significantly degrades the performance of an IEEE 802.11 BSS. Each node has an equal opportunity to access the channel, but nodes with a low data transmit rate occupy the channel for a disproportionately long time each time they transmit. The mean data transmit rate of the BSS – the harmonic mean of all the nodes’ transmit rates – is dominated by the lowest transmit rate.

In [2,3,4,5,6], it has been shown that the effective transmit rate of a node may be improved by replacing a single transmission at a low data transmit rate with a sequence of two higher data rate transmissions, via an intermediate relay node. Figure 1 shows a simple example of a 2 Mbps transmission being relayed



(a) Relaying increases the effective transmit rate



(b) Non geometric variation in transmit rate, due to obstacles

**Fig. 1.** Examples of relaying

using a sequence of two 11 Mbps transmissions, for an effective transmit rate of 5.5 Mbps (less some overhead).

ORP is an Opportunistic Relay Protocol for IEEE 802.11 WLANs. Its most significant advantage over previously proposed protocols is that it does not depend on observing other nodes' transmissions to infer relay availability, considerably simplifying implementation and evaluation. In addition, ORP largely preserves the sense of the IEEE 802.11 DCF and MAC headers, allowing ORP and non-ORP nodes to co-exist. Simulation results show that in an IEEE 802.11b BSS where all low transmit rate frames are potentially eligible for relaying, the overall throughput of the BSS increases about 40%.

In the sections that follow, we first define the basic uplink relay discovery mechanism of the ORP protocol and study its performance analytically. Then we extend the basic ORP mechanism to include both uplink and downlink transmission and examine its performance in simulation. Finally, we define the contribution of our work relative to existing relay protocols.

## 2 Basic ORP Uplink

The basic ORP uplink mechanism allows frames to be relayed from a node (the source) to the access point (AP). To discover a relay, the source optimistically makes a frame available for forwarding, using the duration field in the MAC header to protect the forwarding transmission. If an intermediate node successfully decodes the frame and believes that it can forward it to the AP within the time constraint implied by the duration value, it is a potential relay for the frame. Because there may be more than one potential relay for the frame, a short backoff is used to reduce the risk of relay collision. If exactly one potential relay forwards the frame, the AP sends an ACK directly to the source. Otherwise, the relay fails and the source must retransmit the frame.

Using the duration field to indicate the end-to-end transmit time preserves the sense of the IEEE 802.11 MAC header, making relaying largely transparent to

non-ORP nodes. ORP relaying also preserves the IEEE 802.11 DCF contention behavior. Relaying does not affect the relay's own traffic, because the source has already successfully contended for the channel, nor does relaying affect the relay's contention backoff values in the next contention round. The protocol is presented in more detail below:

Consider a network with transmit rates:  $R_0$ ,  $R_1$ , and  $R_2$ , where  $\frac{1}{R_1} + \frac{1}{R_2} < \frac{1}{R_0}$ . A source node whose current transmit rate is  $R_0$  attempts to send a frame of length  $L$  to the AP using an intermediate node as a relay. The first transmission (from the source to the relay) uses rate  $R_1$  and the second (from the relay to the AP) is intended to use rate  $R_2$ .

The total time for the transmission is:

$$T_{\text{relay}} = T_{R_1}(L) + \text{relay\_backoff} + SIFS + T_{R_2}(L) + \quad (1)$$

$$SIFS + T(ACK) \quad (2)$$

where  $T_R(L)$  is the transmit time for a MAC frame of size  $L$  bits at rate  $R$ ,  $\text{relay\_backoff}$  is a constant discussed below and  $T(ACK)$  is the transmit time for the ACK frame.  $T(ACK)$  is a constant based on the AP transmitting the ACK directly to the source at the lowest available rate.

The source does not know whether there are any nodes that can act as a relay. Nevertheless, it sets the duration field in the MAC header assuming the frame will be forwarded using rate  $R_2$ . As in conventional IEEE 802.11, the duration value reflects the remaining transmit time: the relay backoff, relay transmission and ACK (terms 1 and 2 above). The source then transmits the DATA frame using transmit rate  $R_1$ .

Non-ORP nodes that receive the frame set their network allocation vector (NAV) according to the duration field as in IEEE 802.11 DCF and will not attempt to access the channel during the relay process. Keeping the meaning of the duration field this way allows ORP and non-ORP nodes to co-exist in the same BSS.

ORP nodes examine the frame's duration value. If the frame is a direct transmission, the duration shows that the AP will immediately return an ACK. Otherwise, the duration includes the time allocated for the relay transmission. All of the components of the duration value are known constants except for the frame length  $L$  (given in the frame's PLCP header) and  $R_2$ . Each receiver can therefore determine  $R_2$ , the transmit rate intended by the source for the forwarding transmission.

Each receiver is assumed to know the transmit rate that it currently uses to communicate directly with the AP. A receiver is a potential relay only if its direct transmit rate is at least  $R_2$ . Because there is no coordination among potential relay nodes, more than one node may determine that it is a potential relay. To avoid simultaneous relay transmissions, ORP uses a simple backoff. Each potential relay sets a random backoff timer

$$\text{backoff} = \text{Random}() * \text{slotTime},$$



where  $\text{Random}()$  is uniformly distributed over  $[0..\text{relayCW}]$ . To allow for the worst case backoff, the value of  $\text{relay\_backoff}$  used in the duration calculation is  $\text{relayCW} * \text{slotTime}$ .

When its backoff timer expires, a potential relay checks the channel. If the channel is clear, the relay sends the frame to the AP, using transmit rate  $R_2$ . The duration value gives the transmit time for the ACK, just as with any direct transmission. If the channel is busy, the potential relay assumes that another node is already relaying the frame and drops it.

If exactly one relay wins the backoff and the AP successfully receives the relay transmission, the AP transmits an ACK directly to the source. Because the relay transmission begins with the first backoff timer to expire, the ACK is usually transmitted before the time specified in the duration value (which assumed the maximum backoff). The duration value in the ACK is 0, allowing the next contention period to begin.

If there is no relay node or if two potential relays select the same backoff value and transmit during the same slot, the AP will not receive the frame. If the relay transmission fails, the source times out waiting for the ACK and eventually retransmits the frame using direct transmission.

Relaying will also fail if another node begins transmitting during the relay backoff. This situation can occur if a node can sense the originating transmission, but cannot obtain the duration value from the frame header. The node will defer during the originating transmission, then begin the IEEE 802.11 DCF backoff procedure. However, the case is biased in favor of the relay backoff, which is not preceded by the DIFS and ends as soon as any potential relay begins to transmit.

### 3 Analysis

We present two results that will be useful in further discussion of ORP: the probability of successfully relaying and the effective transmit rate obtained using given combination of transmit rates. In this section, we assume that there is a fixed transmit distance for each transmit rate and there are no packet errors.

#### 3.1 Relay Success

A source at distance  $x$  from the AP attempts to relay using transmit rates  $R_1$  and  $R_2$ , with transmit ranges  $r_1$  and  $r_2$  respectively. The attempt succeeds if there is at least one potential relay and exactly one relay wins the backoff. The probability relaying successfully is:

$$P_{\text{success}}(x) = \sum_{n=1}^{N-1} P(\text{no collision} | \text{relays} = n) P(\text{relays} = n) \quad (3)$$

where  $N$  is the total number of nodes in the BSS.

The first term in equation 3 is the probability that there is no collision, given that there are  $n$  potential relays participating in the relay backoff. Each potential relay chooses a backoff uniformly distributed on  $[1..S]$  where  $S = \text{relayCW}$ . The

relay backoff ends successfully at slot  $i$  if one relay selects slot  $i$  and the other  $n - 1$  relays select any later slot in  $[i + 1..S]$ , so

$$P(\text{no collision} | \text{relays} = n) = \sum_{i=1}^S \frac{n}{S} \left( \frac{S-i}{S} \right)^{n-1}. \quad (4)$$

The second term in equation 3 is the probability that  $n$  nodes in the BSS are potential relays for the transmission. We find this probability geometrically, by computing the area of the relay region, in which a relay node must be located to satisfy transmit range constraints (Figure 2). A closed form for  $A(x)$ , the area of the relay region for a source at distance  $x$  from the AP is found in [2,7].

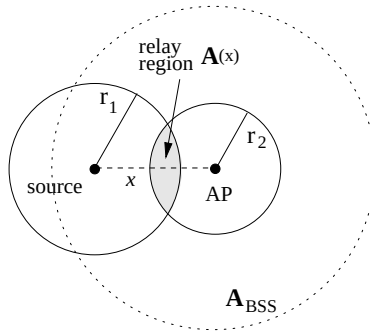


**Fig. 2.** Area of the relay region  $A(x)$  for a source at distance  $x$  from an AP with coverage area  $A_{BSS}$

We assume  $N$  nodes are distributed according to a spatial Poisson process over area  $A_{BSS}$ , the coverage area of the AP. The probability that a source at distance  $x$  from the AP has  $n$  potential relays is the probability that there are exactly  $n$  of  $N - 1$  nodes in a region of area  $A(x)$ .

$$P(\text{relays} = n) = \binom{N-1}{n} \left( \frac{A(x)}{A_{BSS}} \right)^n \left( 1 - \frac{A(x)}{A_{BSS}} \right)^{N-1-n} \quad (5)$$

Multiplying equations 4 and 5 gives equation 3, the probability that a node at distance  $x$  from the AP successfully relays using transmit rates  $R_1$  and  $R_2$ .

### 3.2 Effective Rate

The effective transmit rate  $R_{\text{eff}}$  of a relayed transmission is its apparent transmit rate when viewed as a direct transmission. The time required for a direct transmission is:

$$T_{\text{direct}} = PLCP + \frac{L}{R_{\text{direct}}} + SIFS + T(ACK), \quad (6)$$

where PLCP is the fixed time required for the physical layer convergence protocol (PLCP) preamble and header. The worst case time required using relaying is:

$$T_{\text{relay}} = PLCP + \frac{L}{R_1} + \text{relay\_backoff} + SIFS + PLCP + \frac{L}{R_2} + SIFS + T(ACK). \quad (7)$$

Setting equations 6 and 7 equal and defining  $R_{\text{eff}}$  by analogy with  $R_{\text{direct}}$  gives

$$\frac{L}{R_{\text{eff}}} = \frac{L}{R_1} + \text{relay\_backoff} + SIFS + PLCP + \frac{L}{R_2}.$$

Because of the constant terms,  $R_{\text{eff}}$  depends on the frame length. In general, it is not cost effective to relay short (< 200 byte) frames. In this work, we assume 1500 byte frames, which are representative of TCP traffic.

**Table 1.** Parameters used for computing effective transmit rates

|                  | IEEE 802.11b | IEEE 802.11g<br>(ERP-OFDM) |
|------------------|--------------|----------------------------|
| frame length $L$ | 1500 bytes   | 1500 bytes                 |
| PLCP             | 96 $\mu s$   | 24 $\mu s$ + 6 $\mu s$     |
| SIFS             | 10 $\mu s$   | 10 $\mu s$                 |
| slotTime         | 20 $\mu s$   | 9 $\mu s$                  |
| relayCW          | 15           | 10                         |
| relay_backoff    | 300 $\mu s$  | 90 $\mu s$                 |

To compute specific values of  $P_{\text{success}}$  and  $R_{\text{eff}}$ , we assign values to the various constants as in table 1. The nominal transmit ranges are taken from the published data sheet[8] for the Cisco Aironet 1200 IEEE 802.11b/g AP.

Using the results above, figure 3 shows the probability  $P(x)$  of a node at a distance  $x$  from the AP successfully relaying and obtaining a given  $R_{\text{eff}}$ . Note how the probability of obtaining a 5.5 + 5.5 Mbps relay does not decrease with distance from the AP, because the decreasing size of the relay region  $A(x)$  is offset by the reduced risk of relay collision.

## 4 Further Discussion

In this section, we present two aspects of ORP in more detail: the rate selection process and downlink relaying.

### 4.1 Rate Selection

ORP assumes that each node maintains an estimate of its current direct transmit rate to the AP, but does not presuppose any particular rate adaptation

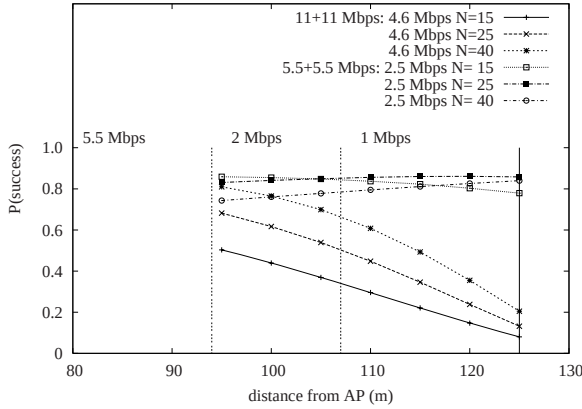


**Fig. 3.** IEEE 802.11b: Probability of successfully relaying for various BSS sizes. In some cases, the probability of success increases with distance, as the decreasing size of the relay region is offset by a lower risk of relay collision.

mechanism. Because shorter frames are generally transmitted without relaying, the direct rate adaptation mechanism is assumed to operate in parallel with multi-rate relaying.

For each relay frame, the source needs to determine whether or not to use relaying and with what combination of transmit rates. Currently, we prescribe a single relay rate combination for each direct transmit rate. The source periodically attempts to relay using this combination. If consecutive relay attempts fail, the source reverts to direct transmission until the next relay attempt. (This mechanism is essentially Auto Rate Fallback(ARF)[9].)

This strategy does not provide the best possible performance, because there will generally be more than one feasible relay rate combination for a given direct transmit rate. Each relay rate combination will provide a different effective transmit rate and will have a different probability of success.

In general, for a given direct transmit rate, only a relatively small number of relay rate combinations will provide a higher effective transmit rate. Unlike direct transmit rates, however, there is not a simple relationship between the effective transmit rate and the probability of the corresponding uplink relay succeeding. A single luckily positioned relay node may make a rate combination feasible when a more conservative one is not, particularly in complex propagation environments.

Nevertheless, the analytic tools provided in section 3 can be used to compute (offline) a plausible ordering of rate combinations for each direct transmit rate, noting that a successful relay transmission provides information about feasible combinations. The decision to attempt to relay must balance the cost of a failed attempt against the benefits of success. Further development of a rate selection mechanism is future work.

## 4.2 Downlink Relaying

The basic ORP mechanism cannot be used for downlink relaying, because the mechanism relies on potential relays knowing the transmit rate at which they communicate with the destination, i.e. the AP. However, current network traffic patterns reflect download traffic and are characterized by a sequence of large downlink frames and short uplink frames. To obtain significant benefit from relaying, it is necessary to relay downlink traffic.

To provide downlink relaying, uplink relaying is used as a *relay discovery* mechanism. The AP records the uplink relay node used by each source and uses this relay for the corresponding downlink transmission. In this case, the address of the relayer must be specified in the frame. We follow CoopMAC II [3] in using the **Address4** field in the MAC header.

The duration value continues to reserve the channel for the complete relay sequence. Because the relay is specified in the MAC header, the relay backoff is unnecessary and the relay forwards the frame immediately. The total time for the relayed transmission is:

$$T_{\text{relay}} = T_{R_2}(L) + SIFS + T_{R_1}(L) + SIFS + T(ACK), \quad (8)$$

and the corresponding effective rate  $R_{\text{eff}}$  is slightly higher.

The AP does not know the transmit rate  $R_1$  that was used between the source and the relay on the uplink path, but it does know the direct transmit rates  $R_0$  and  $R_2$  that it uses to communicate with the source and the relay, respectively. As with uplink relaying, there is currently a single prescribed pair of rates, so  $R_1$  is known.

Even in the case where multiple rate combinations are permitted and  $R_1$  is not known, given  $R_0$  and  $R_2$ , there are (in practice) at only a few reasonable options for transmit rate  $R_1$ , so the AP can begin with the lowest rate and later attempt to increase it.

Currently, we assume an even balance of uplink and downlink relay traffic, so the AP simply records the identity of the relay, which is renewed (and possibly changed) with each uplink transmission.

In more realistic traffic scenarios, relaying short uplink frames is inefficient due to overhead. However, if relay information is cached at the AP, then the cost of using a short uplink frame for relay discovery can be amortized over several downlink transmissions. This approach requires more careful cache management, as the cache may be invalidated due to node mobility, requiring the AP to revert to direct downlink transmission. The design of such a caching mechanism is future work, though we believe that such caching is feasible, particularly in common low mobility scenarios such as offices, conferences and internet cafes.

## 5 Simulation Experiments

We did simulation experiments to investigate the throughput performance of ORP. The results show that ORP provides significant improvement. To focus on

the impact of relaying, we use a simple traffic model and exclude issues of rate adaptation and selection by fixing direct transmit rates and permitting only a single, fixed combination of relay rates. Other parameters are as in Table 1.

The experiments investigate the case of a saturated IEEE 802.11b BSS with an equal mix of uplink and downlink traffic with a “ping-pong” pattern. All nodes in the BSS are assumed to have an infinite number of frames to transmit: each node sends a frame of length 1500 bytes to the AP and the AP responds with a frame of the same length. Each uplink frame provides relay discovery for the corresponding downlink frame.

The direct transmit rate of each node is assigned based on its distance from the access point, providing a bit error rate of  $10^{-5}$  in the absence of interference. Nodes with a direct transmit rate of 1 Mbps use  $R_1 = R_2 = 5.5$  Mbps for an effective transmit rate of 2.5 Mbps. Nodes with a direct transmit rate of 2 Mbps use  $R_1 = R_2 = 11$  Mbps for an effective transmit rate of 4.6 Mbps. If three consecutive relay transmission fail, the source reverts to direct transmission for 40 transmissions, then attempts to relay again.

The simulation experiments used Omnet++ 3.2[10] and mobility-fw 1.0a4[11], to which we added support for 802.11b multi-rate communication. The mobility-fw package uses a propagation model similar to that used in ns-2, but provides a somewhat more detailed model of the air frame. As usual in IEEE 802.11 BSS environments, RTS/CTS is not used. Following [12], the bit error rate (BER) for 1 and 2 Mbps and for 5.5 and 11 Mbps transmissions respectively are given by:

$$BER = 0.5 \exp\left(\frac{-SNIR * BW}{\text{bitrate}}\right) \text{ and } BER = C_{5.5,11} \operatorname{erf}\left(\frac{-SNIR * BW}{\text{bitrate}}\right).$$

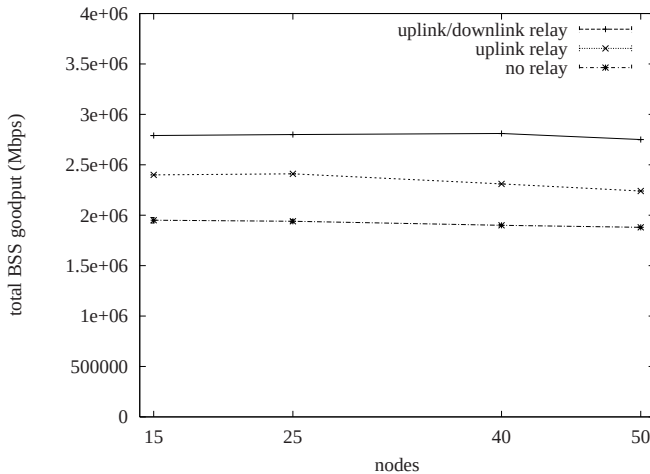


**Fig. 4.** IEEE 802.11b: Overall AP throughput as a function of BSS size (The 95% confidence interval is approximately the size of the data point.)

We record the total goodput of the BSS (i.e. frames successfully transmitted to and from the AP) over a simulation time of 50s, for each of 50 randomly generated static topologies for network sizes ranging from 15 to 50 nodes.

Figure 4 compares the total traffic sent or received at the AP in the case of no relaying, uplink-only relaying, and both uplink and downlink relaying. The impact of nodes using a low data transmit rate is clear: With no relaying, the goodput is less than 2Mbps, even though about half of the nodes transmit at either 5.5 or 11 Mbps. Uplink relaying alone results in a goodput about 20% higher over a range of node densities, while both uplink and downlink relaying results in about a 40% increase.

## 6 Related Work

The authors used geometric arguments to analyze the feasibility of relaying and outlined a relay mechanism in [2]. In this section, we highlight key contributions made in this work relative to previously proposed protocols, including CoopMAC [3], RAAR [6], *r*PCF [4], and *r*DCF[5].

Any relay protocol must obtain two data: the transmit rate between the source and each potential relay and the transmit rate between the potential relay and the AP. The structure of each protocol is determined by which entity (source, relay or AP) collects this data and selects and assigns relays. Table 2 summarizes protocols according to these criteria.

In Cooperative MAC (CoopMAC), nodes use received signal strength (RSSI) measurements to estimate the transmit rates needed to communicate with potential relays and directly observe the transmit rates used potential relays to communicate with the AP. Relay selection is distributed, each source uses its rate information to select a relay node. CoopMAC I uses an RTS/"HTS"/CTS negotiation to inform the intended relay of its role. Like ORP, CoopMAC II uses the Address4 field in the MAC header to indicate the selected relay.

The Relay-based Adaptive Auto Rate (RAAR) protocol is a centralized protocol in which nodes observe the RSSI of their neighbors' transmissions and estimate the appropriate transmission rate for communicating with each neighbor. The estimates are forwarded to the AP, which computes relay assignments distributes them via its periodic beacon transmission.

*r*PCF is intended for IEEE 802.11 PCF networks. Nodes forward RSSI observations to the AP, which explicitly assigns relays. *r*DCF is intended for ad hoc networks. Nodes observe the transmit rates used by their neighbors to determine node pairs for which they might act as a relay. Nodes periodically announce their relay capabilities to their neighbors. ORP uses a similar volunteer relay approach, but does not require an explicit advertisement.

ORP differs from previous work in discovering relay nodes experimentally, by optimistically making frames available for relaying and allowing nodes to select themselves as relays. This approach has three advantages:

First, ORP does not rely on RSSI data to discover relay nodes, avoiding the overhead of maintaining RSSI observations for each potential relay. We believe

**Table 2.** Comparing relay protocols

|               | source-relay | relay-AP | relay selection | relay assignment | uplink downlink | backwards compatibility |
|---------------|--------------|----------|-----------------|------------------|-----------------|-------------------------|
| CoopMac I     | RSSI         | snoop    | source          | RTS-HTS          | uplink only     | no                      |
| CoopMac II    | RSSI         | snoop    | source          | Address4         | uplink only     | no                      |
| RAAR          | RSSI         | AP       | AP              | AP               | both            | potentially             |
| rPCF          | RSSI         | AP       | AP              | AP               | both            | PCF only                |
| rDCF          | n/a          | n/a      | relay           | Address4         | n/a             | ad hoc only             |
| ORP (uplink)  | exp't        | relay    | relay           | none             | uplink only     | non-ORP nodes           |
| ORP(downlink) | exp't        | relay    | relay           | Address4         | both            | non-ORP nodes           |

that transmit rate estimation based on RSSI is less straightforward than suggested in previous work. Computing an SNIR from the RSSI depends on the noise floor estimate, because the RSSI measurement reflects only the total power received at the antenna. This functionality is not provided by default in IEEE 802.11 hardware: the transceiver tries to synchronize whenever the RSS exceeds the receive sensitivity.

As a result, accurate transmit rate estimation may not be feasible in a noisy environment. Moreover, the performance of an RSSI-based approach can be difficult to evaluate in simulation. To infer transmit rates from RSSI data, the relay protocol must incorporate some channel model. The simulator also uses a channel model to approximate the behavior of a real wireless network. If these models are too closely aligned, the simulation results may not provide a good indication of performance.

Second, ORP nodes select themselves as relays, avoiding the overhead of communicating rate information and relay assignments found in other protocols. ORP also provides greater flexibility in managing relays. If relays are not self selecting, then the selector has to determine which nodes support ORP and track node battery levels. An ORP node with a low battery simply does not participate in the relay backoff, while non-ORP nodes need only follow IEEE 802.11 rules to avoid collision.

Third, ORP does not incur a significant performance penalty despite its simple design. Although it is difficult to compare performance results reflecting different simulation environments and experiments, ORP appears to achieve performance comparable to more complex protocols (e.g. [3]).

## 7 Conclusions and Future Work

We introduce ORP, an opportunistic relay protocol that differs significantly from previously proposed protocols in not using RSSI information for relay selection. We derive the probability that a node successfully uses relaying to obtain a given effective transmit rate and present simulation results showing that in a saturated IEEE 802.11b network with both uplink and downlink relaying, the total throughput of the BSS increases about 40%. In addition, because ORP



preserves IEEE 802.11 DCF semantics, ORP and non-ORP nodes can co-exist in an ORP BSS.

These results are promising and there is considerable scope for further performance improvement, particularly though the use of rate selection and relay caching discussed in section 4. We also plan to transition ORP to IEEE 802.11a/g, which provides more and faster relay rates. Finally, we hope to implement and test ORP in realistic environments. In non line-of-sight environments, such as buildings or offices, relaying effectively enables routing around obstacles (figure 1(b)). This non-geometric variation in transmit rate provides opportunities to obtain even greater advantages from ORP relaying.

## References

1. Heusse, M., Rousseau, F., Berger-Sabbatel, G., Duda, A.: Performance anomaly of 802.11b. In: *Proceedings of IEEE INFOCOM*. (2003)
2. Feeney, L.M., Hollos, D., Karl, H., Kubisch, M., Mengesha, S.: A geometric derivation of the probability of finding a relay in multi-rate networks. In: *3rd IFIP Conference on Networking (Networking 2004)*. (2004)
3. Liu, P., Tao, Z., Panwar, S.: A cooperative MAC protocol for wireless local area networks. In: *IEEE Conference on Communications (ICC 2005)*. (2005)
4. Zhu, H., Cao, G.: On improving the performance of IEEE 802.11 with relay-enabled PCF. *MONET* **9**(4) (2004) 423–434
5. Zhu, H., Cao, G.: rDCF: A relay-enabled medium access control protocol for wireless ad hoc networks. In: *Proceedings of IEEE INFOCOM*. (2005)
6. Liu, J., Lin, C.: An opportunistic relay method for increasing throughput in multirate IEEE 802.11 wireless LAN. In: *IEICE Transactions on Communications*. (2005)
7. Cetin, B.: Opportunistic relay protocol for IEEE 802.11 WLANs. Master's thesis, Royal Institute of Technology (2006)
8. Cisco: <http://www.cisco.com>.
9. Kamerman, A., Monteban, L.: WaveLAN-II a high-performance wireless LAN for the unlicensed band. *Bell Labs Technical Journal (Summer)* (1997) 118–133
10. Varga, A.: The OMNeT++ discrete event simulation system'. In: *Proceedings of the European Simulation Multiconference (ESM'2001)*. (2001)
11. Koepke, A., Willkom, D.: <http://www.sourceforge.net>.
12. Ebert, J.P., Aier, S., Kofahl, G., Becker, A., Burns, B., Wolisz, A.: Measurement and simulation of the energy consumption of a WLAN interface. Technical Report TKN-02-010, Technical University Berlin (2002)

# Performance Analysis of IEEE 802.11b Under Multiple IEEE 802.15.4 Interferences

Dae Gil Yoon<sup>1</sup>, Soo Young Shin<sup>2</sup>, Jae Hee Park<sup>1</sup>,  
Hong Seong Park<sup>3</sup>, and Wook Hyun Kwon<sup>2</sup>

<sup>1</sup> Samsung Electro-Mechanics, Maetan-3-dong, Yeongtong-gu,  
Suwon, Gyeonggi, Korea

daegil77.yoon@samsung.com, jaehee01.park@samsung.com

<sup>2</sup> School of Electrical Eng. and Computer Science Seoul National University  
San 56-1 Shillim-dong, Kwanak-gu, Seoul, Korea

wdragon@cisl.snu.ac.kr, whkwon@cisl.snu.ac.kr

<sup>3</sup> Dept. of Electrical and Computer Eng., Kangwon National University  
Chuncheon, Kangwon-Do, Korea

hspark@kangwon.ac.kr

**Abstract.** This paper presents analysis of the performance on the IEEE 802.11b in the presence of multiple IEEE 802.15.4 interferences. To analyze the performance, packet error rate (PER) and throughput are used as performance metrics. The PER is computed by the bit error rate (BER), the collision time and the number of IEEE 802.15.4 interferers in the presence of in-band of the IEEE 802.11b. The throughput of the IEEE 802.11b under multiple IEEE 802.15.4 interferences is obtained from the total IEEE 802.11b packet length received during a specified time. Analytic results of the performance of IEEE 802.11b under multiple IEEE 802.15.4 interferences are verified by simulation results. These results can support coexistence criteria for the IEEE 802.11b and the IEEE 802.15.4.

**Keywords:** Packet Error Rate (PER), Throughput, Coexistence, Interference.

## 1 Introduction

In 2.4 GHz Industrial, Scientific, and Medical (ISM) band, there are several heterogeneous networks, IEEE 802.11b (WLAN) [1], IEEE 802.15.1 (Bluetooth) [2], and IEEE 802.15.4 (Zigbee) [3]. Besides, 2.4 GHz electromagnetic noise is radiated from microwave ovens of home appliances. The increase of 2.4 GHz band utilization and the electromagnetic noise may cause serious interference problems in the IEEE 802.11b devices using the 2.4 GHz band. Since the IEEE 802.11b and the IEEE 802.15.4 have been designed for different purposes, they can be collocated within the communication range of each other [8]. The IEEE 802.11b is used for transmitting data and accessing internet over an access point. On the other hand, the IEEE 802.15.4 is used for the sensor network or the home

network. To construct the home network system, many IEEE 802.15.4 devices are deployed in the home. Because the IEEE 802.11b is exposed to multiple IEEE 802.15.4 interferers in the home, the coexistence performance of the IEEE 802.11b under multiple interferences of the IEEE 802.15.4 needs to be evaluated.

There are some previous studies about coexistence between the IEEE 802.11b and the IEEE 802.15.4 [4] [5] [6] [7] [8]. In [4], the packet error rate (PER) of the IEEE 802.11b under the IEEE 802.15.4 interference is obtained from an experiment without analysis, and vice versa. In [5], the impact of an IEEE 802.15.4 network on the IEEE 802.11b devices is analyzed by the PER. However, the PER in [5] is analyzed without considering the collision time. Further, it is not considering whether bandwidth of the IEEE 802.11b is overlapped by that of the IEEE 802.15.4. In [6], the PER of the IEEE 802.15.4 under the interference of the IEEE 802.11b is evaluated using a simulation without numerical analysis. In [7], the PER of the IEEE 802.15.4 under the interference of the IEEE 802.11b is evaluated using the analysis and the simulation. The PER is obtained from the BER and the collision time in [7]. In [8], the PER of the IEEE 802.11b under the interference of the IEEE 802.15.4 is evaluated using the analysis and the simulation. The PER, the throughput and the safe distance ratio are obtained from the BER and the collision time in [8]. However, PER of the IEEE 802.11b is not considered in presence of multiple IEEE 802.15.4 interferers.

In this paper, the main objective is focused on analyzing and simulating the performance of IEEE 802.11b under multiple IEEE 802.15.4 interferences. To evaluate the performance of the IEEE 802.11b under multiple IEEE 802.15.4 interferers, the PER and the throughput are used as performance metrics. The PER is obtained by the bit error rate (BER), the collision time and number of IEEE 802.15.4 interferers in the presence of in-band of the IEEE 802.11b. The BER is obtained from the signal to noise and interference ratio (SNIR). The collision time is defined as the time that IEEE 802.11b packet is overlapped by IEEE 802.15.4 packets. The throughput of the IEEE 802.11b under multiple interferences of the IEEE 802.15.4 is obtained from the total IEEE 802.11b packet length received during a specified time. Analytic results of the performance of the IEEE 802.11b under multiple IEEE 802.15.4 interferers are verified by simulation results.

This paper is organized as follows. Network topology of IEEE 802.11b devices and IEEE 802.15.4 interferers is shown in Section 2. In Section 3, the interference model between the IEEE 802.11b and multiple IEEE 802.15.4 interferers is described. Section 4 evaluates the performance analysis of the IEEE 802.11b under multiple IEEE 802.15.4 interferers. Comparison between analytic and simulation results are shown in Section 5. Finally, conclusions are presented in Section 6.

## 2 Network Topology of IEEE 802.11b and IEEE 802.15.4

IEEE 802.11b devices and IEEE 802.15.4 interferers may be operated in the proximity of home or office. In this paper, topology of IEEE 802.11b devices and IEEE 802.15.4 interferers is used as illustrated in Fig. 1.

As illustrated in Fig. 1,  $D(W_0, W_1)$ ,  $D(\text{End\_dev\_}k, \text{Coord\_}k)$ , and  $D(W_0, \text{End\_dev\_}k)$  represent the distance between two IEEE 802.11b devices, the distance between two IEEE 802.15.4 devices, and the distance between the IEEE 802.11b WLAN\_0 and the  $k$ -th IEEE 802.15.4 End\_dev, respectively, where  $\forall k \in [1, \dots, 16]$ . In the  $k$ -th IEEE 802.15.4 network that is consisted of an End\_device and a coordinator, the End\_dev\_  $k$  and the coord\_  $k$  are devices that co-exist with the IEEE 802.11b.

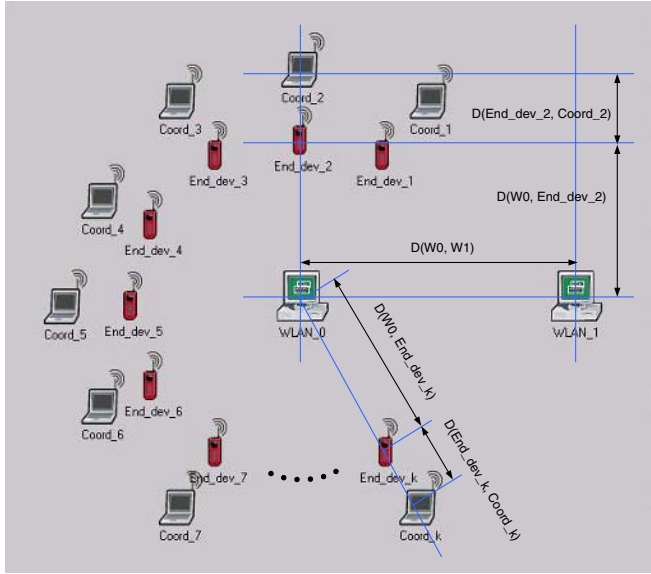


**Fig. 1.** IEEE 802.11b and the IEEE 802.15.4 topology

For home or office environment, indoor propagation model is adopted in this paper [9].

$$L_p(d) = \begin{cases} 20 \log_{10} \left( \frac{4\pi d}{\lambda} \right) & , d \leq d_o \\ 20 \log_{10} \left( \frac{4\pi d}{\lambda} \right) + 10n \log_{10} \frac{d}{d_o} & , d > d_o \end{cases} \quad (1)$$

where  $d$  is the distance between the transmitter and the receiver.  $d_o$  is the length of the line of sight (LOS).  $\lambda$  is the wavelength of the propagating wave;  $c/f_c$  where  $c$  is the light velocity and  $f_c$  is the carrier frequency.  $n$  is the path loss exponent that indicates the rate at which the path loss increases with distance between the transmitter and the receiver.

### 3 Interference Model Between IEEE 802.11b and IEEE 802.15.4

The IEEE 802.11b and the IEEE 802.15.4 channel at the 2.4GHz is illustrated in Fig. 2. In this paper, for simplicity, it is assumed that more than two IEEE

802.15.4 networks can not use same channel, i.e. a channel is allocated for a IEEE 802.15.4 network. If so, a network of the IEEE 802.15.4 is independent of other IEEE 802.15.4 networks and has minimum backoff time. As shown in Fig. 2, 16 networks of the IEEE 802.15.4 can exist and four IEEE 802.15.4 channels can interfere with a channel of the IEEE 802.11b.

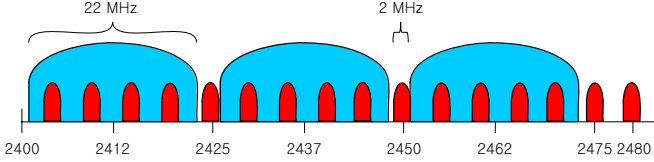


**Fig. 2.** IEEE 802.11b and IEEE 802.15.4 channel

Both standards use carrier sense multiple access with collision avoidance (CSMA/CA) and perform a backoff process before transmitting packets. In this paper, it is assumed that the WLAN\_1 and the End\_dev\_ $k$  transmit data packets without consideration of the channel state. That's why the WLAN\_1 and the End\_dev\_ $k$  consider signal power of each other as noise. On the other hand, the other devices, the WLAN\_0 and the Coord\_ $k$ , only receive packets from the WLAN\_1 and the End\_dev\_ $k$ .

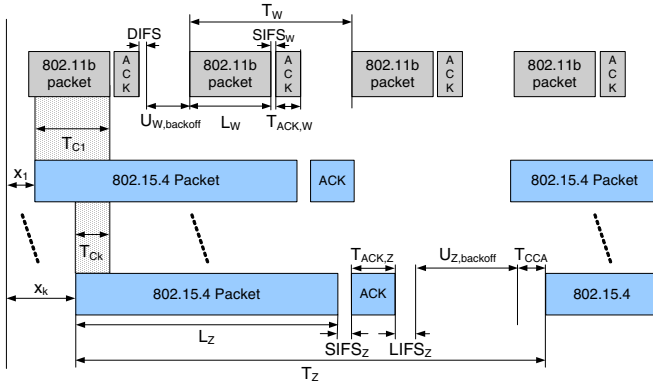


**Fig. 3.** Analytic model for collision time between IEEE 802.11b and IEEE 802.15.4

For simplicity, acknowledgement (ACK) packets of both the IEEE 802.11b and the IEEE 802.15.4 are not considered because this paper focuses on receiving data packets. An analytic interference model among the IEEE 802.11b and the IEEE 802.15.4, can be illustrated as in Fig. 3. Let  $T_X$ ,  $L_X$ , and  $U_X$  be the inter-arrival time, the packet duration, and the average random backoff time, respectively, where the subscript  $X$  is either  $W$  for the IEEE 802.11b or  $Z$  for the IEEE 802.15.4.  $x_k$  is  $k$ -th time offset of the IEEE 802.15.4 packet,  $\forall k \in [1, \dots, 16]$ .

As has been mentioned in assumption, transmitting packets of both standards are independent. Therefore, the backoff time is randomly chosen within the minimum contention,  $CW_{min}$ . Because IEEE 802.15.4 networks are independent, the contention among IEEE 802.15.4 networks doesn't exist in this paper.

The inter-arrival times,  $T_W$  and  $T_Z$ , can be expressed as:

$$T_W = L_W + SIFS_W + T_{ACK,W} + DIFS + U_W \quad (2)$$

and

$$T_Z = L_Z + T_{CCA} + SIFS_Z + T_{ACK,Z} + U_Z \quad (3)$$

where  $T_{CCA}$  denotes the two clear channel assessment (CCA) slot time of the IEEE 802.15.4 and  $U_X = \sigma_X \cdot CW_{min,X}/2$ . Besides, other parameters are listed in Table 1.

**Table 1.** Parameters of the Interference Model

|             |                                                      |
|-------------|------------------------------------------------------|
| $T_W$       | inter-arrival time between two IEEE 802.11b packets  |
| $L_W$       | duration of IEEE 802.11b packet                      |
| $SIFS_W$    | short IFS of IEEE 802.11b                            |
| $DIFS$      | DCF IFS of IEEE 802.11b                              |
| $T_{ACK,W}$ | duration of IEEE 802.11b ACK packet                  |
| $U_W$       | average backoff time of IEEE 802.11b                 |
| $\sigma_W$  | slot time of IEEE 802.11b                            |
| $T_Z$       | inter-arrival time between two IEEE 802.15.4 packets |
| $L_Z$       | duration of IEEE 802.15.4 packet                     |
| $SIFS_Z$    | short IFS of IEEE 802.15.4                           |
| $LIFS$      | long IFS of IEEE 802.15.4                            |
| $T_{ACK,Z}$ | duration of IEEE 802.15.4 ACK packet                 |
| $U_Z$       | average backoff time of IEEE 802.15.4                |
| $\sigma_Z$  | slot time of IEEE 802.15.4                           |

## 4 Performance Analysis of IEEE 802.11b Under IEEE 802.15.4

### 4.1 Packet Error Rate Analysis of IEEE 802.11b Under IEEE 802.15.4

The physical layer of IEEE 802.11b provides dynamic data rate and supports to shift up to 11 Mbps. 11 Mbps data rate is obtained using CCK (Complementary Code Keying) modulation. In this paper, 11 Mbps data rate of the IEEE 802.11b is only considered. Because transmission power of the IEEE 802.11b, 30mW, is higher than that of the IEEE 802.15.4, 1mW, the IEEE 802.11b rarely transmits at lower speeds, 5.5 Mbps, 2 Mbps, and 1 Mbps. The bit error rate (BER) for 11 Mbps can be expressed as follows [9].

$$P_B = 1 - \frac{1}{\sqrt{2\pi}} \int_{-X}^{\infty} \left( \frac{1}{\sqrt{2\pi}} \int_{-(v+X)}^{v+X} \exp\left(-\frac{y^2}{2}\right) dy \right)^{\frac{N}{2}-1} \cdot \exp\left(-\frac{v^2}{2}\right) dv \quad (4)$$

where  $X = \sqrt{2 \cdot E_b/N_o}$ .  $E_b/N_o$  is used to denote the ratio of the average energy per information bit to the noise power spectral density at the receiver input in the case of an additive white Gaussian noise (AWGN) channel.  $N = 8$  for 11 Mbps.

Because the IEEE 802.15.4 signal is considered as the partial band jammer noise for the IEEE 802.11b, the signal-to-interference plus noise ratio can be defined as

$$SNIR = 10 \log_{10} \left( \frac{P_c}{P_{N_o} + P_i} \right) + P_{gain} \quad (5)$$

where  $P_c$  is the power of the desired signal,  $P_{N_o}$  is the noise power and  $P_i$  is the power of the interferer. The  $P_{gain}$  is the processing gain of the IEEE 802.11b [10]. The CCK is a form of M-ary bi-orthogonal modulation and it uses Walsh/Hadamard codes of length 8 for modulation. The symbol rate for 11 Mbps is fixed at 1.375 Msps. For 11 Mbps, 8 bits are encoded into the 8 chip sequence. Thus there is no processing gain for the 11 Mbps data rate.

By replacing  $E_b/N_o$  in Equation (4) with SNIR in Equation (5), the BER of the IEEE 802.11b under the IEEE 802.15.4 can be obtained. The information signal power at receiver is  $P_{R_x} = P_{T_x} \cdot 10^{-\frac{L_p(d)}{10}}$ , where it is assumed that the transmitter power is fixed as  $P_{T_x}$  and the subscript  $x$  is either W for the IEEE 802.11b or Z for the IEEE 802.15.4.

The  $T_{C_k}$  is the collision time that a IEEE 802.11b packet from WLAN\_1 is overlapped by IEEE 802.15.4 packets from the End\_dev\_k in time,  $\forall k \in [1, \dots, 16]$ . Assume that the time offset  $x_k$  is uniformly distributed in  $[0, T_Z]$ , then,  $T_{C_k}$  can be obtained as:

$$T_{C_k} = \begin{cases} L_W - x_k & , x_k < L_W \\ 0 & , L_W \leq x_k < T_Z - L_Z \\ x_k - (T_Z - L_Z) & , T_Z - L_Z \leq x_k < T_Z - L_Z + L_W \\ L_W & , T_Z - L_Z + L_W \leq x_k \leq T_Z \end{cases} \quad (6)$$

The packet error rate (PER) under  $k$ -th IEEE 802.15.4 interference,  $\forall k \in [1, \dots, 16]$ , can be obtained from the BER and the collision time,  $T_{C_k}$ . The PER,  $P_{P_k}$  is expressed as

$$P_{P_k} = 1 - \left\{ (1 - P_B)^{\lceil (L_W - T_{C_k})/b \rceil} \cdot (1 - P_B^I)^{\lceil T_{C_k}/b \rceil} \right\} \quad (7)$$

where  $P_B$  and  $P_B^I$  are BER without and with interference of the IEEE 802.15.4, respectively.  $b$  is the bit duration of the IEEE 802.11b.

Because impact of multiple IEEE 802.15.4 influences is independent to the IEEE 802.11b, the PER of the IEEE 802.11b under multiple interferences of the IEEE 802.15.4,  $P_{multi-P}$ , can be expressed as

$$P_{multi-P} = 1 - (1 - P_{P_1})(1 - P_{P_2}) \cdots (1 - P_{P_k}) \quad , \forall k \in [1, \dots, 16] \quad (8)$$

## 4.2 Throughput of the IEEE 802.11b

The throughput of the IEEE 802.11b under multiple interferences of the IEEE 802.15.4 is defined as the total packets length received during a specified time.

The throughput of the IEEE 802.11b under the multiple IEEE 802.15.4 interferers,  $\rho_W$ , is given by

$$\rho_W = \frac{L_W R}{T_W}(1 - P_{multi\_P}) \tag{9}$$

where  $R$  is data rate.

### 5 Analysis and Simulation Result

The IEEE 802.11b providing data rates of 11 Mbps and the IEEE 802.15.4 using the slotted CSMA/CA are used for the simulation. For the indoor propagation model that used in simulation, the length of LOS,  $d_0$ , is 8 m and the path loss exponent,  $n$ , is 3.3.

**Table 2.** Configuration and Simulation Parameters

| Parameter         | IEEE 802.11b | IEEE 802.15.4     |
|-------------------|--------------|-------------------|
| Transmitted Power | 30mW         | 1mW               |
| Payload size      | 1500 bytes   | 105 bytes         |
| Slot time         | 20 $\mu s$   | 320 $\mu s$       |
| $CW_{min}$        | 31           | 7                 |
| Center frequency  | 2.412 GHz    | 2.405 ~ 2.480 GHz |

The configuration and simulation parameters used in this paper are shown in Table 2. These parameters, maximum payload size, minimum contention window and so on, are chosen for the worst scenario. Analysis and simulation are performed under circumstances as depicted in Fig. 1. The D(W0, W1) and D(End\_dev\_ $k$ , Coord\_ $k$ ) is fixed to 5m and 1 m, respectively. The D(W0, End\_dev\_ $k$ ) is varied from 0m to 7m. Note that every D(W0, End\_dev\_ $k$ ),  $\forall k \in [1, \dots 16]$ , is varied by the same distance, i.e. if D(End\_dev\_1, Coord\_1) is 3m, D(End\_dev\_2, Coord\_2), ..., and D(End\_dev\_ $k$ , Coord\_ $k$ ) are 3m.

Fig. 4 shows the PER of the IEEE 802.11b for 11 Mbps under multiple interferences of the IEEE 802.15.4. In Fig. 4, until four IEEE 802.15.4 interferers, the Num of the IEEE 802.15.4 expresses the number of the IEEE 802.15.4 in the presence of in-band of the IEEE 802.11b. When there are five IEEE 802.15.4 interferers, one of five IEEE 802.15.4 interferers is located in out-band of the IEEE 802.11b. By increasing the number of IEEE 802.15.4 interferers in presence of in-band of the IEEE 802.11b, the PER of the IEEE 802.11b increases. In case of D(W0, End\_dev\_ $k$ ) = 4m, when the number of IEEE 802.15.4 interferers is 1, 2, 3, and 4, the PER of the IEEE 802.11b is  $8.64 \times 10^{-2}$ ,  $4.39 \times 10^{-1}$ ,  $5.80 \times 10^{-1}$ ,  $6.85 \times 10^{-1}$ , respectively. As shown in Fig. 4, D(W0, End\_dev\_ $k$ ) for  $P_{multi\_P} \leq 10^{-3}$  should be larger than 4.7m, 5.5m, 5.7m, 6.2m, when the number of IEEE 802.15.4 interferers is 1, 2, 3, and 4, respectively.



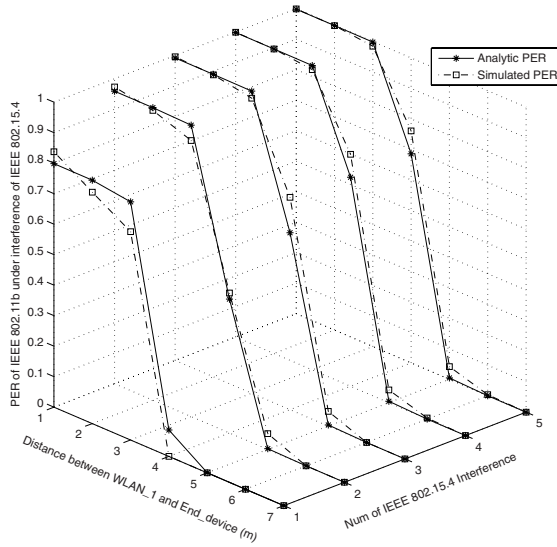


**Fig. 4.** PER of IEEE 802.11b for 11 Mbps under interference of IEEE 802.15.4

As shown in Fig. 4, because any IEEE 802.15.4 interferers in presence of out-band can't affect to the PER of the IEEE 802.11b, the PER of the IEEE 802.11b in the presence of five IEEE 802.15.4 interferers is similar to that of the IEEE 802.11b in the presence of four IEEE 802.15.4 interferers. Conclusively, if  $D(W0, \text{End\_dev}_k)$  is larger than 6.2m, the IEEE 802.11b is not affected by multiple IEEE 802.15.4 interferers in both analytic and simulated result.

Fig. 5 shows the throughput of the IEEE 802.11b for 11 Mbps under the multiple interferences of the IEEE 802.15.4. By increasing the number of IEEE 802.15.4 interferers in presence of in-band of the IEEE 802.11b, the throughput of the IEEE 802.11b decreases. As shown in Fig. 5, when the number of IEEE 802.15.4 interferers is 1, 2, 3, and 4, the throughput of the IEEE 802.11b for  $D(W0, \text{End\_dev}_k) = 4\text{m}$  is  $7.02 \times 10^6$ ,  $4.31 \times 10^6$ ,  $3.23 \times 10^6$ ,  $2.42 \times 10^6$ , respectively. In case of number of the IEEE 802.15.4 interferers=1, 2, 3, and 4,  $D(W0, \text{End\_dev}_k)$  for maximum throughput of the IEEE 802.11b has to be larger than 4.5m, 5m, 5.4m and 6.0m, respectively. When  $D(W0, \text{End\_dev}_k)$  is 4.5m, 5m, 5.4m and 6.0m, the throughput of the IEEE 802.11b is  $7.46 \times 10^6$ ,  $7.67 \times 10^6$ ,  $7.66 \times 10^6$  and  $7.65 \times 10^6$ , respectively.

Similarly with Fig. 4, the throughput of the IEEE 802.11b in the presence of five IEEE 802.15.4 interferers is similar to that of the IEEE 802.11b in the presence of four IEEE 802.15.4 interferers in Fig. 5. Conclusively, if  $D(W0, \text{End\_dev}_k)$  is larger than 6.0m, throughput of the IEEE 802.11b is guaranteed from multiple IEEE 802.15.4 interferences as both analytic and simulated result.

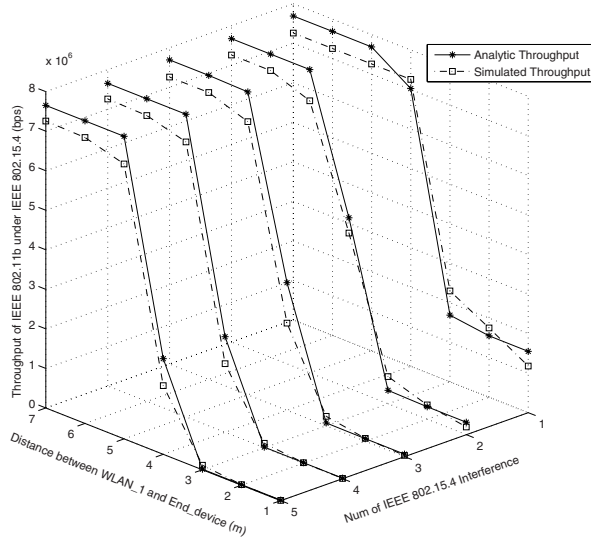


**Fig. 5.** Throughput of IEEE 802.11b for 11 Mbps under interference of IEEE 802.15.4

## 6 Conclusion

In this paper, the Packet error rate (PER) and the throughput of the IEEE 802.11b under multiple IEEE 802.15.4 interferences are analyzed to evaluate the coexistence performance.

The PER can be obtained by the bit error rate (BER), the collision time and the number of IEEE 802.15.4 interferers in the presence of in-band of the IEEE 802.11b. Since the bandwidth of the IEEE 802.11b, 22MHz, is larger than that of the IEEE 802.15.4, 2MHz, the IEEE 802.15.4 signal is considered as the partial band jammer noise for the IEEE 802.11b. The BER of the IEEE 802.11b is given by closed equation from the complementary code keying (CCK) for 11 Mbps modulation scheme. The collision time is defined as the time that IEEE 802.11b packet is overlapped by IEEE 802.15.4 packets from multiple IEEE 802.15.4 interferers. The collision time is obtained from the interference model between the IEEE 802.11b and IEEE 802.15.4. The collision time is calculated under assumption that the packet transmission of the IEEE 802.11b device and the IEEE 802.15.4 devices is independent. Regardless of number of interferers, if distance between a IEEE 802.11b device and multiple IEEE 802.15.4 interferers is larger than 6.2m, PER of the IEEE 802.11b is less than  $10^{-3}$ , i.e. the PER is guaranteed from multiple IEEE 802.15.4 interferers.

The throughput of the IEEE 802.11b under multiple IEEE 802.15.4 interferences is defined as the total packets length received during a specified time in the analysis. The simulated throughput is calculated from the number of received

IEEE 802.11b packets, duration of IEEE 802.11b packets and total simulation time. Regardless of number of interferers, if distance between a IEEE 802.11b device and multiple IEEE 802.15.4 interferers is larger than 6.0m, the throughput of the IEEE 802.11b is about  $7.65 \times 10^6$  and it is guaranteed from multiple IEEE 802.15.4 interferers.

These analytic results are verified from simulation results. These results can be suggested coexistence criteria for the IEEE 802.11b and the IEEE 802.15.4.

## References

1. IEEE Std.802.11: IEEE Standard for Wireless LAN Medium Access Control(MAC) and Physical Layer(PHY) Specification (1997)
2. IEEE Std.802.15.1: IEEE Standard for Wireless Medium Access Control(MAC) and Physical Layer(PHY) Specifications for Wireless Personal Area Networks (WPANs) (2002)
3. IEEE Std.802.15.4: IEEE Standard for Wireless Medium Access Control (MAC) and Physical Layer(PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs) (2003)
4. Sikora, A.: Coexistence of IEEE 802.15.4 (ZigBee) with IEEE 802.11 (WLAN), Bluetooth, and Microwave Ovens in 2.4 GHz ISM-Band (2004)
5. I. Howitt and J.A. Gutierrez: IEEE 802.15.4 low rate wireless personal area network coexistence issues, in Proc. of Wireless Communications and Networking (WCNC), Vol. 3, (2003) 1481-1486
6. N. Golmie and D. Cypher: Performance analysis of low rate wireless technologies for medical applications, Computer Communications, Vol. 28, (2005) 1266-1275
7. Soo Young Shin and Wook Hyun Kwon: Packet Error Rate Analysis of IEEE 802.15.4 under IEEE 802.11b Interference, Wired/Wireless Internet Communications, (WWIC), (2004) 279-288
8. Dae Gil Yoon and Wook Hyun Kwon: Packet error rate of IEEE 802.11b under IEEE 802.15.4 Interference, Vehicular Technology Conference, (VTC 2006-Spring), Vol. 3, (2006) 1186-1190
9. M.Borgo and A.Zanella: Analysis of the hidden terminal effect in multi-rate IEEE 802.11b networks, in Proc. of Wireless Personal Multimedia Communications, (2004)
10. R.E Ziemer R.L. Peterson and D.E. Borth: Introduction to Spread Spectrum Communications, Prentice Hall Publications, (1995)

# Characterization of Service Times Burstiness of IEEE 802.11 DCF

Francesco Vacirca and Andrea Baiocchi

University of Roma "La Sapienza", INFOCOM Dept.,  
Via Eudossiana 18, 00184, Rome, Italy

francesco@net.infocom.uniroma1.it, andrea.baiocchi@uniroma1.it

**Abstract.** First order metrics (throughput, average delay) of the IEEE 802.11 DCF MAC protocol have been extensively analyzed. The service process of the same protocol has not received the same attention, although it is known that it might be bursty. We develop a simple and accurate Markov model that allows a complete characterization of the service process of an 802.11 BSS. A major result of simulations is that correlation between consecutive service times is negligible, hence the service process can be safely described as a renewal process. The analytic model highlights that service burstiness lies essentially in the doubling of the contention window after a collision up to very large values. The trade-off between average throughput and service burstiness is obtained from the model.

## 1 Introduction

Performance evaluation of a single-hop, Independent Basic Service Set (IBSS) IEEE 802.11 Distributed Coordination Function (DCF) has been largely focused on average, long-term metrics, like saturation throughput (e.g. see [1,2,3,4]), non saturated average throughput (e.g. [5]), delay analysis (e.g. [6] and [7]), average throughput of long-lived TCP connections (e.g. [8] and [9]) and short-lived TCP connections (e.g. [10]).

We aim at characterizing the 802.11 DCF from an external point of view, i.e. as a server of upper layer data units. To this end we focus on an IBSS made up on  $n$  stations, possibly including an Access Point (AP), within full visibility of one another, so that carrier sense is fully functional. Saturation traffic is considered, i.e. each station always has a packet to send. We characterize the service process of the network at MAC layer, i.e. the sequence of times between two consecutive service completions. In this context, service of a MAC frame is completed when it is successfully delivered to its destination or when it is discarded after the maximum number of transmission attempts has been reached, as envisaged by the IEEE 802.11 DCF standard [13]. Service completion can be viewed both from an individual side (a tagged station service completion) or from a collective standpoint (service completion irrespective of the originator of the served MAC frame).

We define a rigorous analytical model able to describe the service process from both points of view, i.e. the probability distribution of the service times and of the number of frames served in between two service completion epochs of a tagged station. The results of the model are shown to be quite accurate as opposed to simulations. Simulation software is used also to give evidence of a major result: in spite of the system

memory entailed by the back-off stage of each station (see [4]), service times appear to be essentially uncorrelated. That finding makes the use of regenerative models fully apt to capture the statistics of the service time process.

On the protocol side, we highlight by exploiting the model that very large service times and bursts of interposed frames from other than the tagged source are not negligible. As a matter of example, service times larger than 1 s can be achieved with probability about  $10^{-3}$ . With a similar probability, *hundreds* of frames of other stations can be served in between two consecutive frames served from a same tagged station, with an overall population of 15 stations, i.e. there can be quite long intervals when a tagged station does not receive service at all: hence the burstiness.

It is known that 802.11 DCF gives a preferential treatment to stations that just transmitted successfully. E.g. in [11], short term fairness comparison between CSMA/CA and ALOHA protocols is studied. We make this notion quantitative and give analytical tools to evaluate how it affects service offered by 802.11 DCF. So, we can pinpoint that a major cause of burstiness lies in the very large value of the maximum contention window as compared to the default value of the minimum one (typically, 1023 as opposed to 31). We refer to the maximum contention window as *large* because of a practical (not conceptual or theoretical) remark: a single 802.11 IBSS can be hardly conceived to offer service to more than a few tens of *simultaneous* traffic flows. Although there is no difficulty to evaluate 802.11 analytical models with up to hundreds of stations, when we use them under saturation traffic this turns out as a very unrealistic scenario, to have so many contending, *simultaneously* active stations. Once we recognize that reasonable values of  $n$  are under a few tens, 1023 appears an excessive value for the maximum contention window. By exploiting the model, we evaluate the trade-off between average long term throughput and burstiness of service times in realistic scenarios. We show that a minor degradation of throughput yields a great improvement of the latter. It is well known that variability of service times affects adversely queue performance of backlogged traffic inside stations (e.g. mean queue delays are proportional to the coefficient of variation of the service times). In view of support of real time and streaming services on WLANs, service time excessive jitter is a problem as well. Moreover burstiness in the service process can degrade TCP performance due to ACK compression.

The rest of the paper is organized as follows. In Section 2 modeling assumptions are stated. The transient Markov chains of the analytical model are laid out in Section 3. Section 4 applies the discrete time Markov chain to the analysis of the service times and also presents numerical results. Conclusions are drawn in Section 5.

## 2 802.11 DCF Markov Model

The model of 802.11 DCF is derived under the usual assumptions:

- *Symmetry*: stations are statistically indistinguishable, i.e. traffic parameters (input frame rate, frame length) and multiple access parameters (e.g. maximum retry limit) are the same.
- *Proximity*: every station is within the coverage area of all others, i.e. there are no hidden nodes.
- *Saturation*: stations always have packets to send.

Along with these we introduce two simplifying hypotheses:

- *Independence*: states of different stations are realization of independent random processes.
- *Geometric Back-off*: back-off counter probability distribution is geometric ( $p$ -persistent model of the DCF, [15]).

The last two hypotheses are useful to keep the analytical model simple, hence practical. The first one is essential to allow the use of a one dimensional state space of the Markov chain model. The second one is not essential yet it reduces the number of states at a minimum. Both of them are justified by the more than satisfactory results of the model as compared to simulations. Simulation results are obtained by means of a matlab simulator that reproduces the 802.11 DCF system under the *Symmetry*, *Proximity* and *Saturation* hypotheses.

Let  $m$  be the maximum retry number, i.e. the maximum number of transmission attempts before discarding a frame. Let us consider the Markov chain describing the back-off stage of a tagged station (*Symmetry*) in a full visibility IBSS made up of  $n$  stations. The state of the Markov chain is  $X(t) \in \{0, 1, \dots, m\}$ . Figure 1 depicts the

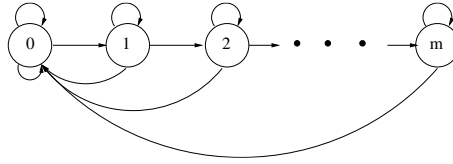


Fig. 1. 802.11 DCF Markov chain

Markov chain of the tagged station evolution. For  $i < m$  a transition from the state  $i$  to the state  $i + 1$  represents the event that the tagged station attempts to access the channel but the transmitted packet collides. If the tagged station does not attempt to access the channel, the state does not change. If the tagged station accesses the channel successfully, a transition from  $i$  to 0 occurs. If  $i = m$ , a transition from  $m$  to 0 occurs, when the tagged station attempts to access the channel (successfully or unsuccessfully). Non null transition probabilities are given by:

$$\begin{aligned} \phi_{i,i} &= 1 - \tau_i \\ \phi_{i,i+1} &= \tau_i(1 - (1 - \tau)^{n-1}) \text{ for } i < m \\ \phi_{i,0} &= \begin{cases} \tau_i(1 - \tau)^{n-1} & \text{for } i < m \\ \tau_m & \text{for } i = m \end{cases} \end{aligned}$$

where  $\tau_i$  is the tagged station access probability conditional on state  $i$  and  $\tau$  is the average other station access probability, independent of the chain state, obtained as  $\tau = \sum_j \pi_j \cdot \tau_j$ , where  $\pi_j$  are the steady state Markov chain probabilities. Considering that the number of embedded epochs the tagged station remains in a state is geometrically distributed,  $\tau_i$  is derived imposing that the average extracted backoff from the

geometric distribution is equal to the average backoff counter extracted from an uniform distribution as specified by IEEE 802.11 standard. So, we let  $\tau_i = 2/(W_i + 1)$  with  $W_i = \min\{CW_{max}, 2^i(CW_{min} + 1) - 1\}$ , for  $i = 0, \dots, m$ . Usual values of  $CW_{min}$  and  $CW_{max}$  are 31 and 1023 respectively.

Solving the Markov chain we obtain that the steady state probability  $\pi_i$  are:

$$\pi_i = \frac{\frac{p^i}{\tau_i}}{\sum_{j=0}^m \frac{p^j}{\tau_j}} = \frac{p^i(W_i + 1)}{\sum_{j=0}^m p^j(W_j + 1)}$$

where  $p$  is the tagged station collision probability conditional on the tagged station transmitting:  $p = 1 - (1 - \tau)^{n-1}$ . It is easily verified that  $\pi_i$  above coincides with the steady state probability of the tagged station staying in back off stage  $i$  as calculated from the two-dimensional Markov chain in [2]. Also the analytic expressions of the average transmission probability  $\tau$  and the average throughput are just the same for the original two-dimensional model and for our simplified one<sup>1</sup>. For these reasons, model validation against simulation results is not reported here.

### 3 802.11 DCF Service Transient Markov Chain

Let  $t_k$  denote the  $k$ -th back-off decrement time; it occurs either after an idle time lasting a slot time or after a transmission attempt followed by a slot time. At each transmission attempt, either a frame is successfully delivered, or a collision occurs<sup>2</sup>. In the former case, a frame has been served, i.e. we have a frame service completion epoch. In the collision case, frame delivery is attempted again after a back off time, except of those frames whose maximum number of attempts has been exhausted. For those frames service is complete as well, although ending up with a failure. Let  $t_k^{(a)}$  be the service completion epochs (either with success or failure) as seen from the overall system point of view, i.e. irrespectively of the specific station that completes its frame service; let also  $t_k^{(s)}$  be the service completion epochs (either with success or failure) as seen by a tagged single station. The sequence  $\{t_k^{(a)}\}$  is obtained by sampling appropriately the full sequence  $\{t_k\}$  and the sequence  $\{t_k^{(s)}\}$  turns out as a proper sampling of the sequence  $\{t_k^{(a)}\}$ . Figure 2 depicts a sample time evolution of the 802.11 medium access at the considered sampling points.

The  $k$ -th service completion times for the tagged station and for the collective ensemble of contending stations are denoted respectively as  $\Theta_{s,k} = t_k^{(s)} - t_{k-1}^{(s)}$  and  $\Theta_{a,k} = t_k^{(a)} - t_{k-1}^{(a)}$ . We assume that at steady state they are distributed as a common random variable:  $\Theta_{s,k} \sim \Theta_s \forall k$  and  $\Theta_{a,k} \sim \Theta_a \forall k$ . In the following we develop a regenerative model of service completions allowing us to compute the statistics of  $\Theta_s$  and  $\Theta_a$ . Such models are based on a variation of the ergodic Markov chain in Section 2.

Figure 3 depicts the two transient Markov chains. The state of the Markov chain is  $Y(t) \in \{0, 1, \dots, m+2\}$ , where the last two states denote failure ( $m+1 \equiv \text{KO}$ ) and

<sup>1</sup> This is even a stronger simplification than the one in [5].

<sup>2</sup> We assume ideal physical channel, so that no frame loss due to receiver errors takes place.

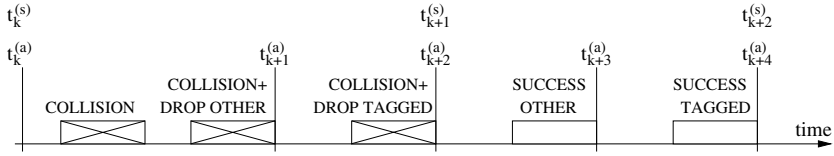


Fig. 2. 802.11 DCF medium access evolution

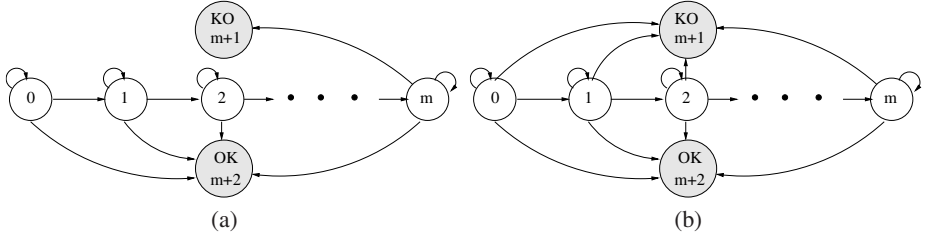


Fig. 3. Single (a) and All (b) stations transient Markov chains

frame delivery success ( $m + 2 \equiv \text{OK}$ ) respectively. We are interested in a transient behavior of the chain where the initial probability vector at time 0 is the  $(m + 1)$ -dimensional vector  $\alpha$  and the last two states are absorption ones<sup>3</sup>.

Let us define some notation.

- $\Phi$  = the one step Markov chain transition probability matrix.
- $\Psi$  = the substochastic submatrix of the one step Markov chain transition probability matrix relevant to the first  $m + 1$  states; it has positive elements only on the diagonal and super-diagonal.
- $\varphi_{m+1}$  and  $\varphi_{m+2}$  = the transition probability vectors from each of the transient states into the absorption states  $m + 1$  (i.e. failure of delivery, packet drop due to maximum retry limit) and  $m + 2$  (i.e. success of delivery) respectively.
- $\mathbf{D}_1 = \text{diag}[\varphi_{m+1}]$ ,  $\mathbf{D}_2 = \text{diag}[\varphi_{m+2}]$  and  $\mathbf{D} = \mathbf{D}_1 + \mathbf{D}_2$ .

Let  $T$  be the number of transition to absorption, given that the initial probability distribution is  $[\alpha \ 0 \ 0]$ . Then, it can be verified that:

$$\mathcal{P}(T = t; S_T = j; A = m + k) = \alpha \Psi^{t-1} \mathbf{D}_k \quad t \geq 1, j = 0, 1, \dots, m, k = 1, 2$$

where  $S_T$  is the transient state from which absorption occurs,  $A$  is the resulting absorption state. The marginal distribution of each of these variables can be obtained easily. In particular, the probability distribution of  $T$  is given by  $f_T(t) = \alpha \Psi^{t-1} \mathbf{D} \mathbf{e}$ ,  $t \geq 1$ , where  $\mathbf{e}$  is a column vector of 1's of size  $m + 1$ .

There remains to identify the vector  $\alpha$  and the values of the entries of the one step transition probability matrix  $\Phi$ . They both depend on the subset of the back-off counter decrements times we consider.

<sup>3</sup> We assume initialization cannot take place directly in one of the two absorption states.



### 3.1 Tagged Station Service Transient Markov Chain

As far as regards the Markov chain related to the time series  $\{t_k^{(s)}\}$ , the associated Markov chain is depicted in Figure 3(a). The transition probability  $\varphi_{i,i}$  is the probability that the tagged station remains in the state  $i$ -th:

$$\varphi_{i,i} = 1 - \tau_i$$

A state transition from  $i$  to  $i+1$  and from the state  $m$  to the state  $m+1$  (i.e. transmission failure absorption state) occurs when the tagged station attempts to access the channel, but at least one of the other stations collides with it:

$$\varphi_{i,i+1} = \tau_i(1 - (1 - \tau)^{n-1}), \quad i = 0, 1, \dots, m$$

The transition probability from state  $i$  to  $m+2$  (i.e. transmission success absorption state) is:

$$\varphi_{i,m+2} = \tau_i(1 - \tau)^{n-1}, \quad i = 0, 1, \dots, m$$

All other entries of the matrix  $\Phi$  are null, except of  $\varphi_{m+1,m+1} = \varphi_{m+2,m+2} = 1$ .

The initial probability vector  $\alpha$  is  $[1, 0, \dots, 0]$ , since tagged station always restarts from the backoff stage 0, both after a successfully transmission and after a packet drop due to the maximum retry limit  $m$ .

### 3.2 All Stations Service Transient Markov Chain

The derivation of the Markov chain transition probabilities related to the time series  $\{t_k^{(a)}\}$  is more involved. In this case we assume that a station other than the tagged one is in the state  $m$  with probability  $\pi_m$  derived from the ergodic Markov chain in Section 2. With probability  $\pi_m$ , in fact a station is in its last backoff stage and if a further collision occurs the packet is dropped because the maximum retry limit  $m$  is reached. Supposing that other stations are independent of one another, the transition probabilities  $\varphi_{i,j}$ s are given by:

$$\varphi_{i,i} = (1 - \tau_i)(1 - \tau)^{n-1} + (1 - \tau_i) \sum_{k=2}^{n-1} \binom{n-1}{k} \tau^k (1 - \tau)^{n-1-k} (1 - \pi_m)^k$$

This probability is composed by two components; the first one reflects the case no station accesses the channel, the second one the case the tagged station does not access the channel and two or more of the other stations access the channel but none of the transmitted packets is discarded (none of the transmitting stations is in the  $m$  backoff stage).

With probability  $\varphi_{i,m+1}$ , the Markov chain is absorbed in the  $m+1$  state. In this case, we have to distinguish between the case the tagged station is in the last backoff stage or in the other stages. In the first case the transition probability is given by:

$$\varphi_{m,m+1} = \tau_m[1 - (1 - \tau)^{n-1}] + (1 - \tau_m) \sum_{k=2}^{n-1} \left\{ \binom{n-1}{k} \tau^k (1 - \tau)^{n-1-k} [1 - (1 - \pi_m)^k] \right\}$$

i.e., when the tagged station does not access the channel and two or more of the other stations access the channel and at least one of the transmitted packets is discarded or when the tagged station accesses the channel and at least one of the other stations collides with it.

When the backoff stage of the tagged station is lower than  $m$ , the transition probability is given by:

$$\begin{aligned} \varphi_{i,m+1} = (1 - \tau_i) \sum_{k=2}^{n-1} \left\{ \binom{n-1}{k} \tau^k (1 - \tau)^{n-1-k} [1 - (1 - \pi_m)^k] \right\} + \\ \tau_i \sum_{k=1}^{n-1} \left\{ \binom{n-1}{k} \tau^k (1 - \tau)^{n-1-k} [1 - (1 - \pi_m)^k] \right\} \end{aligned}$$

In this case, the transition occurs if: i) the tagged station does not access the channel and two or more of the other stations access the channel and at least one of the transmitted packets is discarded or ii) the tagged station accesses the channel, at least one of the other stations collides with it and one of the colliding stations (among other stations) is in the  $m$  state.

The transition probability  $\varphi_{i,i+1}$  for  $i < m$  is given by:

$$\varphi_{i,i+1} = \tau_i \sum_{k=1}^{n-1} \left\{ \binom{n-1}{k} \tau^k (1 - \tau)^{n-1-k} (1 - \pi_m)^k \right\} \text{ for } 0 \leq i < m$$

i.e. the tagged station collides but no colliding packet is discarded.

A transition to the  $m + 2$  state (OK state) occurs when a successful transmission events occurs:

$$\varphi_{i,m+2} = \tau_i (1 - \tau)^{n-1} + (1 - \tau_i)(n - 1)\tau(1 - \tau)^{n-2}$$

Otherwise the transition probability is 0. For this Markov chain, finding the vector  $\alpha$  is more involved; however, since it is not used in the remainder of the paper, its derivation is not reported.

## 4 802.11 DCF Service Time Characterization

We want to characterize the burstiness in the 802.11 DCF service time. Three different metrics are defined: i) the tagged station service time distribution, ii) the distribution of the number of service completions of stations other than the tagged one between two consecutive tagged station service completions; iii) correlation between the tagged station service intervals and the correlation between all stations service intervals.

Exploiting previous models, we are able to fully characterize the first two issues, whereas the third one cannot be analyzed with renewal process models, since, by definition, they are not able to reproduce correlated events. However, by means of simulation results we verify that correlation between consecutive samples is very low. The absolute value of the autocorrelation at time lag 1 of sequences  $\{t_k^{(a)}\}$  and  $\{t_k^{(s)}\}$  is always below 10% and decreases quickly when the time lag increases.

#### 4.1 Service Time Distribution

Up to now, we confined ourselves to the realm of embedded Markov epochs, to obtain the probability distribution of the absorption time  $T$  in terms of number of embedded points. If each transition takes a different time and we are interested in the overall actual time (not just number of transitions), we need to de-normalize the probability distribution of  $T$ .

Let then  $f_{i,j}(s)$  be the Laplace transform of the probability density function of the time required to make a transition from state  $i$  to state  $j$  and let  $\mathbf{G}(s)$  be the  $(m+3) \times (m+3)$  matrix whose entry  $g_{i,j}(s)$  is  $f_{i,j}(s)\varphi_{ij}$ ,  $i, j = 0, 1, \dots, m+2$ ; note that  $\mathbf{G}(1) = \mathbf{\Phi}$ . Let also  $\mathbf{D}_{m+k}(s) = \text{diag}[g_{0,m+k}(s) \ g_{1,m+k}(s) \ \dots \ g_{m,m+k}(s)]$  for  $k=1,2$ . It can be checked that

$$f_{\Theta}(s; S_T = j; A = m+k) = \alpha [\mathbf{I} - \mathbf{G}(s)]^{-1} \mathbf{D}_{m+k}(s) \quad (1)$$

for  $j = 0, 1, \dots, m, k = 1, 2$ , where  $f_{\Theta}(s; S_T = j; A = m+k)$  is the Laplace transform of the probability density function of the absorption time  $\Theta$  with absorption from  $j$  towards  $m+k$ , i.e. the time required to complete a transmission cycle ending up with a failure ( $k = 1$ ) or a success ( $k = 2$ ) and leaving the state of the tagged station at stage  $j$ .

The Laplace transform of the probability density function of the absorption time is found by summing up over  $j$  and  $k$  in eq. (1). Then

$$f_{\Theta}(s) = \alpha [\mathbf{I} - \mathbf{G}(s)]^{-1} [\mathbf{g}_{m+1}(s) + \mathbf{g}_{m+2}(s)] \quad (2)$$

where  $\mathbf{g}_{m+k}(s) = \mathbf{D}_{m+k}(s)\mathbf{e}$ ,  $k = 1, 2$ .

Let now consider the Markov chain that represents the service completion times of the tagged station, i.e. the time it takes for a tagged station frame to be successfully delivered or discarded because of exceeding the number of retransmission attempts. Then, we have  $\alpha = [1, 0, \dots, 0]$  and the entries of the matrix  $\mathbf{\Phi}$  are as in Section 3.1. The forward transitions, i.e. those from state  $k$  to state  $k+1$  ( $k = 0, 1, \dots, m-1$ ), require the time to perform a collision,  $T_c$ , which is constant if we assume a constant data frame payload length; therefore,  $g_{k,k+1}(s) = \varphi_{k,k+1}e^{-sT_c}$ ,  $k = 0, \dots, m-1$ . The time of the loop transition of each of the transient state has the same probability distribution: let  $h(s)$  be its Laplace transform; then  $g_{k,k}(s) = h(s)\varphi_{k,k}$ ,  $k = 0, \dots, m$  with  $h(s) = [p_e e^{-s\delta} + p_s e^{-sT_s} + p_c e^{-sT_c}]$ , where  $p_e$ ,  $p_s$  and  $p_c$  are the probabilities that the stations other than the tagged one stay idle, transmit a frame with success or make a collision, respectively. Finally, the time required for a transition towards the absorbing state  $m+2$  (success) is always equal to  $T_s$ , i.e. the time for a successful frame transmission and acknowledgment<sup>4</sup>; the time related to the transition to the absorbing state  $m+1$  (failure) is instead equal to  $T_c$  and only occurs from state  $m$ .

The inverse matrix in eq. (2) can be explicitly calculated by exploiting the special structure of  $\mathbf{\Phi}$  and hence of  $\mathbf{G}(s)$ :

$$f_{\Theta_s}(s) = \sum_{j=0}^m \left[ \frac{1}{1 - h(s)\varphi_{j,j}} \prod_{k=0}^{j-1} \frac{e^{-sT_c}\varphi_{k,k+1}}{1 - h(s)\varphi_{k,k}} \right] [e^{-sT_s}\varphi_{j,m+2} + e^{-sT_c}\varphi_{j,m+1}] \quad (3)$$

<sup>4</sup> We adopted 802.11b DCF mode standard parameter values with 11 Mbps data rate such that  $\delta=20 \mu s$  and  $T_c=T_s=1.589 ms$  corresponding to a 1500 bytes payload.

Moments of  $\Theta_s$  can be found by deriving  $f_{\Theta_s}(s)$ .  $f_{\Theta_s}(s)$  can be numerically inverted by using standard methods (e.g., see [14]). Figure 4 depicts the complementary cumulative distribution function (CCDF) of the service delay obtained by inverting Eq. (3) (Figure 4(a)) and the empirical CCDF obtained by means of simulations (Figure 4(b)) when the number of competing stations  $n$  is 15. When the max retry limit is low, the model is not able to reproduce successfully the service delay statistics in the considered probability range since the impact of the geometric backoff assumption dominates the delay statistics. When  $m$  increases, the model reproduces successfully the CCDF. From the analysis of the CCDF, we note that the variability of the service time is quite high. E.g, when  $m$  is 7 (standard 802.11 retry limit), 1 packet over 1000 experiences a delay higher than 1 second indicating a high level of dispersion of the service delay.

Figure 5(a) depicts the ratio  $\mu$  between the variance of  $\Theta_s$  and the squared mean of  $\Theta_s$  obtained by means of Eq. (3) and as derived from simulation results;  $\mu$  is a good

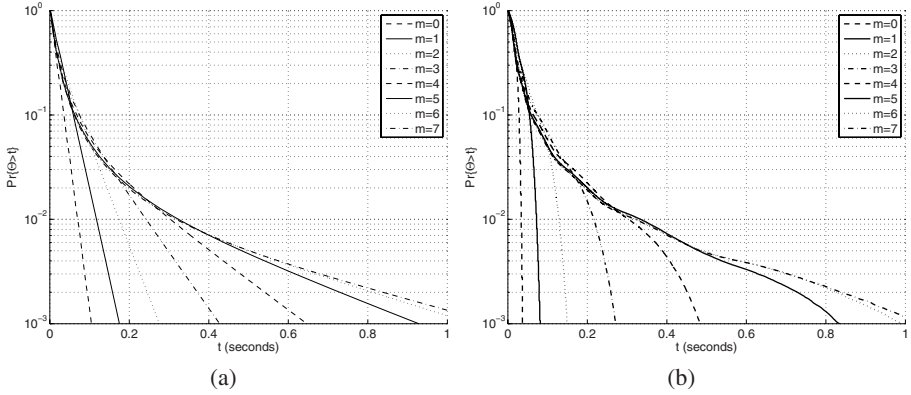


**Fig. 4.** Analytical (a) and Simulative (b) CCDF of the service delay for  $n=15$  and  $CW_{max} = 1023$  varying the maximum retry limit  $m$

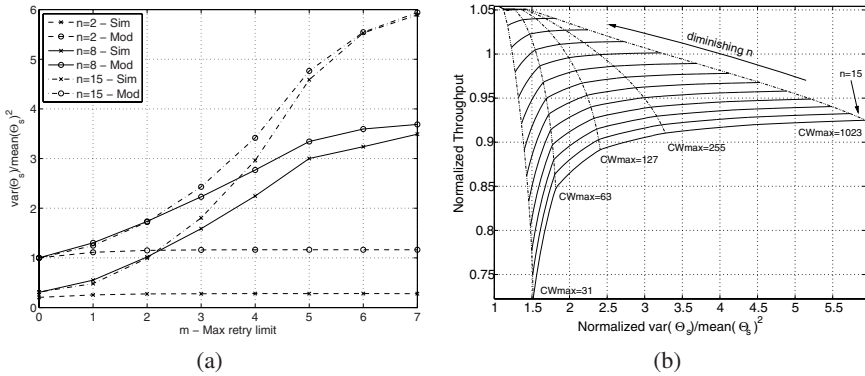


**Fig. 5.** Coefficient of variation of the service time,  $\mu$  - Validation against simulative results (a) and Tradeoff against normalized throughput (b)

indication of the dispersion degree of the service delay with respect to the mean service delay. Even in this case, discrepancies between model and simulation results are due to the geometrical distribution assumption that leads to an overestimation of the variance of the service time. The error vanishes as the maximum retry limit gets closer to realistic values (standard default is  $m=7$ ).

Figure 5(b) plots the tradoff between performance penalty and service time jitter obtained by varying  $CW_{max}$  for all values of  $n \in [2, 15]$ . Performance are measured as long term average throughput normalized with respect to the case  $n=1$  and  $CW_{max}=1023$ . Jitter is measured by the squared coefficient of variation of the service time  $\mu$  normalized with respect to  $\mu$  in case  $n=1$  and  $CW_{max}=1023$ . Dashed lines through the graphs join together points where  $CW_{max}$  has the same value, from 31 to 1023. The key result is that most of the right portion of the curves is almost flat, pointing out that reducing even substantially the service time jitter can be achieved in spite of a minor throughput degradation. By decreasing  $CW_{max}$ , it is possible to limit the dispersion of the service delay around the mean service delay, without reducing drastically the mean throughput. As a matter of example, using  $CW_{max}=127$  instead of 1023 when  $n$  is 15, the delay dispersion is more the halved against a throughput reduction of about 4%.

## 4.2 Burstiness

As a further step, we characterize the burstiness of the service process. We exploit the Markov chain related to the time series  $\{t_k^{(a)}\}$  to evaluate the distribution of the number of service completion of stations other than the tagged one between two successive time epochs of the sequence  $\{t_k^{(s)}\}$ , i.e. two successive tagged station service completions.

Let  $\mathbf{D}_o$  and  $\mathbf{D}_s$  two diagonal matrices; the  $i$ -th diagonal element of  $\mathbf{D}_o$  ( $\mathbf{D}_s$ ) is defined as the probability that stations other than tagged one end up with their services (the tagged station terminates its service) conditional on the tagged station being in state  $i$ . Then the  $i$ -th element of the vector  $\alpha(\mathbf{I} - \Psi)^{-1}\mathbf{D}_o$  is the probability that stations other than the tagged one terminates their service with the tagged station in state  $i$ ; this is just the probability distribution of the state of the tagged station left over by a service termination of another station. Therefore the probability that  $k$  services of other stations occur before the tagged station is served, i.e., between two services of the tagged station, is given by:

$$q_k = \alpha[(\mathbf{I} - \Psi)^{-1}\mathbf{D}_o]^k(\mathbf{I} - \Psi)\mathbf{D}_s\mathbf{e} \text{ for } k \geq 0 \quad (4)$$

Figure 6(a) depicts  $q_k$  derived from the analytical model against simulation results, for different values of  $n$  and  $m$  fixed to 7. When the number of competing stations  $n$  is large enough, the model is able to reproduce the burstiness level of the service process. When  $n$  is 2, the model does not reproduce correctly  $q_k$ ; in this scenario, the independence hypothesis does not hold, since the two competing stations' evolutions are correlated. Figure 6(b) depicts  $q_k$  obtained by means of the analytical model, for different values of  $CW_{max}$ , for  $m = 7$  and  $n = 15$ . Moreover the solid bold line depicts  $q_k$  when using a short term fair random scheduler, i.e. a random scheduler that chooses the next served station independently of previous served stations with the same probability; in this case  $q_k = (1 - 1/n)^k 1/n$ . The figure highlights that decreasing

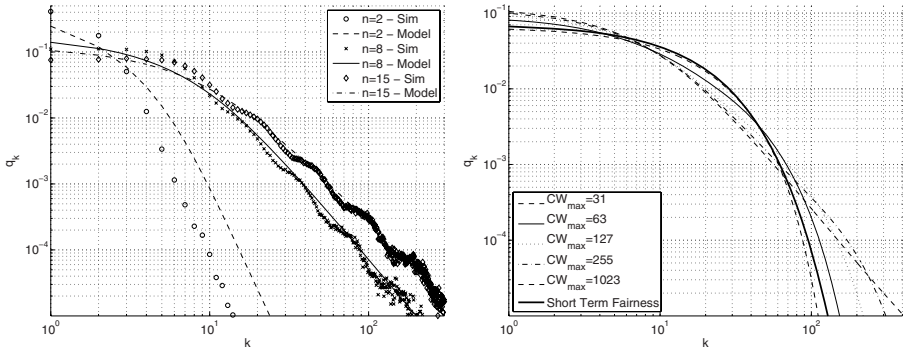


**Fig. 6.** Probability that  $k$  services of other stations occur between two services of the tagged station,  $q_k$  - Comparison between model and simulative results (a) and model results (b)

$CW_{max}$ , the 802.11 behaves as a short term fair scheduler. The heavier right tail of the 802.11 scheduling distribution with large values of  $CW_{min}$  highlights the burstiness of the 802.11 service process.

## 5 Final Remarks

The service times burstiness of IEEE 802.11 DCF is analyzed in this work. A simple analytical model able to describe the service process from both an individual station point of view (tagged station service completions) and from a collective standpoint (all stations service completions) is derived. The model is exploited to obtain the service time distribution of the tagged station service process and the probability density function of the number of services of other stations (different from tagged one) that occur between two services of the tagged station.

Analytical results highlight the impact of the maximum contention window  $CW_{max}$  on the service process burstiness. On the one hand, reducing  $CW_{max}$  has a strong impact on the service time burstiness that decreases significantly, on the other hand the system throughput degrades since the collision probability increases. However analytical results show that the impact on the system throughput is negligible with respect to the gain in the service burstiness reduction.

## References

1. G. Bianchi, "IEEE 802.11 Saturation throughput analysis," IEEE Communications Letters, Vol. 2, no. 12, December 1998, pp. 318-320.
2. G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Function," IEEE Journal of Selected Areas in Telecommunications, Vol. 18, No. 3, pp. 535-547, Mar. 2000.
3. G. Bianchi and I. Tinnirello, "Remarks on IEEE 802.11 DCF performance analysis," IEEE Communications Letters, Vol. 9, no. 8, August 2005, pp. 765-767.
4. G. Sharma, A.J. Ganesh, and P.B. Key, "Performance Analysis of Contention Based Medium Access Control Protocols," IEEE INFOCOM 2006, Barcelona (Spain), 23-28 April, 2006.

5. M. Garetto and C.F. Chiasserini, "Performance analysis of the 802.11 Distributed Coordination Function under sporadic traffic," *proc. of Networking 2005*, Waterloo, Canada, May 2005.
6. G. Wang, Y. Shu, L. Zhang, O.W.W. Yang, "Delay analysis of the IEEE 802.11 DCF", *proc. of the 14th International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC 2003*, September 7-10, 2003, Beijing, China.
7. P. E. Engelstad, O. N. Osterbo, "Analysis of the Total Delay of IEEE 802.11e EDCA and 802.11 DCF," in *proc. of IEEE ICC 2006*, Istanbul, June 2006.
8. Boris Bellalta, Michela Meo, Miquel Oliver, "Comprehensive Analytical Models to Evaluate the TCP Performance in 802.11 WLANs," in *proc. of WWIC 2006*, pp. 37-48.
9. J. Yu, S. Choi, "Modeling and analysis of TCP dynamics over IEEE 802.11 WLAN," in *proc. WONS 2007*, Obergurgl, Austria.
10. D. Miorandi, A. Kherani and E. Altman, "A queueing model for HTTP traffic over IEEE 802.11 WLANs," *Computer Networks*, Vol. 50-1, 2006, pp. 63-79.
11. C. Koksall, H. Kassab and H. Balakrishnan, "An analysis of short term fairness in wireless media access protocols," in *Proc. ACM SIGMETRICS*, Santa Clara, CA, June 2000.
12. O. Tickoo and B. Sikdar, "On the impact of the IEEE 802.11 MAC on traffic characteristics," *IEEE Journal on Sel. Areas in Comm.*, Vol. 21, no. 2, February 2003, pp. 189-203.
13. IEEE std 802.11, "Wireless LAN Media Access Control (MAC) and Physical layer (PHY) Specifications," 1999 <http://standards.ieee.org/getieee802/>.
14. J. Abate and W. Whitt, "The Fourier-series method for inverting transforms of probability distributions," *Queueing Systems* 10, pp. 5-88, 1992.
15. F. Cali, "Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit," *IEEE Transaction on Networking*, Vol. 8(6), pp. 785-799, Dec. 2000.

# ID-Based Multiple Space Key Pre-distribution Scheme for Wireless Sensor Networks

Tran Thanh Dai and Choong Seon Hong\*

Networking Lab, Department of Computer Engineering, Kyung Hee University, Korea  
daitt@networking.khu.ac.kr, cshong@khu.ac.kr

**Abstract.** Providing security services for wireless sensor networks plays a vital role in secure network operation especially when sensor networks are deployed in hostile areas. In order to pave the way for these mechanisms, cryptographic keys must be agreed on by communicating nodes. Unfortunately, due to resource constraints, the key agreement problem in wireless sensor networks becomes quite intricate. To deal with this problem, many public-key unrelated proposals have been proposed so far. One prominent branch of these proposals is based on random key pre-distribution. Inspired by this trend, in this paper, we propose a new random key pre-distribution scheme that is comparable to Du et al.'s scheme [2] in terms of network resiliency and memory usage. On the other hand, our later analysis shows that our scheme outperforms Du et al.'s scheme in terms of computational and communication overhead.

**Keywords:** ID-based, random key pre-distribution, key agreement, security, wireless sensor networks.

## 1 Introduction

A typical wireless sensor network (WSN) may contain a large number of microsensor nodes, which are connected by a wireless medium, controlled and managed by one or several powerful control nodes (often called base stations). These sensor nodes are tiny in size and capable of capturing various physical properties, such as temperature, humidity, or pressure, and mapping the physical characteristics of the environment to quantitative measurements. The captured and pre-processed information is delivered to base stations as well as other nodes through immediate neighboring nodes. WSNs encourage several novel and existing applications such as environmental monitoring; health care; infrastructure management; public safety; medical; home and office security; transportation; and military.

Deployment of a WSN can be in random fashion (e.g., scattered from an airplane) or planted manually. When being deployed in hostile environments, WSNs are vulnerable to different kinds of malicious attacks. In such contexts, providing security services based on solving the key agreement problem becomes one of the major concerns. Unfortunately, due to resource constraints, the key agreement problem in

---

\* This work was supported by MIC and ITRC Project. Dr. CS Hong is corresponding author.



WSNs becomes quite intricate. To deal with this problem, many public-key unrelated proposals which are considered more reasonable in cost than public key based approaches have been proposed so far. One prominent branch of these proposals is based on random key pre-distribution [2], [4], [5], [6]. Another outstanding branch is ID-based key pre-distribution schemes [1], [11] which have the following properties: (i) No previous communication is required; (ii) Key pre-distribution procedure consists of simple computations; (iii) In order to establish the key, each party should input its partner's identifier only into the secret key sharing function.

Inspired by these observations, in this paper, we propose a highly resilient, robust, resource-efficient, and ID-based random key pre-distribution scheme. On the one hand, our scheme as being analyzed later is much like Du et al.'s scheme [2] (*Du's scheme* for short) in terms of network resiliency with the same memory cost. In other words, when the number of compromised nodes is less than a threshold, the probability that any nodes except these compromised nodes is security influenced is negligible. This property means that an attacker's gain is decreased for small scale network breach and this gain has a significant security impact only when the attacker mounts a successful attack on a considerable proportion of the network which is considered to be detected easily. On the other hand, our scheme significantly improves resource usage in terms of computational and communication overhead compared to Du's scheme.

The rest of this paper is organized as follows: section 2 mentions the related work; section 3 summarizes our keystone, the Matsumoto-Imai scheme; section 4 describes our ID-based random key pre-distribution scheme; section 5 analyzes the resiliency of our scheme against node capture attack; section 6 presents the performance analysis in terms of memory usage, communication overhead, and computational overhead; section 7 concludes the paper and states our future work.

## 2 Related Work

In this section, we briefly review several noticeable random key pre-distribution schemes for WSNs that have been published in the literature so far.

Eschenauer et al. [4] are the first to propose a random key pre-distribution scheme that relies on probabilistic key sharing among the nodes of a DSN. The main idea is that a random pool of keys is selected from the key space. Each sensor node then receives a random key ring from the key pool before deployment. After deployment, any two neighboring nodes able to find a common key within their respective key rings using *shared-key discovery phase* can use that key as their shared secret to initiate communication and to set up the secure connection. In the case that those nodes could not find a common key, they can resort to *path-key establishment phase* to solve the key agreement issue.

Chan et al. [5] further exploited the idea in [4] to developed three mechanisms for key establishment using the framework of pre-distributing a random set of keys to each node. The first one is q-composite keys scheme. This scheme is mainly based on [4]. The difference between this scheme and [4] is that q common keys, instead of just a single one, are needed to establish secure communication between a pair of nodes. The second one is multi-path key reinforcement scheme applied in conjunction with [4] to yield greatly improved resiliency against node capture attack by trading off

some network communication overhead. The third one is random pairwise keys scheme. The purpose of this scheme is to allow node-to-node authentication between communicating nodes.

Du et al. [2] presented a multiple space key pre-distribution scheme for wireless sensor networks. This scheme first uses Blom's key generation scheme [12] as a building block to generate multiple key spaces, a pool of tuple  $(D, G)$ , where matrices  $D$  and  $G$  are as defined in Blom's scheme. Then this pool is used as a pool of keys as in [4] to establish a common secret key between any pair of nodes.

Chan et al. [13] proposed a variant of random key pre-distribution scheme for key agreement problem in the clustered DSN. Accordingly, the DSN is sub-grouped into clusters. Different clusters in different regions are assigned different probabilities of node compromise based on the hostile level of those regions. Within each cluster, the scheme in [4] is applied. This scheme is claimed to isolate the effect of node compromise into one specific subgroup and offer an effective scalable security mechanism that increases the resiliency to the attacks on the sensor subgroups.

### 3 Matsumoto-Imai Scheme (MI Scheme)

First of all, in this paper, we assume that each sensor node has a unique identification whose range is from 1 to  $N$  where  $N$  is the maximum number of deployable nodes. Each of the unique identifications is represented by  $m = \log_2(N)$  bit effective ID in sensor nodes' memory.

This section explains how the sensor nodes' secret keying material is generated and how sensor nodes use this material to establish pairwise keys in the manner of the MI scheme [1].

A central server first generates  $l(m \times m)$  symmetric matrices  $M_\omega$ s over finite field  $GF(2)$ . These  $M_\omega$ s are kept secret and must not be disclosed to both attackers and sensor nodes.  $M_\omega$  is used to generate the  $\omega$ -th bit of a pairwise key between any pair of neighboring nodes, so  $l$  is the length of this key. The central server then computes the keying material for each node  $S_i$  as follows:

$$\Phi_i^\omega = y_i M_\omega \quad (\omega = \overline{1, l}) \quad (1)$$

$$\Phi_i = [\Phi_i^1 \quad \Phi_i^2 \quad \dots \quad \Phi_i^l]^T \quad (2)$$

where  $y_i$  ( $i = \overline{1, N}$ ) is the  $m$ -dimensional vector, effective ID of node  $S_i$ .  $\Phi_i$  needs to be kept secret in the node  $S_i$  and should not be disclosed to attackers or other sensor nodes.

Therefore, when nodes  $S_i$  and  $S_j$  need to find the pairwise key between them, they first exchange their effective IDs  $y_i$  and  $y_j$  respectively, then use  $\Phi_i$  and  $\Phi_j$  to compute their pairwise key as follows:

$$S_i: \quad K_{ij}^\omega = \Phi_i^\omega y_j^T \quad (\omega = \overline{1, l}), \quad K_{ij}^T = \Phi_i y_j^T \quad (3)$$

$$S_j: \quad K_{ji}^\omega = \Phi_j^\omega y_i^T \quad (\omega = \overline{1, l}), \quad K_{ji}^T = \Phi_j y_i^T \quad (4)$$

where symbol  $T$  denotes transposition operation. Fig. 1 illustrates how the pairwise key  $K_{ij} = K_{ji}$  is generated.

This scheme has a noteworthy property that as long as no more than  $m-1$  nodes are compromised, the entire network is theoretically secure. In other words, an attacker needs to compromise at least  $m$  nodes in order to compute any pairwise key of any two uncompromised neighboring sensor nodes using their effective IDs.

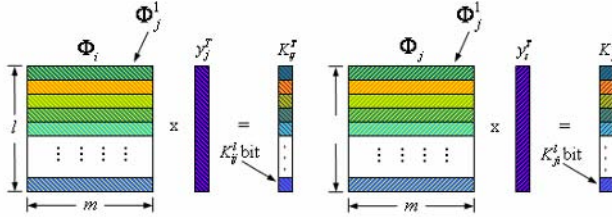


Fig. 1. Pairwise key generating in MI scheme

## 4 Our ID-Based Multiple Space Key Pre-distribution Scheme

To enhance network resiliency against node capture attack, we propose an ID-based key pre-distribution scheme that uses IM scheme as a keystone in combination with the idea of multiple key spaces proposed in [2]. Accordingly, the entire network is depicted in the graph theory language. There is an edge between two neighboring sensor nodes (two vertices in graph theory) if and only if they can establish a pairwise key between themselves. Using IM scheme we guarantee to create a complete graph. To obtain our aim of key agreement and enhance resilience, all we need is a connected graph, rather than a complete graph since the latter is a very wasteful use of security.

Some terms need to be clarified before detailing our proposed scheme. We define a key space  $\Omega_i$  as a 3-tuple  $(M_i, l, m)$  of  $l$  ( $m \times m$ ) matrices  $M_{i\omega}$ , where  $M_{i\omega}$  is defined as in IM scheme. A node is said to choose a key space  $\Omega_i$  if it carries the secret information generated from  $\Omega_i$  using MI scheme. Two nodes can derive their pairwise key if they have a key space in common.

### 4.1 Keying Information Pre-distribution Phase

During this phase, we have to pre-distribute keying material to each node such that after deployment neighboring nodes can derive a pairwise key between them using this material. We also select the security parameters  $\mu$ ,  $\lambda$ , and  $m$ , where  $2 \leq \mu < \lambda$ . these parameters are chosen with the security and performance in mind which will be discussed later. This phase is performed as follows:

**Step 1 (Generating 3-tuples  $(M_i, l, m)$ ).** A central server generates  $\lambda$  key spaces. Each key space  $\Omega_i$  consists of  $l$  ( $m \times m$ ) symmetric matrices  $M_{i\omega}$ s as defined in IM scheme.

**Step 2 (Generating  $\Phi_i$  matrices).** We randomly choose  $\mu$  distinct key spaces from  $\lambda$  key spaces for each node. For each space  $\Omega_i$  chosen by node  $S_j$ , we first compute keying material  $\Phi_{ji}$  using equations (1), (2) and then store  $\Phi_{ji}$  at this node. Therefore, each node  $S_j$  has  $\mu$  distinct values of  $\Phi_{ji}$ s. Using MI scheme; two nodes can derive a pairwise key if they have both chosen a common key space.

## 4.2 Pairwise Key Establishment Phase

After deployment, each node needs to discover whether it shares any key space with its neighbors. To do so, each node instantly broadcasts a message containing the following information: the node's effective ID and the indices of the key spaces it carries.

Suppose that nodes  $S_i$  and  $S_j$  are neighbors, and then they have received the above-mentioned broadcast messages. If they figure out that they have an identical index of a key space (or identical key space  $\Omega_s$ ), they can easily compute their pairwise key using equations (3) and (4) of MI scheme respectively. Conversely, there is the case that two nodes who even are neighbors could not establish a pairwise key. To tackle this problem, there are two possible methods that can be used. The first one has already presented in [2]. However this method is vulnerable to node capture attack. Specifically once an attacker can successfully compromise one node on the key path during the key path process, the promising pairwise key  $K$  will be disclosed. The second one is a combination of the  $(k, n)$  secret sharing method [3] and disjoint path finding methods. Accordingly,  $S_i$  first discovers the secured disjoint paths to  $S_j$  and then uses the secret sharing method to split  $K$  into pieces. After that, each piece is sent on one of the secured disjoint paths as described in [2]. Finally,  $K$  can be re-constructed if  $S_j$  receives no less than  $k$  pieces. For fairness, when making comparison with Du's scheme, only the first method is taken into account.

## 4.3 Selecting $\mu, \lambda$

The problem here is that given the size and the density of a network, how we can select the values for  $\mu$  and  $\lambda$  such that the entire network is securely connected with high probability  $P_{gc}$ ? The approach is that we first compute  $P_{rlc}$  (the required probability of two neighboring nodes sharing at least one key space in order to obtain the desired global connectivity  $P_{gc}$ ). Then we compute  $P_{alc}$  (the actual probability of two neighboring nodes sharing at least one key space) using  $\mu$  and  $\lambda$ . Afterward, the values of  $\mu$  and  $\lambda$  could be found to satisfy the following inequality

$$P_{alc} \geq P_{rlc} \quad (5)$$

Using the result shown in [4], we can obtain the expected degree of a node  $d$  (i.e., the average number of secure links between that node and its neighbors) in order to achieve a given value of  $P_{gc}$  when  $N$  is large:

$$d = (N-1) \left[ \frac{\ln(N) - \ln(-\ln(P_{gc}))}{N} \right] \quad (6)$$

Using a given density of sensor network deployment and wireless connectivity constraints, the expected number of a node's neighbors  $n$  can be estimated. Therefore, the value of  $P_{rlc}$  can be estimated as:

$$P_{rlc} = \frac{d}{n} \quad (7)$$

After the values of  $\mu$  and  $\lambda$  have been selected, the actual probability  $P_{alc}$  is determined as follows. Since  $P_{alc} = 1 - P[\text{two neighbors do not share any key space}]$ , we have:

$$P_{alc} = 1 - \frac{\binom{\lambda}{\mu} \binom{\lambda - \mu}{\mu}}{\binom{\lambda}{\mu}^2} = 1 - \frac{((\lambda - \mu)!)^2}{(\lambda - 2\mu)! \lambda!} \quad (8)$$

For better visualization, we plot the values of  $P_{alc}$  in Fig. 2 where  $\lambda$  varies from  $\mu$  to 100 and  $\mu = 2, 4, 6, 8$ .

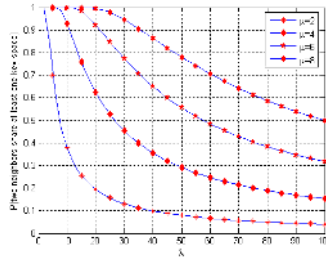


**Fig. 2.** Probability of finding at least one common key space between two nodes when  $\mu$  spaces are randomly chosen from  $\lambda$  spaces

Combining equations (5), (6), (7), and (8), we easily derive the following inequality:

$$1 - \frac{((\lambda - \mu)!)^2}{(\lambda - 2\mu)! \lambda!} \geq (N - 1) \left[ \frac{\ln(N) - \ln(-\ln(P_{gc}))}{nN} \right] \quad (9)$$

Correspondingly, to obtain a certain  $P_{gc}$  of the entire network connectivity with size  $N$  and the expected number of neighbors for each node  $n$ , all we have to do is selecting values of  $\mu$  and  $\lambda$  such that inequality (9) is satisfied.

## 5 Security Analysis

In this section, we evaluate our proposed scheme concerning its resiliency against node capture. Our evaluation is conducted by finding the answer to two questions:

(i) Given that  $b$  nodes are captured, what is the probability that at least one key space is broken? To successfully break one key space, an attacker needs to capture at least  $m$  nodes that contain this key space's keying material. Hence, the answer to this question quantitatively shows when the network starts to become insecure. (ii) Given that  $b$  nodes are captured, what fraction of the additional communications (communications among un-captured nodes) also becomes compromised? The answer to this question shows how much payoff an attacker can obtain after having captured a certain number of nodes.

### 5.1 Probability of at Least One Key Space Being Broken

Let  $B_i$  denote the event that key space  $\Omega_i$  is broken, where  $i = \overline{1, \lambda}$ , and  $C_b$  denote the event that  $b$  nodes are captured in the network. Moreover, let  $B_i \cup B_j$  denote the joint event that either  $\Omega_i$  or  $\Omega_j$ , or both is broken. Thus, we have

$$P(\text{at least one space is broken} | C_b) = P(B_1 \cup B_2 \cup \dots \cup B_\lambda | C_b).$$

Since,  $P(B_1 \cup B_2 \cup \dots \cup B_\lambda | C_b) \leq \sum_{i=1}^{\lambda} P(B_i | C_b)$  and owing to the fact that each key space has an equal chance to be broken,  $\sum_{i=1}^{\lambda} P(B_i | C_b) = \lambda P(B_1 | C_b)$ . Hence,

$$P(\text{at least one space is broken} | C_b) \leq \lambda P(B_1 | C_b). \quad (10)$$

Our task now is reduced to calculate  $P(B_1 | C_b)$  - the probability of key space  $\Omega_1$  being compromised when  $b$  nodes are compromised. The probability that each compromised node contains information about  $\Omega_1$  is  $\rho = \frac{\mu}{\lambda}$ . Let  $X$  denote the number of compromised nodes containing information about  $\Omega_1$  after  $b$  nodes have been compromised. Then,  $X$  is a binomial random variable with parameters  $(b, \rho)$ . Since the event  $B_1$  can only occur after at least  $m$  nodes are compromised, we have the following result:

$$P(B_1 | C_b) = \sum_{k=m}^b \binom{b}{k} \rho^k (1-\rho)^{b-k}. \quad (11)$$

Combining inequality (10) and equation (11), we derive the following result:

$$\begin{aligned} P(\text{at least one space is broken} | C_b) &\leq \lambda \sum_{k=m}^b \binom{b}{k} \rho^k (1-\rho)^{b-k} \\ &= \lambda \sum_{k=m}^b \binom{b}{k} \left(\frac{\mu}{\lambda}\right)^k \left(1 - \frac{\mu}{\lambda}\right)^{b-k}. \end{aligned} \quad (12)$$

## 5.2 The Faction of Additional Network Communications Being Compromised

To understand how resilient our proposed scheme is, it is better to investigate the influence caused by the event that an attacker has already captured  $b$  nodes over the rest of the network. In other words, we like to find out the fraction of additional communications (communications among uncompromised nodes) that an attacker can compromise based on the information obtained from the  $b$  captured nodes. In order to evaluate this fraction, what we have to do is to compute the probability that any one of the additional secure communication links is compromised after  $b$  nodes have been captured. Note that the additional secure communication links here are the communication links secured by pairwise keys that are established by two uncompromised neighboring nodes.

Let  $s$  denote an additional secure communication link, and  $pk$  denote the pairwise key used for this link. Let  $H_i$  denote the joint event that  $pk$  belongs to key space  $\Omega_i$  ( $pk \in \Omega_i$ ) and space  $\Omega_i$  is compromised. Hence, we have:

$$P(s \text{ is broken} | C_b) = P(H_1 \cup H_2 \cup \dots \cup H_\lambda | C_b).$$

Since  $s$  can only use one key and events  $H_1, \dots, H_\lambda$  are mutually exclusive and equally likely. Therefore, we have:

$$P(s \text{ is broken} | C_b) = \sum_{k=1}^{\lambda} P(H_i | C_b) = \lambda P(H_1 | C_b).$$

However,  $P(H_1 | C_b) = \frac{P((pk \in \Omega_1) \cap (\Omega_1 \text{ is compromised}) \cap C_b)}{P(C_b)}$ . Because the event  $(pk \in \Omega_1)$  is independent of the event  $C_b$  or the event  $(\Omega_1 \text{ is compromised})$ , we have:

$$\begin{aligned} P(H_1 | C_b) &= \frac{P(pk \in \Omega_1) \cdot P((\Omega_1 \text{ is compromised}) \cap C_b)}{P(C_b)} \\ &= P(pk \in \Omega_1) \cdot P(\Omega_1 \text{ is compromised} | C_b). \end{aligned}$$

$P(\Omega_1 \text{ is compromised} | C_b)$  is computed by equation (11).  $P(pk \in \Omega_1)$  is the probability that link  $s$  uses a key from key space  $\Omega_1$ . Since the choice of a key space from  $\lambda$  key spaces is equally probable, we have:  $P(pk \in \Omega_1) = \frac{1}{\lambda}$ . Therefore,

$$P(s \text{ is broken} | C_b) = \lambda P(H_1 | C_b) = \lambda \cdot \frac{1}{\lambda} \cdot P(B_1 | C_b) = \sum_{k=m}^b \binom{b}{k} \left( \frac{\mu}{\lambda} \right)^k \left( 1 - \frac{\mu}{\lambda} \right)^{b-k}. \quad (13)$$

The above equation shows that, given that  $b$  nodes are compromised, the fraction of the additional secure communication links compromised is equal to the probability of one key space being compromised.

### 5.3 Further Discussion

Our scheme together with other random key pre-distribution schemes [2], [4], [5], [6] is still vulnerable to several kinds of attacks that uniquely occur in random based schemes such as node replication attack [7] and key-swapping collusion attack [8]. To thwart the node replication attack, the method proposed in [9] can be used in cooperation with our scheme. Regarding the key-swapping collusion attack, there has been no radical proposal to prevent it so far. More efforts need to be put into this attack.

In [2], the authors proposed using two-hop neighbors in order to improve security. The same technique can also be applied to our scheme. However, using two-hop neighbors is vulnerable to man-in-the-middle attack if an intermediate node is compromised before or during the pairwise key establishment process. Fortunately, this attack and the mechanism to thwart it have already been presented in [10].

## 6 Performance Analysis

In this section we evaluate our proposed scheme with respect to memory usage, communication overhead, and computational overhead using Du's scheme as a benchmark.

### 6.1 Memory Usage

For each key space, according to MI scheme, each node  $S_i$  has to spend  $m \times l$  bits on storing the value of  $\Phi_i$ . Thus the total memory usage (KB)  $MU$  for each node with  $\mu$  chosen key spaces is:

$$MU = \frac{m \times l \times \mu}{8 \times 1024} \quad (14)$$

Since the value of  $m$  is equal to the value of  $\lambda + 1$  in [2]; the value of  $\mu$  is equal to that of  $\tau$  in [2]; and  $\frac{l}{8 \times 1024}$  is the memory unit of equation (5) in [2], hence memory consumption of our scheme for pairwise key establishment purpose is exactly identical to that of Du's scheme.

### 6.2 Communication Overhead

In this subsection, we like to compare the communication overhead of our scheme with that of Du's scheme. Note that to establish a pairwise key, the data that each node in our scheme needs to transmit is its effective ID and the indices of the key spaces in it, while in [2] the data needed to transmit is the node's ID, the indices and the seed of the column of  $G$ . Based on that, we draw a comparison as shown in fig. 3. From the figure, some observations are straightforwardly drawn: (i) if we choose the value of  $m$  such that the inequality  $m < 2 \times (\text{length of a pairwise key})$  is assured, then the communication overhead of each node in our scheme is always less than that



in [2]. For example, if  $m = 50$  (as chosen in [2]) and pairwise key length is 64 bits, then from the figure, the extra communication overhead for each node in [2] in comparison with our scheme is about 10 bytes. It is well known that transmitting a single bit costs as much power as executing 1000 instructions, then the communication overhead of our scheme is far less than that of [2]. (ii) The extra communication overhead of [2] in comparison with our scheme is directly proportional to the length of the pairwise key. Thus, although increasing in key size means an increase in security level but also in communication overhead.

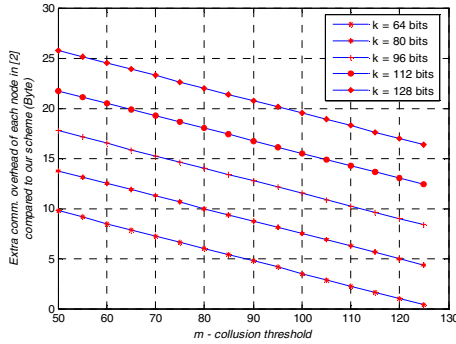


Fig. 3. Extra communication overhead of each node in [2] compared to our scheme

### 6.3 Computational Overhead

In this subsection, we compare the computational overhead of our scheme with that in [2]. In our scheme, to compute a pairwise key, each node needs to perform a multiplication of a  $(l \times m)$  matrix and an  $(m \times 1)$  effective ID. Therefore, each node needs  $l \times m$  single-precision multiplications while each node in [2] needs to do  $2 \times (m-1) \times l^2$  single-precision multiplications. It follows that the computational overhead of our scheme is far less than that in [2]. The numbers in fig. 4 reinforce our argument.

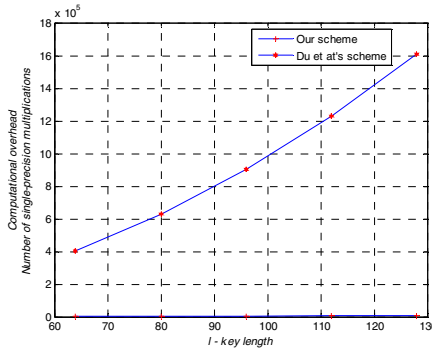


Fig. 4. Computational overhead in each node with various key lengths

## 7 Conclusions and Future Work

This paper proposes a new key pre-distribution scheme for WSNs that can be considered as a refinement of two types of schemes: ID-based key pre-distribution scheme and random key pre-distribution scheme. As a result, our scheme possesses a number of attractive properties. First, our scheme is scalable and flexible in terms of network size. Second, our scheme substantially improves network resiliency against node capture attack compared to schemes [4], [5], [11], and are comparable to Du's scheme. Furthermore, we have argued that network resiliency can be further improved using a combination of multi-hop neighbors method and a method to thwart man-in-the-middle attack that we proposed in [10]. We have also investigated the performance of our scheme to show its efficiency. Accordingly, our scheme is the same as Du's scheme in terms of memory usage. Moreover, under a certain condition, our scheme is more efficient than Du's scheme concerning communication overhead. Finally, computational overhead of our scheme is argued to be far less than that of Du's scheme.

In the preceding discussion, we have shown that our scheme is still vulnerable to node replication attack and key-swapping collusion attack. Therefore, in our future work, we would like to explore additional mechanisms to efficiently and radically thwart these attacks.

## References

1. T. Matsumoto and H. Imai, "On the KEY PREDISTRIBUTION SYSTEM: A Practical Solution to the Key Distribution Problem", CRYPTO'87, LNCS Vol. 293, pp.185-193, Aug. 1987
2. W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks", ACM Trans. Info. Sys. Sec., Vol. 8, No. 2, pp.228-258, May 2005
3. A. Shamir, "How to share a secret", Communications of the ACM, Vol. 22, No. 11, pp.612-613, Nov. 1979
4. L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks", Proc. of the 9th ACM Conference on Computer and Communications Security, pp.41-47, Nov. 2002
5. H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks", Proc. IEEE Symposium on Security and Privacy, pp.197-213, May 2003
6. D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks", Proc. of the 10th ACM Conference on Computer and Communications Security (CCS'03), pp.52-61, Oct. 2003
7. H. Fu, S. Kawamura, M. Zhang, and L. Zhang, "Replication attack on random key pre-distribution schemes for wireless sensor networks", Proc. IEEE on SMC Information Assurance Workshop, pp.134-141, Jun. 2005
8. T. Moore, "A collusion attack on pairwise key predistribution schemes for distributed sensor networks", Proc. IEEE on Pervasive Computing and Communications Workshops (PERCOMW'06), Mar. 2006
9. B. Parno, A. Perrig, and V. Gligor, "Distributed detection of node replication attacks in sensor networks", Proc. of IEEE Symposium on Security and Privacy, pp.49-63, May 2005

10. T. T. Dai, C. T. Hieu, Md. M. Rahman, and C. S. Hong, "A Robust Pairwise Key Predistribution Scheme Resistant to Common Attacks for Wireless Sensor Networks", Proc. of 7th WISA 2006, pp.121-135, Jeju Island, Korea, Aug. 2006.
11. T. T. Dai, C. T. Hieu, and C. S. Hong, "An Efficient ID-based Bilinear Key Predistribution Scheme for Distributed Sensor Networks", LNCS 4208, pp.260-269, Sep. 2006.
12. R. Blom, "An optimal class of symmetric key generation systems", EUROCRYPT '84, LNCS Vol. 209, pp.335-338, 1985
13. S. P. Chan, R. Poovendran, and M. T. Sun, "A key management scheme in distributed sensor networks using attack probabilities", Proc. IEEE GLOBECOM 2005, pp.1007-1011, 2005
14. W. Stallings, "Cryptography and Network Security: Principles and Practice", Prentice Hall, 2nd edn, Jul. 1998
15. A. Menezes, P. Van Oorschot, and S. Vanstone, "Handbook of Applied Cryptography", CRC Press, Inc., 1996

# Distributed Event Localization and Tracking with Wireless Sensors<sup>\*</sup>

Markus Wälchli<sup>1</sup>, Piotr Skoczylas<sup>2</sup>, Michael Meer<sup>1</sup>, and Torsten Braun<sup>1</sup>

<sup>1</sup> Institute of Computer Science and Applied Mathematics  
University of Bern - Switzerland

<sup>2</sup> School of Computer and Communication Sciences  
EPFL - Switzerland

**Abstract.** In this paper we present the distributed event localization and tracking algorithm DELTA that solely depends on light measurements. Based on this information and the positions of the sensors, DELTA is able to track a moving person equipped with a flashlight by dynamically building groups and electing well located nodes as group leaders. Moreover, DELTA supports object localization. The gathered data is sent to a monitoring entity in a fixed network which can apply pattern recognition techniques to determine the legitimacy of the moving person. DELTA enables object tracking with minimal constraints on both sensor hardware and the moving object. We show the feasibility of the algorithm running on the limited hardware of an existing sensor platform.

**Keywords:** Sensor network, monitoring, tracking, hardware.

## 1 Introduction

Composed of hundreds or thousands of tiny battery-powered devices, equipped with an array of sensors and a wireless radio to communicate, sensor networks are utilized to monitor and interact with the environment. The target application scenario we aim at is terrain observation during night. By the help of a sensor network a certain area (e.g. public building) is observed and exceptional behavior should be detected. DELTA detects and tracks light sources (typically from flashlights) and sends the data to a management station in a fixed network, where pattern recognition techniques can be applied to determine whether the person is present legitimately or not. The basic operations are depicted in Fig. 1. To be able to identify movement patterns online, the sensor network has to provide meaningful data in real time.

DELTA is a fully distributed object tracking framework avoiding heavy data load to the base station. Furthermore, DELTA uses the sensor light measurements in decisions and computations like leader election and localization. DELTA

---

<sup>\*</sup> The work presented in this paper was supported by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

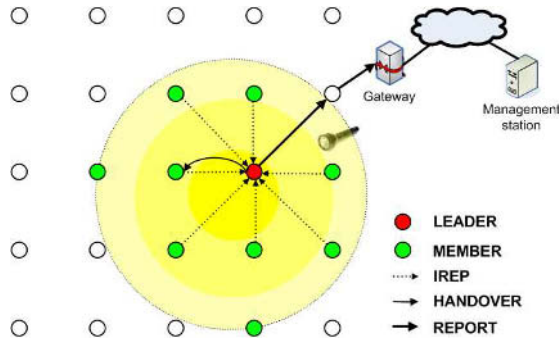


**Fig. 1.** DELTA observing an event and sending the data to the management station for further processing

maintains groups of nodes that are dynamically formed around objects as soon as these objects appear. A measurement-based election algorithm determines a group leader that is responsible for group maintenance, data gathering and processing, as well as reporting to the base station. The base station is connected to the Internet and may trigger alarm in case unusual behavior is detected. The alarm may be sent to any responsible authority. DELTA is designed to run on tiny nodes and has been implemented on the ESB sensor platform [1].

A requirement of DELTA is the knowledge of location information by the sensor nodes. This can be achieved by GPS or any other location service [2]. It could also be configured during deployment. The dependency on the node positions is no restriction as a terrain monitoring application intrinsically needs to know where the sensor nodes are placed.

In the next section we discuss related work in the areas of event localization and tracking. In section 3 we introduce the DELTA concepts and algorithms. The used hardware platform is described in section 4. Configuration data and the evaluation are provided in section 5. The paper ends with conclusions and outlook on future work in section 6.

## 2 Related Work

Existing event monitoring applications can mainly be divided in two categories: event detection and event tracking. While event localization is an intrinsic part of many detection applications, it is only rarely considered in tracking applications. With DELTA we aim at filling that gap as the recognition of the movement pattern of a person needs precise and detailed information.

[3] uses the sensor nodes only to collect data. That data is routed to a base station equipped with more computational resources where the event location is computed. Avoiding the computational limitations of sensor nodes, the algorithm accepts heavy network load, in particular in the direction of the base station. Other approaches [4], [5], [6] maintain predefined and static clusters of sensor

nodes to localize appearing events. Creation and maintenance of these clusters leads to communication overhead, especially in sensor networks with temporal disconnections and failures of nodes. Sextant [7] is a distributed approach using Bézier regions to represent the locations of nodes and events. Sextant disseminates network properties in a restricted area causing too much delay for real time tracking.

EnviroTrack [8], [9] is a distributed event tracking algorithm. It supports event detection and tracking, but not localization. A moving object is tracked by dynamically established groups of nodes. DELTA performs a similar set of basic operations like EnviroTrack adding beneficial features like the usage of the sensor readings and event localization. The SensIt project [10], [6] considers localization, tracking and classification algorithms. In [10] signal energy based multilateration similar to the approach we propose was mentioned, but the algorithm was neither implemented there nor in later publications. The SensIt project is tailored to powerful sensor nodes like PDAs making their set of algorithms not really applicable in our context.

Some existing localization algorithms [3], [11] depend on the possibility to distinguish two kinds of signals transmitted by an event. The distance of the event is derived from the time difference of arrival (TDOA) of two different signals. For example, [3] uses the time difference of arrival between the shock wave and the muzzle blast generated by a gun to estimate the distance to a sniper. In many cases the dependency on two different kinds of signals is restrictive and not easy to fulfill. In contrast, we propose a localization algorithm that uses a multilateration scheme that solely depends on the values of the sensor readings of the nodes observing an event [12].

### 3 Architecture of the DELTA Framework

A key problem of event tracking is the overhead of identifying and organizing the event relevant sensor nodes in a distributed manner with as little communication overhead as possible while providing a satisfactory degree of accuracy. Considering the application aimed at, persons are assumed to move quickly and the cones of light their flashlights produce may fluctuate even more.

To deal with these frequently changing conditions we use the sensor light readings to establish tracking groups fast and efficiently. Furthermore, the measurements are used to estimate the location of the moving object (light source). Details are explained in the next section. A limitation of existing tracking algorithms we overcome with DELTA is the dependency on the assumption that the communication range CR of the sensor nodes is significantly higher than their sensing range SR. In DELTA we ease this restriction with a 'passive heartbeat' mechanism: The leader periodically broadcasts heartbeats. To be able to estimate the location of a moving event, the group leader needs the light measurements and positions of its neighbors. Accordingly, the nodes that overhear the heartbeat respond their data. This message is overheard by the two hop neighbors which are thus inherently informed about the existence of a leader.

If needed, the presence of the leader can be disseminated even deeper into the network by the usage of passive heartbeats. Smart algorithms to distribute that information remain for future work.

### 3.1 State Diagram of DELTA Nodes

As DELTA is a distributed approach, the sensor nodes collaborate to compute the moving object location before sending the data to the base station. To achieve this different roles are assigned to the nodes. The states and state changes of the individual nodes and their roles are depicted in Fig. 2. The dashed lines show state changes caused by event observations. The solid lines show actions caused by messages from other nodes.

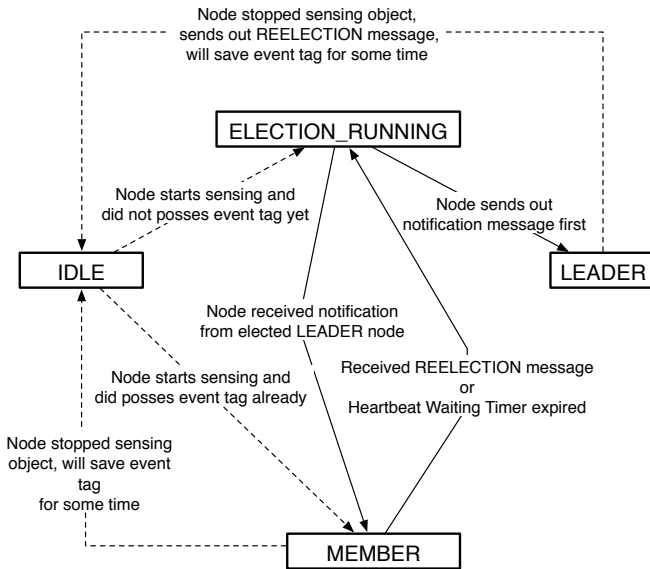


**Fig. 2.** State diagram of a DELTA node's roles

One sensor node is the leader of a tracking group. The leader is responsible for maintaining group coherence, localization of the target position, and communication with the base station. All direct neighbors of the leader are group members and deliver their relevant tracking and localization data to the leader. All other sensors are idle. They check their sensors frequently to detect a target at appearance. Additionally, by overhearing the communication of other nodes they are informed about a target moving into their direction. In DELTA, the roles are assigned dynamically. If an object is sensed, but no leader has been established yet the election running state is entered.

### 3.2 Distributed Leader Election and Group Maintenance

Initially, all DELTA nodes are in the IDLE state. As soon as a target has been observed by a node, it switches to election running mode and schedules a timer according to the intensity of its light measurement. When the timer expires a heartbeat message is broadcast to inform about the presence of the group leader. The calculation of the timer is crucial as it determines the leader node. It partly depends on the hardware platform and discussed in more detail there (see section 4). Additionally, the leader node sets several state variables concerning the newly formed group: An event tag to identify the tracked event. It is used to announce the tracking group to the base station as well as to maintain group coherence. A round number which is increased whenever the leader broadcasts heartbeat message. The round number is used to check the freshness of the messages, i.e. whether they are relevant or whether they can be ignored. Furthermore, there is a TTL field defining the depth, the leader information has to be disseminated into the network. Finally, the leader node is responsible to ensure a controlled handover of the leadership once the observation of the moving object fails. The leader will then immediately broadcast a leader reelection message and switch to IDLE state.

Group members wait for periodic heartbeat messages to be informed about the presence of the leader. If DELTA applications have larger sensing ranges than communication ranges, not every node that senses the moving target is a direct neighbor of the leader. The information response messages to the leader as well as passive heartbeats are used to inform such nodes about the tracking group leader. Otherwise, concurrent tracking groups may evolve leading to confusion and message overhead in the direction of the base station (multiple leaders that report). The message flow with which DELTA overcomes the restriction  $\frac{SR}{CR} < 1$  or even  $\frac{SR}{CR} < \frac{1}{2}$  is depicted in Fig. 3.

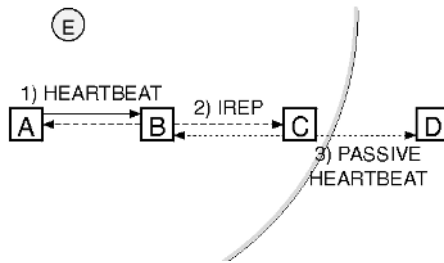


**Fig. 3.** Group communication in a DELTA network with  $\frac{SR}{CR} > 1$  and the TTL of the heartbeat messages from the leader set to 3

As mentioned above, a leader sends periodic heartbeat messages to inform its neighboring members about its existence and to request the needed information from them. The neighboring members answer with IREP messages containing



that information. These messages are moreover overheard by all two-hop neighbors of the leader which are thus informed about the existence of the leader. Passive heartbeat messages are then retransmitted until the TTL value expires. Implementation details concerning medium access, communication pattern, and light measurements are provided in section 4.

The leader election process aims at quickly determining a single leader node which is able to cover a moving target reliably. Reliability includes several aspects: The leader should be able to keep its leading state as long as possible, thus minimizing the number of reelections and handovers. Therefore, its location should be close to the target position or the path the target is moving at. Furthermore, the leader should have enough battery power left to be able to bear the burden of increased communication and computation compared to normal group members. The election process needs to be fast to avoid periods when no leader is present. In contrast to EnviroTrack, the leader election delay of DELTA is deterministic (see section 4).

## 4 Hardware Platform and Implementation Details

As sensor platform the ESB sensor boards [1] are used. These nodes consist of a chip with a TI MSP430 microcontroller, 2kB of RAM, 60kB flash memory, and a low power consuming radio transceiver (868MHz) operating at a transmission rate of 19.2kb/s by default. Furthermore, the sensor boards are equipped with a number of sensors like luminosity, temperature, vibration, etc. The boards have mainly two restrictions: the comparatively low transmission rate and the resource limitations of the memory and the processing unit. This is mainly caused by the miniaturization of the implemented hardware. The sensors have to work with at most 3V DC and should consume as little energy as possible.

All experiments are based on the light sensor. The provided light measurement software was reimplemented as it allows only binary decisions (light on/off), what is not appropriate for our purpose. The light sensor (TSL245) is associated to a interrupt-capable register and an interrupt is generated on each positive edge of the output frequency of the TSL245 (see Fig. 4). For each interrupt a counter is incremented counting that frequency. This solution implies high costs in case of high irradiance. Therefore, the spectrum is limited to a frequency of 100kHz. All above is just considered as maximal bright. The output frequency of the TSL245 in a standard office on the desk during day is around 2kHz.

To detect moving light sources an exponentially weighted moving average filter has been implemented with  $\bar{x}_k = \alpha \bar{x}_{k-1} + (1 - \alpha)x_k$ . The calculation of the mean  $\bar{x}_k$  thus only requires the storage of the past value of  $\bar{x}_{k-1}$  and the actual light measurement  $x_k$ . A light irradiance change is considered to be significant if the currently measured value differs more than a configurable threshold  $T$  from the average. Currently,  $T$  is set to 50. The advantage of having a moving average is the adaptivity to changing brightness in the environment. The moving average converges to the actual brightness, avoiding the permanent throwing of events during day, building works, etc. For the current evaluation, the value of  $\alpha$  is set to 0.9.

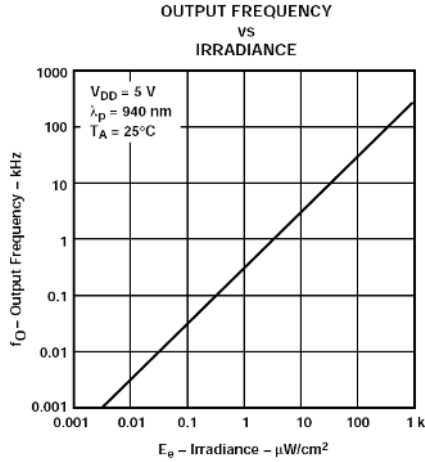


**Fig. 4.** TAOS TSL245 infrared to frequency converter. The output frequency of the TSL245 is linear to the measured light irradiance. The figure is from the TSL245 data sheet.

As mentioned above, the computation of the leader election timer is crucial for the performance of DELTA. On the ESB platform we calculate the light irradiance every 200ms for exactly 100ms. As we limit the TSL245 output frequency to 100kHz, we get light values from a spectrum between 0 and 10'000. Nodes with high irradiance should compute short delays whereas nodes with low irradiance should compute long delays. The delay is computed as follows:

$$\Delta t[ms] = \frac{I_{MAX} - I_C}{10}$$

$$\Delta round[ms] = round(i) \cdot SAMPLE\_FREQUENCY$$

$$\Delta t = \begin{cases} \Delta t & , \Delta t < \Delta round \\ \Delta t = \Delta t - \Delta round & , \text{else} \end{cases}$$

$I_C$  is the currently measured irradiance.  $I_{MAX}$  is the maximum value of 10'000. Accordingly,  $\Delta t$  generates a delay between zero and one second. The  $SAMPLE\_FREQUENCY$  is the light measurement frequency of 200ms. The round variable is set to 0 when the election is initialized and then incremented each time the light value is measured (all 200ms). The computation of the delay allows the filtering of non-continuous irradiance peaks as long as the value is not too high, i.e. the timer does not expire before the next light measurement is done.

The ESB sensor boards include a TR1001 radio module. The provided software runs the radio with 19.2kbps. For our purpose this bandwidth is too slow as it causes high collision probabilities in case of message bursts as with the heartbeat/IREP message flow of DELTA. We therefore changed the software to run with ASK modulation and 76kbps. To ensure 1-hop communication while

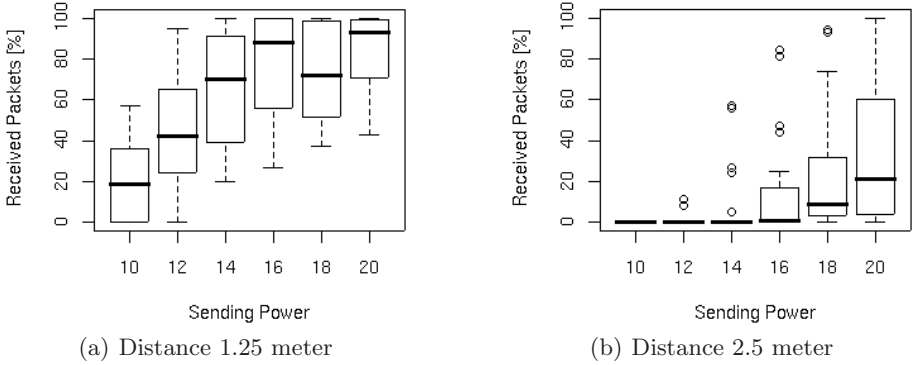


**Fig. 5.** Fraction of received messages for different transmission power levels

minimizing energy consumption we adjusted the radio power to have communication at 1.25 meters, while having no communication at the doubled distance. The radio power adjustment results are depicted in Fig. 5.

We concluded that a sending power of 16 is the best choice according to the evaluation results. In dense networks the burst of IREP messages cannot be handled efficiently by CSMA with random backoff given a delay of 2 ms to switch from receive to transmit state and the approximately 14 ms to transmit a message. As the leader does not need all IREP messages in dense networks to calculate the target position we implemented the following mechanism: Within the heartbeat message the leader schedules  $n$  nodes from which it received IREP messages in previous rounds ( $n \leq 8$ ). Thus, the first  $n \cdot 14ms$  are reserved for the scheduled nodes to transmit their IREP messages. Not scheduled nodes send their IREP message after this time using common CSMA with random backoff. Obviously all nodes compete for the medium when a new leader is elected, as the leader has no neighbor information at that time.

## 5 Evaluation

For the evaluation EnviroTrack [8] was chosen as reference algorithm. The reason for this choice is the similarity of both concepts in distributed group establishment and maintenance. DELTA adds a number of features like light measurement based leader election and object localization.

### 5.1 Reference Implementation of DELTA and EnviroTrack in the Simulator

To compare DELTA to the original EnviroTrack algorithm as well as to support our real world experiments, we implemented both DELTA and EnviroTrack in the OMNeT++ network simulator [13].

The simulation settings from the original EnviroTrack evaluations have been taken to make our implementations and the original DELTA implementation comparable. The goal of their simulated network was to track T-72 battle tanks moving through an off-road environment. For the simulations a realistic object path neither with sharp turns nor following just a straight line was used. Only the tracking performance of both, DELTA and EnviroTrack have been evaluated. DELTA has been evaluated with a TTL of 1 (just heartbeats like EnviroTrack) and a TTL of 2. We vary the speed of the target object and the ratio between sensing range and communication range. All settings are repeated eight times and a 95% confidence interval is used. The sensor network consists of 160 nodes, arranged in a grid of 8 times 20.

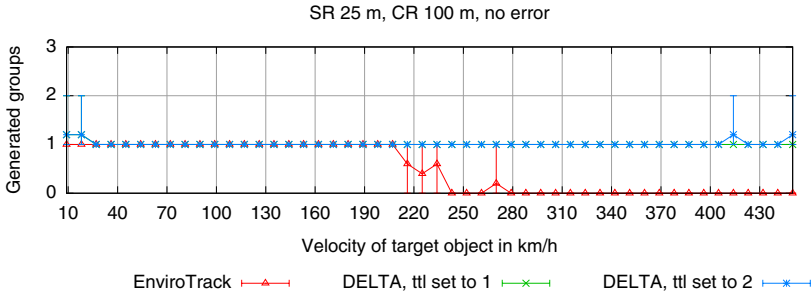


**Fig. 6.** DELTA and EnviroTrack with  $\frac{SR}{CR} = \frac{1}{4}$ . The average, minimum and maximum number of groups are shown.

In Fig. 6 results tailored to EnviroTrack are shown, i.e. the CR is significantly higher than the SR. Both algorithms have no problems in such scenarios. With this simulation we just show that DELTA performs not worse than EnviroTrack in a scenario tailored to EnviroTrack. In contrary, DELTA introduces shorter election delays (not depicted) and is therefore able to track objects at even higher speeds. Furthermore, DELTA performs equally well with the TTL set to one and two. This is obvious, as the ratio between SR and CR is too small to involve multiple groups. Consequently, the passive heartbeat mechanism does not provide any improvements here. The results are comparable to the original results from the EnviroTrack paper.

Fig. 7 shows what happens if the ratio between SR and CR becomes larger. In this scenario the SR is still smaller than the CR, but nevertheless the number of coexistent groups increases for both EnviroTrack and DELTA with TTL set to 1. This shows that EnviroTrack is not suitable for dense sensor networks where the SR may become equal to or is even larger than the CR. With such network properties we are however confronted by the application we aim at. The decreasing number of leaders in EnviroTrack under higher speeds is due to the inability of EnviroTrack to build groups in time.

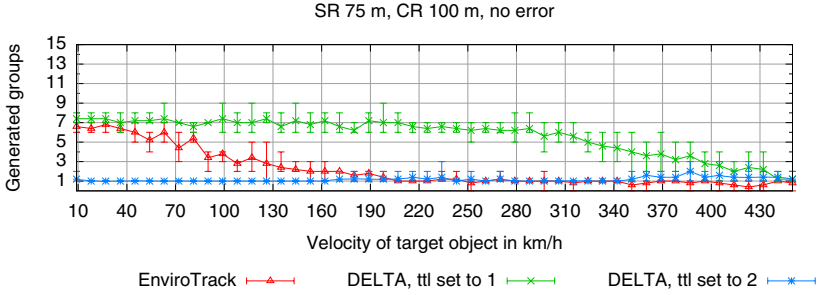


**Fig. 7.** DELTA and EnviroTrack with  $\frac{SR}{CR} = \frac{3}{4}$ . The average, minimum and maximum number of groups are shown.

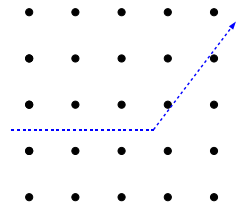
## 5.2 Real World Performance of DELTA and EnviroTrack

Both DELTA and EnviroTrack have been implemented on the ESB sensor nodes. In the following the test setup and the results of the measurements are described.

**Tracking Performance of DELTA and EnviroTrack.** The tests have been performed indoor in a completely dark room to avoid any external influences while tracking the moving light source. 25 nodes were placed in a 5x5 grid with a grid spacing of 1.25m. The transmission power has been reduced to 16 to



(a) Network setup



(b) Event path

just allow a node to communicate reliably with its grid neighbors. Two lamps, common office equipment with a 25W bulb and a 40W bulb, have been used as light sources. The lamp was held about 1.5m over ground pointing 1.5m in front of the moving person. The directly illuminated area consisted of a circle with an approximately 4 meters diameter. The person walked through the sensor field at constant speed of about 0.2 m/s.

Fig. 8 shows the percentage of leaders concurrently elected for DELTA and EnviroTrack. DELTA has significantly less concurrent leaders. Multiple concurrent leaders result in multiple reports about the same object consuming energy and bandwidth. In particular a much higher load to the base station is the consequence what affects the overall network lifetime. With DELTA a higher message load in the tracking area is accepted, while the proportional overloaded paths to

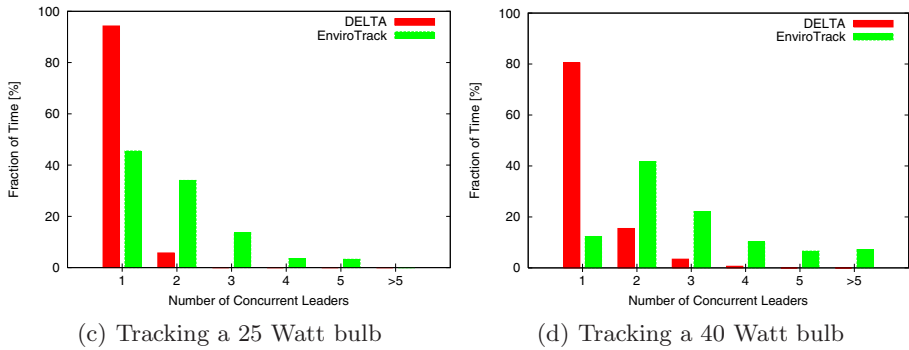


**Fig. 8.** Percentage of time a certain number of leaders are elected

the base station are relieved. The amount of the message overhead of DELTA has not yet been considered and remains to be evaluated in future work. The on-demand establishment of the time slots for the IREP messages performs well. During all the simulations enough IREP messages were received at the leader to enable the computation of the event location. However, the localization remains to be finished in future work.

## 6 Conclusions and Future Work

The DELTA algorithm proposes a solution for object tracking and localization by combining features that are not yet available in other algorithms. Groups are dynamically created. DELTA works in smart dust environments with small radio ranges and proportionally high sensing ranges. The leader election procedure of DELTA is quick and very precise. It favors well suited candidates adaptively. The on-demand setup of the 'slotted' medium access for the IREP messages works well. The results show that the usage of light measurements and IREP messages help to establish and maintain single groups to track the moving light source.

To distribute data in a larger area an optimized broadcast protocol could be used. A possible solution [12] could be implemented. Furthermore, it remains to be shown that measurement-based multilateration works on resource constraint sensor nodes. The light measurements are very exact, therefore we expect good results. The message overhead of DELTA remains to be investigated. Furthermore, we aim at deploying the algorithm in a realistic scenario to test its performance under real conditions. Finally, the occurrence of coexisting light sources and object classification remain to be investigated.

## References

1. Scatterweb: Sensor platform (2007) <http://www.scatterweb.net>.
2. Langendoen, K., Reijers, N.: Distributed localization in wireless sensor networks: a quantitative comparison. *Computer Networks* **43**(4) (2003) 499–518

3. Simon, G., Balogh, G., Bap, G., Maróti, M., Kusy, B., Sallai, J., Lédeczi, A., Nádas, A., Frampton, K.: Sensor network-based countersniper system. In: SenSys, Baltimore, Maryland, USA (2004)
4. Zou, Y., Chakrabarty, K.: Sensor deployment and target localization in distributed sensor networks. *ACM Transactions on Embedded Computing Systems (TECS)* **3**(1) (2004) 61–91
5. Kumar, M., Schwiebert, L., Brockmeyer, M.: Efficient data aggregation middleware for wireless sensor networks. In: IEEE International Conference on Mobile Ad-hoc and Sensor Systems, Fort Lauderdale, Florida, USA (2004) 1579–1581
6. Brooks, R.R., Ramanathan, P., Sayeed, A.M.: Distributed target classification and tracking in sensor networks. *Proc. IEEE* **91**(8) (2003) 1163–1171
7. Guha, S., Murty, R.N., Sirer, E.G.: Sextant: A unified node and event localization framework using non-convex constraints. In: MobiHoc'05, Urbana-Champaign, Illinois, USA (2005) 205–216
8. Abdelzaher, T., Blum, B., Evans, D., George, J., George, S., Gu, L., He, T., Huang, C., Nagaraddi, P., Son, S., Sorokin, P., Stankovic, J., Wood, A.: Envirotrack: Towards an environmental computing paradigm for distributed sensor networks. In: Proc. of 24th International Conference on Distributed Computing Systems (ICDCS), Tokyo, Japan (2004)
9. Luo, L., Abdelzaher, T.F., He, T., Stankovic, J.A.: Envirosuite: An environmentally immersive programming framework for sensor networks. *ACM Transaction on Embedded Computing System (TECS)* **V** (2006) 1–31
10. Li, D., Wong, K.D., Hu, Y.H., Sayeed, A.M.: Detection, classification and tracking of targets. *IEEE Signal Processing Magazine* **19**(2) (2002) 17–29
11. Capkun, S., Hamdi, M., Hubaux, J.P.: Gps-free positioning in mobile ad hoc networks. In: Proceedings of HICSS. (2001) 3481–3490
12. Wälchli, M., Scheidegger, M., Braun, T.: Intensity-based event localization in wireless sensor networks. In: Proceedings of IFIP Third Annual Conference on Wireless On Demand Network Systems and Services (WONS'06), Les ménuires, France (2006)
13. Varga, A.: Omnet++ simulator (2006) <http://www.omnetpp.org/>.

# Cross-Layer Distributed Diversity for Heterogeneous Wireless Networks

H. Javaheri, G. Noubir, and Y. Wang

College of Computer and Information Science  
Northeastern University, Boston, MA 02115, USA  
{hooman, noubir, yin}@ccs.neu.edu

**Abstract.** In this paper, we introduce a cross-layer diversity framework for multi-air interface wireless communication devices. As an initial step, we focus on devices, of the cellular phones type, that have both long-range relatively low-rate communication air-interface (e.g., GSM) and short-range high-rate communication air-interface (e.g., IEEE802.11). The devices can cooperate through the energy-efficient high-rate interface to improve the performance of the long-range interface. Within this framework we propose a distributed signal-combining technique that accounts for the limited bandwidth of the short-range communication: *Threshold Maximum Ratio Combining*. We analytically derive the probability distribution function of the signal to noise ratio (SNR) of the combined signals as a function of the number of involved devices and show that significant improvement of the SNR is achievable which translates into a reduction of the overall system outage probability.

## 1 Introduction

Diversity has been used for many years to increase the robustness and efficiency of wireless communication systems [1,2,3]. However, to the best of our knowledge very little research has been done on cross-layer receive diversity for distributed cooperative systems with multiple air-interfaces (e.g., GSM and IEEE802.11) and that accounts for the unique characteristics of each of the interfaces.

For example, consider the scenario depicted in Figure 1. Three mobile users each with a GSM phone suffer from the typical channel-fading that impairs urban cellular communication. The cooperation of these three devices can significantly boost the signal to noise ratio (SNR) making use of both energy-combining gain and fading independence. This SNR improvement results in coverage and capacity increase. Furthermore, it reduces interference because the base stations do not have to increase their transmission power to overcome the fades in order to reach mobile nodes.

Unlike traditional diversity paradigms [1,2,3], our approach considers a distributed setup using the local high-rate wireless network. We account for the constraints of the local resources such as bandwidth, computation and energy. In this paper, we propose and analyze a novel technique in this setup to boost the SNR (and therefore the system throughput) of the system using the diversity and energy combining gains and still satisfying the local bandwidth limit.



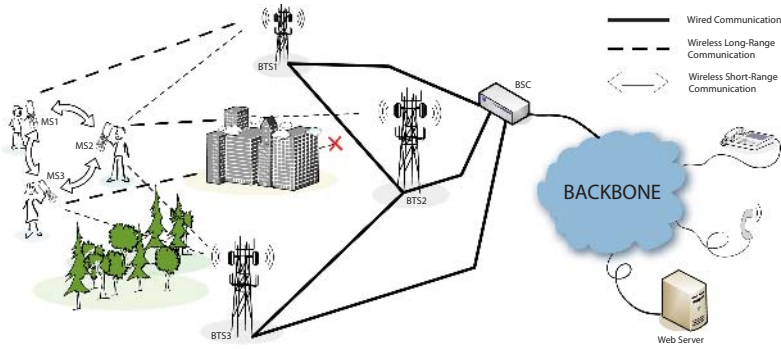


**Fig. 1.** Example of setup for distributed cross-layer diversity

**Related Work:** A lot of the previous work on diversity focusses on combining techniques for multiple-antennas/receivers that are co-located on a single device [1, 2, 3]. Given that all the antennas are directly interconnected, the combining algorithms have full access to the signals from all the antennas. Co-locating multiple antennas on a single *mobile* device is impractical for the current cellular systems. This is not only due to the form factor limitation and cost, but also because of the spatial separation needed between the antennas to achieve channel independence (See Section 2). Recently, distributed cooperative communication has received a significant interest [4, 5, 6, 7, 8, 9, 10, 11] from the wireless communications research community. However, most of the work is focused on *transmit diversity* (uplink) [4, 5, 6, 8, 10], while there is very little research on the specific case of a cooperative *receive diversity* where a long-range air-interface subject to a fading channel is combined with a reliable bandwidth-limited short-range air-interface.

**Contributions:** We introduce a networking framework for distributed cross-layer cooperation. This framework can be implemented on today's cell phone with minimal access to the baseband of the cellular link. A practical technique is proposed for distributed combining that we call *Threshold Maximum Ratio Combining* (TMRC) which enables distributed diversity while accounting for the local communication constraints. We characterize the SNR performance of TMRC as a function of the number of nodes involved and an energy threshold parameter. We derive a closed form formula for the probability distribution function of the combined SNR. This allows us to determine the outage probability of the system and characterize the tradeoff between performance and bandwidth constraints. Our analysis indicates that a significant SNR/Outage improvement can be achieved with today's GSM/IEEE802.11 devices.

The paper is organized as follows. In Section 2, we present the framework and discuss the rationale for distributed cross-layer diversity. In Section 3, we introduce the TMRC technique, derive a closed form formula for the SNR distribution and analyze the performance tradeoff. In Section 4, we propose an abstract protocol for implementing and enhancing TMRC. Finally, in Section 5, we introduce and discuss RMRC a more generalized form of TMRC.

## 2 Cross-Layer Distributed Diversity Framework

The performance of a long-range link is limited by channel fading caused by multi-path propagation and mobility. This is a critical problem in cellular communication as it results in dead-signal areas and poor localized system performance. Consider a scenario in the GSM system where a Base-Transceiver-Station ( $BTS_1$ ) is transmitting to a Mobile-Station ( $MS_1$ ), Figure 1. A GSM downlink (the uplink is similar) consists of one time-slot (out of 8) on a 200KHz frequency band. Assume that  $MS_1$  and  $MS_2$  have activated a distributed cooperation functionality and they are working in the following way.  $MS_1$  informs  $MS_2$  which 200KHz frequency band and time slot to listen to.  $MS_2$  samples the signal from GSM RF channel and forwards the sampled signal to  $MS_1$  over the short-range, low-power and high-rate link (i.e., IEEE802.11).  $MS_1$  now combines the two signals. This cooperation relies on five key aspects: *Diversity*, *Local bandwidth limit*, *Cross-layer cooperation*, *Modes of operation* and *Fairness*.

**Diversity:** Implementing an efficient diversity mechanism on a single small device is difficult [4], because the antennas need to be spatially separated beyond a theoretical lower bound to obtain channel independence. In a uniform scattering environment with omnidirectional antennas, the minimum space for independent fading is approximately  $0.4\lambda$  (where  $\lambda$  is the carrier wavelength) [3]. For GSM systems operating over the 850MHz and 1900MHz bands (in the US), the separation has to be at least  $0.14m$ . Furthermore, in cellular system directional antennas are usually used at the based stations. This requires an even larger separation due to the small multi-path angle. As a result, it is usually impractical to implement multiple antennas on a single mobile phone. In our setup the two mobile stations are well separated, thus the links  $BTS_1$  to  $MS_1$  and  $BTS_1$  to  $MS_2$  are independent. Therefore, signal combining from neighboring devices will provide a diversity gain in addition to the typical antenna gain.

**Local Bandwidth Limit:** In our framework, the RF signals received from the long-range interface have to be converted in order to be forwarded through the local short-range interface. We are currently prototyping our framework using the GNU Software Defined Radio (SDR) [12] with the Universal Software Radio Peripheral (USRP) [13]. The long-range RF signal will be down-converted to the intermediate frequency (IF) by the RF front end, sampled by analog-to-digital converter (ADC) and forwarded through the short-range interface. However transmitting the sampled analog waveforms requires a significant bandwidth. For example, in the GSM systems a frame consists of 8 time slots and takes 4.6ms. So each time slot is about  $577\mu s$ . The symbol rate of GMSK in GSM is 270.833Ksps. Assume the ADC samples 8 times per symbol and with a dynamic range of 12 bits [14]. Therefore, to transmit a single time slot RF signal in every frame requires a data rate of 3261Kbps. IEEE802.11g (54Mbps) can support around 15 links while 802.11b (11Mbps) can only support 3 links.

**Distributed Cross-Layer:** Having multiple devices to combine the RF channel signals using separate air-interfaces is a novel communication paradigm where the physical layer is virtual and distributed over a set of nodes. It raises

interesting questions on how to maximize the performance while satisfying the constraints of the short-range communication link in terms of bandwidth and energy consumption. Several combining techniques have been introduced in the past for centralized diversity (e.g., Maximum Ratio Combining, and Selective Combining [15, 1, 3]). We introduce a new diversity combining technique called Threshold Maximum Ratio Combining which is a special case of a larger set of randomized combining techniques (See Section 5).

**Modes of Operation:** Nodes can cooperate in two ways, Figure 2:

- *Master-Slave:* A node currently communicating over its long-range communication interface becomes a master. Surrounding nodes that are willing to help, become assisting nodes or slaves. The master node collects sampled signals from the slaves and dictates the cooperation strategy to satisfy the short-range communication requirements while maximizing the SNR of the combined signals at master side.
- *Peer-to-Peer:* The cooperation happens according to a distributed multi-round and multi-hop algorithm. All the nodes are assumed to operate with the common goal to help neighbors obtain higher throughput meanwhile being helped by others.

Both operation modes are constrained by the limitations on local resources. In this paper, we focus on the Master-Slave mode.

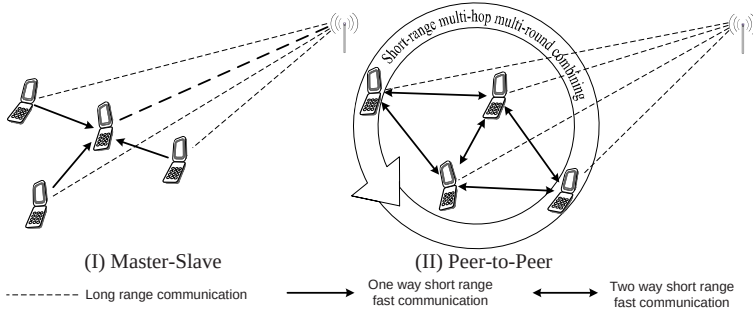


**Fig. 2.** Modes of operation

**Mitigating Selfish Behavior:** For such cooperation to be practical, there is a need to develop mechanisms that reward cooperating nodes, detect and punish selfish behavior [16, 17]. Such mechanisms are very important for the success of cross-layer distributed diversity but their study is outside the scope of this paper.

### 3 Diversity Combining for Distributed Systems

#### 3.1 Traditional Maximum Ratio Combining(MRC)

Rayleigh fading is the typical fading model for cellular systems [2] and our analysis is based on this model. Let  $\gamma_i$  be the random variable for the SNR from

the  $i^{th}$  branch and  $m$  be the number of diversity branches. Assume each fading channel is independent and identically distributed (i.i.d.) with the same noise power spectral density  $N_0/2$ . It has been shown that  $\gamma_i$  has an exponential distribution [3] (i.e.,  $p(\gamma_i) = \frac{1}{\bar{\gamma}} e^{-\frac{\gamma_i}{\bar{\gamma}}}$  where  $\bar{\gamma}$  is the average SNR and  $p(x)$  denotes the probability density function of random variable  $x$ ). MRC is basically a weighted sum of all branches. By choosing the weights to be the square root of the SNR of each branch, which leads to the combined SNR [15]:

$$\gamma_{\Sigma} = \gamma_1 + \cdots + \gamma_m$$

### 3.2 Threshold Maximum Ratio Combining (TMRC)

In MRC, the branch with higher SNR gets the higher weight and the branch with low SNR gets the lower weight. If the local resources are limited, it is better to keep the high SNR branches and discard the lower ones. The Threshold Maximum Ratio Combining is a simple extension of MRC adapting it to account for the limited bandwidth available in cooperative networks. In TMRC, each assisting node transmits the data to the master node *if and only if* its SNR is above a threshold  $\gamma_T$  which is preset by the master node according to the channel condition and local bandwidth. The master node then collects all the signals from the assisting nodes and combines them using MRC. In this paper, we show that under the same assumptions of MRC, the combined signal distribution with  $m$  assisting nodes and a threshold  $\gamma_T$  is as follows (See Theorem 2):

$$p_{TMRC}(\gamma) = \sum_{i=1}^m \binom{m-1}{i-1} \cdot C^{m-i} \cdot g^{(i)}(\gamma), \quad \gamma \geq 0 \quad (1)$$

where

$$g^{(i)}(x) = \frac{(x - (i-1) \cdot \gamma_T)^{i-1}}{(i-1)! \cdot \bar{\gamma}^i} \cdot e^{-\frac{x}{\bar{\gamma}}}, \text{ for } x \geq (i-1) \cdot \gamma_T \text{ and } C = 1 - e^{-\frac{\gamma_T}{\bar{\gamma}}}$$

Based on this distribution, we can further measure the performance of TMRC (e.g., outage probability, energy gain, bandwidth and energy consumption).

The structure of the proof is as follows. We first derive the signal distribution from each assisting node. Theorem 1 shows the probability distribution of the combined SNR for a simplified case in which all the nodes (including the master node) only use the signal above the threshold  $\gamma_T$ . Theorem 2 gives the distribution of the combined SNR for a more practical protocol. It differentiates the master node from the assisting nodes in the way that the master node uses the signal in the full SNR range because it does not consume any local bandwidth. And  $g^{(i)}(x)$  is derived in Lemma 2.

### 3.3 Probability Distribution of SNR for TMRC

We keep the MRC assumption that the nodes long-range signals' SNR are i.i.d. with parameter  $\bar{\gamma}$  and  $p(\gamma) = \frac{1}{\bar{\gamma}} e^{-\frac{\gamma}{\bar{\gamma}}}$ , where  $\gamma \geq 0$ . In TMRC, each assisting

node does not relay the signal below the threshold  $\gamma_T$ . The master node collects the signals from each assisting nodes. Let  $p_T(\gamma)$  be the distribution for each branch at the master side. We have  $p_T(\gamma)$  equals  $p(\gamma)$  when  $\gamma \geq \gamma_T$ , and  $p_T(\gamma)$  equals 0 when  $0 < \gamma < \gamma_T$  because the master will not get any signal from the assisting node in that range. The formal definition of  $p_T(\gamma)$  is given in Equation 2.

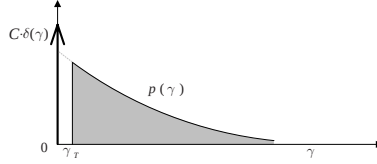


**Fig. 3.** PDF of  $\gamma$  of each branch at master side in TMRC

Let  $\delta(x)$  denote the Dirac delta function defined as follows:

$$\delta(x) = \begin{cases} \infty, & x = 0 \\ 0, & x \neq 0 \end{cases} \text{ and } \int_{-\infty}^{+\infty} \delta(x) dx = 1.$$

We define the distribution of SNR  $T$  for each branch under TMRC, Figure 3,

$$p_T(\gamma) = \begin{cases} \int_0^{\gamma_T} \frac{1}{\bar{\gamma}} e^{-\frac{\tau}{\bar{\gamma}}} d\tau \cdot \delta(\gamma), & \gamma = 0 \\ 0, & 0 < \gamma < \gamma_T \\ \frac{1}{\bar{\gamma}} e^{-\frac{\gamma}{\bar{\gamma}}}, & \gamma \geq \gamma_T \end{cases}$$

Let  $f(x) = \begin{cases} \frac{1}{\bar{\gamma}} e^{-\frac{x}{\bar{\gamma}}}, & x \geq \gamma_T \\ 0, & x < \gamma_T \end{cases}$  and  $C = \int_0^{\gamma_T} \frac{1}{\bar{\gamma}} e^{-\frac{\tau}{\bar{\gamma}}} d\tau = 1 - e^{-\frac{\gamma_T}{\bar{\gamma}}}$ , so the probability distribution function for the SNR of each branch can be written as:

$$p_T(\gamma) = C \cdot \delta(\gamma) + f(\gamma) \quad (2)$$

**Definition 1.** For a function  $h(x)$ , recursively define  $h^{(i)}$ , where  $i \geq 0$ , as follows,  $h^{(0)}(x) = 1$ ,  $h^{(1)}(x) = h(x)$  and  $h^{(i)}(x) = h(x) * h^{(i-1)}(x)$ , where  $*$  is the convolution operator.

**Lemma 1**

$$f^{(i)}(x) = \frac{(x - i \cdot \gamma_T)^{i-1}}{(i-1)! \cdot \bar{\gamma}^i} \cdot e^{-\frac{x}{\bar{\gamma}}}, \quad \text{for } x \geq i \cdot \gamma_T, \quad i \geq 1 \quad (3)$$

*Proof.* By induction.

Base case:  $i = 1$ ,

$$f^{(1)}(x) = \frac{(x - \gamma_T)^0}{0! \cdot \bar{\gamma}} \cdot e^{-\frac{x}{\bar{\gamma}}} = \frac{1}{\bar{\gamma}} e^{-\frac{x}{\bar{\gamma}}}, \quad x \geq \gamma_T$$

Hypothesis: assume Equation 3 holds for  $i = j$ .

On  $i = j + 1$ ,

According to the definition of  $f^{(j+1)}(x)$ , we have

$$\begin{aligned}
 f^{(j+1)}(x) &= f(x) * f^{(j)}(x) \\
 &= \int_{\gamma_T}^{x-j \cdot \gamma_T} f(\tau) \cdot f^{(j)}(x - \tau) d\tau, \quad \tau \geq \gamma_T; x - \tau \geq j \cdot \gamma_T \\
 &= \int_{\gamma_T}^{x-j \cdot \gamma_T} \left( \frac{1}{\bar{\gamma}} e^{-\frac{\tau}{\bar{\gamma}}} \right) \cdot \left( \frac{(x - \tau - j \cdot \gamma_T)^{j-1}}{(j-1)! \cdot \bar{\gamma}^j} \cdot e^{-\frac{x-\tau}{\bar{\gamma}}} \right) d\tau \\
 &= \frac{(x - (j+1) \cdot \gamma_T)^j}{j! \cdot \bar{\gamma}^{j+1}} \cdot e^{-\frac{x}{\bar{\gamma}}}, \quad x \geq (j+1) \cdot \gamma_T
 \end{aligned}$$

□

Let  $X_1$  and  $X_2$  be two independent random variables and  $X_{1+2} = X_1 + X_2$ . By basic probability rule, the distribution of the sum of two independent random variables is the convolution of their distributions. So  $p_{x_{1+2}}(x) = p_{x_1}(x) * p_{x_2}(x)$ . It can be generalized to the sum of  $m$  independent random variables:

$$p_{T_{\Sigma_m}}(\gamma) = p_T^{(m)}(\gamma) = (C \cdot \delta(\gamma) + f(\gamma))^{(m)} \quad (4)$$

**Theorem 1.** *The distribution of  $\gamma_{T_{\Sigma_m}}$  which is the sum of  $m$  channels under i.i.d. Rayleigh fading with threshold  $\gamma_T$  is:*

$$p_{T_{\Sigma_m}}(\gamma) = \sum_{i=0}^m \binom{m}{i} \cdot C^{m-i} \cdot f^{(i)}(\gamma), \quad \gamma \geq 0$$

*Proof.* Expand the equation (4) and simplify using the Dirac function property:

$$\delta(x) * f(x) = f(x)$$

□

$$\text{Let } g^{(i)}(x) = \begin{cases} p(x) * f^{(i-1)}(x), & x \geq (i-1) \cdot \gamma_T \\ 0, & x < (i-1) \cdot \gamma_T \end{cases}, \quad i \geq 1$$

**Lemma 2**

$$g^{(i)}(x) = \frac{(x - (i-1) \cdot \gamma_T)^{i-1}}{(i-1)! \cdot \bar{\gamma}^i} \cdot e^{-\frac{x}{\bar{\gamma}}}, \quad \text{for } x \geq (i-1) \cdot \gamma_T, \quad i \geq 1$$

*Proof*

$$\begin{aligned}
 g^{(i)}(x) &= p(x) * f^{(i-1)}(x) \\
 &= \int_0^{x-(i-1) \cdot \gamma_T} p(\tau) \cdot f^{(i-1)}(x - \tau) d\tau, \quad \tau \geq 0; x - \tau \geq (i-1) \cdot \gamma_T \\
 &= \int_0^{x-(i-1) \cdot \gamma_T} \left( \frac{1}{\bar{\gamma}} e^{-\frac{\tau}{\bar{\gamma}}} \right) \cdot \left( \frac{(x - \tau - (i-1) \cdot \gamma_T)^{i-2}}{(i-2)! \cdot \bar{\gamma}^{i-1}} \cdot e^{-\frac{x-\tau}{\bar{\gamma}}} \right) d\tau
 \end{aligned}$$

$$= \frac{(x - (i - 1) \cdot \gamma_T)^{i-1}}{(i - 1)! \cdot \bar{\gamma}^i} \cdot e^{-\frac{x}{\bar{\gamma}}}, \quad x \geq (i - 1) \cdot \gamma_T \quad \square$$

To get the distribution  $p_{TMRC}(\gamma)$  of the cooperative network with size  $m$ , we need to add the  $\gamma$  of the master node to the  $\gamma_{T_{\Sigma m}}$  of the  $m - 1$  assisting nodes.

**Theorem 2.**  $\gamma_{TMRC}$  is the sum of  $m$  channels under i.i.d. Rayleigh fading in which  $m - 1$  branches are with threshold  $\gamma_T$  and one master node which always uses its received signal, i.e.,  $\gamma_{TMRC} = \gamma + \gamma_{T_{\Sigma m-1}}$ . The distribution of  $\gamma_{TMRC}$  is:

$$p_{TMRC}(\gamma) = \sum_{i=1}^m \binom{m-1}{i-1} \cdot C^{m-i} \cdot g^{(i)}(\gamma), \quad \gamma \geq 0$$

*Proof*

$$\begin{aligned} p_{TMRC}(\gamma) &= p(\gamma) * p_{T_{\Sigma m-1}}(\gamma) \\ &= p(\gamma) * \left( \sum_{i=0}^{m-1} \binom{m-1}{i} \cdot C^{m-i-1} \cdot f^{(i)}(\gamma) \right) \\ &= \sum_{i=1}^m \binom{m-1}{i-1} \cdot C^{m-i} \cdot g^{(i)}(\gamma), \quad \gamma \geq 0 \end{aligned}$$

□

By taking the limit of  $\gamma_T \rightarrow 0$ , we get

$$\lim_{\gamma_T \rightarrow 0} p_{TMRC}(\gamma) = \frac{\gamma^{m-1}}{(m-1)! \cdot \bar{\gamma}^m} \cdot e^{-\frac{\gamma}{\bar{\gamma}}}.$$

It is also the distribution of  $\gamma_{\Sigma}$  of the traditional MRC. Therefore, MRC can be viewed as a special case of TMRC with zero threshold.

### 3.4 Performance, Bandwidth and Energy Tradeoffs

In wireless communication systems, transmissions account for most of the energy consumption. For simplicity purpose, as a first step, we will use the amount of transmissions to account for the system constraints. Since MRC achieves the full diversity order and TMRC as a variation of MRC intentionally discards the low SNR signals at assisting nodes, intuitively its performance should be in-between MRC and no diversity. TMRC has the advantage of being able to satisfy the local bandwidth requirements. We discuss the tradeoff between the performance and bandwidth in terms of outage probability and energy gain.

Let  $\gamma_0$  be the minimum SNR for acceptable performance of the demodulator. The outage probability is defined as

$$P_{out} = p_{TMRC}(\gamma < \gamma_0) = \int_0^{\gamma_0} p_{TMRC}(\gamma) d\gamma.$$

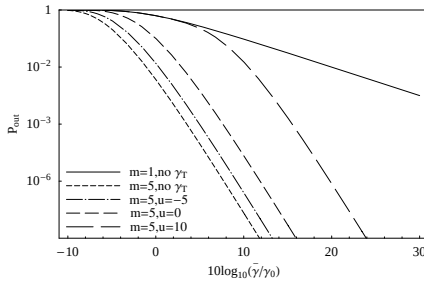


**Fig. 4.** The outage prob. of the master node with different  $\gamma_T$  in TMRC compared to MRC and no diversity (case  $m = 1$ ). [ $u$  denotes  $10\log_{10}(\gamma_T/\gamma_0)$ ].

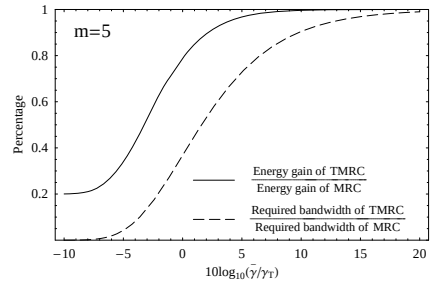


**Fig. 5.** Percentages of the energy gain and the bandwidth requirement of TMRC over MRC when setting the threshold  $\gamma_T$  to different values

When setting  $\gamma_T = 0$ , as expected, the outage probability of the master node in TMRC is the same as the one in MRC for the same  $m$ . As we increase  $\gamma_T$  the outage probability increases until  $\gamma_T$  reaches infinity where it becomes the same as the one with no diversity. This is reasonable because when  $\gamma_T$  is set to infinity basically no assisting node transmits; the master only uses its own received signal which is the case of no diversity. Figure 4 shows examples of  $m = 5$  (i.e., 4 assisting nodes) with different  $\gamma_T$ ,  $m = 5$  in MRC (full diversity) and  $m = 1$  (no diversity). Note that if  $\gamma_T$  is in the order of  $\gamma_0$ , the outage probability of TMRC is very close to MRC.

The increase in averaged SNR of the combined signal over the average SNR of each branch is called *Energy Gain* or *Array Gain*. In TMRC, it starts as the full diversity MRC case when  $\gamma_T = 0$ , it goes down as  $\gamma_T$  increases and finally converges to the no diversity case. We also observe that the required bandwidth drops as  $\gamma_T$  increases and finally down to 0 when  $\gamma_T$  reaches infinity. But with the same  $\gamma_T$  the drop percentage of energy gain is always less than that of bandwidth requirement. It means we can always lower the amount of wireless communication at the expense of less loss of energy gain. For the case of  $m = 5$ , Figure 5, if we set  $\gamma_T = \bar{\gamma}$  which is at position 0, the energy gain of TMRC is 79% of MRC while it needs only 36% of the local bandwidth of MRC. This justifies the performance and bandwidth tradeoff in TMRC.

## 4 Protocols for TMRC Implementation

The packet containing the sampled signal is large (See Section 2) and requires significant bandwidth. Even though in TMRC the low SNR signals are dropped, inspired by [6] we can still further lower the bandwidth requirement by introducing another threshold  $\gamma_D$ . When an assisting node finds the received signal SNR is beyond  $\gamma_D$ , it believes that the signal is good enough for demodulation and does not need any further combining. So it just sends the demodulated bits to





## 5 Generalization of TMRC

Generalizing the idea of TMRC, we introduce a class of techniques called Randomized Maximum Ratio Combining (RMRC). In RMRC each node transmits the sampled signal to the master in a randomized way with the probability determined by the signal SNR.

Let  $TX(\gamma)$  denote the function generating the probability of transmission. It takes the SNR as the input and outputs the probability of replaying the signal to the master. First, let  $TX(\gamma)$  be an exponential function, Figure 7(I),  $TX'(\gamma) = 1 - e^{-c' \cdot \gamma}$ , where  $c'$  is a parameter. When  $\gamma$  is low, the probability of transmitting the sampled signal is small. As  $\gamma$  increases, the probability of transmitting goes up by following an exponential function. Similarly, we can define  $TX(\gamma)$  to be a linear function, Figure 7(II),  $TX''(\gamma) = c'' \cdot \gamma$  if  $\gamma < 1/c''$ ;  $TX''(\gamma) = 1$  if  $\gamma \geq 1/c''$ , where  $c''$  is a parameter. Actually if we review the TMRC, we find that it is exactly a special case in RMRC. In this case, Figure 7(III),  $TX'''(\gamma) = 0$  if  $\gamma < \gamma_T$ ;  $TX'''(\gamma) = 1$  if  $\gamma \geq \gamma_T$ . Similarly, we can get the distribution of  $\gamma$  for each branch.

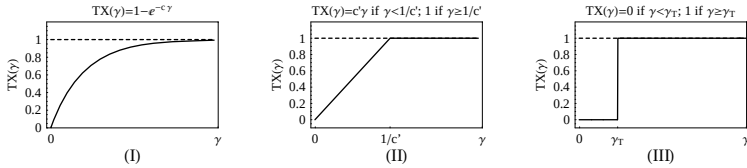


Fig. 7. Potential  $TX(\gamma)$  functions for RMRC combining

For  $TX'(\gamma)$ , we have

$$p(\gamma) = C' \cdot \delta(\gamma) + \frac{1}{\bar{\gamma}} e^{-\frac{\gamma}{\bar{\gamma}}} \cdot (1 - e^{-c' \cdot \gamma}), \quad C' = \int_0^{+\infty} \frac{1}{\bar{\gamma}} e^{-\frac{\tau}{\bar{\gamma}}} \cdot e^{-c' \cdot \tau} d\tau = \frac{1}{c' \cdot \bar{\gamma} + 1}$$

For  $TX''(\gamma)$ , we have

$$p(\gamma) = \begin{cases} C'' \cdot \delta(\gamma) + \frac{1}{\bar{\gamma}} e^{-\frac{\gamma}{\bar{\gamma}}} \cdot (c'' \cdot \gamma), & \gamma < \frac{1}{c''} \\ \frac{1}{\bar{\gamma}} e^{-\frac{\gamma}{\bar{\gamma}}}, & \gamma \geq \frac{1}{c''} \end{cases},$$

$$C'' = \int_0^{\frac{1}{c''}} \frac{1}{\bar{\gamma}} e^{-\frac{\tau}{\bar{\gamma}}} \cdot (1 - c'' \cdot \tau) d\tau = c'' \bar{\gamma} \cdot e^{-\frac{1}{c'' \bar{\gamma}}} - c'' \cdot \bar{\gamma} + 1$$

RMRC techniques can be analyzed in the exact same manner as TMRC. Due to the lack of space, we will skip this part.

## 6 Conclusion

In this paper, we introduce a cross-layer distributed diversity framework where neighboring nodes collaborate to boost their performance. We consider the specific case of cellular nodes with two wireless communication interfaces a long-range

low-rate and a short-range high-rate. We introduce a novel diversity combining technique called TMRC that takes into account the bandwidth constraints of the short-range communication air-interface. We analyze and derive a closed form formula for the probability distribution function of the SNR for TMRC. This allows us to characterize the outage probability and energy gain as a function of cooperating nodes, local bandwidth, and channel conditions. We show that TMRC leads to significant channel boosting and provide an abstract protocol for implementing it. In the future, we plan to develop protocols that optimize the tradeoffs in involving more assisting nodes (higher  $m$ ) or reducing the TMRC threshold ( $\gamma_T$ ) while satisfying the local bandwidth constraint. We are currently working on a prototype for cross-layer distributed diversity using the GNU Radio and USRP SDR platform.

## References

1. Proakis, J.: Digital Communications 4 edition. McGraw-Hill (2000)
2. Tse, D., Viswanath, P.: Fundamentals of Wireless Communication. Cambridge University Press (2005)
3. Goldsmith, A.: Wireless Communications. Cambridge University Press (2005)
4. Andrew Sendonaris, E.E., Aazhang, B.: User cooperation diversity— part i and part ii. IEEE Transactions on Communications **51**(11) (2003) 1927–1948
5. Laneman, J.N., Tse, D.N., Wornell, G.W.: Cooperative diversity in wireless networks: Efficient protocols and outage behavior. IEEE Trans. on Info. Theory (2004)
6. Laneman, J.N., Wornell, G.W.: Exploiting distributed spatial diversity in wireless networks. Proc. Allerton Conf. Communications, Control, and Computing (2000)
7. Dubois-Ferrière, H., Estrin, D., Vetterli, M.: Packet combining in sensor networks. SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems (2005)
8. Hunter, T.E., Nosratinia, A.: Diversity through coded cooperation. IEEE Trans. on Wireless Commun. **5**(2) (2006)
9. Fitzek, F.H.P., Katz, M.D.: Cooperation in Wireless Networks: Principles and Applications: Real Egoistic Behavior is to Cooperate! Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)
10. Jaime Adeane, M.R.D.R., Wassell, I.J.: Characterisation of the performance of cooperative networks in ricean fading channels. 12th International Conference on Telecommunications (2005)
11. Chen, J., Jia, L., Liu, X., Noubir, G., Sundaram, R.: Minimum energy accumulative routing in wireless networks. Proceedings of IEEE Infocom (2005)
12. The GNU Software Defined Radio, <http://www.gnu.org/software/gnuradio/>
13. The Universal Software Radio Peripheral, <http://www.ettus.com/>
14. A GMSK modulation and demodulation implementation for GNU Radio, <http://noether.uoregon.edu/~jl/gmsk/> (2005)
15. Brennan, D.G.: Linear diversity combining techniques. Proceedings of the IEEE **91**(2) (2003)
16. Buttyan, L., Hubaux, J.P.: Stimulating cooperation in self-organizing mobile ad hoc networks. Mob. Netw. Appl. **8**(5) (2003) 579–592
17. Feldman, M., Chuang, J.: Overcoming free-riding behavior in peer-to-peer systems. SIGecom Exch. **5**(4) (2005) 41–50
18. Bjørn A. Bjerke, John G. Proakis, K.L., Zvonar, Z.: A comparison of gsm receivers for fading multipath channels with adjacent- and co-channel interference. IEEE Journal on Selected Areas In Communications **18**(11) (2000)

# Location-Aware Signaling Protocol for WWAN and WLAN Interworking\*

SungHoon Seo, SuKyoung Lee, and JooSeok Song

Department of Computer Science, Yonsei University, Seoul, Korea  
hoon@emerald.yonsei.ac.kr, sklee@cs.yonsei.ac.kr,  
jssong@emerald.yonsei.ac.kr

**Abstract.** With the rapid improvement of wireless networking technologies, current mobile devices can be potentially equipped with multiple interfaces to access different kinds of wireless networks. Thus, there have been many efforts to provide a seamless roaming between heterogeneous networks. However, most previous studies do not address how to select the best possible interface in terms of energy consumption. Therefore, in this paper, we propose to take advantage of accurate location positioning via WWAN interface, so that Mobile Nodes (MNs) obtain the information of hotspot range and completely turn off WLAN interface during idle state, leading to save energy consumption. Simulation results show that our mechanism outperforms existing approaches in terms of energy consumption and signaling overhead.

**Keywords:** Network interworking, Vertical handover, Signaling protocol.

## 1 Introduction

Seamless integration of heterogeneous wireless networks is one of the key steps towards building 3G communication systems and beyond, where currently, it is the center of attention to integrate WLANs (Wireless Local Area Networks) providing high data rate into WWANs (Wireless Wide Area Networks) with full coverage. In fact, during the last decade, these two wireless access technologies have spread so widely that current mobile devices are often equipped with various wireless interfaces in multi-stack or multi-mode configurations.

In heterogeneous wireless networks, one of the main challenges is to maintain MN's active connections across different types of networks, referred to as vertical handover. There have been several works about vertical handover [1,2,3,4]. The authors of [1] introduced new performance metrics to provide seamless mobility for designing handover decision function. In [2], the authors proposed an

---

\* 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 Advancement) (IITA-2006-C1090-0603-0028).

end-to-end mobility management system that reduces unnecessary handover by obtaining the condition of different networks. Various network-layer based inter-network handover techniques are addressed and their performance is evaluated in a realistic network testbed in [3]. Most recently, a GPS-based location-aware vertical handover scheme is introduced in [4].

In most of the methods proposed in [1,2,3], note that MNs must turn on their WLAN interface to receive the periodic beacons from Access Points (APs) in the idle state, while at the same time, the WWAN interface at the node checks the paging channels from BSs (Base Stations), resulting in significant energy consumption. Even though in [4], the WLAN interface is closed down whenever it leaves a hotspot area so that both the WLAN and WWAN interfaces are not active all the time, there is still additional energy consumption due to continual GPS position scan resulting in limiting the energy capability in MNs. Therefore, in this paper, we propose Location-Aware Signaling protocol (LAS) for interworking between WWAN and WLAN heterogeneous networks, where during the idle state, the WLAN interface is turned off without any periodic wake-up (i.e., entering inactive state) in order to save energy consumption. Especially, the existing forward and reverse paging channels (F-PCH and R-PCH) of WWAN are exploited to notify the MN whose position is located within the coverage of AP hotspot without additional positioning device, i.e., GPS receiver.

Noting that WLAN is currently used primarily for long-lived multimedia data service [4,5], it is beneficial to make the WLAN interface enter the inactive state because there is no need to wake up the WLAN interface for the momentary traffic that had better be downloaded through WWAN. Thus, we devise a dwell-timer which is a timer for managing incoming traffic to be served by WLAN interface instead of WWAN interface. When incoming traffic toward to the MN exists, the MN receives the traffic via WWAN interface firstly with starting the dwell-timer. As soon as the dwell-timer expires, the MN turns on its WLAN interface and switches point of attachment from its WWAN interface. Using of the dwell-timer makes MN to process long-lived traffic more effectively with accurate information of AP location provided by WWAN signalings.

The rest of the paper is organized as follows. In Section 2, we describe a new protocol to utilize the paging messages from WWAN for WLAN. Section 3 provides the performance evaluation of proposed mechanism and discusses results in terms of energy consumption and overhead due to signaling through the WWAN PCHs. Finally, Section 4 concludes the paper.

## 2 Location-Aware Signaling Protocol

When connected to a WLAN, the WLAN interface typically stays in idle state almost 70% of the overall time [6]. Even in idle state with power-saving mode, it wakes up periodically to receive beacons from APs in order to check frames buffered for delivery to the interface, consuming its energy continuously. Usually, the energy consumption rate of WLAN and GPS interfaces are around 10 and 5 times higher than that of WWAN interface in the idle state, respectively [7,8].

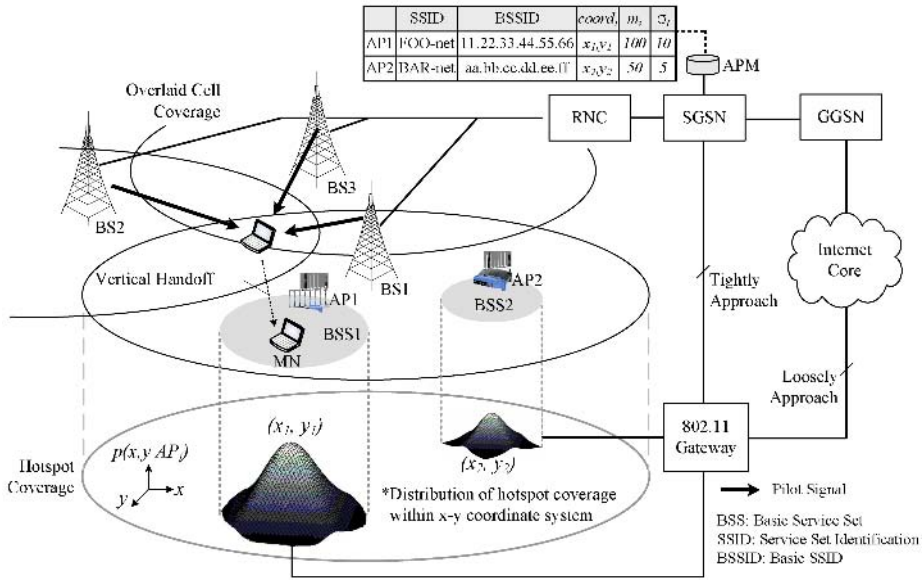


Fig. 1. Heterogeneous architecture for AP Mapping (APM) in 3G system

Thus, we propose to turn off the WLAN interface without any periodic wake-up during the idle period where the WWAN interface is assumed to listen PCH continuously to detect paging messages directed to the WLAN interface in addition to the messages addressed to it. Architectural details of our energy enhanced LAS protocol is described in this Section.

2.1 Overall Interworking Architecture

In order to utilize the PCH of cellular network for WLAN, we base our system model on tightly-coupled approach that makes the WLAN appear to the 3G core network as another 3G access network as shown in Fig. 1. Then, we can assume that the location information of each AP in an 802.11 access network is able to be registered with the corresponding SGSN at the initialization phase of the BS or when additional APs are deployed into the BS coverage as shown in Fig. 2(a). When AP registration is completed, SGSN stores AP information in local APM and delivers its information only to the BS whose coverage includes AP's BSS.

Initially, the propagation range of  $i$ -th AP in mean ( $m_i$ ) and variance ( $\sigma_i$ ) are chosen based on manual measurement when APs are deployed. The  $i$ -th AP is set with its own SSID and BSSID (i.e., MAC address of AP) and locates at  $(x_i, y_i)$ . In our system, every MN reports the result of attachment to an AP by signaling through WWAN PCHs, so that the propagation range of AP ( $m_i$  and  $\sigma_i$  of  $i$ -th AP in APM) is dynamically updated according to current network condition.

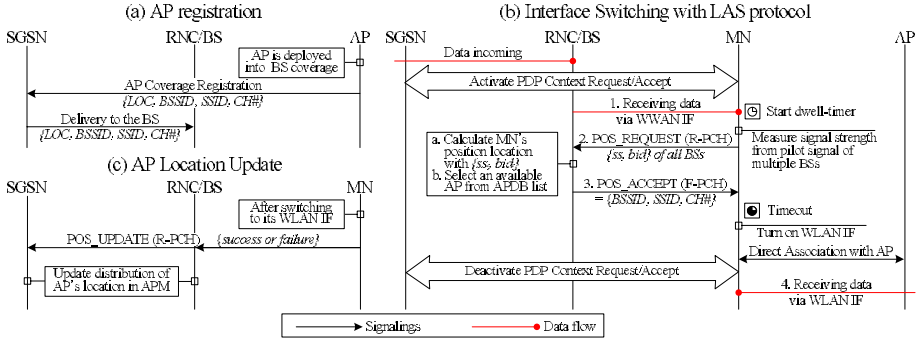


Fig. 2. Signaling procedure for downlink data traffic

## 2.2 Signaling Procedure for Utilizing Paging Messages

As paging messages for WLAN, three types of new message structures are defined by using the existing Data Burst Message (DBM) [9]. The Service Option (SO) field of DBM is used to differentiate the control information for WLAN with the existing DBMs (e.g. SMS: Short Message Service is one of the DBM applications identified with its SO field).

- *POS\_REQUEST* : From the pilot signals from  $i$ -th BS, MN obtains signal strength ( $SS_i$ ) and BS identifier ( $bid_i$ ). Then MN generates a *POS\_REQUEST* message putting the pairs of  $ss_i$  and  $bid_i$ , and sends this message to the BS through the reverse-PCH. Thus, the BS calculates the MN's position location ( $x, y$ ) with a set of ( $ss_i, bid_i$ ) pairs. Based on the calculated ( $x, y$ ), the BS selects a candidate AP from its APM.
- *POS\_ACCEPT* : The BS sends this message including information of the selected candidate AP (BSSID, SSID, and channel number). Then, the MN directly connects to the candidate AP via its WLAN interface without probing or active scanning delay for initialization of WLAN interface.
- *POS\_UPDATE* : After the MN turns on its WLAN interface and switching point of attachment to WLAN, MN reports the result whether the candidate AP is able to connect or not to the BS. Then, BS updates its APM field for candidate AP according to the result of success or failure.

As shown in Fig 2(b), the detailed signaling steps of LAS protocol are as follows:

- **Step 1.** When there is incoming data for MN, the MN receives the data via its WWAN interface with activated PDP context. At the same time, the MN starts its dwell-timer and receives pilot signals to measure signal strength for each BS as long as the dwell-timer is on.
- **Step 2.** The MN sends a *POS\_REQUEST* message to currently serving BS via the R-PCH. Then, the BS calculates the MN's position location and selects a candidate AP from its APM with received set of ( $ss, bid$ ).



- **Step 3.** The BS sends a *POS\_ACCEPT* message to the MN via the F-PCH. If a timeout occurs after receiving the *POS\_ACCEPT*, the MN turns on its WLAN interface and associates with the candidate AP directly without probe or active scanning delay. If no *POS\_ACCEPT* is received, it is not necessary to turn on the WLAN interface.
- **Step 4.** When the association between the MN and the AP is completed, the PDP context is deactivated to make the WWAN interface go to standby mode. Then, the incoming data starts to be received via its WLAN interface.

In Step 2, the BS calculates the probability of MN's position location based on more than one signal strength (*SS*) measured at MN being sent to current serving BS as in [10]. Let  $d_j$  denote distance between  $(x, y)$  and  $j$ -th BS among  $N$  number of BSs. The standard deviation and the long-term median of  $ss_j$  at  $d_j$  are denoted by  $\sigma_j^2$  and  $E_j$ , respectively. The pdf of  $ss_j$  at  $(x, y)$  is given by  $p(x, y|ss_j) = C_j \cdot \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(ss_j - E_j)^2}{2\sigma_j^2}\right)$ . For all BSs, the probability density function of *SS* at  $(x, y)$  is  $p(x, y|SS) = C \prod_{j=1}^N p(x, y|ss_j)$  where  $C$  and  $C_j$  are constants. Then,  $(x, y)$  can be estimated as  $E[x] = \int_{-\infty}^{\infty} x \left\{ \int_{-\infty}^{\infty} p(x, y|SS) dy \right\} dx$  and  $E[y] = \int_{-\infty}^{\infty} y \left\{ \int_{-\infty}^{\infty} p(x, y|SS) dx \right\} dy$ .

BS selects one candidate AP from APM based on the distribution function for hotspot area in each APs. Let  $p(x, y|AP_i)$  denote probability density function of connection availability when MN locates at position  $(x, y)$  and  $i$ -th AP locates at position  $(x_j, y_j)$ . With  $E[x]$  and  $E[y]$ , we calculate  $p(x, y|AP_i)$  as follows

$$p(x, y|AP_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(d_i(\theta) - m_i)^2}{2\sigma_i^2}\right) \quad (1)$$

where  $d_i(\theta)$  denotes the distance between  $(x, y)$  and  $(x_j, y_j)$ , and  $\theta$  is the direction of MN from AP. To distinguish direction of  $\theta$  degree in Eq. 1, we apply distance function  $w_i(\theta)$  to reduce or expand weighted distance for each  $i$ -th AP as  $d_i(\theta) = w_i(\theta) \sqrt{(x_i - E[x])^2 + (y_i - E[y])^2}$ . Among the APs listed in APM, the BS selects one candidate AP with the maximum  $p(AP_i)$  against MN's position location as  $\max[p(x, y|AP_i)] \forall i$ , if  $p(x, y|AP_i) > 0$ .

In Sec. 2.2, MN turns on its WLAN interface after it receives *POS\_ACCEPT*. However, even though the MN turns on its WLAN interface and tries to switch point of attachment to the candidate AP, it is possible that the MN can not connect to the candidate AP because of signaling failure. As shown in Fig. 2(c), we thus propose an algorithm for updating APM via *POS\_UPDATE* message to cope with dynamical distribution of hotspot as follows:

---

After MN switches WLAN interface to  $AP_j \Rightarrow$  sends *POS\_UPDATE* to serving BS  
if (connection successful)

$\Rightarrow$  BS decreases weighted distance in  $\theta$  degree,  $w_j(\theta)$

else (connection failure)

$\Rightarrow$  BS increases weighted distance in  $\theta$  degree,  $w_j(\theta)$

---



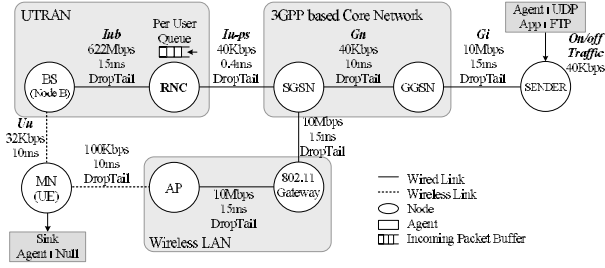


**Fig. 3.** Simulation topology for WWAN (3G) and WLAN integrated networks

### 3 Simulation Parameters and Results

We evaluate the performance of proposed LAS protocol utilizing WWAN PCH for WLAN and GPS interfaces. In terms of energy consumption and signaling overhead, we compare LAS mechanism with RSS and GPS based schemes through simulation, where RSS and GPS based scheme performs interface decision by using only received signal strength and by the current location of MN obtained from GPS receiver as introduced in [4], respectively.

#### 3.1 Simulation Topology

The simulation results presented in this paper are obtained by using EURANE UMTS extension-1.09 [11] for *ns-2* simulator [12]. All simulations are performed using integrated 3G and WLAN topology shown in Fig. 3, where downlink data is generated by exponential on/off traffic. Cellular and hotspot range are 500- and 50-meter radius, respectively. All the results are measured after statistical averaging on ten sample of simulations, lasting 3600 seconds each.

#### 3.2 Evaluation Parameters

For the performance comparison of LAS, RSS, and GPS based scheme, we provide cost parameters of energy consumption and signaling overhead as follows:

1) *Energy consumption*: We define energy consumption rate of multiple interfaces at MN as  $W_{st}^{if}$  where type of interface,  $if \in \{3G, WLAN, GPS\}$  and its operation state,  $st \in \{off, idle, connected\}$ . Based on the state transition of MN's interfaces as shown in Fig. 4, total energy consumption for all equipped network interface at a MN,  $E_{total}$  is given by

$$E_{total} = \sum_{if} \left[ W_{off}^{if} \cdot t_f + W_{idle}^{if} \cdot t_i + W_{connected}^{if} \cdot t_c \right] \quad (2)$$

where  $t_f$ ,  $t_i$ , and  $t_c$  are periods for off, idle, and connected states, respectively.

2) *Signaling overhead*: Let  $R_{pg}$ ,  $R_{bc}$ , and  $R_{gs}$  be the number of signalings for paging messages via 3G interface, for beacon frames via WLAN interface, and

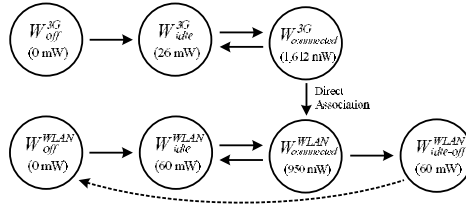


**Fig. 4.** State transition of interfaces and energy consumption rates for each state [7]

for satellite scanning via GPS interface, respectively. Then, the total number of received signaling packets at a MN over the period  $t$ ,  $R_{signaling}(t)$  is

$$R_{signaling}(t) = R_{pg}(t) + R_{bc}(t - \delta) + R_{gs}(t) \quad (3)$$

where  $t = t_i + t_c$  and  $\delta$  indicates the inactive period.

### 3.3 Performance Results

We first present simulation results of total energy consumption for MN's wireless interfaces,  $E_{total}$  obtained from Eq. 2. Figs. 5(a) and (b) are plots of  $E_{total}$  as a function of inactive period ( $t_{off}$ ) ranging from 30 to 360 sec for active periods ( $t_{on}$ ) are 120 and 360 sec, respectively. These results show the effect of energy saving for LAS scheme where it requires average 55.8% (62.9%) and 50.1% (50.0%) less energy than RSS and GPS based scheme for  $t_{on}=120$  sec (360 sec), respectively. We know that these improvements of LAS scheme are achieved by reducing energy consumption for receiving beacons or position scanning which is addressed by RSS or GPS based scheme even when MN is on inactive period.

It is noted that RSS and GPS based schemes address additional control information (e.g., beacon and coordinates) even when network interfaces stay in idle. However, these extras impose signaling overhead upon the scarce of radio resource. Now, we investigate the overhead generated by signaling obtained from Eq. 3. Figs. 6(a) and (b) are plots of total number of received signaling packet,

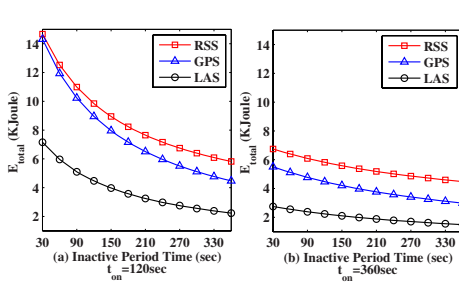


**Fig. 5.** Total energy consumption vs. inactive period for different active periods

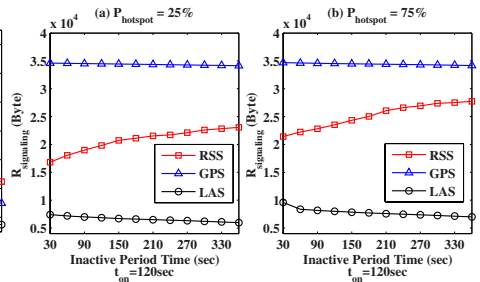


**Fig. 6.** Total amount of signalings of interfaces vs. proportion of hotspot area

$R_{\text{signaling}}$  as a function of active and inactive period same as that of Fig. 5(a) for different proportions of hotspot area per a cell ( $P_{\text{hotspot}}$ ) in 25% and 75%, respectively. These results show that average signaling gains for LAS scheme are 68.3% (69.0%) and 80.8% (77.4%) compared to RSS and GPS based scheme for  $P_{\text{hotspot}}=25\%$  (75%), respectively. We know that GPS based scheme requires relatively higher overhead for receiving coordinates, where  $R_{\text{signaling}}$  of GPS based scheme is a factor of 5.2 (4.4) and 1.7 (1.4) higher than that of LAS and RSS based schemes for  $P_{\text{hotspot}}=25\%$  (75%), respectively. Also, we can see that the more hotspot is deployed ( $P_{\text{hotspot}}=25$  to 75), the larger amount of signaling as increasing 18.1% and 20.9% for LAS and RSS based scheme. This indicates that LAS scheme brings smaller signaling imposition than RSS based scheme.

## 4 Conclusions

In this paper, we propose Location-Aware Signaling protocol for WWAN and WLAN integrated networks, where during the idle state, the WLAN interface is turned off without any periodic wake-up in order to save energy consumption. We exploit WWAN paging channels to notify MN of the presence of incoming traffic for the MN and the availability of AP. The simulation results show that our proposed mechanism has better performance than the existings in terms of energy consumption and signaling overhead.

## References

1. J. McNair and F. Zhu, "Vertical Handoffs in Fourth-Generation Multinetwork Environments", IEEE Wireless Communication, vol.11, no.3, pp.8–15, Jun. 2004.
2. C. Guo, Z. Guo, Q. Zhang and W. Zhu, "A Seamless and Proactive End-to-End Mobility Solution for Roaming Across Heterogeneous Wireless Networks", IEEE Journal on Selected Areas in Communications, vol.22, iss.5, pp.834–848, Jun. 2004.
3. R. Chakravorty, P. Vidales, K. Subramanian, I. Pratt and J. Crowcroft, "Performance Issues with Vertical Handovers - Experiences from GPRS Cellular and WLAN Hot-spots Integration", Proc. IEEE PerCom 2004, pp.155–164, Mar. 2004.
4. M. Ylianttila, J.M. and K.P., "Analysis of Handoff in a Location-aware Vertical Multi-access Network", Computer Networks, vol.47, iss.2, pp.185–201, Oct. 2005.
5. T. S. Rappaport, C. Na, J. K. Chen, "Convergence of Cellular and Wireless LAN: Hotspot Traffic Statistics and User Trends", CTIA show, Mar. 22, 2004.
6. R. Chary, R. Banginwar and J. Gilbert, "Power Management Technologies for WLAN enabled Handheld Devices", Intel Developer Forum, Fall 2003.
7. E. Shih, P. Bahl, and M. J. Sinclair, "Wake on Wireless: An Event Driven Energy Saving Strategy for Battery Operated Devices", MobiCom, pp.160–171, Sep. 2002.
8. Specification of GPS-9534, [http://www.leadtek.com/gps/gps\\_9534\\_2.html](http://www.leadtek.com/gps/gps_9534_2.html).
9. 3GPP, "Technical Specification Group Terminals - Technical realization of the Short Message Service (Release 6)", TS 23.040, v6.5.0, <http://www.3gpp.org>.
10. R. Yamamoto, H. Matsutani, H. Matsuki, T. Oono, and H. Ohtsuka, "Position Location Technologies using Signal Strength in Cellular Systems", VTC, vol.4, pp.2570–2574, May 2001.
11. Enhanced UTRAN Extensions for ns-2, <http://www.ti-wmc.nl/eurane>.
12. The Network Simulator - ns-2, <http://www.isi.edu/nsnam/ns>.

# Performance Evaluation of Non-persistent CSMA as Anti-collision Protocol for Active RFID Tags

E. Egea-López, J. Vales-Alonso, A.S. Martínez-Sala, M.V. Bueno-Delgado,  
and J. García-Haro

Department of Information Technologies and Communications  
Polytechnic University of Cartagena, Spain  
{esteban.egea, javier.vales, alejandros.martinez, mvictoria.bueno,  
joang.haro}@upct.es

**Abstract.** In this paper we propose the use of non-persistent CSMA as an anti-collision procedure for RFID active tags. Current proposals for both passive and active tags are based on the framed slotted ALOHA protocol, which does not scale well requiring additional procedures for frame length adaptation. However, active RFID devices already include carrier sense capabilities with no additional cost and, thus, CSMA may be employed seamlessly. Nevertheless, selecting the contention micro-slots of CSMA in the classical way (*i.e.*, with a uniform distribution and an exponential back-off algorithm) does not result in an efficient identification process, as we will demonstrate. Fortunately, better choices can be found. Recently, an optimal distribution for the selection of micro-slots for event-driven sensor networks has been computed, as well as a practical implementation: the Sift distribution. In this work we propose the application of the quasi-optimal Sift distribution along with CSMA for active tag identification. By means of an analytical study, we evaluate the average time needed for identification with this mechanism and compare it with the current ISO 18000-7 and EPC “Gen 2” standard. The results reveal that the Sift-based non-persistent CSMA outperforms both of them. Moreover, it also scales much better, without the need for further adaptation mechanisms.

**Keywords:** Radio Frequency Identification (RFID), anti-collision protocol, non-persistent CSMA, EPC “Gen 2”, ISO 18000-7, performance evaluation, active RFID tag.

## 1 Introduction

Radio Frequency Identification (RFID) systems are one of the enabling technologies for the ubiquitous computing paradigm [1]. Its foreseen application range spans from replacement of bar-code systems to location of containers in large cargo vehicles. A wide range of RFID technologies have been in study to match such a broad range of applications. All of them share a common architecture: a

basic RFID cell consists of a *reader* device (aka *master* or *interrogator*) and a (potentially large) set of RFID tags, which reply to the queries or enforce the commands from the interrogator. RFID devices are classified according to the source of energy of the tags: **passive** ones do not have a power source and obtain the energy from the reader signal (via induction), whereas **active** ones incorporate their own battery. On the one hand, passive tags are targeted to be inexpensive and, thus, very simple, usually read-only, devices. Their coverage typically ranges from centimetres to a couple of metres. On the other hand, active tags are more complex devices, with more sophisticated capabilities (usually integrating a microprocessor and memory) and they can be read and written from distances in excess of 100 meters [1]. Whereas passive RFID systems are the most deployed and have been studied for years [2,3,4], active RFID systems have devoted little academic attention, and only recently a standard is available [5].

In both cases, the *tag collision* problem arises: in a RFID cell, if multiple tags are to be identified simultaneously, reply messages from tags can collide and cancel each other. Thus, an anti-collision mechanism is needed. Since, in a typical application, items (with attached tags) enter and leave the reader coverage area, the goal of this mechanism is to communicate with the tags as quickly and reliably as possible, ensuring that all tags have been identified. An additional goal for active tags is to save energy in order to maximise the battery lifetime. Therefore, the tag identification problem deals with identifying multiple objects with minimal delay and power consumption, reliability, line-of-sight independence and scalability. Unlike classical medium access protocols, channel utilisation and fairness are not usually issues in RFID systems.

For passive tags, mainly due to the limitations of the devices, the protocols are very simple and most of them fall into the following two categories [2]:

- *Splitting algorithms.* The set of tags to be identified is split in disjoint smaller subsets until the number of tags in a subset becomes one. It is done either by the tags selecting a random number or by the reader sending a string that matches only a subset of tags identification number (ID). Algorithms of this type can be viewed as a tree search.
- *Probabilistic algorithms.* The other major family of protocols is based on Framed Slotted ALOHA (FSA) [6]. In this case, after receiving a signal from the reader, the tags randomly select a slot out of  $K$  (the frame length) and send their ID. This mechanism is very simple, but when the number of tags increases, it needs some mechanism to adapt the frame length ( $K$ ) [3].

What are the approaches used with active tags? The answer is not straightforward due to some existing confusion around this technology. On one hand, there is a lack of scientific literature that specifically addresses the collision-resolution problem for active tags. The ISO 18000-7 standard [5] deals with it and proposes framed slotted ALOHA as an anti-collision protocol, suggesting a frame length

adaption mechanism but without specifying a particular one and leaving it open to the vendors. In addition, the EPCglobal organisation, leader of development of industry-driven standards for this field, has settled the EPC “Gen 2” as a reference standard [7]. The anti-collision procedure of “Gen 2” is supposed to be independent of the class of device, that is, passive or active. With minor changes, EPC “Gen 2” has also chosen framed slotted ALOHA and does suggest a specific algorithm for frame length adaptation.

On the other hand, a typical active tag has the capabilities of an on-board microprocessor and a sophisticated transceiver and may use Bluetooth or IEEE 802.11 protocols or Wireless Sensor Networks (WSN) MAC protocols [8]. It is clear that these protocols are designed with different requirements in mind and, at the moment, the cost of these devices is still possibly too high. Therefore, it seems that the possible choices are: very simple approaches suitable for passive tags or very sophisticated proposals designed for different purposes.

Is any of these choices efficient? In this paper we explore an intermediate solution: the use of non-persistent Carrier Sense Multiple Access (CSMA) as anti-collision mechanism for active RFID tags. With this mechanism, after receiving an identification request, nodes would listen to the channel for a randomly selected number of contention slots<sup>1</sup>. A node would transmit if the channel remains idle during this interval. Otherwise, it would defer transmission. As we will show, if micro-slots are selected uniformly (the classical approach), the identification of tags does not have to be more efficient than a framed slotted ALOHA. However, an optimised distribution for the selection of CSMA contention micro-slots has been proposed for Wireless Sensor Networks [9]. This distribution minimises the probability of collision when  $N$  stations become simultaneously backlogged, which is exactly the main problem of RFID identification. Using this distribution, the identification process is faster and more scalable than the FSA proposals, and even simplifies the implementation of the protocol. It should be remarked that the use of CSMA is feasible for active RFID devices: A typical low cost chip for active RFID [1] *already* integrates carrier sensing capabilities without any additional cost, even though in the application domain of active RFID (vehicle and container tracking and management) it is already assumed the need for more complex and expensive tags. Moreover, it may reduce the cost compared to devices that use complex protocols like IEEE 802.11.

In this paper we support this solution by studying analytically the performance of quasi-optimal non-persistent CSMA as an anti-collision mechanism for RFID identification. We compute the average number of identification cycles needed to identify all the tags present in a coverage cell for CSMA and ISO 18000-7 or EPC “Gen 2” and compare them. The rest of this paper is organised as follows: Section 2 briefly reviews the related work. In Section 3 the different proposals to be compared are reviewed and analysed. Section 4 provides a comparison and discussion of the proposals. Finally, section 5 concludes and outlines possible future works.

---

<sup>1</sup> We will refer to a contention slot as *micro-slot*, to distinguish it from a slotted ALOHA one.

## 2 Related Work

Most of the anti-collision protocols focus on passive RFID tags [2]. In this case, the limitations of the device, usually impose the use of very simple protocols, all the burden of the identification process lying on the reader. The different proposals fall into the two following categories: (i) splitting algorithms or (ii) probabilistic protocols. In the first group, a well-known protocol, called QT memoryless [2] exemplifies its operation: the reader sends a string prefix and all the tags whose ID match that prefix reply. The reader appends a new digit to the string prefix subsequently. If there is no collision in the response, a tag has been identified. This type of algorithms can be observed as a tree search and are deterministic, meaning that all tags are identified with probability 1 within a bounded time. However, this time can be very long, depending on the length of the tag IDs.

In the second group, framed slotted ALOHA is practically a unanimous choice. For instance, the I-Code protocol [3] is used with passive tags. Surprisingly, most of the proposals for active systems have also selected this approach: the ISO 18000-7 standard [5] as well as the EPC “Gen 2” protocol [7]. The latter, which is expected to be a *de facto* standard, is to be used with both active and passive tags. Unlike ISO 18000-7, EPC does suggest a procedure to adapt the frame length.

Vogt [3] analyses the identification process of framed slotted ALOHA as a Markov chain and derives two procedures to dynamically adapt the frame length. It is assumed that tags are not acknowledged and all tags participate in every identification round. In this paper, we use a slightly modified analysis, considering that identified tags do not keep on participating, since the two major proposals, ISO 18000-7 and EPC “Gen 2”, state that tags retire after being acknowledged.

As said in Section 1, we propose the use of CSMA with the optimal probability distribution ( $p^*$ ) for the selection of CSMA contention micro-slots derived in [9]. This distribution maximises the probability of success when  $N$  stations become simultaneously backlogged, but depends on the number of slots in use ( $K$ ) and the number of nodes ( $N$ ) contending. Since the latter is usually unknown (also in RFID), an approximation is also provided, the Sift distribution, which not only keeps close to the optimal for a wide range of its configuration parameters but it is also scalable. The authors of reference [9] discuss different applications in wireless sensor networks, but RFID is not mentioned. In this paper we show that RFID is a major field of application of this optimised distribution.

Finally, both CSMA and framed slotted ALOHA have been extensively studied [10,6], but as classical MAC protocols, focusing on the channel utilisation and access delay. In RFID, on the contrary, the appropriate performance metric is the identification delay. We evaluate the performance of the protocols regarding this metric.

## 3 Analysis of Proposals

In this section, the different proposals to be compared will be reviewed and analysed.

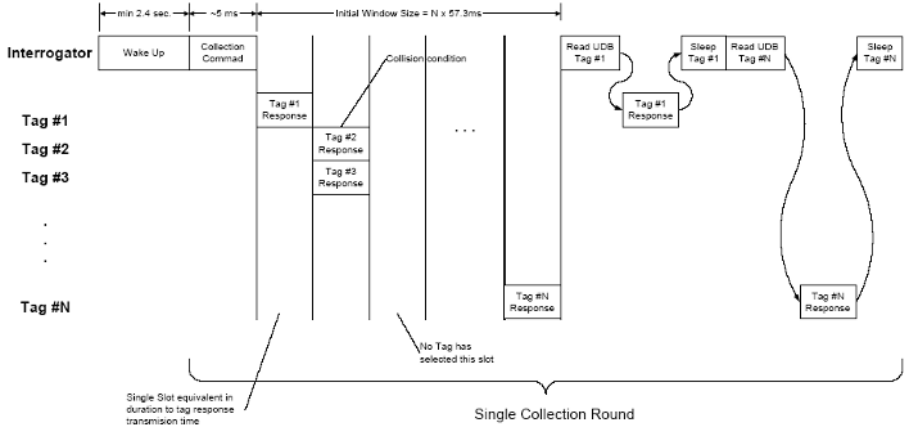


Fig. 1. Anti-collision procedure of ISO 18000-7 (from [5])

### 3.1 Framed Slotted ALOHA

Both ISO 18000-7 and EPC “Gen 2” [5,7] define a similar anti-collision procedure that we call generically: Framed slotted ALOHA (FSA). In both cases, a population of  $N$  tags start the identification process after receiving a collection command from the interrogator. At this moment, nodes randomly select a slot with a uniform distribution and transmit their ID at the selected slot. We refer to the number of possible slots to choose as *frame length*,  $K$ . If two or more nodes select the same slot, a collision occurs. For each slot with a single reply, the interrogator sends an ACK packet to put the tag to sleep, preventing it from participating again in the identification process. The acknowledged tags (already identified) withdraw from contention in the following rounds. Fig. 1 illustrates the process. We refer to a collection command plus the  $K$  slots as an *identification cycle*. Although shown in Fig. 1, we assume no transmission of data is done and tags only identify themselves. After three collection rounds without reply, the interrogator assumes that all nodes have been identified.

As explained in [3], the identification process can be modelled as a (homogeneous) Markov process  $\{X_s\}$ , where  $X_s$  denotes the number of tags *unidentified* at the  $s$  identification cycle. Thus, the state space of the Markov process is  $\{N, N-1, \dots, 0\}$ . The probability distribution of the random variable  $\mu_r$  that indicates the number of slots being filled with exactly  $r$  tags is:

$$P_{K,N}(\mu_r = m) = \frac{\binom{K}{m} \prod_{i=0}^{m-1} \binom{N-ir}{r} G(K-m, N-mr, r)}{K^N} \quad (1)$$

where  $m = 0 \dots K$  and

$$G(M, l, v) = M^l + \sum_{i=1}^{\lfloor \frac{l}{v} \rfloor} \left\{ (-1)^i \prod_{j=0}^{i-1} \left\{ \binom{l-jv}{v} (M-j) \right\} (M-i)^{l-iv} \frac{1}{i!} \right\} \quad (2)$$



**Table 1.** Average number of identification cycles versus number of tags

| Slots / Tags | 10   | 20   | 30   | 40   | 50               | 60               | 70               | 80               | 90               | 100                 |
|--------------|------|------|------|------|------------------|------------------|------------------|------------------|------------------|---------------------|
| 4            | 8.2  | 60   | 630  | 8159 | $1.1 \cdot 10^5$ | $1.6 \cdot 10^6$ | $2.5 \cdot 10^7$ | $3.8 \cdot 10^8$ | $6.0 \cdot 10^9$ | $9.6 \cdot 10^{10}$ |
| 8            | 3.67 | 8.56 | 19.6 | 49.4 | 138.0            | 413.9            | 1304.2           | 4244.6           | 14127            | 47797               |
| 16           | 2.44 | 4.11 | 6.15 | 8.93 | 13.03            | 19.3             | 29.41            | 46.0             | 73.81            | 121.3               |
| 32           | 1.89 | 2.76 | 3.60 | 4.47 | 5.424            | 6.50             | 7.76             | 9.26             | 11.0             | 13.2                |
| 64           | 1.54 | 2.15 | 2.61 | 3.06 | 3.465            | 3.90             | 4.32             | 4.77             | 5.23             | 5.72                |

Since all the acknowledged tags in a cycle withdraw from contention, the transition matrix  $H$  and transition probabilities are given by

$$h_{ij} = \begin{cases} P_{K,N-i}(\mu_1 = j - i), & i < j \leq i + K \\ 1 - \sum_{k=i+1}^{i+K} h_{i,k}, & i = j \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where  $i = 0 \dots N$ . Since this is an absorbing Markov chain, the average number of identification cycles equals the average number of steps to absorption, which is given by

$$\mathbf{t} = F\mathbf{c} \quad (4)$$

where  $\mathbf{t}$  is a column vector and  $t_s$  is the expected number of steps (cycles, in our case) before the chain is absorbed given that the chain starts in state  $X_s$ ,  $F$  is the *fundamental matrix* of  $H$  and  $\mathbf{c}$  is a column vector all of whose entries are 1 (see [11]). Thus, if the starting state is  $X_1$ , that is, all the  $N$  tags to be identified, the average number of cycles to identify all the tags is  $t_1$ .

Table 1 shows the average number of cycles versus number of tags ( $N$ ) for different frame lengths (in number of slots). It shows that with a fixed framed length, the number of cycles increases exponentially with the number of tags. Therefore, a simple mechanism like framed slotted ALOHA does not scale well, requiring a frame adaptation mechanism as the number of tags increases.

### 3.2 Carrier Sense Multiple Access

The operation of the identification protocol when using CSMA would be as follows: after receiving a collection command from the reader all  $N$  tags listen to the channel for a number of *micro-slots* chosen randomly from a set of  $K$ . If the channel remains idle after the number of selected micro-slots, a node sends

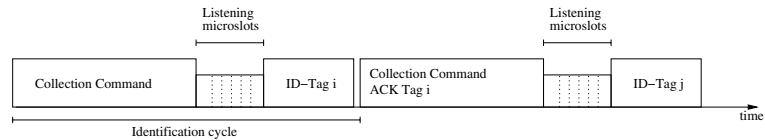


**Fig. 2.** Anti-collision procedure with CSMA

its ID. Otherwise, it withdraws until the next collection command. If there is no collision, the reader sends an ACK-Collection command, which indicates the node already identified and asks for more IDs. The remaining nodes start the process again. Fig. 2 illustrates this mechanism.

The probability of success  $\pi_p(N)$  when  $N$  nodes select a contention micro-slot using probability distribution  $p$ , where  $p_r$  is the probability each contender independently picks slot  $r$ , is [9]:

$$\pi_p(N) = N \sum_{s=1}^{K-1} p_s \left(1 - \sum_{r=1}^s p_r\right)^{(N-1)} \quad (5)$$

In this case, the transition matrix  $H$  for the Markov process  $\{X_s\}$  defined previously is

$$h_{ij} = \begin{cases} \pi_p(N-i), & j = i+1 \\ 1 - \pi_p(N-i), & i = j \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

The average number of steps until absorption is computed as previously.

Let us assume first that the slots are chosen uniformly. In this case  $p_r = \frac{1}{K}$ . Fig. 3(a) and 3(b) show the results using a uniform distribution for different number of micro-slots. Again, the actual duration of an identification cycle depends on the number of micro-slots, the packet length and the transmission rate. The parameters for the computation of the average time are given in Section 4. Like framed slotted ALOHA, this procedure does not scale well either. In fact, its performance is worse and together with the additional device complexity it may be one of the reasons why it has never been proposed as an anti-collision procedure for RFID systems.

Let us assume now the Sift distribution is used, which is an approximation to the optimised distribution derived in reference [9]. In this case,  $p_r = \frac{\alpha^{-r}(1-\alpha)\alpha^K}{1-\alpha^K}$  for  $r = 1 \dots K$  and  $\alpha = M^{\frac{-1}{K-1}}$ .  $M$  is a parameter of the Sift distribution, pre-configured before deployment and representing the maximum number of contenders (as expected by the designer). The results shown in Fig. 4(a) and 4(b) reveal that the number of cycles increases almost linearly with the number of tags, unlike the exponential increment of framed slotted ALOHA. Therefore, this procedure scales well. In addition, by increasing the number of micro-slots the number of cycles tends to the minimum necessary ( $N$  cycles), but it implies increasing the duration of a cycle and may be even counterproductive: as seen in Fig. 4(b), 8 micro-slots are enough to handle the entire range of tags.

These results show that after choosing carefully the distribution for the contention window CSMA becomes a scalable technique for the identification of RFID tags. In Section 4, the different proposals for active tags are compared and discussed.

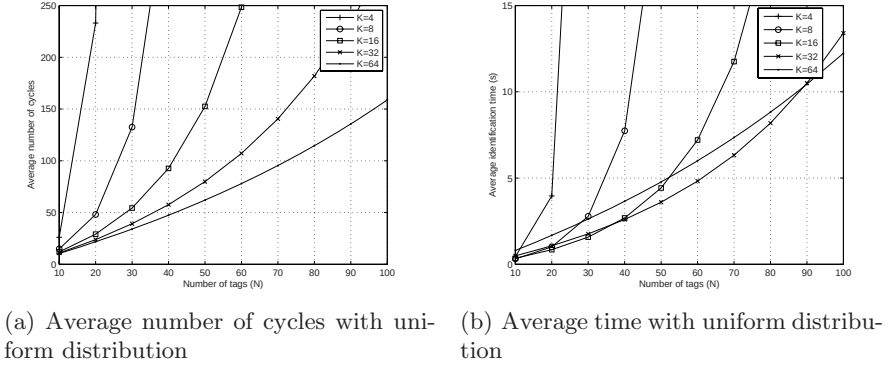


**Fig. 3.** Performance of identification process with uniform distribution

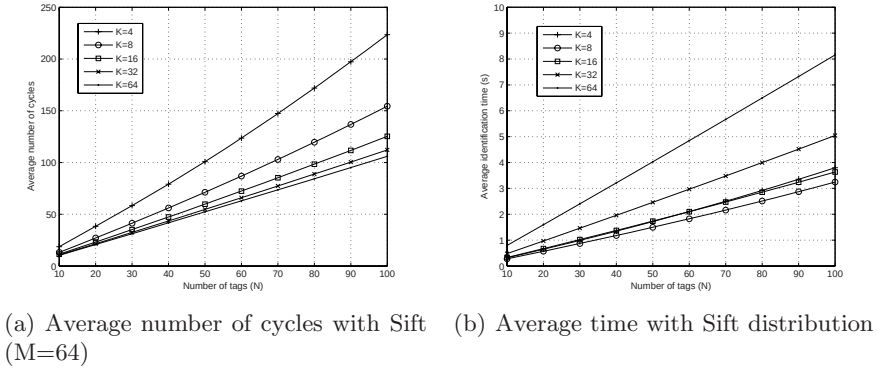


**Fig. 4.** Performance of identification process with Sift distribution

## 4 Comparison of Proposals

In order to compare the different proposals the identification cycles are translated to absolute time, since the actual duration of an identification cycle depends on the number of slots. First, we compare non-persistent CSMA with Sift distribution versus ISO 18000-7. The parameters are chosen from the specification [5]. An identification cycle lasts a collection command (5 ms) plus each slot (8 ms). Finally, for each identified tag the interrogator sends an ACK packet (5 ms) before starting a new cycle. For the CSMA cycle, we assume the same duration for the interrogator commands (5 ms) and ID packets (8 ms) plus the time for *all* the micro-slots, though the expected successful slot comes earlier. In fact, the performance depends to a great extent on the minimum time needed to perform the carrier sense, that is, the duration of the contention micro-slot. The duration and accuracy of carrier sensing (Clear Channel Assessment, CCA) depends on the technology, device and implementation [12]. There are many

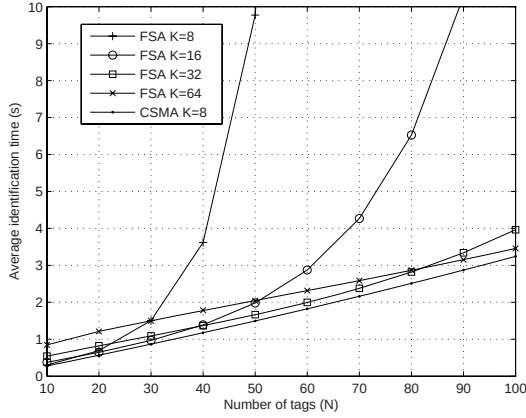


**Fig. 5.** Comparison of ISO 18000-7 versus CSMA with Sift distribution

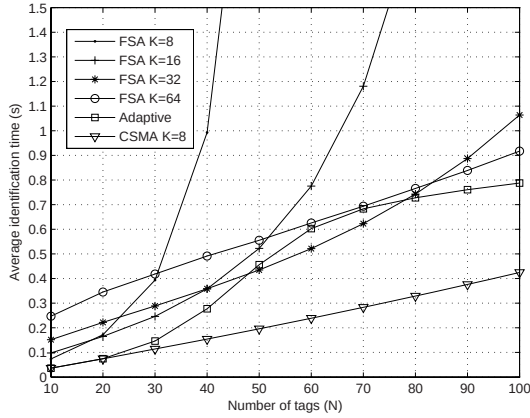


**Fig. 6.** Comparison of EPC “Gen 2” versus CSMA with Sift distribution

possibilities, but we assume that devices use coherent CCA, that is, the channel is busy when the packet preamble is detected. Thus, we set the micro-slot time to the duration of preamble. For ISO 18000-7, preamble is around 1 ms and so it is the time per micro-slot. Indeed it can be considered a conservative value, since current devices can perform this task in less time [12]. Even though, as can be seen in Fig. 5, non-persistent Sift-based CSMA with 8 micro-slots (and Sift parameter  $M=64$ ) outperforms the procedure proposed by ISO 18000-7 for every frame length. When the number of tags is low, the improvement is not significant, but, as the number increases, framed slotted ALOHA becomes unstable and the frame length must be adapted. On the contrary, non-persistent Sift-based CSMA can handle all the range of tags seamlessly.

Finally, we compare with EPC “Gen 2”. In this case, according to the specification [7], empty slots and slots with collision are shorter than slots with correct tag ID packets. However, we provide an approximation of the average identification time, assuming that the duration of all slots is the same and equals 2.505 ms, which is the time needed for the correct identification of a single tag at 40 Kbps. Thus, this is a conservative estimate, since empty and collision slots are actually shorter (0.575 ms). For CSMA, we again assume that in a cycle we have a tag ID packet (1.4 ms) plus an interrogator ACK-Collection (0.55 ms), plus the duration of the entire contention window, as before. In this case, we set the micro-slot time to 100  $\mu$ s, which is again the duration of the preamble. In Figure 6 we depict the results for EPC and non-persistent Sift-based CSMA ( $M=64$ ). In addition to the average number of cycles previously computed, we have simulated the EPC frame adaptation mechanism recommended in the specification [7]. It is also included in Figure 6 labelled as “Adaptive”. Obviously, for fixed frame length the results are the same as in Fig. 5 but with another time scale. However, this figure shows in addition that non-persistent Sift-based CSMA outperforms EPC with frame adaptation as well. Moreover, in this case, the improvement is even better (around 50 % for almost every number of tags) due to the shorter micro-slot, as discussed before.

In summary, CSMA allows for a quicker identification of tags in all the cases. The actual improvement depends on the duration of the contention micro-slot. However, it is more important to remark that CSMA also *scales much better* than framed slotted ALOHA, even with frame adaptation, which simplifies the implementation of reader and tags.

## 5 Conclusions

In this paper we propose the use of non-persistent Sift-based CSMA as an anti-collision procedure for RFID active tags. Current proposals directly use the approach for passive tags, that is, framed slotted ALOHA, which does not scale well and needs an additional procedure for frame adaptation. However, active devices already include carrier sense capabilities with no extra cost and, thus, CSMA may be used seamlessly. In fact, compared to some commercial products that use more sophisticated protocols, like IEEE 802.11 or Bluetooth, CSMA may even reduce the cost of active devices while achieving the goal of the anti-collision procedure.

To support our proposal, we have evaluated the average time needed for identification with this mechanism and compared it with the current ISO 18000-7 and EPC “Gen 2” standards. The results show that CSMA outperforms both of them. For instance, the average identification time can be decreased by 50 % compared to EPC with frame adaption. In fact, the performance improvement depends on the duration of the contention micro-slots. More important is the fact that, in both cases, non-persistent Sift-based CSMA also scales much better: configured with 8 micro-slots CSMA can effectively handle a range that spans from a few tags to hundreds of them, without the need for additional

adaptation mechanisms, which simplifies the implementation of reader and tags. Even though, the number of contention microslots may also be adapted if necessary. As future work we are currently investigating adaptation mechanisms for CSMA as well as different techniques to optimise the frame length of slotted ALOHA.

## Acknowledgements

This work has been funded by the Spanish Ministerio de Educación y Ciencia with the projects DEP2006-56158-C03-03/EQUI and m:ciudad (FIT-330503-2006-2, partially funded by ERDF) and by the Spanish Research Council with the ARPaq project (TEC2004-05622-C04-02/TCM).

## References

1. Stanford, V., "Pervasive Computing Goes the Last Hundred Feet with RFID Systems", *IEEE Pervasive Computing*, vol. 2, no. 2, pp. 9–14, April–June 2004.
2. Shih, D., Sun, P., Yen, D., Huang, S., "Taxonomy and survey of RFID anti-collision protocols", *Elsevier Computer Communications*, vol. 29, pp. 2150–2166, 2006.
3. Vogt, H., "Efficient Object Identification with Passive RFID Tags", *Lecture Notes in Computer Science*, vol. 2414, pp. 98–113, 2002.
4. Zhou, F., Chen, C., Jin, D., Huang, C., Min, H., "Evaluating and Optimizing Power Consumption for Anti-Collision Protocols for Applications in RFID Systems", in *Proc. Int. Symp. on Low Power Electronics and Design 2004*, pp. 357–362, 2004.
5. ISO/IEC 18000-7:2004 Information technology–Radio frequency identification for item management–Part 7: Parameters for active air interface at 433 MHz, 2004.
6. Wieselthier, J. E., Ephremides, A., Michaels, L. A., "An exact analysis and performance evaluation of framed ALOHA with capture", *IEEE Transactions on Communications*, vol. 37(2), pp. 125–137, 1988.
7. Class 1 Generation 2 UHF Air Interface Protocol Standard Version 1.0.9: "Gen 2". Available online at: <http://www.epcglobalinc.org/standards>
8. Zhao, F., y Guibas, L., *Wireless Sensor Networks. An information processing approach*. Morgan Kaufmann, 2004.
9. Tay, Y., C., Jamieson, K., Balakrishnan, H., "Collision-Minimizing CSMA and its Applications to Wireless Sensor Networks", *IEEE Journal on Selected Areas in Communications*, vol. 22(6), pp. 1048–1057, 2004.
10. Kleinrock, L., Tobagi, F., "Packet Switching in Radio Channels: Part I-Carrier Sense Multiple Access and their throughput-delay characteristics", *IEEE Transactions on Communications*, vol. 23, pp. 1400–1416, 1975.
11. Grinstead, C. M., Snell, J. L., *Introduction to Probability*, 2nd Edition, American Mathematical Society, 2003.
12. Ramachandran, I., Das, A., Roy, S., "Clear Channel Assessment in Energy-constrained Wideband Wireless Networks", *IEEE Wireless Communications Magazine*, forthcoming.

# Multicast Overlay Spanning Tree Protocol for Ad Hoc Networks

Georgios Rodolakis, Amina Meraihi Naimi, and Anis Laouiti\*

INRIA and Ecole Polytechnique, France

Georges.Rodolakis@inria.fr, Amina.Naimi@inria.fr, Anis.Laouiti@inria.fr

**Abstract.** In this paper we present an extension to the OLSR unicast routing protocol to support multicast routing in mobile ad hoc networks. The proposed protocol is based on Multicast Overlay Spanning Trees (MOST). The main benefits of this approach are twofold. Firstly, it implies that only nodes interested in taking part in the multicast communication need to participate in the protocol operation, which is transparent to other OLSR nodes. In addition, the MOST approach scaling properties achieve the theoretical performance bounds concerning the capacity of multicast communication in massive ad hoc networks. We perform simulations of the MOST protocol under the ns-2 simulator to compare with the theoretical results, and we present a fully working implementation for real network environments.

## 1 Introduction

Multicast offers an elegant way to establish group communication between users by using the concept of multicast groups, which are defined by their corresponding address. Interested clients can join and leave those groups in order to send and/or receive data from other group members. Moreover, the mechanisms which enable multicast communication ensure that an efficient strategy is used to deliver the data packets to all the members simultaneously. Therefore, multicast communication is adequate for a large class of applications, such as video-conferences, multi-player games, streaming applications *etc.* The previously described requirements make multicast routing an important and difficult challenge in the Internet, and even more so in ad hoc networks. In fact, mainly due to the dynamic nature of the routes, multicast protocols developed for wired networks cannot operate in the harsher mobile environment. This creates a need for protocols which are specially adapted to ad hoc networks.

Multicast ad hoc protocols can be classified according to the underlying routing structure to tree-based protocols and mesh-based protocols. The routing structure can be either group shared, or source dependent. Some tree-based protocols are MAODV [16] which is an extension to the unicast routing protocol AODV [15] based on a group shared tree, MOLSR [12] which is an extension to OLSR unicast routing protocol [6], based on a Dijkstra tree, and Adaptive

---

\* Anis Laouiti is currently with GET/INT, France.

Demand-driven Multicast Routing Protocol (ADMR) [11]. A protocol which is based on overlay trees is AMRoute [14]. As an example of mesh-based routing protocols we mention On-Demand Multicast Routing Protocol (ODMRP) [13] and Core-Assisted Mesh Protocol (CAMP) [8].

In contrast to the previously described protocols our work is motivated by analytical results on the achievable capacity of multicast communication in ad hoc networks. We show how to use minimum spanning trees to perform efficient multicast routing. The performance of minimum spanning tree multicast in ad hoc networks was studied in [10] and it was shown via analysis to be nearly optimal in case the number of multicast clients of each group is small compared to the network size. The advantage of this approach is that only multicast nodes need to participate in the multicast protocol. We present an overlay group shared tree based routing protocol called MOST which works in conjunction with the unicast OLSR protocol. The MOST protocol being based on overlay trees guarantees robustness, while it achieves good performance since OLSR provides topology information allowing to construct an optimal multicast tree. In addition, the MOST protocol is designed with the aim to achieve the theoretical capacity bounds.

The rest of the paper is organized as follows. In Section 2 we summarize the theoretical results concerning multicast scaling in ad hoc networks. In Section 3 we present the MOST protocol specification. A description of the protocol implementation is provided in Section 4. In Section 5 we present ns-2 simulations of the proposed protocol, which are compared to the theoretical analysis.

## 2 Asymptotic Multicast Properties in Ad Hoc Networks

One of the advantages of multicast routing is that it reduces the total bandwidth required to communicate with all group destinations, since some links can be common to several destinations. In wired networks, the gain of multicast communication has been studied in [4,5], by estimating the ratio of the number of links in a multicast tree to  $n$  destinations over the average unicast hop distance between two random nodes. The resulting normalized multicast cost has been found experimentally to scale in  $n^{0.8}$ . The gain of multicast is reflected by how far the normalized multicast cost deviates from linear growth. However, the topology of mobile ad hoc networks is significantly different and one would expect a much different scaling law. Indeed the average unicast hop distance in wired networks is usually of the order  $\log N$ , where  $N$  is the total number of nodes in the network, while in ad hoc networks the average distance grows proportionally to  $\sqrt{N/\log N}$ , since the optimal neighbor degree increases in  $O(\log N)$  when the capacity increases [9].

In [10], performance bounds are established on the expected size of multicast trees as a function of the number of multicast destinations  $n$ , both via analytical methods and via simulation. In random mobile ad hoc networks, the gain of multicast communication compared to unicast is significantly larger than in wired networks. For instance, a scaling law in  $O(\sqrt{n})$  holds for the normalized



multicast cost. The implications of this scaling law consist in a significant increase of the total capacity of the network for data delivery, which will be proportional to  $\sqrt{\frac{nN}{\log N}}$ . It is shown that when the number of multicast clients for each group is small compared to the network size, minimum spanning trees lead to asymptotically optimal performance in terms of network bandwidth utilization and achieve the theoretical capacity bounds. Therefore, the theoretical analysis can be applied in the design of an efficient multicast protocol based on the overlay minimum spanning tree approach.

### 3 Multicast Overlay Spanning Tree (MOST) Protocol

In this section, we present an extension to the OLSR unicast routing protocol, called MOST (Multicast Overlay Spanning Tree), in which we take into consideration the previously described theoretical results. First we present the main algorithms used for the multicast tree construction, followed by the protocol description and specification.

#### 3.1 Overlay Tree Construction Algorithms

As we saw previously, it is more efficient to consider minimum spanning trees. We discuss two algorithms for the overlay tree construction, which achieve optimal normalized multicast cost. The algorithms do not require any more information than what is provided by a link state unicast routing protocol, like OLSR.

---

##### **Algorithm 1.** Basic Minimum Spanning Tree Algorithm

---

**Input:** Network graph.

**Output:** Overlay tree.

1. Find shortest paths between all pairs of multicast nodes.
  2. Build complete graph on multicast nodes with costs  $c_{ij} = \{\text{length of shortest path between } i \text{ and } j\}$ .
  3. Build minimum spanning tree on the complete graph, rooted at the source node.
- 

In algorithm 1 the construction of the minimum spanning tree (step 3) can be implemented using Prim's algorithm. The resulting tree is an overlay multicast tree, since it consists only of multicast nodes and its links are in fact tunnels in the actual network. Multicasting is achieved when each node forwards multicast packets to its successors in the overlay tree. It must be noted that in a fully distributed protocol, each node must be able to compute the same minimum spanning tree independently from the others. Therefore, we impose an ordering to the multicast nodes based on their IP addresses when executing Prim's algorithm. The computed minimum spanning tree will be directed and rooted to the node with the smallest IP address. However, this fact has no practical importance in the protocol's operation, where the tree will be treated as a shared tree

The algorithm's expected complexity can be improved because it is not necessary in practice to compute all shortest paths from each node to all other nodes to build the minimum spanning tree. We propose Algorithm 2 as a faster alternative to compute minimum spanning overlay trees. The algorithm is essentially equivalent to Algorithm 1, but the shortest paths are calculated in conjunction with the minimum spanning tree. Hence, it is not necessary to compute shortest paths between all pairs of multicast nodes. In fact, according to tests in wireless network topologies, this algorithm has an average running time comparable to a Dijkstra algorithm, even when the number of clients increases.

---

**Input:** Weighted Graph  $G(V, E, w)$ , Multicast Node Set  $S$ , Root Node  $s$ .  
**Output:** Predecessor Table  $\pi$ .

- We denote  $G(V, E, w)$  the network graph, where  $V$  is the node set,  $E$  is the edge set, and each edge  $e$  is associated with a cost  $w(e)$ . We also denote  $S$  the set of multicast nodes. The array  $d$  associates each node with a distance to the multicast overlay tree, *i.e.*,  $d[v]$  corresponds to the minimum distance of node  $v$  to the multicast nodes that are already part of the tree. This distance is initialized to 0 for the root node and to  $\infty$  for all other nodes. The array  $\pi$  associates each node with a predecessor multicast node. When this table has been computed, it contains the information needed to represent the overlay tree, since each multicast node will be associated with another multicast node (except from the root). The predecessors of the other nodes in the graph need only be maintained during the computations.

The algorithm manages a set  $F$  of multicast nodes that have not been covered yet by the tree, and a min-priority queue  $Q$  which includes all nodes, with the

priority attribute being equal to their distance  $d$ . In each iteration the algorithm chooses a node with the smallest distance to the overlay tree (step 6), and checks whether it is a multicast node (step 7). In this case, the node's distance is updated to 0 (because the node is added to the overlay tree) and it is removed from the set  $F$ . Afterwards, for each chosen node, steps 11–15 check its adjacent nodes on whether their distance can be improved, and update the predecessors appropriately, similarly to Dijkstra's algorithm. However, in this case there are two important differences: the overlay predecessors can only be multicast nodes, hence steps 14–15 perform an additional check; moreover, previously extracted non-multicast nodes might be re-inserted in the priority queue in case their distance to the tree has improved due to the addition of new overlay nodes. The iteration ends when multicast nodes have been covered, hence the improvement in the average case complexity.

### 3.2 Specification of MOST Protocol

We now present the MOST multicast routing protocol which is based on overlay group shared trees. The protocol must be used in conjunction with a link state protocol, hence we choose to develop an extension to OLSR. One of the advantages of the overlay approach is that only the multicast nodes need to participate in the construction of the multicast tree, while the other nodes serve merely as relays and are not necessarily aware of the multicast communication. This fact facilitates the development of a peer to peer protocol which can be run only by the participating multicast nodes, hence it could be downloaded dynamically by a node whenever it decides to join a multicast communication.

The Optimized Link State Routing protocol [6] is a proactive table-driven routing protocol. An OLSR node uses Hello messages in order to detect neighbors and informs the entire network of its local topology by broadcasting TC messages. Broadcast traffic is relayed via Multi-Point Relay (MPR) nodes. MPR nodes are elected by their neighbors because they cover their two-hop neighborhood. That way broadcast traffic consumes less resources in order to be forwarded to all destinations. A node can either advertize the full neighbor link set, or the advertized link set can be limited to MPR links, *i.e.*, the neighbors that have elected this node as an MPR, while still guaranteeing that the shortest paths can be computed.

In a fully distributed spanning tree design, the tree computation is performed independently by each group member. To proceed to the correct computation of the overlay tree, multicast nodes need to know the membership of their multicast group. Therefore, when a node wants to join or leave a group, it broadcasts a Join or Leave message to the entire network, via the optimized MPR-flooding mechanism used in OLSR. Join messages are sent periodically according to the *Join\_Interval*, which is set by default to be equal to the TC message interval, *i.e.*, 5 seconds. The total protocol overhead is limited in these messages, independently of the number of groups and sources. In fact, MOST is well suited for managing numerous groups of small size, with arbitrary sources. Each group member must compute periodically the multicast tree to discover and maintain

its overlay neighbors. In order to compute the multicast tree, the node needs information about network topology which is delivered by OLSR. The overlay neighbor set is a subset of clients sharing the same tree with that member, which are linked to it via unicast tunnels. The distance between overlay neighbors can be of one or several hops. Each client receives/retransmits multicast data from/to its overlay neighbors. The computed tree is a group shared tree, hence it must always be the same for all the clients. However, because of changes in the network topology and group membership, there is no guarantee that all clients hold the same tree. Consequently, to avoid loops there is a need to maintain a duplicate table in each client node. Moreover, some redundancy is introduced in data forwarding after tree updates to avoid packet losses.

**Managing Join and Leave Messages.** The Join message contains the multicast group(s) address(es) that the sender has joined. The message format is the OLSR message format [6]. The Join message contains a list of multicast group addresses. Each multicast node maintains a membership table with the members of all the groups it belongs to. Upon receipt of a new Join message, each concerned node adds the new client to its membership table. The entries in the membership table are also associated with an expiration time, which is determined by the *VTime* field in the Join message header. After this time period the entries are removed, unless another Join message is received, in which case the expiration time is updated. When a node wants to leave a group, it broadcasts a Leave message to the entire network but keeps acting as a group member if it receives data for a predefined transition period. Beyond this period, received packets are not retransmitted. Leave messages follow the same format as Join messages, hence the message contains the group addresses that the originator wishes to leave. We note that in case there is a multicast node failure or disconnection from the network, this event will be accounted for in the OLSR topology table. Therefore, other multicast nodes will act accordingly, as if the problematic node had left all multicast groups, and the multicast communication will not be affected.

**Tree Computation and Maintenance.** For each group, each client computes periodically the corresponding overlay tree, according to its membership table. Therefore, the client needs to maintain a table with its overlay neighbors. The update period is set by default to the Hello interval, *i.e.*, 2 seconds, which has been found empirically to provide optimal performance. If a new node joins the group or a client leaves the group, the tree is updated immediately. Whenever the overlay neighbors change after an update, the client considers both new, as well as older overlay neighbors for a transition period of 1 second, that is half the update period. The transition period is introduced to make it possible for all the clients to take into account tree changes, and to improve the packet delivery ratio. After that period, only new neighbors are considered. Finally, in order to be able to perform the overlay tree construction according to the previously described algorithm, nodes switch to full-OLSR mode and start advertizing their complete neighbor set whenever they join a multicast group. The reason for this

is that it must be possible to compute the distance between each pair of group members.

**Transmission and Forwarding of Data Packets.** Unlike common multicast protocols where data packets are transmitted in a broadcast mode, in MOST data packets are encapsulated in unicast packets before being forwarded to the overlay tunnels. Unicast transmissions present important additional benefits when the subnetwork layer in use is IEEE 802.11 [3], as is the case in most actual wireless networks. Firstly, the packet delivery ratio is significantly improved, since packets are retransmitted in case of collisions, while this is not the case for broadcast (or multicast) packets. Secondly, in most 802.11 variants (including b and g), the broadcast frames are transmitted by default at a lower rate than unicast frames. For instance, when 802.11b operates at a unicast data rate of 11 Mbps, the default broadcast rate is 2 Mbps, and most wireless card drivers do not offer the possibility to change it. Therefore, unicast tunnels can actually increase the available bandwidth, since data is sent at a much higher rate. Under these MAC constraints, MOST outperforms protocols using broadcast frames.

When a client receives a multicast data packet<sup>1</sup>, it checks whether the packet has already been received. If this is the case, the duplicate packet is dropped. Otherwise, the client forwards the packet to each of its overlay neighbors, except the one from which the packet was received. Source nodes act also as group members, hence they simply send their data in unicast to their overlay neighbors.

## 4 Implementation Overview

In this section we outline our complete implementation of the MOST protocol for Linux. An overview of the architecture is depicted in Figure 1. The implementation consists of two modules: MDFP and OOLSR.

MDFP (Multicast Data Forwarding Protocol) is a forwarding protocol that enables point to multipoint data transfer. Multicast packets are captured and encapsulated in order to be forwarded inside a multicast tree. This module was developed for use with the MOLSR multicast protocol [1], and we adapted it to also support MOST. OOLSR (Object oriented OLSR) is INRIA's implementation of the OLSR protocol in *C++* [2]. The core of MOST was implemented as an extension inside this module.

The OOLSR module with MOST extension is in charge of sending and processing Join and Leave messages, as well as computing and maintaining the overlay multicast tree, based on the network topology. The MDFP module is in charge of the actual forwarding of multicast data packets in the overlay tree. For this purpose, it performs encapsulation and decapsulation of data packets, and maintains a table in order to detect duplicate packet receptions. As shown in Figure 1, the two modules constantly exchange information. The OOLSR daemon provides MDFP with up to date overlay neighbors information, which is

---

<sup>1</sup> In fact all packets are received in unicast, so we refer here to the encapsulated multicast content.

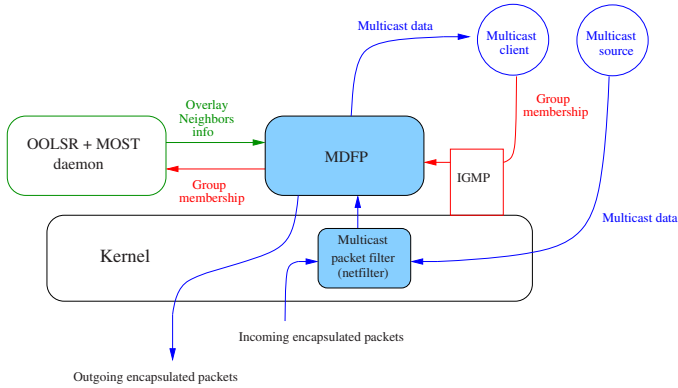


**Fig. 1.** Overview of multicast implementation

all that is needed to perform the transmission and forwarding of multicast data. Conversely, MDPF communicates to the OOLSR daemon the group membership information concerning the node's OLSR interfaces. In fact, multicast client applications update the interfaces' IGMP information (*cf.* Internet Group Management Protocol [7]), and this information is interpreted by MDPF. Incoming multicast data packets are captured by the *netfilter* module in the kernel, following predetermined rules (such as a predetermined UDP port number). MDPF decapsulates the packets and passes them to the client applications, while it re-encapsulates them in order to forward them to the overlay neighbors. Similarly, data transmitted by a local multicast source is also captured by netfilter and processed by MDPF. Finally, we note that the OOLSR module (including the MOST extension) can be loaded as a plugin in ns-2, hence the simulator shares the same source code as the real implementation.

## 5 Simulation Results

In this section we perform ns-2 simulations in various scenarios aiming to verify the theoretical minimum spanning tree analysis. We also present some protocol performance measures in a mobile ad hoc network environment. In Table 1, we summarize the parameters that are common for all our simulations.

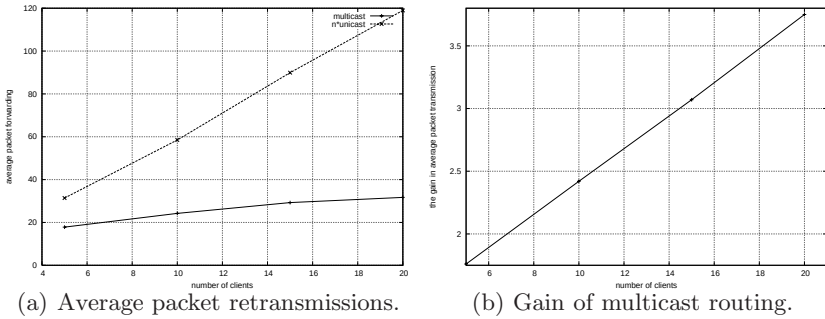
**Comparison of Multicast and Unicast Performance.** We measure the average multicast cost for various group sizes, by counting the total number of times each packet is relayed to reach all destinations, using MOST. The unicast cost is determined in the same manner, by repeating the same simulations and considering OLSR unicast transmissions between each source-client pair.

The simulation environment consists of a randomly generated topology of 100 wireless nodes forming an ad hoc network, in an area of  $1500m \times 1500m$ . We consider group sizes ranging from 5 to 20 nodes (not including the source). Multicast groups and sources are chosen at random. The source node sends to

**Table 1.** Common simulation parameters

|                      |                          |
|----------------------|--------------------------|
| MAC Protocol         | IEEE 802.11b (11Mb rate) |
| Propagation model    | Two ray ground           |
| Transmission range   | 250m                     |
| Packet size          | 1200 bytes               |
| Traffic type         | CBR                      |
| Number of iterations | 5                        |

the group CBR traffic of 64kbps, with packets of 1200 bytes, for 150 seconds of simulated time. To obtain reliable results, simulations are conducted several times with 5 different seeds. The mean results are depicted in Figure 2.


**Fig. 2.** Comparison of multicast versus unicast to all destinations

The analysis in [10] allows to compute an upper bound on the number of multicast packet retransmissions as a function of the number of group clients  $n$ . Namely, an upper bound for this parameter is:  $\sqrt{2n} \times d_u$ , where  $d_u$  is the average unicast distance between two nodes in hops. In Figure 3, we compare the average number of retransmissions measured through simulations to the theoretical bound (where border effects are ignored). Although the analysis is performed in an asymptotic setting, we notice that the upper bound is also valid in these simulations.

**Multicast Performance versus Throughput and Group Size.** Simulations are conducted to determine the maximum source rate that maintains an acceptable delivery ratio in a multicast group. By acceptable we mean that its value is higher than 95%. We consider static topologies again since the goal is to find the saturation point of the network. We consider a 200 wireless nodes network in a  $1800m \times 1800m$  area, with one multicast group. We vary the number of clients as well as the source bit rate and we measure the packet delivery ratio, as shown in Figure 4. We notice that the source node can transmit with a rate of up to 200kbps with a delivery ratio higher than 99%. From a 250kbps rate, the performance remains good for small groups but it decreases for large group sizes.

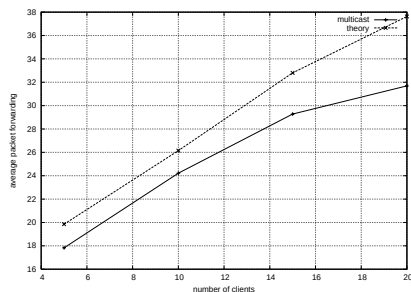


**Fig. 3.** Simulation results vs theoretical upper bound

We also run other simulations by fixing the number of clients to 10 and varying the number of active multicast groups in a 300 nodes network. In each group a source is transmitting CBR traffic with a rate of 64kbps. As shown in Figure 5 we notice that the delivery ratio is very high until network saturation, with 8 groups of 10 clients.

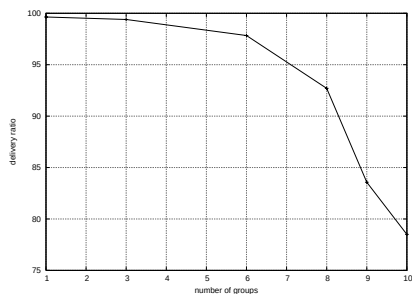


**Fig. 4.** Delivery ratio (%) vs number of groups in the network

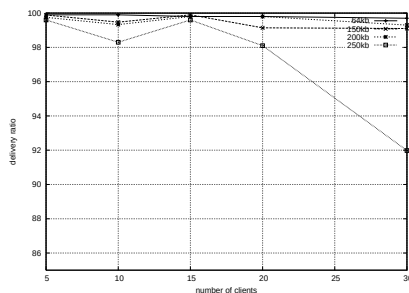


**Fig. 5.** Delivery ratio (%) vs group size for different source rates

**Multicast Performance with Mobility.** To evaluate the protocol performance with mobility, we consider a scenario consisting of a randomly generated topology of 200 wireless nodes in an area of  $1850m \times 1850m$ , and one multicast group in which an arbitrary source node sends a CBR traffic of 64kbps for 300 seconds. We run simulations in which we vary the group members, the number of clients (from 5 to 20 nodes, not including the source) and the maximum mobility speed (from  $1m/s$  to  $10m/s$ ). The mobility model is the Random Way-point model with a pause time of 10 seconds: nodes choose a random point in the network area and move to it with a constant speed chosen at random between  $1m/s$  and the maximum defined value; after they reach their destination, they remain idle for a period equal to the pause interval and then the same procedure is repeated. Moreover, we consider the following OLSR parameters: a Hello interval of 1 second and TC interval of 5 seconds; we note that the performance of MOST with mobility can be improved by using smaller intervals, at the cost



of higher OLSR control message overhead. The simulations are again repeated several times with 5 different seeds. We measure the multicast packet delivery ratio and the traffic load caused by duplicate packets, and we depict the obtained results in Figures 6 and 7.

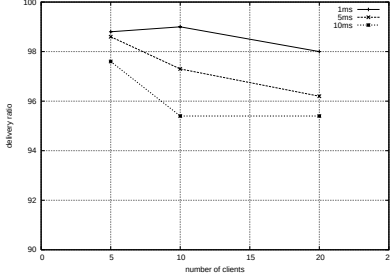


**Fig. 6.** Delivery ratio (%) vs group size with different mobility speeds

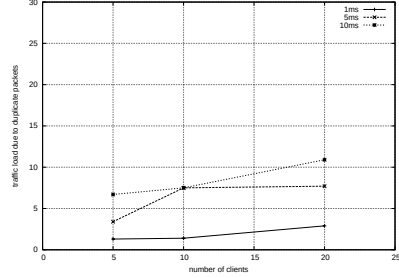


**Fig. 7.** Duplicate traffic load (%) vs group size with different speeds

As we can see, by varying the speed from  $1m/s$ ,  $5m/s$  and up to  $10m/s$ , the delivery ratio remains acceptable, *i.e.*, higher than 95 % for all groups of up to 20 clients. On the other hand, traffic load due to duplicate packets is higher when the mobility speed or the group size increase. This can be explained by the fact that any change in the topology due to mobility can affect the shared tree. In fact, each client is aware of these changes since it uses the OLSR protocol, and when it recalculates the overlay tree it enters a transition period, during which old and new overlay tree neighbors are maintained. A compromise can be found between the overhead due to duplicate packets and packet delivery ratio by setting a suitable transition period length. We notice that the duplicate traffic load in the network remains small compared to the total traffic load (10% in the worst case), hence the important advantage of improving the packet delivery ratio comes only with a modest performance cost.

## 6 Conclusion

In this paper we presented MOST, a Multicast Overlay Spanning Tree routing protocol which is an extension to OLSR ad hoc routing. The protocol is fully distributed in the sense that each group member computes and maintains the shared multicast tree independently. The MOST protocol being based on overlay trees guarantees robustness while it achieves good performance, with additional important advantages in the case of IEEE 802.11 networks. The protocol was tested through ns-2 simulations, which verify the corresponding analytical results stating that multicasting can reduce the overall network load by a factor  $O(\sqrt{n})$ , for  $n$  multicast group members. Furthermore, MOST has been tested in real network environments, hence we intend to provide measurement studies of the protocol performance in future work.

## References

1. MDFP, <http://hipercom.inria.fr/smolsr-molsr/>.
2. OOLSR, <http://hipercom.inria.fr/oolsr/>.
3. IEEE 802.11 Standard, Wireless LAN Medium Access Control and Physical layer Specifications, 1997.
4. C. Adjih, L. Georgiadis, P. Jacquet, and W. Szpankowski. Is the internet fractal: The multicast power law revisited. In *SODA*, 2002.
5. J. C.-I. Chuang and M. A. Sirbu. Pricing multicast communication: A cost-based approach. *Telecommunication Systems*, 17(3):281–297, 2001.
6. T. Clausen and P. Jacquet (editors). Optimized link state routing protocol (OLSR). RFC 3626, October 2003. Network Working Group.
7. W. Fenner. Internet Group Management Protocol, Version 2, RFC 2236, 1997.
8. J. Garcia-Luna-Aceves and E. L. Madruga. The core-assisted mesh protocol. *Journal on Selected Areas in Communications*, 17(8):1380–1294, August 1999.
9. P. Gupta and P. R. Kumar. Capacity of wireless networks. *IEEE Transactions on Information Theory*, IT-46(2):388–404, 2000.
10. P. Jacquet and G. Rodolakis. Multicast scaling properties in massively dense ad hoc networks. In *SaNSO, Fukuoka, Japan*, 2005.
11. J. Jetcheva and D. B. Johnson. Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks. In *Proceedings of the Second Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2001)*, 2001.
12. A. Laouiti, P. Jacquet, P. Minet, L. Viennot, T. Clausen, and C. Adjih. Multicast Optimized Link State Routing, INRIA research report RR-4721, 2003.
13. S. Lee, W. Su, and M. Gerla. On demand multicast routing protocol in multihop wireless mobile networks. *ACM/Baltzer Mobile Networks and Applications, special issue on Multipoint Communication in Wireless Mobile Networks*, 2000.
14. M. Liu, R. R. Talpade, A. McAuley, and E. Bommaiah. AMRoute: Adhoc Multicast Routing Protocol. *UMD TechReport 99-8*.
15. C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (AODV) routing, RFC 3561, 2003.
16. E. Royer and C. Perkins. Multicast Ad hoc On-Demand Distance Vector (MAODV) Routing, IETF, Internet Draft: draft-ietf-manet-maodv-00.txt, 2000.

# Detection of Packet Forwarding Misbehavior in Mobile Ad-Hoc Networks

Oscar F. Gonzalez, Michael Howarth, and George Pavlou

Center for Communications Systems Research, University of Surrey, Guildford, UK  
o.gonzalez-duque@surrey.ac.uk, m.howarth@surrey.ac.uk,  
g.pavlou@surrey.ac.uk

**Abstract.** Mobile Ad Hoc networks (MANETs) are susceptible to having their effective operation compromised by a variety of security attacks. Nodes may misbehave either because they are malicious and deliberately wish to disrupt the network, or because they are selfish and wish to conserve their own limited resources such as power, or for other reasons. In this paper, we present a mechanism that enables the detection of nodes that exhibit packet forwarding misbehavior. We present evaluation results that demonstrate the operation of our algorithm in mobile ad hoc environments and show that it effectively detects nodes that drop a significant fraction of packets.

**Keywords:** mobile ad hoc network, misbehavior detection, packet forwarding.

## 1 Introduction

The wireless nature and inherent features of mobile ad hoc networks makes them vulnerable to a wide variety of attacks by misbehaving nodes. Such attacks range from passive eavesdropping, where a node tries to obtain unauthorized access to data destined for another node, to active interference where malicious nodes hinder network performance by not obeying globally acceptable rules. For instance, a node can behave maliciously by not forwarding packets on behalf of other peer nodes. However, when a node exhibits malicious behavior it is not always because it intends to do so. A node may also misbehave because it is overloaded, broken, compromised or congested in addition to intentionally being selfish or malicious [3,11].

Misbehavior can be divided into two categories [3]: routing misbehavior (failure to behave in accordance with a routing protocol) and packet forwarding misbehavior (failure to correctly forward data packets in accordance with a data transfer protocol). In this paper we focus on the latter. Our approach consists of an algorithm that enables packet forwarding misbehavior detection through the *principle of conservation of flow* [25]. Our scheme is not tightly coupled to any specific routing protocol and, therefore, it can operate regardless of the routing strategy adopted. Our criterion for judging a node is the estimated percentage of packets dropped, which is compared against a pre-established misbehavior threshold. Any node dropping packets in excess of this threshold is deemed a misbehaving node while those below the threshold are considered to be correctly behaving.

Our scheme detects misbehaving nodes (whether selfish, malicious or otherwise) capable of launching two known attacks: the simplest of them is the black hole attack. In this attack a

misbehaving node drops all the packets that it receives instead of normally forwarding them. A variation on this is a gray hole attack, in which nodes either drop packets selectively (e.g. dropping all UDP packets while forwarding TCP packets) or drop packets in a statistical manner (e.g. dropping 50% of the packets or dropping them with a probabilistic distribution). Both types of gray hole attacks seek to disrupt the network without being detected by the security measures in place.

In this paper we first present a framework and a relevant algorithm and protocol that deal with this attack. We then demonstrate through simulations that an appropriate selection of the misbehavior threshold allows for a good discrimination between misbehaved and well-behaved nodes, as well as providing robustness against different degrees of node mobility in a network that is affected by black hole and/or gray hole attacks.

The rest of this paper is organized as follows. Section II describes related work in the area of MANET security. Section III specifies our assumptions on the network and security models and clarifies the terminology adopted. Section IV describes our algorithm for packet forwarding misbehavior detection, and Section V presents a performance evaluation. Finally, the paper is concluded in Section VI.

## 2 Related Work

In this Section we initially look at ways of protecting routing protocols against misbehaving nodes and then review work that attempts to detect misbehavior in data packet forwarding.

Some research effort has been focused on the development of new routing protocols whose objective is to protect the network from security threats that were not addressed by work preceding them. The Secure Routing Protocol (SRP) [7] and Authenticated Routing for Ad hoc Networks (ARAN) [8] assume the existence of *a priori* relationships in a network: in the case of SRP between the two communicating nodes, and for ARAN between each node in the network and a certificate server. Both protocols perform an end-to-end authentication and intermediate nodes are not allowed to reply to route requests even if they know a route to the destination. However, *a priori* relationships in MANETs may not exist. These approaches secure the path discovery and establishment functionality of routing protocols, and our approach complements them by securing the data forwarding functionality.

The routing protocol proposed in [16] offers resilience to disruption or degradation of the routing service by an algorithm that allows the detection of a malicious link after  $\log n$  faults have occurred on a path, where  $n$  is the hop length of the path. In [19] each node is able to detect signs of intrusion locally and neighboring nodes collaborate to further investigate malicious behavior. In both these approaches a node uses its own data to identify another node as an intruder. In contrast, in our approach a node detects anomalies in packet forwarding based on data acquired by other nodes in the network as well as on its own data, thus potentially obtaining a more balanced evaluation of a node's behavior.

Secure routing protocols have also been proposed based on existing ad hoc routing protocols. These eliminate some of the optimizations introduced in the original routing protocols because they can be exploited to launch different types of attacks. Examples of such protocols are the secure efficient distance vector (SEAD) routing [6]

which is based on the destination sequenced distance vector (DSDV) [12], the secure ad-hoc on-demand distance vector (SAODV) routing protocol [9, 10] based on AODV [13, 14], and the secure on-demand routing protocol for ad hoc networks (Ariadne) [2] based on the dynamic source routing (DSR) protocol [15] and the timed efficient stream loss-tolerant authentication (TESLA) protocol proposed in [17]. A second work extending DSR to provide it with security mechanisms is CONFIDANT (Cooperation Of Nodes: Fairness In Dynamic Ad-hoc NeTworks) [18]. As with SRP and ARAN these protocols can be coupled with our approach, which is not routing protocol dependent, to offer an improved security solution.

Attack patterns have also been the object of research. In [4] and [5] the authors propose a framework for misuse detection which divides the nodes in a network into two categories: insiders and outsiders. Insiders are always well-behaved nodes in charge of performing network tasks. Such nodes belong to trusted users. Outsiders only communicate using the network and are always invigilated by a subset. Unfortunately, this framework assumes the existence of trust relationships prior to a MANET formation, which goes against their dynamic and spontaneous nature. In this regard, our protocol does not assume prior relationships between nodes in a network and it only evaluates their behavior after the MANET has been formed and the packet transmission has commenced.

Other work has tried to protect security mechanisms against attacks, such as [11] where the authors distinguish between two types of threats: threats to the basic mechanisms of ad hoc networks, such as routing, and threats to the security mechanisms such as key establishment and management. Their proposed algorithm uses certificates stored in repositories to find a certificate chain between the communicating nodes. Although simulations show an acceptable performance, this method requires a considerable amount of memory in every node in the network.

There has also been some work that aims to protect data packet forwarding against malicious attacks in order to provide reliable network connectivity. The final part of this section describes some approaches that detect malicious behavior in the data forwarding phase. WATCHERS (Watching for Anomalies in Transit Conservation: a Heuristic for Ensuring Router Security) [25] is a protocol designed to detect disruptive routers in fixed networks through analysis of the number of packets entering and exiting a router. Although WATCHERS is based on the principle of conservation of flow in a network in the same way as our proposed algorithm, its design focuses only on fixed networks and is not applicable to mobile ad hoc networks.

The Secure Message Transmission (SMT) and Secure Single Path (SSP) protocols are both introduced in [20]. In SMT a message that is to be sent towards a destination is first divided in  $N$  parts and then sent by  $N$  independent paths. Each part carries a limited amount of redundancy in such a way that only  $M$  parts, where  $M < N$ , are needed at the destination to recover the whole message. In SSP one path is used at a time and the source tries a different path each time the communication is not successful. However, SMT is very bandwidth-intensive, and these protocols do not attempt to find the source of the packet loss. Our protocol, on the contrary, identifies any source(s) that appear to be causing packet losses, allowing for their isolation at a later stage if necessary.

In [24] the authors look at traffic transmission patterns between any two communicating nodes in order to facilitate verification by a receiver. Such traffic patterns can

be analyzed, and a misbehaving node can be found, if they are used in concert with suboptimal techniques at the medium access control (MAC) layer that preserve the statistical regularity from hop to hop. This work, however, has a very narrow scope of application due to its MAC layer assumptions to preserve statistical regularity, and thus it is very unlikely for it to be useful in scenarios other than military applications.

SCAN (self-organized network layer security in mobile ad hoc networks) [3] focuses on securing packet delivery. SCAN nodes monitor their neighbors by listening to packets that are forwarded to them. The SCAN node maintains a copy of the neighbor's routing table and determines the next-hop node to which the neighbor should forward the packet; if the packet is not overheard as being forwarded, it is considered to have been dropped. In contrast, in our algorithm nodes do not need to overhear transmissions to and from any neighbor in order to detect misbehavior. In SCAN each node must possess a valid token to be able to interact with the network. SCAN develops this idea from [21] where tokens are used to give a valid key to a new node entering the network. Similar techniques have also been studied in various papers such as [22], where they are used to achieve distribution of trust throughout a network.

Finally, in [23] a system that can mitigate the effects of packet dropping is proposed. This is composed of two mechanisms that are kept in all network nodes: a watchdog and a pathrater. The watchdog mechanism identifies any misbehaving nodes by promiscuously listening to the next node in the packet's path. If such a node drops more than a predefined threshold of packets the source of the communication is notified. The pathrater mechanism keeps a rate for every other node in the network it knows about. A node's rate is decreased each time a notification of its misbehavior is received. Then, nodes' rates are used to determine the most reliable path towards a destination. However, the watchdog might not detect a misbehaving node in the presence of ambiguous collisions, receiver collisions or nodes capable of controlling their transmission power. Such weaknesses are the result of using promiscuous listening to determine whether a node has forwarded a packet or not. Our approach does not have these same weaknesses since it is based on metrics obtained from nodes that are actually sending and receiving packets to and from the node whose behavior is under evaluation, as explained in section IV.A.

### 3 Model Assumptions

The physical layer of a wireless network is often vulnerable to denial of service (DoS) attacks such as frequency jamming. Spread spectrum and frequency hopping are examples of techniques that have been studied as means of preventing this type of attacks. The link layer is also subject to attacks where nodes gain unfair access to the medium or where they disrupt communications, for example by dropping packets related to typical handshake processes. We disregard attacks aimed at the physical and link layers.

We assume bidirectional communication symmetry in every direct link between a pair of nodes. This means that if a node  $v_2$  receives a packet from node  $v_1$ ,  $v_1$  can also receive a packet from  $v_2$ . This is a sensible assumption since our approach needs MAC protocols with collision avoidance mechanisms to work properly, such as the extensively deployed IEEE 802.11, MACA (Multiple Access with Collision Avoidance)

[26] and MACAW (MACA for Wireless LANs) [1], which require bidirectional communication for reliable transmission.

We assume the MAC layer protocol to be reliable (e.g. IEEE 802.11). This is required to provide confidence that a data packet has been successfully transmitted to the next-hop node, and enables us to apply the principle of flow conservation (see Section IV.A).

## 4 Detecting Packet Forwarding Misbehavior

Our work provides a novel methodology to secure the data forwarding functionality in mobile ad hoc networks. We propose an approach that takes advantage of the principle of flow conservation in a network. This states that all bytes/packets sent to a node, and not destined for that node, are expected to exit the node. In this Section we first present, from a theoretical point of view, how this principle works assuming it is implemented in an ideal network, and then we demonstrate that by making some reasonable assumptions and adaptations, our algorithm can cope with the practical problems that are encountered in real MANETs.

### 4.1 The Principle of Flow Conservation

We now formally introduce the principle of flow conservation over an ideal static network model:

- Let  $v_j$  be a node such that  $v_j \in V$ , where  $V = \{v_1, v_2, v_3 \dots v_N\}$  is the set of all nodes in the network,  $N$  is the total number of nodes in the network, and  $j = 1, 2, 3 \dots N$ .
- Let  $U_j$  be the subset of nodes in the network which are neighbors of  $v_j$ , i.e.  $U_j$  is the neighborhood of  $v_j$ . It follows that  $v_j \notin U_j$  and also  $U_j \subset V$ .
- Let  $\Delta t$  be the period of time elapsed between two points in time  $t_0$  and  $t_1$  such that  $\Delta t = t_1 - t_0$ .
- Let  $T_{ij}$  be the number of packets that node  $v_i$  has successfully sent to node  $v_j$  for  $v_j$  to forward to a further node;  $v_i \in U_j$ ,  $v_j \in U_i$ ,  $i \neq j$  and  $T_{ij}(t_0) = 0$ .
- Let  $R_{ij}$  be the number of packets that node  $v_i$  has successfully received from node  $v_j$  that did not originate at  $v_j$ ;  $v_i \in U_j$ ,  $v_j \in U_i$ ,  $i \neq j$  and  $R_{ij}(t_0) = 0$ .

If all nodes  $v_j \in V$  remain static for a period of time  $\Delta t$  during which no collisions occur in any of the transmissions over an ideal (noiseless) wireless channel, then for a node  $v_j$ :

$$\sum_{\forall i|v_i \in U_j} R_{ij}(t_1) = \sum_{\forall i|v_i \in U_j} T_{ij}(t_1) \quad (1)$$

Equation (1) states the fundamental premise of the flow conservation principle in an ideal static network applied to packets rather than raw bytes. It states that if all neighbors of a node  $v_j$  are queried for i) the amount of packets sent to  $v_j$  to forward and ii) the amount of packets forwarded by  $v_j$  to them, the total amount of packets sent to and received from  $v_j$  must be equal.

In practice networks exhibit conditions that are far from ideal. First of all, the wireless channel is error prone and packets get lost while in transit. Secondly, collisions happen when the network uses protocols where nodes have to compete for the medium, such as when the link layer protocol is based on the distributed coordination function (contention period) of the IEEE 802.11 a/b standard. In order to allow equation (1) to hold we propose to use a reliable MAC protocol such as IEEE 802.11, MACA or MACAW.

The use of a reliable MAC protocol in conjunction with the conservation of flow principle means that we are not susceptible to problems that arise when overhearing other nodes' transmissions. Thus, problems such as ambiguous collisions, receiver collisions, and the ability of a node to control its transmission power do not exist in our approach. *Ambiguous collisions* occur when a node  $v_1$  is trying to determine if another node  $v_2$  is properly forwarding a packet. It may happen that node  $v_2$  forwards the packet to a further node  $v_3$ , which is out of the transmission range of  $v_1$ , while a second transmission prevents  $v_1$  from overhearing the forwarded packet, thus  $v_1$  will not know if the packet was forwarded. On the other hand, in the *receiver collision* problem  $v_2$  forwards the packet to  $v_3$  at which point a collision occurs. Node  $v_1$  is unaware of such a collision and assumes that the packet was forwarded even if  $v_2$  does not attempt a retransmission. Another problem is caused by nodes capable of controlling their transmission power. Thus,  $v_2$  can transmit with enough power for  $v_1$  to overhear but not enough power for  $v_3$  to receive it, leaving  $v_2$  unaware of the situation. All these weaknesses are due to the fact that overhearing is used by nodes to check for misbehavior. In our algorithm the nodes that maintain statistics that are used to determine whether the forwarding was properly made are the nodes actively involved in the transmission process, i.e. the transmitter and the receiver of each transmission.

However, a node may exhibit malicious behavior even if it is not purposefully doing so. For example, an overloaded node may lack the CPU cycles, buffer space or bandwidth to forward packets. For these reasons equation (1) cannot be applied in a rigorous manner and a threshold needs to be established to account for packets dropped by a node through no fault of its own. Equation (2), which holds for well behaved nodes, reflects this change.

The  $\alpha_{threshold}$  factor can take values between 0 and 1 and as we shall see plays an important role in the detection power of our proposed algorithm, i.e. the capability of the algorithm to detect misbehaving nodes. The lower  $\alpha_{threshold}$  is the more likely it is that our algorithm detects any malicious behavior. However, it also means that the probability of a false detection increases, as it can be inferred from our simulations (Section V). A false detection occurs when the result of a single evaluation of a node mistakenly determines that the node appears to be misbehaving. Therefore, fine tuning is required to reach a fair point in this tradeoff.

$$(1 - \alpha_{threshold}) \sum_{\forall i|v_i \in U_j} R_{ij}(t_1) \leq \sum_{\forall i|v_i \in U_j} T_{ij}(t_1) \quad (2)$$

## 4.2 Adapting Conservation of Flow to Mobile Networks

In MANETs the neighborhood  $U_j$  of a node  $v_j$  changes dynamically over time, making it difficult to determine those nodes that have transmitted or received packets to or from a



node  $v_j$ . Our scheme overcomes this problem by means of a limited broadcast that tracks down nodes that have been in contact with node  $v_j$  as explained in this section.

The core parts of our algorithm are detailed in the pseudocode shown in figure 1. A node  $v_i$  maintains a table with two metrics  $T_{ij}$  and  $R_{ij}$ , which contains an entry for each node  $v_j$  to which  $v_i$  has respectively transmitted packets to or received packets from. Node  $v_i$  increments  $T_{ij}$  on successful transmission of a packet to  $v_j$  for  $v_j$  to forward to another node, and increments  $R_{ij}$  on successful receipt of a packet forwarded by  $v_j$  that did not originate at  $v_i$ .

All nodes in the network continuously monitor their neighbors and update the list of those they have heard recently (Fig. 1.a). If the ID of an overheard node is not included in the table of overheard nodes a new entry is created. Otherwise, the existing entry is updated with a timestamp corresponding to the time the node was last overheard. Upon the creation of a new entry, a node schedules a task/event to check the behavior of the node whose ID has been saved in the new entry. Nodes randomly select a period of time between  $T_{\min}$  and  $T_{\max}$  to schedule the behavior checking task.

#### a. OVERHEARING

```

if node  $v_k$  overhears a node  $v_i \in U_k$ 
. if node  $v_i$  is not in  $v_k$ 's table of overheard nodes
. . add node  $v_i$  to  $v_k$ 's table of overheard nodes
. . schedule an event to check  $v_i$ 's behavior
. else
. . update last time node  $v_i$  was heard
. endif
endif

```

#### c. REQUEST HANDLING

```

if node  $v_k$  receives a metrics request for node  $v_i$ 
. if node  $v_k$  has node  $v_i$  in its table of overheard nodes
. . rebroadcast metrics request packet (MREQ)
. . reschedule any event to check  $v_i$ 's behavior
. . if node  $v_k$  has metrics for node  $v_i$ 
. . . send a metrics reply (MREP) to requesting node
. . endif
. else
. . ignore request
. endif
endif

```

#### b. INITIATE BEHAVIOR CHECK

```

if in node  $v_k$  an event to check node  $v_i$ 's behavior is triggered
. send a metrics request (MREQ) with node  $v_i$ 's ID
. schedule event to check  $v_i$ 's behavior at  $t+T_{\max}$ 
endif

```

#### d. REPLY HANDLING

```

if a request was sent out
. while there are more replies to be received for node  $v_i$ 
. . receive reply
. . acknowledge reply reception (send MACK)
. . add received metrics to totals
. endwhile
.
. if  $(1 - \alpha_{\text{threshold}}) \sum_{v_i \in U_j} R_{ij} > \sum_{v_i \in U_j} T_{ij}$ 
. .
. . node  $v_i$  is misbehaving (detection)
. else
. . node  $v_i$  is not misbehaving (non-detection)
. endif
endif

```

Fig. 1. Our algorithm pseudocode

When a scheduled task is triggered in node  $v_k$  to check  $v_j$ 's behavior (Fig. 1.b), node  $v_k$  broadcasts a *metrics request packet (MREQ)* with TTL = 1 in the IP header. An MREQ includes the ID of the node emitting the request (SRC\_ID), the ID of the node whose behavior is to be checked (CHK\_ID), an MREQ\_ID and a timestamp indicating the time at which the task was triggered. If a node sees an MREQ that has the same MREQ\_ID and SRC\_ID of a packet seen before, the MREQ is dropped. This technique prevents flooding packets from traversing a zone of the network more than once. The timestamp, on the other hand, is used to resolve conflicts when two nodes start a behavior check on the same node at almost the same time. In such cases,

nodes can see which of the packets corresponds to the earlier triggered task and disregard the other. This does not require accurate synchronization of the nodes' clocks; approximate synchronization is all that need be assumed. Finally, after the MREQ packet is broadcast, a task is scheduled to be triggered a period of time  $T_{\max}$  (maximum elapsed period of time without checking an active node's behavior) later. This means it is highly unlikely that the same node will originate two successive checks of another node, and gives other nodes a chance to perform the behavior check.

The handling of requests (Fig. 1.c) is the heart of our limited broadcast algorithm. When a node receives an MREQ it first checks if the `CHK_ID` is in its table of overheard nodes; if it is not the node ignores the MREQ and discards the check. However, if the `CHK_ID` appears in its table then it rebroadcasts the MREQ with `TTL = 1` in the IP header. Thus, every MREQ travels only one hop at a time, and is then analyzed and rebroadcast if the protocol so determines. By following this algorithm, our protocol is capable of tracking down nodes that have been in contact with the checked node, as illustrated in Figure 2. In the Figure, node  $v_7$  is first in position  $a$  where it can be overheard by nodes  $v_1, v_2, v_3, v_6, v_8, v_{11}, v_{12}$  and  $v_{13}$ . Each of these nodes makes an entry in their table of overheard nodes when  $v_7$  first transmits and each of them schedules a task to check its behavior. At some point in time,  $v_7$  decides to move following the path depicted in Figure 2 coming in contact with nodes  $v_{14}, v_{17}, v_{18}, v_{19}, v_{20}, v_{23}, v_{24}$  and  $v_{25}$ . It finally stops in position  $b$ . In the Figure the scheduled behavior check initiation task (Fig. 1.b) in  $v_8$  is the first to be triggered and the limited broadcast commences. All nodes that have overheard node  $v_7$  re-broadcast the MREQ, whereas nodes such as  $v_4, v_9$  and  $v_{15}$  also receive the MREQ but ignore it because they have not overheard node  $v_7$ .

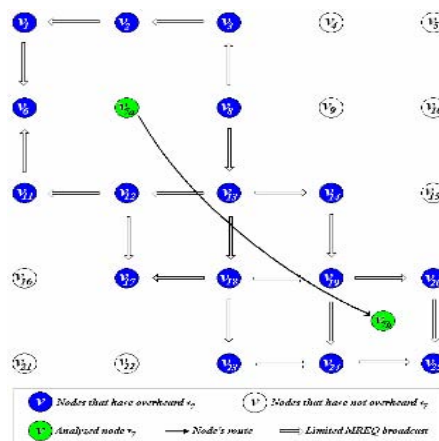


**Fig. 2.** Example of limited broadcast to track down nodes that have overheard node  $v_7$

Once a node has decided whether to continue or not broadcasting a MREQ, it re-schedules any pending task to check the behavior of the checked node specified in the `CHK_ID` field of the MREQ. The new behavior checking task is scheduled in the same way as when a new entry is made in the table of overheard nodes, i.e. a period

of time is randomly selected between  $T_{\min}$  and  $T_{\max}$ . In this way if the random selection is uniformly distributed the average frequency with which an active node's behavior is checked is:

$$avg\_freq = \frac{1}{(T_{\min} + T_{\max})/2} = \frac{2}{(T_{\min} + T_{\max})} \quad (3)$$

The last task a node performs when it receives a MREQ is to check if it has any metrics ( $R_{ij}$  or  $T_{ij}$ ) relating to the node being checked. If any of the metrics has a value other than zero the node returns a metrics reply packet (MREP) (Fig. 1.c) containing the metrics, but if the value of both metrics is zero then the node does not send back any response. In our scheme a metrics reply packet is returned to the node that originated the MREQ following the reverse of the MREQ's path. This approach avoids the high overhead produced by reactive routing protocols when they perform route discovery for many MREP packets.

Reply handling is executed in the node that initiated the MREQ. This node,  $v_g$  in Fig. 2, waits for a period of time in order to give all nodes with metrics about the checked node the opportunity of replying. When the time expires, the node checks the behavior of the analyzed node by verifying that equation (2) holds (Fig. 1.d). If it does not, it flags the checked node as a misbehaving one; this is a detection. Using a single detection to accuse a node is not sufficient since such an algorithm may lead to false accusations against correctly behaving nodes. A scheme in which multiple detections by different nodes are necessary to accuse a node is fairer to well-behaved nodes, while keeping a high probability of correctly accusing misbehaving nodes. Thus, a system whose goal is to accuse misbehaving nodes could use an approach similar to the distributed consensus mechanism proposed in SCAN [3].

## 5 Evaluation

We perform our simulations using the GloMoSim simulation package. The results presented for each value are the average of 10 simulation runs. Unless explicitly stated otherwise, our simulation parameters take the following values: i) nodes move according to the random waypoint mobility model with a mean node speed of  $5\text{ms}^{-1}$  and a standard deviation of  $2.89\text{ms}^{-1}$ , ii) the mean pause time is 30 seconds, iii) the wireless transmission range is 100 meters, iv) the link capacity is 2 Mbps, v) the MAC layer protocol is the IEEE 802.11 DCF, vi) the underlying routing protocol is AODV, and vii) the simulation time is 1200 seconds. The network was set-up with 40% of nodes misbehaving. Nodes check the behavior of active nodes within a period chosen uniformly between 40 and 60 seconds, and keep any overheard node in their tables for 120 seconds. The principal metrics in our tests are the percentage of detections and the network traffic overhead. These metrics are assessed in terms of misbehavior threshold and node speed.

We initially consider our misbehavior detection algorithm in terms of the misbehavior threshold, which is the parameter  $\alpha_{\text{threshold}}$  in equation (2), i.e. the maximum percentage of packets that a node is allowed to drop without being detected as a misbehaving node. In order to see the effect of the misbehavior threshold on nodes, simulations were carried out with networks containing 20 and 60 nodes, and areas of

40 000m<sup>2</sup> (200m\*200m) and 120 000m<sup>2</sup> (346.41m\*346.41m) respectively. We varied both the packet drop probability of misbehaving nodes and the misbehavior threshold between 0% and 100%.

Figures 3 and 4 show the percentage of detections as a function of the misbehavior threshold for nodes exhibiting different probabilities of misbehavior for networks with 20 and 60 nodes respectively. The lower the threshold is the more packets nodes need to forward to be considered well-behaved. However, since characteristics inherent to MANETs such as mobility and the noisy wireless medium can cause some packets to be lost (including packets of our own protocol), it also means that an increasing number of correctly behaving nodes can be falsely detected as misbehaving ones. Finally, it can also be seen from Fig. 3 and Fig. 4 that selecting a misbehavior threshold equal to a node's misbehaving probability prevents our approach from identifying misbehaving nodes with certainty, i.e. the probability of detection is approximately 50%. These occurrences are all contained in the zone between 40% and 60% probability of detection in the figures.

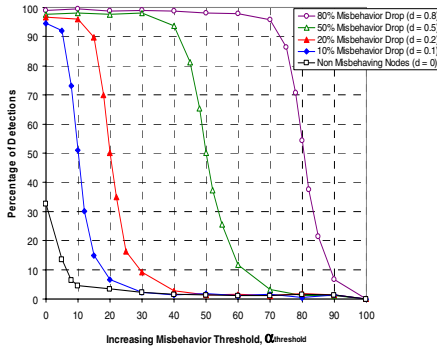


**Fig. 3.** Percentage of positive detections as a function of the increasing misbehavior threshold (20 node network)

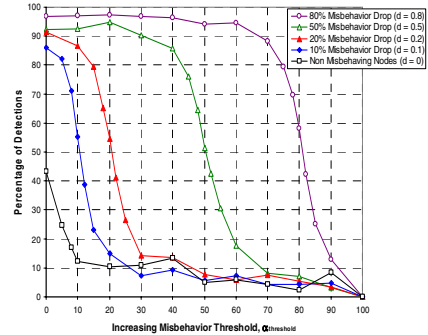


**Fig. 4.** Percentage of positive detections as a function of the increasing misbehavior threshold (60 node network)

Selecting an acceptable or tolerable level of misbehavior  $x$  in a network is a policy decision. This policy then allows a value of  $\alpha_{\text{threshold}}$  to be set depending on the desired detection probability. For example, for a detection probability of  $>90\%$  our results suggest that  $\alpha_{\text{threshold}}$  should then be set at approximately  $x-0.1$  for the 20 node network and  $x-0.15$  for the 60 node network.

The final set of results assesses the network overhead generated by our misbehavior detection algorithm. In this set of simulations misbehaving nodes drop packets with a 50% probability ( $d=0.5$ ), the misbehavior threshold  $\alpha_{\text{threshold}}$  is 40%, the node speed varies between 0ms<sup>-1</sup> (a static network) and 20ms<sup>-1</sup> giving a standard deviation of 0.58ms<sup>-1</sup>. Fig. 5 displays the mean number of packets sent over each link per behavior check. The network resources are calculated by adding one each time a packet crosses a different link: thus a MREQ packet broadcast that traverses three hops (links) contributes three packet-links to the total. Results are displayed for networks

containing 20, 40 and 60 nodes, which are distributed over areas of 40 000m<sup>2</sup> (200m\*200m), 80 000m<sup>2</sup> (282.84m\*282.84m), and 120 000m<sup>2</sup> (346.41m\*346.41m) respectively.

The network overhead per behavior check shown in Fig. 5 is the sum of the overhead produced by the MREQ, MREP and MACK packets per behavior check. It is least when the nodes are stationary and increases with the mean node speed. Whereas this increment is moderate for the 20 and 40 node networks, it is very significant for the 60 node network. This can be explained as follows. First of all, the bigger the network is and the faster a node moves the more nodes it comes in contact with. This means that it is more probable that a MREQ is rebroadcast by a greater number of nodes. On top of this, in a dynamic large network it is more likely that a node that rebroadcasts a MREQ moves out of the wireless range of nodes that need to send back a MREP through it towards the node that originated the MREQ. This causes MREP retransmissions which introduces more overhead into the network.

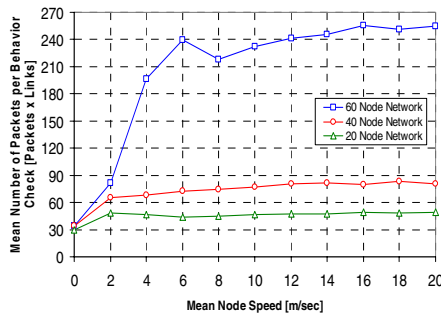


Fig. 5. Network overhead as a function of the increasing mean node speed

## 6 Conclusions

The self-regulating nature of MANETs requires that they be able to monitor the behavior of the network. Limited resources mean that there is an incentive for nodes to misbehave by not correctly forwarding packets (selfish nodes); nodes may also misbehave for other reasons.

In this paper we have presented an algorithm that is capable of detecting misbehavior. The algorithm does not require high density networks in which many nodes can overhear each others' received and transmitted packets, but instead uses statistics accumulated by each node as it transmits to and receives data from its neighbors.

We have shown that we can detect nodes that misbehave by dropping a significant percentage of packets. Detection is successful in spite of inherent packet losses in MANETs caused by noisy links, mobility, and packet losses due to routing protocol behavior. To avoid falsely accusing correctly behaved nodes of misbehavior, a collaborative consensus mechanism such as that described in [3] can be used. This will be considered in future work.

## References

- [1] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LANs," Proceedings of the ACM SIGCOMM Conference on Communications Architectures, Protocols and Applications, vol. 24, issue 4, pp. 212-225, 1994.
- [2] Y. C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks," Proceedings of the 8<sup>th</sup> ACM International Conference on Mobile Computing and Networking, pp. 12-23, September 2002.
- [3] H. Yang, J. Shu, X. Meng, and S. Lu, "SCAN: Self-organized network-layer security in mobile ad hoc networks," IEEE Journal on Selected Areas in Communications, vol. 24, issue 2, pp. 261-273, February 2006.
- [4] D. Subhadrabandhu, S. Sarkar, and F. Anjum, "A framework for misuse detection in ad hoc networks – part I," IEEE Journal on Selected Areas in Communications, vol. 24, pp. 274-289, February 2006.
- [5] D. Subhadrabandhu, S. Sarkar, and F. Anjum, "A framework for misuse detection in ad hoc networks – part II," IEEE Journal on Selected Areas in Communications, vol. 24, , pp. 290-304, February 2006.
- [6] Y. C. Hu, D. B. Johnson, and A. Perrig, "SEAD: secure efficient distance vector routing for mobile wireless ad hoc networks," Proceedings of the 4<sup>th</sup> IEEE workshop on Mobile Computing Systems & Applications, pp. 3-13, June 2002.
- [7] P. Papadimitratos, and Z. J. Haas, "Secure routing for mobile ad hoc networks," Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference, pp. 193-204, January 2002.
- [8] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer, "A secure routing protocol for ad hoc networks," Proceedings of the 10<sup>th</sup> IEEE International Conference on Network Protocols, pp. 78-87, November 2002.
- [9] M. Guerrero-Zapata, and N. Asokan, "Securing ad hoc routing protocols," Proceedings of the 3<sup>rd</sup> ACM Workshop on Wireless Security, pp. 1-10, 2002.
- [10] M. Guerrero-Zapata, "Secure ad hoc on-demand distance vector (SAODV) routing," Internet Draft, IETF Mobile Ad Hoc Networking Working Group, February 2005, *draft-guerrero-manet-saodv-05.txt*
- [11] J. P. Hubaux, L. Buttyán, and S. Čapkun, "The quest for security in mobile ad hoc networks," Proceedings of the 2<sup>nd</sup> ACM International Symposium on Mobile Ad Hoc Networking & Computing, pp. 146-155, 2001.
- [12] C. E. Perkins, and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," Proceedings of the ACM SIGCOMM Conference on Communications Architectures, Protocols and Applications, vol. 24, issue 4, pp. 234-244, 1994.
- [13] C. E. Perkins, and E. M. Royer, "Ad-hoc on-demand distance vector routing," Proceedings of the 2<sup>nd</sup> IEEE Workshop on Mobile Computer Systems & Applications, pp. 90-100, 1999.
- [14] C. E. Perkins, "Ad hoc on-demand distance vector (AODV) routing," Request For Comments (RFC) 3561, July 2003, available at: <http://www.ietf.org/rfc/rfc3561.txt>
- [15] D. B. Johnson, D. A. Maltz, and Y. C. Hu, "The dynamic source routing protocol for mobile ad hoc networks," Internet Draft, IETF MANET Working Group, July 2004, *draft-ietf-manet-dsr-10.txt*

- [16] B. Awerbuch, D. Holmes, C. Nita-Rotaru, and H. Rubens, "An on-demand secure routing protocol resilient to Byzantine failures," *Proceedings of the 3<sup>rd</sup> ACM Workshop on Wireless Security*, pp. 21-30, 2002.
- [17] Perrig, R. Canetti, J. D. Tygar, and D. Song, "Efficient authentication and signing of multicast streams over lossy channels," *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 56-73, May 2000.
- [18] S. Buchegger, and J. Le Boudec, "Performance analysis of the CONFIDANT protocol," *Proceedings of the 3<sup>rd</sup> ACM Symposium on Mobile Ad Hoc Networking & Computing*, pp. 226-236, 2002.
- [19] Y. Zhang, and W. Lee, "Intrusion detection in wireless ad-hoc networks," *Proceedings of the 6<sup>th</sup> ACM International Conference on Mobile Computing and Networking*, pp. 275-283, August 2000.
- [20] P. Papadimitratos, and Z. Haas, "Secure data communication in mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, issue 2, pp. 343-356, February 2006.
- [21] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, "Providing robust and ubiquitous security support for mobile ad-hoc networks," *Proceedings of the 9<sup>th</sup> IEEE International Conference on Network Protocols*, pp. 251-260, November 2001.
- [22] L. Zhou, and Z. Haas, "Securing ad hoc networks," *IEEE Network Magazine*, vol. 13, issue 6, November/December 1999.
- [23] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile ad hoc networks," *Proceedings of the 6<sup>th</sup> ACM International Conference on Mobile Computing and Networking*, pp. 255-265, August 2000.
- [24] R. Rao, and G. Kesidis, "Detecting malicious packet dropping using statistically regular traffic patterns in multihop wireless networks that are not bandwidth limited," *Proceedings of the 2003 IEEE Global Telecommunications Conference*, vol.5, pp. 2957-2961, 2003.
- [25] K. A. Bradley, S. Cheung, N. Puketza, B. Mukherjee, and R. A. Olsson, "Detecting disruptive routers: a distributed network monitoring approach," *Proceedings of the 1998 Symposium on Security and Privacy*, pp. 115-124, May 1998.
- [26] P. Karn, "MACA – a new channel access method for packet radio," *ARRL/CRRL Amateur Radio 9<sup>th</sup> Computer Networking Conference*, pp 134-240, September 1990.

# Reliable Geographical Multicast Routing in Vehicular Ad-Hoc Networks

Maria Kihl<sup>1</sup>, Mihail Sichitiu<sup>2</sup>, Ted Ekeroth<sup>1</sup>, and Michael Rozenberg<sup>1</sup>

<sup>1</sup> Dep. of Communication Systems, Lund University, Sweden

<sup>2</sup>Dep. of Electrical and Computer Engineering, North Carolina State University, USA

**Abstract.** Vehicular ad-hoc networks (VANETs) offer a large number of new potential applications without relying on significant infrastructure. Many of these applications benefit from multi-hop relaying of information, thus requiring a routing protocol. Characteristics unique to VANETs (such as high mobility and the need for geographical addressing) make many conventional ad hoc routing protocols unsuitable. Also, some envisioned applications have end-to-end QoS requirements. In this paper we propose a new multicast routing protocol specifically designed for VANETs. Its purpose is to provide a routing service for a future reliable transport protocol. We evaluate its performance using realistic network and traffic models. It is shown that it is possible to implement a reliable multicast routing protocol for VANETs.

## 1 Introduction

For many years research projects have been focused on issues regarding inter-vehicle communication (IVC) systems [1][2][3]. The objective of those projects has been to create the “fully connected vehicle”. By letting vehicles communicate both with each other and with base stations along the road, accidents can be avoided and traffic information can be made available to the driver. Of course, ultimately, the vision is to have in-vehicle Internet access as well. A couple of years ago the term VANET (Vehicular Ad-hoc Network) was introduced, combining mobile ad-hoc networks (MANETs) and IVC systems.

Vehicular Ad-hoc Networks (VANETs) are envisioned to both decrease the number of deaths in traffic and improving the travel comfort by, for example, increasing inter-vehicle coordination. Understandably, the most commonly considered applications are related to public safety and traffic coordination. Collision warning systems and vehicle platooning are two applications that projects work on. Also, traffic management applications, traveller information support and various comfort applications have the potential to make travel (considerably) more efficient, convenient and pleasant.

Most VANET applications require that data is transmitted in a multi-hop fashion, thus prompting the need for a routing protocol. In many aspects, a VANET can be regarded as a MANET. However, the inherent nature of a VANET imposes the following three constraints for a routing protocol:

1. Short-lived links.
2. Lack of global network configuration.
3. Lack of knowledge about a node's neighbors.



The first issue is due to the mobility of the vehicles. Studies have shown that the lifetime of a link between two nodes in a VANET is in the range of seconds [4]. Similar to a MANET, no central coordinator can be assumed in a VANET. Finally, although a hello protocol (as in OSPF) can be used to discover the neighbors of a node, this may be an expensive and difficult to tune solution. The routing protocol should discover the neighbors as needed. It is also preferable that the routing protocol works for a wide range of applications and traffic scenarios. Several papers propose solutions for specific VANET applications [5][6][7]. Some VANET applications require unicast routing. For example, some envisioned comfort applications, as on-board games and file transfer, will likely need unicast routing with fixed addresses. Many papers have proposed unicast protocols for VANETs. Some papers suggest that VANETs should use already existing unicast protocols for MANETs, as AODV [8][9] or cluster-based protocols [10][11]. Other papers propose new unicast protocols for VANETs [12][13]. However, many VANET applications require position-based multicasting (e.g., for disseminating traffic information to vehicles approaching the current position of the source). A natural match for this type of routing are the geocasting protocols

[6][14] that forward messages to all nodes within a Zone of Relevance (ZOR). The geocast concept has been studied for VANETs since the beginning of 1990s [15]. In [16] a geocasting protocol for VANETs was described; in this approach a node forwards a message after a delay that depends on the distance from the last sender. Variants of this protocol have been proposed in [17][18].

The major problem with flooding-based geocasting protocols is that the flooding mechanism is commonly based on broadcast, and it is, thus, best effort. However, some applications will require multicast transmission with end-to-end QoS. Flooding-based geocast protocols are not intended for these types of applications. Therefore, there is a need to develop multicast protocols for VANETs that can support end-to-end QoS mechanisms implemented in a transport layer protocol.

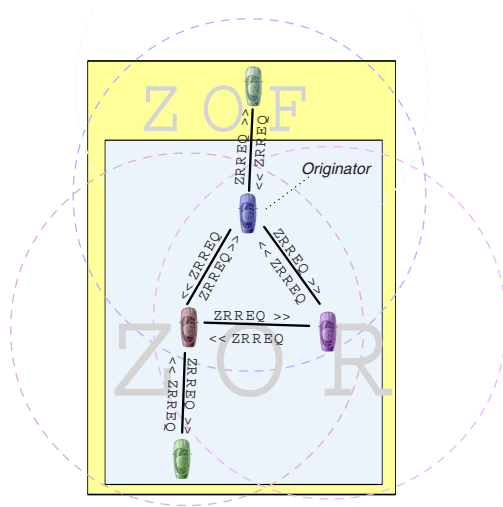


**Fig. 1.** ZRRREQ messages are flooded from the originator (source) vehicle

In this paper we present a RObust VEHicular Routing (ROVER) protocol, that offers reliable geographical multicast. The protocol uses a reactive route discovery process within a ZOR. We evaluate the protocol with a realistic simulation setup. We consider a generic data transfer application, in which a vehicle sends a data message to all vehicles within a specified ZOR. The results show that ROVER delivers the data with reasonable delays to 100% of the intended vehicles for almost all scenarios. Also, ROVER could be used by applications that require end-to-end QoS, by implementing a transport layer protocol that uses the multicast tree set up by ROVER.

## 2 Rover

In this section we will describe the routing protocol ROVER (RObust VEHicular Routing). In short the main difference between geocasting and ROVER is similar to the difference between flooding and a MANET reactive protocol such as AODV: both in ROVER and in AODV only control packets are flooded in the network - the data packets are unicast, potentially increasing the efficiency and reliability. Each vehicle is assumed to have a unique Vehicle Identification Number (VIN). Also, the vehicles are assumed to have a GPS receiver and access to a digital map. The objective of the protocol is to transmit a message,  $M$ , from an application,  $A$ , to all other vehicles within an application-specified ZOR,  $Z$ . The ZOR is defined as a rectangle (although other definitions can be easily accommodated) specified by its corner coordinates. Thus, a message is defined by the triplet  $[A, M, Z]$ . When a vehicle receives a message, it accepts the message if, at the time of the reception, it is within the ZOR. Similar to geocasting protocols we also define a Zone Of Forwarding (ZOF) as a zone including the source and the ZOR. All vehicles in the ZOF are part of the routing process, although only vehicles in the ZOR deliver the message to their corresponding application layer (specified by  $A$ ).

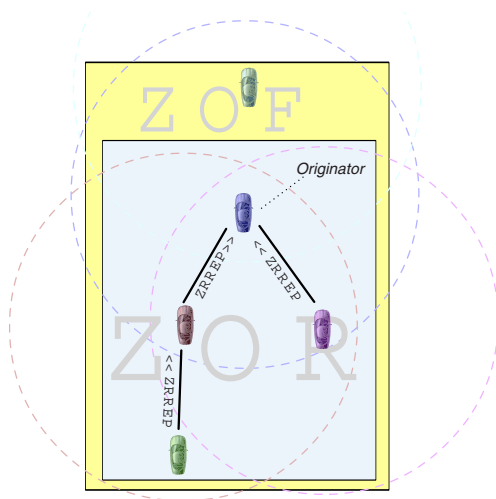


**Fig. 2.** ZRREP messages are unicasted the one-hop neighbors from where the ZRREQ was first received

## 2.1 Route Discovery

The first time the routing layer receives a packet  $[A, M, Z]$  from the application layer, a route discovery process is triggered. The process is also initiated if the previous ZOR is no longer valid. The objective of the route discovery process is to build a multicast tree from the source vehicle to all vehicles within the ZOR  $Z$ .

As shown in Figure 1, the route discovery process is initiated when the originator vehicle floods a Zone Route Request (ZRREQ) message containing its  $VIN$ , location, the current ZOR, and a route sequence number,  $SS$ , throughout the ZOF.

Any vehicle that receives a ZRREQ for the first time for this session sequence number accepts the message if the vehicle is within the ZOF, and is not too far away from the sender. The reason for including the distance to the sender is to build a robust multicast tree. The *Cutoff Distance* is calculated as  $\alpha \cdot R$  where  $R$  is the (assumed) maximum radio range and  $0 < \alpha \leq 1$ . In this paper we have used  $\alpha = 2/3$ .

If a vehicle accepts a ZRREQ, it replies to the one-hop vehicle that forwarded the ZRREQ with a Zone Route Reply (ZRREP) message, containing its  $VIN$ . It also stores the information  $[SS, Z]$  in a routing table. Finally it re-broadcasts the ZRREQ, including the original  $VIN$ ,  $ZOR$ , and  $SS$ . The vehicles in ZOF but not in ZOR do not reply to ZRREQ messages unless they receive a reply themselves. The sequence number  $SS$  in conjunction with the  $VIN$  of the source vehicle (originator) is used as a unique identifier in the routing tables formed by the route discovery process.

After flooding the ZRREQ throughout the ZOF, unlike for AODV, the ZRREP messages are not sent back to the source. Instead they are only transmitted to the node transmitting the ZRREQ. All recipients of a ZRREP message store the  $VIN$  of the vehicle that sent the ZRREP and the corresponding  $SS$  and source  $VIN$ . Data packets from the same source  $VIN$  and  $SS$  will be forwarded to the sender of the ZRREP. This way all nodes store the local information needed to build a multicast tree rooted at the source node. Once the tree is formed, i.e., after the ZRREP are sent to parents in the tree, data can be disseminated in the tree (as shown in Figure 2).

## 2.2 Data Transfer

Since each vehicle stores next-hop(s) information about the source  $VIN$  and  $SS$ , data will be forwarded through the tree as a function of those numbers. The source forwards the data packets immediately after it receives a ZRREP message. The source (and all forwarding nodes in the multicast tree) unicasts the message  $M$  to all the vehicles from which it received a ZRREP. The message is also stored in a buffer for a short time in case it receives a ZRREP after it receives the message. Thus, each message is propagated through the multicast tree according to the “route table” stored during the route discovery process. All receivers also deliver the packet if they are within the ZOR. Since the data is transferred using unicast, it benefits of the normal MAC-layer acknowledgments.

## 2.3 Route Timeout

As vehicles move, the ZOR for a certain application will change in time. However, if a vehicle sends several messages to the same ZOR application within a short time,

there is no need to perform a route discovery for each message. For example, for vehicles travelling at 90km/h, the ZOR may only change by 25m in one second. If the initial ZOR is several kilometers large, the same ZOR can be used. We considered a ZOR invalid when the source vehicle moved for more than 25m from the initial route discovery position.

### 3 Simulation Environment

We have evaluated ROVER using the simulation package Jist/SWANS [19][20] with the STRAW module [21]. Jist/SWANS is a simulator for mobile ad-hoc networks, similar to ns-2, implemented in JAVA. STRAW uses real maps from the Topologically Integrated Geographic Encoding and Referencing (TIGER) system available from the US Census Bureau Geography [22]. We enhanced the simulation setup in several respects and implemented ROVER as a new routing module. At the time we performed the simulations, the development of Jist/SWANS is an ongoing project, and the STRAW module is developed for city scenarios with low speeds and a road grid. It also had a number of incomplete protocol specifications (e.g., missing sequence number in 802.11). Furthermore, since the original protocol stack uses unicast with fixed addresses, we had to make a number of modifications to the original Jist/SWANS/STRAW packages.

#### 3.1 A Data Transfer Application

To evaluate the performance of the proposed routing protocol, we used a generic data transfer application. In this application a vehicle sends a message to vehicles behind it. The vehicle that sends out the message will be referred to, in the rest of the paper, as the Source Vehicle (SV). When an SV sends a message, the application determines a suitable ZOR. In this paper, the ZOR will be a rectangle directly behind the SV, with length  $L$  meters and width  $W$  meters.  $W$  is large enough to cover all lanes of the current road that goes in the same direction as the SV. The message should then be delivered to all vehicles within the ZOR, as fast and as reliable as possible.

#### 3.2 Road and Traffic Models

STRAW uses real road maps by default. Since the objective of the investigations was to evaluate the proposed routing protocol, we wanted to have a very simple road model to avoid any effects caused by the specific road map used. Therefore, we constructed a straight highway in TIGER format and then used this road in the simulations. The highway is of length 10km and with 3 lanes in each direction. The maximum allowed speed on the highway is 120 kilometers per hour.

Vehicles move according to a car-following model [23]. We implemented lane changing behavior. Originally STRAW did not implement this feature, and we observed cases with one lane was heavily congested while the other one was not. In our setup vehicles may change lane if the vehicle in front of them moves too slow.

### 3.3 Communication Models

At the physical layer we used the Rayleigh fading model supplied by SWANS. This model has a gradual transition from 100% to 0% reception rate as the distance between the sender and receiver increases. The physical layer data rate we considered was 54Mbps, consistent with the 802.11a data rates (which in turn are similar to the Dedicated Short Range Communications (DSRC) standard [24]).

At the MAC layer we used the CSMA/CA scheme used in IEEE 802.11 (similar to DSRC). At the network and transport layer we used a slightly modified version of IP and UDP. In particular, since we used geographical addressing, instead of the normal IP addresses we used VIN numbers for the vehicles and ZOR and ZOF (specified by the coordinates of the corners) to specify the destinations.

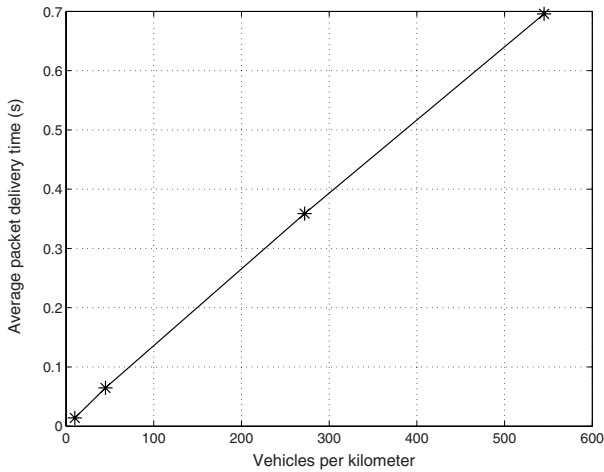


**Fig. 3.** Varying number of vehicles

### 3.4 Simulation Setup

In the beginning of a simulation, a number  $N$  of vehicles are placed on the highway at regular intervals. All vehicles attempt to travel with the maximum posted speed while using the car following model. We placed the vehicles uniformly on the highway. Three seconds into the simulation a vehicle sends a message to vehicles in a ZOR behind itself. In our implementation the ZOR specifies the following:

- The VIN of the source node.
- The current location of the source node (absolute coordinates).
- The extent of the ZOR (relative to the source node).
- The direction of the movement of the source node.

The maximum deviation of the direction of a vehicle from the direction of the SV such that it can still be considered in the ZOR. Nodes that deviate from more than this specified value are in ZOF but not in ZOR. In our implementation we used 180

degrees (i.e., all vehicles in the ZOR will deliver their packets to the application layer). The ZOF (for this application) is specified as the ZOR and an additional buffer zone 15 meters wide.

The default simulation parameters (shown in parenthesis) and the range of values we investigated are shown in Table 1. During the simulations we varied one parameter at a time while maintaining the rest fixed at the default value.

**Table 1.** Simulation parameters

| Parameter                    | Value                |
|------------------------------|----------------------|
| Number of vehicles/km        | 10, 45, (272), 545   |
| Radio transmission range [m] | 100, 200, (300), 400 |
| Length of the ZOR [km]       | 0.5, (1.5), 2.5, 3.5 |

## 4 Performance Metrics

We used two performance metrics when evaluating ROVER. The first metric we considered was the packet delivery ratio, *PDR*, i.e. the percentage of vehicles that (1) are within the ZOR when the message is sent and (2) receives the message. To measure the *PDR* for each message from the *SV* we counted the number of vehicles in the ZOR at the time the message was generated and compared it with the number of vehicles that receive the message. Since more vehicles can enter the ZOR before the message is transmitted throughout the ZOR, *PDR* can be (slightly) larger than 100%. The second metric was the average packet delivery time,  $T_d$ , i.e., the average delay between the time a message is sent by the *SV* until the vehicles receives the message.

## 5 Results and Discussion

In this section we present the performance results for varying the vehicle density, the transmission range and the size of the zone of relevance. The results shown here are averages from 30 runs (with different seeds), and all confidence intervals are within 10% of the average.

### 5.1 Packet Delivery Ratio (PDR)

The results show that ROVER delivers 100% of the messages for almost all scenarios. It is only when the vehicle density is very low (10 vehicles/km) that a message sometimes cannot reach all vehicles within the ZOR. In this case, the average distance between the vehicles is 100 meters, which means that if a ZZREQ or a ZZREP message is lost, a part of the multicast tree may be lost. However, this result is not only due to ROVER, since all routing protocol would probably encounter problems for such low vehicle densities. For all other scenarios, the *PDR* is 100%.

It can be noted that the cutoff mechanism in the route discovery process has a major impact on the performance. It is crucial that the multicast tree is robust and

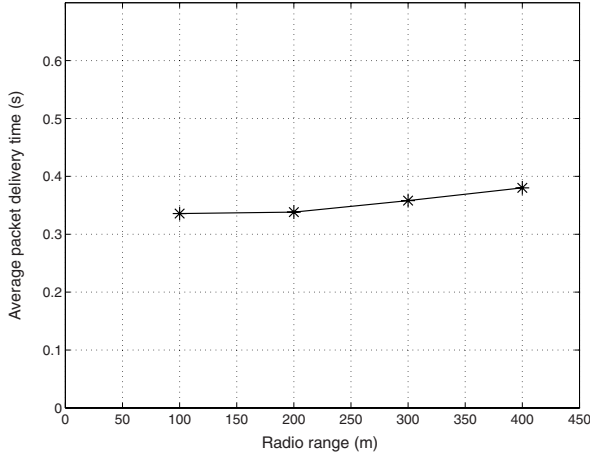


**Fig. 4.** Varying radio range

therefore it is important that linked nodes are relatively close to each other due to the fading channel. It is better with several short (reliable) hops than a few long (unreliable) hops.

## 5.2 Packet Delivery Time ( $T_D$ )

The packet delivery time,  $T_D$ , shows the average time it takes for the message to reach all cars within the ZOR. Of course, the results depend on the chosen scenario.

### 5.2.1 Vehicle Density

As the number of cars increases,  $T_D$  also increases, see Figure 3. The Route Discovery Process is based on flooding. With more cars, packet collisions and backoff times increase at the MAC layer and the effect is longer delays on the application layer. Several papers (see, for example, [16]) have suggested an improved flooding mechanism in which a node has a waiting time before forwarding a packet. The waiting time depends on the distance to the previous sender and nodes further away from the sender will forward the packet sooner than nodes close to the sender. We implemented this feature in ROVER, but could not see any obvious improvements in the performance.

### 5.2.2 Radio Transmission Range

One could expect that a longer radio range would decrease the packet delivery time, due to fewer hops. However, our results for these scenarios showed that the radio range is *not* a major factor in the delivery time, see Figure 4. As the transmission range increases, each transmission will be heard by more nodes. Therefore, the risk of packet collisions and hidden terminals increases. Also, one major part of the packet delivery time is the protocol handling delay in the nodes. This delay will of course not be shorter just because the radio range increases.

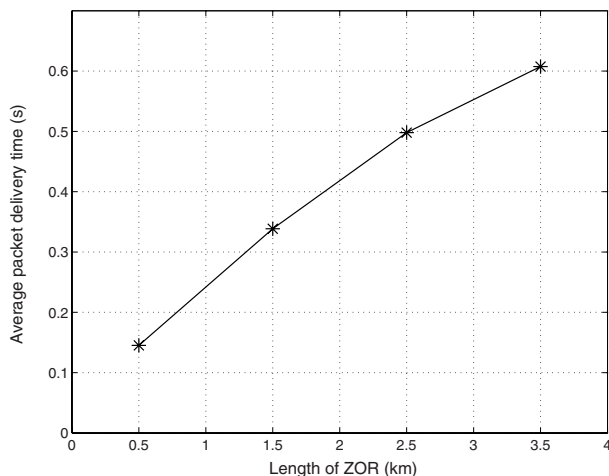


**Fig. 5.** Varying length of ZOR

### 5.2.3 Zone of Relevance

As expected, the delivery time is proportional to the length of the ZOR, see Figure 5. More hops are needed to cover the larger area and therefore the delivery time increases. Remarkable is that even for a ZOR as long as 3.5 km, the delivery time is as low as 600 ms and 100% of the vehicles within the ZOR receives the data. Therefore, ROVER is well suited for VANET applications that require multicast with end-to-end QoS.

## 6 Conclusions

Vehicular ad-hoc networks have the potential to both reduce accidents as well as enhance the comfort of the driver and passengers. Different applications will have different enforce different requirements on the network protocols used. In this paper we have focused on those applications that have requirements on the end-to-end QoS. For those applications there will be a need for a reliable transport protocol. In order for a reliable transport protocol to work properly, a routing protocol is needed that maintains some information about sender and receivers.

Therefore, we in this paper have presented ROVER, a new multicast routing protocol for vehicular ad hoc networks. The protocol uses geographical addressing to form a multicast tree within a zone of relevance. The tree is formed on-demand and can be used to forward multiple data packets from the same source. Therefore, it can be used by a reliable transport protocol to ensure end-to-end QoS. We have evaluated the performance of the protocol in a realistic environment with detailed models both for the vehicular traffic as well as for the physical environment.



## References

1. M. Williams, "PROMETHEUS-the european research programme for optimising the road transport system in europe", In Proc. of the IEE Colloquium on 'Driver Information' (Digest No.127), 1988, pp. 1–9.
2. S. Shladover, C. Desoer, J. Hedrick, M. Tomizuka, J. Walrand, W. Zhang, D. McMahon, H. Peng, S. Sheikholeslam, and N. McKeown, "Automatic vehicle control developments in the PATH program", *IEEE Transactions on Vehicular Technology*, vol. 40, no. 1, pp. 114–130, 1991.
3. W. Enkelmann, "Fleetnet - applications for inter-vehicle communication," in Proc. of the IEEE Intelligent Vehicles Symposium, 2003, pp. 162–167.
4. S. Wang, "The effects of wireless transmission range on path lifetime in vehicle-formed mobile ad hoc networks on highways", In Proc. of the IEEE International Conference on Communications, 2005, pp. 3177–3181.
5. S. Biswas, R. Tatchikou, and F. Dion, "Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety", *IEEE Communications Magazine*, vol. 44, pp. 74–82, 2006.
6. C. Maihöfer, C. Cseh, W. Franz, and R. Eberhardt, "Performance evaluation of stored geocast", In Proc. of IEEE 58th Vehicular Technology Conference, 2003, pp. 2901–2905.
7. J. Bronsted and L. Kristensen, "Specification and performance evaluation of two zone dissemination protocols for vehicular ad-hoc networks", In Proc. of the 39th Annual Simulation Symposium, 2006.
8. T. Kosch, C. Schwingenschlögl, and L. Ai, "Information dissemination in multi-hop inter-vehicle networks", In Proc. of the IEEE 5th International Conference on Intelligent Transportation Systems, 2002, pp. 685–690.
9. A. Ho, Y. Ho, and K. Hua, "A connectionless approach to mobile ad hoc networks in street environments", In Proc. of the 2005 IEEE Intelligent Vehicles Symposium, 2005, pp. 575–582.
10. R. Santos, A. Edwards, R. Edwards, and N. Seed, "Performance evaluation of routing protocols in vehicular ad-hoc networks", *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 1, pp. 80–91, 2005.
11. T. Little and A. Agarwal, "An information propagation scheme for VANETs", In Proc. of the 8th International IEEE Conference on Intelligent Transportation Systems, 2005, pp. 155–160.
12. C. Lochert, H. Hartenstein, J. Tian, H. Füssler, D. Hermann, and M. Mauve, "A routing strategy for vehicular ad hoc networks in city environments", in Proc. of the IEEE Intelligent Vehicles Symposium, 2003, pp. 156–161.
13. G. Korkmaz, E. Ekici, F. Özgüner, and U. Özgüner, "Urban multi-hop broadcast protocol for inter-vehicle communication systems", In Proc. of the first ACM workshop on Vehicular ad hoc networks, 2004, pp. 76–85.
14. Y. Ko and N. Vaidya, "Flooding-based geocasting protocols for mobile ad hoc networks," *Mobile Networks and Applications*, no. 7, pp. 471–480, 2002.
15. W. Kremer, "Realistic simulation of a broadcast protocol for a inter vehicle communication system (IVCS)", In Proc. of the 41st IEEE Vehicular Technology Conference, 1991, pp. 624–629.
16. L. Briesemeister, L. Schäfers, and G. Hommel, "Disseminating messages among highly mobile hosts based on inter-vehicle communication", In Proc. of the IEEE Intelligent Vehicle Symposium, 2000, pp. 522–527.

17. H. Alshaer and E. Horlait, "An optimized adaptive broadcast scheme for inter-vehicle communication", In Proc. of the IEEE 61st Vehicular Technology Conference, 2005, pp. 2840–2844.
18. A. Bachir and A. Benslimane, "A multicast protocol in ad hoc networks inter-vehicle geocast", In Proc. of the 58th IEEE Vehicular Technology Conference, 2003.
19. R. Barr, Z. J. Haas, and R. van Renesse, "Jist: An efficient approach to simulation using virtual machines", *Software - Practice and Experience*, vol. 35, pp. 539–576, 2004.
20. SWANS website, <http://jist.ece.cornell.edu>.
21. D. Choffnes and F. Bustamante, "An integrated mobility and traffic model for vehicular wireless networks", In Proc. of the Second ACM International Workshop on Vehicular Ad Hoc Networks (VANET 2005), Cologne, Germany, Sep 2005.
22. US Census Bureau Geography, <http://www.census.gov/geo/www>.
23. R. Rothery, "Car following models", *Trac Flow Theory*, 1992.
24. "Standard specification for telecommunications and information exchange between roadside and vehicle systems - 5GHz band dedicated short range communications (DSRC) medium access control (MAC) and physical layer (PHY)," ASTM E2213-03, Sept. 2003.

# A Flexible Tree-Based Routing Protocol with a Mesh Relaying Node in Multi-hop Wireless Mesh Networks \*

Hyun-Ki Kim and Tae-Jin Lee

School of Information and Communication Engineering,  
Sungkyunkwan University, Suwon 440-746, Korea  
{hidden01, tjlee}@ece.skku.ac.kr

**Abstract.** The IEEE 802.11 standard defines a medium access control protocol for wireless local area networks (WLANs). In typical WLANs, access points (APs) are connected with one another in a wired manner. As a way to loosen this restriction wireless mesh networks (WMNs), which are rather independent of a backbone infrastructure, is emerging. Due to their inherent characteristics unlike ad hoc networks, it is unreasonable that the ad hoc network routing protocols are directly applied for the WMNs without any modification. The tree-based routing protocol (TRP) defined in the IEEE 802.11s is one of the routing protocols used in WMNs. The TRP, however, forwards data on the only one active path. In this paper, we propose a flexible tree-based routing protocol with a mesh relaying node (TRP-MRN) in WMNs. The TRP-MRN provides a concurrent multi-path establishment algorithm. We analyze the end-to-end delay of the TRP and the TRP-MRN and show that the TRP-MRN can effectively reduce the end-to-end delay and provide alternative routes via simulations.

## 1 Introduction

Wireless local area network (WLAN) technology has been growing rapidly over the last few years. WLANs are currently deployed indoors as hot spots such as a cafe, building, and conference room. WLANs allow users to access and share data, applications, or other network resources in the same way as wired local area networks (LANs). WLANs consist of two or more stations connected together via radio frequency channel, which is different from wired LANs. IEEE 802.11 WLANs have a nominal transmission rate of between 11 and 54Mbps. In typical WLANs, access points (APs) are physically connected to the wired networks using Ethernet cabling. The APs act as gateways/bridges between the wired networks and the stations associated with them [5].

In parallel with the prosperity of WLANs, wireless mesh networks (WMNs) have been emerging as a key technology in recent years. Because radio frequency channels are used to connect APs with one another, WMNs are not only easy and cost-effective in deployment, but also has good scalability in coverage area and capacity [7]. Future

---

\* This research was supported by a grant under the Brain Korea 21 Initiative of the Korean Ministry of Education and Human Resources and by grant No.R01-2006-000-10402-0 from the Basic Research Program Korea Science and Engineering Foundation of Ministry of Science & Technology. Dr. Tae-Jin Lee is the corresponding author.

wireless technology is expected to incorporate ad-hoc networks model in some form [8], [9], [10], [12]. Since WMNs can be dynamically self-organized and self-configured, they can automatically establish ad hoc networks and also maintain the wireless mesh connectivity. Due to their advantages over other wireless networks, WMNs are making a rapid progress and inspiring numerous applications [4].

WMNs consist of two types of nodes, that is, mesh points (MPs) and mesh clients such as notebook computers, personal digital assistants (PDAs), Tablet PCs, mobile phones, and so on. The mesh clients are usually formed using one type of radio and the MPs act as APs and support WLAN mesh services other than the routing capability as in traditional wireless routers. To further improve the flexibility of mesh networking, the MPs are usually equipped with multiple wireless interface cards functioning over a variety of wireless access technologies [2]. The MPs have minimal mobility and form the mesh backbone for mesh clients. The mesh clients equipped with wireless network interface cards (NICs) can access directly to WMNs through MPs. The mesh clients without wireless NICs can connect WMNs through Ethernet [3], [4].

The IEEE 802.11s WMN standard presents two types of routing protocols: tree-based routing protocol (TRP) [1], [13], [14], [15], [16] and radio-aware optimized link state routing (RA-OLSR) [3], [17], [18]. The TRP keeps routes for transferring data towards the root MP node in a proactive manner. In the TRP, routes must be established before data can be sent along the routes. So the route discovery time in proactive routing protocols is less than that in on-demand routing protocols. But a proactive routing protocol such as TRP has no choice of alternative routes until they succeed in discovering alternative routes.

To avoid service disruption, WMN routing protocols must be robust to link failures. Since a tree-based routing protocol keeps only one active path from a source node to a destination node, it is extremely important that broken links be recovered quickly. In order to make failures of links and nodes less disruptive in WMNs, we propose a flexible tree-based routing protocol with a mesh relaying node (TRP-MRN). The TRP-MRN is easily applied to WMNs where an MRN is set up and the MRN can link adjacent clusters a cluster set of mesh clients under an MP to each other. In the TRP, a source node transfers data to the root node via only one active route when a source node transfers data to the destination node associated with a different cluster. On the other hand, in the TRP-MRN, a source node makes an inquiry about paths to the destination node associated with a different cluster to the root node and to the MRN in the same cluster. The concurrency in the route establishment process has the following merits.

1. A route establishment failure in one link does not slow down the entire route establishment process.
2. If several routes are available for data delivery, data is delivered along the route with less cost.

Although path inquiry results in 88 bytes overhead, our analysis and simulation results show that the TRP-MRN can effectively reduce the end-to-end delay and provide alternative routes. The rest of this paper is organized as follows. The existing TRP is discussed in Section 2. The proposed TRP-MRN is described in Section 3. The analysis and simulation results are presented in Section 4. Section 5 concludes our paper.

## 2 Existing Tree-Based Routing Protocol

In proactive routing protocols, a route should be established before data can be transferred. If the discovery process for a route reaches a node where sufficient quality is not available for data delivery, it cranks back several hops and then the discovery is continued from the node on the route. So this process results in an increased time in the route establishment process.

### 2.1 TRP Protocol Overview

The existing TRP in IEEE 802.11s draft is discussed as follows, and the major steps are summarized in Fig. 1. In Step 1, if an MP in a WMN is configured as a root node, other MPs may proactively establish paths to the root node.

In Step 2, topology formation begins when the root node broadcasts the root announcement message that contains a distance metric and a sequence number. The value of the metric is set to zero before broadcast. Any MP hearing the message establishes a path to the root MP node. When the MP selects a parent node for its path to the root node, it registers itself with the root node by sending a “Gratuitous RREP” containing its own address as well as all connected devices. In addition, the MP caches others as potential parents. If the path establishment procedure succeeds, the MP directly updates its routing table and identifies itself as being a directly connected child of the root node. The MP also cumulates the metric associated with the path. Then, the MP rebroadcasts the message with the updated metric. Thus, the tree-based paths are built away from the root node to the child MPs as each MP updates the metric to the root node and rebroadcasts the cumulative metric to its neighbors to the root node.

In Step 3, once an MP selects a parent MP, it maintains the path by periodically sending a maintenance RREQ to the parent MP and receiving an maintenance RREP back. It also proactively broadcasts topology RREQs and compares the topology RREPs with its current metric to ensure that it always has the best path to the root node, i.e., the MP sends out a topology RREQ with the time-to-live (TTL) optionally set to 1 including the root node address, and “Destination Only (DO)” flag bit set to 0 to re-validate the topology RREPs heard from neighbor MPs. A TTL stands for the remaining number of times a message may be forwarded. If a DO flag is set to 1, only a destination MP can generate a RREP. If any intermediate MP receives a RREQ with a DO flag set to 0 for a destination MP and the intermediate MP already has a valid path to the destination MP, it forwards a unicast RREP to the source MP. The interval of topology RREQs/RREPs interval is even longer than that of maintenance RREQs/RREPs. If the current parent is unable to provide the best path, the MP switches to another parent on the list of potential parent nodes. If the link with the current parent is broken, the MP broadcasts the topology RREQ immediately and send out a RERR down the topology so that it makes all nodes remove the entry from its routing table. If there is any potential parent node on the list, it does not need to update the topology paths. But if any potential parent nodes are not found for a pre-defined period, a RERR is forwarded to all children MPs.

In Step 4, when an MP selects a parent MP, it registers itself with the root node by sending a “Gratuitous RREP” message. The message is containing its own address as

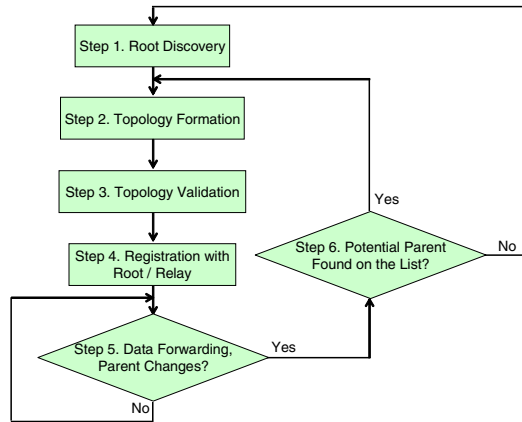


**Fig. 1.** The major steps in TRP. The root node and MPs maintain separate state machines in the TRP.

well as all connected devices. Each intermediate MP on the path receiving the “Gratuitous RREP” message, updates the routing table entry (i.e., a proxy registration procedure to the root node). The MPs are not required to learn about the device addresses. Thus the root node does not learn about all MPs and devices until Step 4 is completed.

In Step 5, data forwarding, topology maintenance and optimization take place. All nodes on the same path to the root node send data to their parent node until the data reaches the root node. Topology maintenance uses directed, unicast RREQ messages at regular intervals. An MP increments the sequence number of each maintenance RREQ before forwarding it. The sequence number of each maintenance RREQ is used by nodes to drop duplicate maintenance RREQs. The recipient MP must reply with a RREP if it is still alive and has a current path to the root node.

In Step 6, if an MP changes its present parent MP with a potential parent MP on the list, it always send a re-registration in the form of “Gratuitous RREP” to the root node so that each intermediate MP can update their routing tables and then the root node issues a RERR on the previous path. But if any potential parent MP is not found on the list, the MP sends out a topology RREQ and moves to Step 1.

## 2.2 TRP Routing Procedure

When a source node wants to send data to a destination node, and if it has no path to the destination node, it sends the data to the root node. Then the root node identifies whether the data is intended for a node within the same cluster or not. If it cannot find an entry in its routing table, it may forward the data on its uplink. If necessary, it must broadcast a RREQ to discover the destination node. If the destination node is within the same cluster, the destination node wishing to receive data acquires the route to the source device on-demand in the way similar to the ad-hoc on demand distance vector routing (AODV) [2]. This hybrid routing protocol allows an initial frame to be

forwarded along a tree (i.e., an optimal metric path) and then the subsequent frames to be forwarded through an on-demand route for the source-destination pair.

### 3 Proposed Tree-Based Routing Protocol with a Mesh Relaying Node

As WLAN service area expands, more than two clusters may become contiguous to each other (see Fig. 2). In this case, when a source node in a cluster transfers data to a destination node in another cluster, it transfers the data on the only active route via the root node connecting the clusters. In the TRP, the failure to discover a route results in delay in the route establishment process due to the time to find an alternative valid route starting at the node where the discovery failure occurred. Our flexible TRP-MRN, however, reduces the delay establishing the routes concurrently since it involves only a minimal modification to the TRP. The placement of MRNs is to be strategically decided based on coverage, so that some of the nodes in different clusters can connect to MRNs directly. In this section we describe the proposed TRP-MRN in detail.

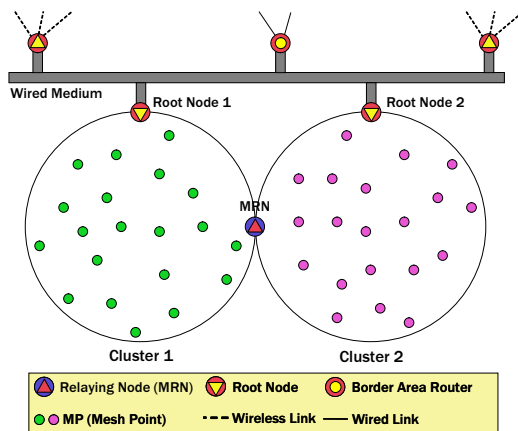


Fig. 2. An example of a mesh network topology with an MRN

#### 3.1 TRP-MRN Protocol Overview

Fig. 3 shows the flowchart of steps in the proposed TRP-MRN. Step 1 through Step 4, Step 9, and Step 10 in the TRP-MRN are similar to those in the existing TRP. Step 5 through Step 8 are dedicated to the proposed TRP-MRN. In Step 4, the MP which tries to register is waiting for registration acknowledgment (ACK) from the "Gratuitous RREP" messages were sent for registration until the timer expires.

In Step 5, if any registration ACKs are not received for a predefined period, it returns to Step 1, and then the MP has to wait for the root/relay announcement messages again. If only one registration ACK is received, it skips to Step 9 because there is only one valid route. If two registration ACKs are received, the MPs proceed to the next Step.

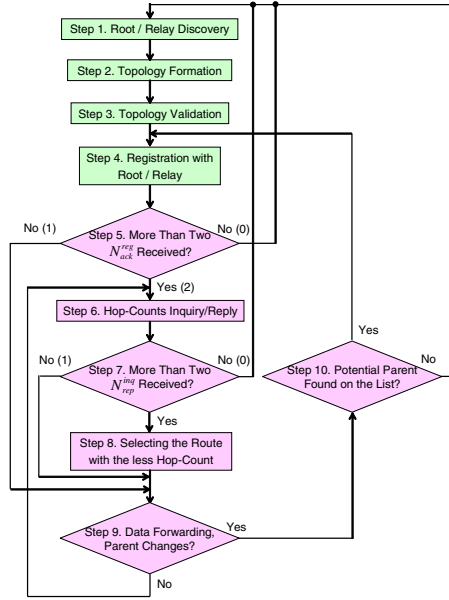


**Fig. 3.** The major steps in the proposed TRP-MRN. (Note that, we use  $N_{ack}^{reg}$  to represent the number of registration ACKs. Similarly,  $N_{rep}^{inq}$  is used to represent the number of the replies about hop-count queries).

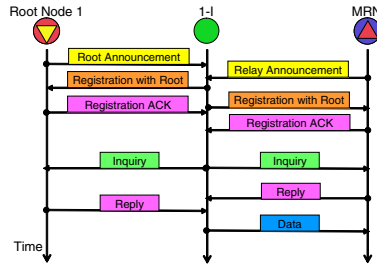


**Fig. 4.** Proposed TRP-MRN protocol. The intermediate nodes on the routes are omitted to simply explain the protocol.

The procedure that a source node inquiries of the root node and the MRN about hop-count or cost to a destination works as in Step 6 in Fig. 1. Each of them already knows the hop-count of cost to all nodes having registered with them due to Step 1 through Step 4, therefore they replies to the source node. The existing RREQ (46 bytes) and RREP (42 bytes) messages are used for the inquiry and the reply, respectively for compatibility. The messages are sent along the tree-based established in Step 1 through Step 4.

In Step 7, the source node is waiting for the replies about hop-count or cost from the inquiry messages were sent until the timer expires. Like Step 4, if any replies do not



arrive at the source node for the period, it returns to Step 1 for the same purpose. If only one reply is received, it also skips to Step 9. If two registration ACKs are received, Step 8 starts.

In Step 8, the MP selects the path with the minimum hop-count or cost among the two registration ACKs. As stated above, Step 9 and Step 10 follow the corresponding processes of the TRP. The MP maintains two routes to the root node and the MRN independently of each other. Fig. 4 illustrates a TRP-MRN protocol.

3.2 TRP-MRN Routing Table Setup

Fig. 5 and Fig. 6 illustrates routing tables for the root node and for the MRN respectively. Each entry in the routing tables stores the corresponding next-hops from which the root and relay announcement message was received, the sequence numbers of the messages, and the hop-counts to the root and the MRN. Each node in the cluster keeps two corresponding next-hops to its own children MPs as well as to the root and the MRN. The routing tables are refreshed every time when the maintenance RREQs are broadcasted.

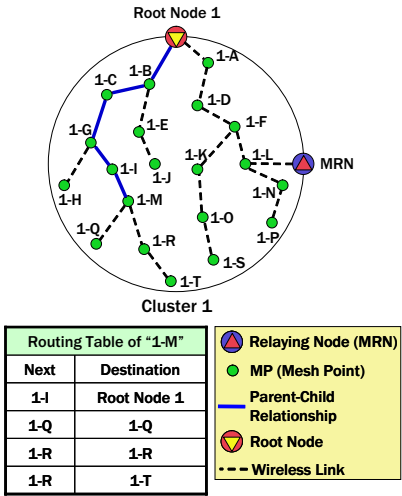


Fig. 5. Routing table of node 1-M for the root node in the cluster 1

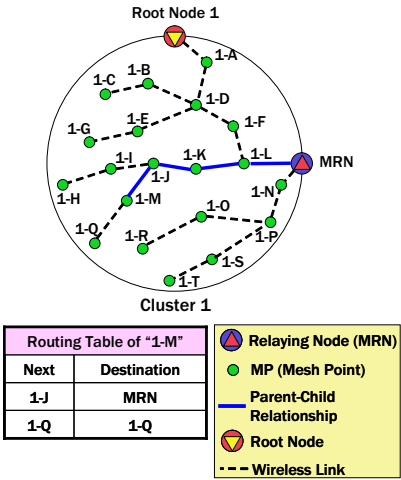


Fig. 6. Routing table of node 1-M for the MRN in the cluster 1

4 Performance Evaluation

4.1 Analysis of End-to-End Delay

In this subsection, we analyze and compare the end-to-end delay of the proposed TRP-MRN with the TRP. The analysis is general in the sense that it does not limit to any specific topology in WMNs. For analytical tractability, we ignore the queueing delay

and processing delay in the following discussion. To calculate the end-to-end delays of the TRP and the TRP-MRN, we define the following notations:

- $D_{TRP}^{total}$ : End-to-end delay of TRP
- $D_{relaying}^{total}$ : End-to-end delay in case data is forwarded via a MRN
- $D_{TRP-MRN}^{total}$ : End-to-end delay of TRP-MRN
- $D_{inq}^{hop}$ : End-to-end delay in case a source inquires the hop-count to a destination to a root/MRN
- $D_{rep}^{hop}$ : End-to-end delay in case a root/MRN replies to the source
- $H_{total}^{root}$ : Total hop-count in case data is forwarded from a source to a destination via a root
- $H_{total}^{relaying}$ : Total hop-count in case data is forwarded from a source to a destination via a MRN
- $H_{source}^{root}$ : Hop-count in case an inquiry message is forwarded from a source to a root
- $H_{source}^{relaying}$ : Hop-count in case an inquiry message is forwarded from a source to a MRN
- $D_i^{trans,root}$ : Transmission delay of the node  $i$  in case data is forwarded via a root,  $1 \leq i \leq H_{total}^{root}$
- $D_i^{prop,root}$ : Propagation delay of the node  $i$  in case data is forwarded via a root,  $1 \leq i \leq H_{total}^{root}$
- $D_i^{total,root}$ : End-to-end delay of the node  $i$  in case data is forwarded via a root,  $1 \leq i \leq H_{total}^{root}$
- $D_j^{trans,relaying}$ : Transmission delay of the node  $j$  in case data is forwarded via a MRN,  $1 \leq j \leq H_{total}^{relaying}$
- $D_j^{prop,relaying}$ : Propagation delay of the node  $j$  in case data is forwarded via a MRN,  $1 \leq j \leq H_{total}^{relaying}$
- $D_j^{total,relaying}$ : End-to-end delay of the node  $j$  in case a data is forwarded via a MRN,  $1 \leq j \leq H_{total}^{relaying}$
- $T_{source}^{root}$ : Cumulative distance from a root to a source
- $T_{source}^{relaying}$ : Cumulative distance from a MRN to a source
- $T_{total}^{root}$ : Total distance in case data is forwarded from a source to a destination via a root
- $T_{total}^{relaying}$ : Total distance in case data is forwarded from a source to a destination via a MRN
- $R_{wireless}$ : Wireless transmission rate
- $R_{wired}$ : Wired transmission rate
- $V$ : Velocity of electromagnetic wave propagation
- $M_{body}$ : MAC frame body size
- $M_{h,fcs}$ : MAC frame header and frame check sequence (FCS) size
- $P_{pre}$ : Physical layer (PHY) preamble size
- $P_h$ : PHY header size
- $H_{inq}$ : Inquiry message size
- $H_{rep}$ : Reply message size

Based on the above definition, the total end-to-end delay when data is forwarded via the root node is given

$$\begin{aligned}
 D_{TRP}^{total} &= \sum_{i=1}^{H_{total}^{root}} D_i^{total} \\
 &= \sum_{i=1}^{H_{total}^{root}} (D_i^{trans} + D_i^{prop}) \\
 &= (P_{pre} + P_h + M_{h_{fcs}} + M_{body}) \times \left( \frac{H_{total}^{root}}{R_{wireless}} + \frac{1}{R_{wired}} \right) + \frac{T_{total}^{root}}{V},
 \end{aligned} \tag{1}$$

where root nodes are assumed to be connected via one-hop wired link. The total end-to-end delay when the data is forwarded via the MRN is found as

$$\begin{aligned}
 D_{relaying}^{total} &= D_{inq}^{hop} + D_{rep}^{hop} + \sum_{j=1}^{H_{total}^{relaying}} D_j^{total} \\
 &= D_{inq}^{hop} + D_{rep}^{hop} + \sum_{j=1}^{H_{total}^{relaying}} (D_j^{trans} + D_j^{prop}) \\
 &= \frac{(P_{pre} + P_h + M_{h_{fcs}}) \times 2 + H_{inq} + H_{rep}}{R_{wireless}} \times \max(H_{source}^{root}, H_{source}^{relaying}) \\
 &\quad + \frac{\max(T_{source}^{root}, T_{source}^{relaying})}{V} \times 2 + \frac{P_{pre} + P_h + M_{h_{fcs}} + M_{body}}{R_{wireless}} \\
 &\quad \times H_{total}^{relaying} + \frac{T_{total}^{relaying}}{V}
 \end{aligned} \tag{2}$$

Since the source node selects the minimum hop-count to the destination node as in Step 8 in Fig. 3, the end-to-end delay of the TRP-MRN is, therefore, given by

$$D_{TRP-MRN}^{total} = \min(D_{TRP}^{total}, D_{relaying}^{total}). \tag{3}$$

## 4.2 Simulations

In this subsection, we compared the performance of our proposed TRP-MRN with the existing TRP. We consider two main performance metrics: mean hop-count and mean end-to-end delay. Without losing generality, we concentrate on unidirectional transmissions, where packets are sent from/to a root node. Simulation topologies are obtained by generating 1000 random WMNs with one MRN, two root nodes, and 40 MP nodes. In each WMN, nodes are uniformly distributed in each of the two clusters with a diameter of 200 m. The number of source-destination (S-D) pairs in each WMN is 1000. The capacity of wireless links is assumed to be 5 Mbps [11]. The average end-to-end delay of both TRP-MRN and TRP under error-free wireless channel is investigated. Each node

**Table 1.** Simulation Parameters

| Parameter                                      | Value                 |
|------------------------------------------------|-----------------------|
| Number of MPs per cluster                      | 40                    |
| Transmission range                             | 50 m                  |
| Wireless transmission rate ( $R_{wireless}$ )  | 5 Mbps                |
| Wired transmission rate ( $R_{wired}$ )        | 50 Mbps               |
| Propagation speed( $V$ )                       | 200,000 km/s          |
| MAC frame data size ( $M_{body} + M_{h,fcs}$ ) | 351, 1351, 2351 bytes |
| PHY preamble size ( $P_{pre}$ )                | 18 bytes              |
| PHY header size ( $P_h$ )                      | 6 bytes               |
| Inquiry message size ( $H_{inq}$ )             | 46 bytes              |
| Reply message size ( $H_{rep}$ )               | 42 bytes              |

**Table 2.** Density of Nodes

| The radius of clusters (m) | The number of nodes in a cluster |
|----------------------------|----------------------------------|
| 100                        | 40                               |
| 150                        | 90                               |
| 200                        | 160                              |
| 250                        | 250                              |
| 300                        | 360                              |
| 350                        | 490                              |
| 400                        | 640                              |

is assumed to have the transmission range of 50 m. Once the topology of a WMN is formed, the routing tables are constructed. Each entry of the routing tables provides the shortest path to the root node and MRN based on the distance metrics. Note that we calculate the shortest paths, since our proposal is to compare the paths made from TRP and TRP-MRN by their hop-counts rather than to analyze other routing cost metrics for multi-hop wireless networks. The parameters defined in the IEEE 802.11a/s standard [3], [6] are adopted in the simulations (see Table 1). Let's assume that the wired line linking two root nodes is equivalent to one hop. In Fig. 7 and 8, we studied the performance of TRP-MRN and TRP when the radius of clusters varies from 150 m to 400 m. The radius of clusters and the number of nodes are shown in Table 2. They were selected so that the node density can be managed approximately constant, which would properly reflect the scalability of routing protocols. Scalability is a very important aspect of WMN routing protocol design. A good routing protocol should scale well in large networks. Wider topology causes longer cumulative distance between S-D pair, and thus increases the mean end-to-end delay. However, the delay of TRP-MRN is much smaller than that of TRP. Since the hop-count in TRP-MRN is reduced significantly by inquiries before data delivery, the end-to-end delay in TRP-MRN is lower than that of TRP. This is why the delay under TRP-MRN increases slowly as the topology becomes wider. As we can see, the delay of TRP-MRN is close to TRP when the topology is small.

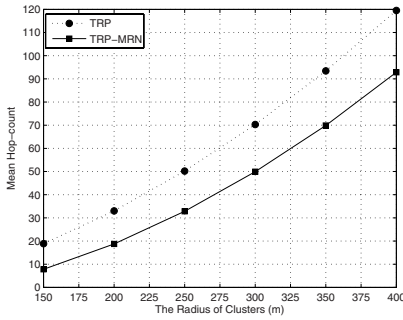


**Fig. 7.** Mean hop-count for various cluster sizes

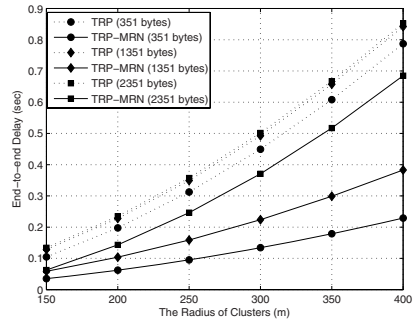


**Fig. 8.** Mean end-to-end delay for various cluster sizes

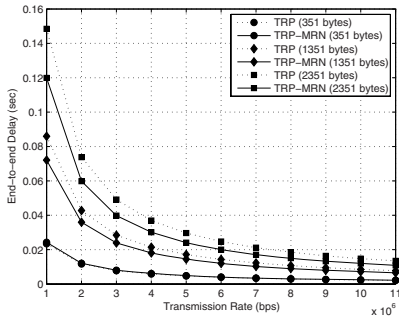


**Fig. 9.** Mean end-to-end delay for various transmission rates

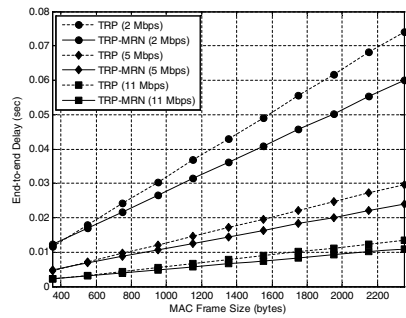


**Fig. 10.** Mean end-to-end delay for various MAC frame sizes

Fig. 9 shows the end-to-end delay with the different transmission rate. The lower the transmission rate, the shorter the delay is. As shown in the figure, TRP-MRN exhibits the delay gain of roughly 21.4% when the transmission rate is 1 Mbps and the MAC frame size is 2351 bytes. The end-to-end delay with MAC frame size is shown in Fig. 10, where we vary the MAC frame size. As one can see, the end-to-end delay is directly proportional to the size of MAC frames.

## 5 Conclusion

WMN routing protocols are different from those in wired networks while WMNs share common features with ad hoc networks. In spite of the existence of several available ad hoc routing protocols, it may not be appropriate that ad hoc routing protocols are used for WMNs without any modification. The IEEE 802.11s draft introduces TRP as one of routing protocols to be used in WMNs. TRP, however, forwards data on the only one active path, which requires changes in the TRP, which is rigid in selecting paths. In this paper, we have proposed TRP-MRN in WMNs. The TRP-MRN can not only

be applied to WMNs easily but also flexibility in route selection, and hence enhanced routing performance. We have shown that the TRP-MRN can effectively reduce the end-to-end delay particularly under low transmission rate and large MAC frame size conditions. In the future, we plan to provide a solution to decide how many MRNs are to be placed and where they should be placed for the optimum performance in various WMN topologies.

## References

1. R. Vaishampayan and J. J. Garcia-Luna-Aceves, "Robust Tree-based Multicasting in Ad hoc Networks," in Proc. of IEEE PCC, pp. 647-652, Apr. 2004.
2. C. E. 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, Feb. 1999.
3. IEEE 802.11 Working Group of the LAN/MAN Committee, Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Amendment: ESS Mesh Networking, IEEE P802.11s/D1.0, Dec. 2006.
4. I.F. Akyildiz and X. Wang, "A Survey on Wireless Mesh Networks," IEEE Communications Magazine, vol. 43, pp. S23-S30, Sept. 2005.
5. Becta, "Wireless Local Area Networks (WLAN)," <http://www.becta.org.uk/>, 2006.
6. IEEE Std. 802.11, Wireless LAN Media Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.
7. T. Tsai and J. Chen, "IEEE 802.11 MAC Protocol over Wireless Mesh Networks: Problems and Perspectives," in Proc. of AINA, vol. 2, pp. 60-63, 2005.
8. J. Jun and M. L. Sichitiu, "The Nominal Capacity of Wireless Mesh Networks," IEEE Wireless Communications, vol. 10, pp. 8-14, Oct. 2003.
9. I. F. Akyildiz, X. Wang, and Weilin Wang, "Wireless Mesh Networks: A Survey," <http://www.sciencedirect.com>, Mar. 2005.
10. M. Frodigh, S. Parkvall, C. Roobol, P. Johansson, and P. Larsson, "Future-generation Wireless Networks," IEEE Personal Communications Magazine, pp. 10-17, Oct. 2001.
11. J. Li, C. Blake, D. S. J. D. Couto, H. I. Lee, and R. Morris, "Capacity of Ad Hoc Wireless Networks," in Proc. of Mobile computing and networking, pp. 61-69, 2001.
12. A. Raniwala and T. Chiueh, "Architecture and Algorithms for an IEEE 802.11-based Multi-channel Wireless Mesh Network," in Proc. of IEEE INFOCOM, Mar. 2005.
13. J. J. Garcia-Luna-Aceves, M. Spohn, "Source-tree Routing in Wireless Networks," in Proc. of International Conference Network Protocols, pp. 273-282, Oct. 1999.
14. W. Jia, W. Tu, W. Zhao, and G. Xu, "Multi-shared-trees Based Multicast Routing Control Protocol Using Anycast Selection," The International Journal of Parallel, Emergent and Distributed Systems, vol. 20, no. 1, pp. 69-84, Mar. 2005.
15. P. Hsiao, A. Hwang, H. T. Kung, and D. Vlah "Load-balancing Routing for Wireless Access Networks," in Proc. of IEEE INFOCOM, vol. 2, pp. 986-995, Apr. 2001.
16. Sajama and Z. J. Haas, "Independent-tree Ad Hoc Multicast Routing (ITAMAR)," in Proc. of IEEE Vehicular Technology Conference, vol. 2, pp. 600-604, Oct. 2001.
17. T. Clausen (ed.) and P. Jacquet (ed) "Optimized Link State Routing Protocol," Internet Engineering Task Force (IETF) RFC 3626, Oct. 2003.
18. P. Jacquet, P. Mühlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized Link State Routing Protocol for Ad Hoc Networks," in Proc. of IEEE International Multi Topic Conference, pp. 62-68, Dec. 2001.

# An Interference-Robust Transmission Method for OFDMA Uplink

Ki-Chang Lee<sup>1</sup> and Tae-Hyun Moon<sup>2</sup>

<sup>1</sup> Dept. of Information & Communication Engineering, Hoseo University,  
Asan, Korea

`kclee@office.hoseo.ac.kr`

<sup>2</sup> Aheadcomm Inc., Sangdaewon-Dong, Jungwon-Gu, Sungnam City,  
Kyunggi-Do, Korea

`ceo@aheadcomm.co.kr`

**Abstract.** This paper deals with a simple way of reducing the multiple access interference (MAI) introduced by symbol timing misalignment in an OFDMA uplink system. To alleviate the MAI, we provide a new OFDMA transmitter equipping with a simple symbol repetition coupled with a cyclic time shift. It is found that this scheme can provide the MAI-robust reception without decreasing the bandwidth efficiency when the system is fully loaded.

## 1 Introduction

Orthogonal frequency division multiplexing (OFDM) has been primarily used in broadcasting systems or in single user links [1]-[4]. Recently, its generalization to multiple access systems has been proposed for broadband wireless multiple access systems, such as IEEE 802.16 wirelessMAN standard and DVB return channel terrestrial (DVB-RCT) [5]-[7]. A combination of OFDM with frequency-division multiple access (FDMA) called orthogonal frequency-division multiple access (OFDMA) is regarded as a promising solution for improving the performance of wireless uplink systems [5]-[7]. One of the main drawbacks is the multiple access interference (MAI) caused by symbol timing misalignments, which destroys the orthogonality between users and degrades the system performance especially for a typical OFDMA uplink scenario [8]. In cyclically prefixed OFDM, the MAI due the symbol timing misalignment can be mitigated by choosing an appropriate synchronization time instant over the cyclic prefix (CP) with the help of a sophisticated OFDM timing estimator [9].

In this paper, we propose a simple MAI-robust uplink system when OFDMA is used as an access scheme. We find that the timing misalignments caused by late-arriving symbol and early-arriving symbol of the interfering users are mitigated by using the inherent CP and two-symbol repetition coupled with a cyclic time shift (CTS), respectively, without sacrificing transmission efficiency by using a high-order modulation scheme.

## 2 OFDMA Uplink System

We consider the uplink of an OFDMA system employing  $N$  subcarriers and accommodating a maximum of  $K$  simultaneously active users. Each user transmits on a set of  $M = N/K$  assigned subcarriers. The OFDMA uplink system with  $K$  users assigns the  $d$ -th user the subband  $\mathcal{B}_d$  defined as  $\mathcal{B}_d = \{k | M(d-1) \leq k \leq M \cdot d - 1\}$ , where  $k$  is a subcarrier index and  $M$  is the number of banded subcarriers assigned to each user. For the  $d$ -th user and  $k \in \mathcal{B}_d$ , the data symbol  $\{X_l^d(k)\}$  are processed by the IDFT during the symbol duration  $T$ . For transmission, the last  $N_g$  samples are preceded as the CP and  $N + N_g$  samples are transmitted during the duration  $T_S = T + T_g$ , where  $N_g$  is the length of the CP and  $T_g$  is its duration.

The transmitted signal by each user goes through multipath fading channel and the received signal during a duration  $T_S$  can be expressed as

$$y(t) = \sum_{d=1}^K \sum_{l=-\infty}^{\infty} \sum_{k \in \mathcal{B}_d} H_l^d(k) X_l^d(k) r(t - lT_S - \tau_d) e^{j2\pi k g(t - \tau_d)/T} + w(t) \quad (1)$$

where  $w(t)$  is the zero-mean additive white Gaussian noise (AWGN) with two-sided power spectral density of  $N_0/2$ ,  $\tau_d$  is the symbol timing error for the  $d$ -th user,  $H_l^d(k)$  represents the channel transfer function of the  $d$ -th user during the  $l$ -th symbol period,  $r(t)$  is assumed as a rectangular function over  $[-T_g, T)$ , and  $g(t)$  is defined as

$$g(t) = \begin{cases} (t + T) - lT_S, & lT_S - T_g \leq t < lT_S \\ t - lT_S, & lT_S \leq t < lT_S + T. \end{cases} \quad (2)$$

At the receiver, let us assume that the symbol timing error  $\tau_j$  for the  $j$ -th desired user is perfectly recovered. Then, the reconstructed  $p$ -th output for the  $j$ -th user is given by

$$\begin{aligned} Y_l^j(p) &= \frac{1}{T} \int_{lT_S + \tau_j}^{lT_S + \tau_j + T} y(t) e^{-j2\pi p(t - lT_S - \tau_j)/T} dt \\ &= H_l^j(p) X_l^j(p) + M_l^j(p) + W_l(p) \end{aligned} \quad (3)$$

where  $p \in \mathcal{B}_j$ ,  $W_l(p)$  is the AWGN with mean zero and variance  $\sigma_W^2$ , and  $M_l^j(p)$  is the MAI component of other interfering users. Here, let  $T_{d,j}$  denote the symbol timing misalignment of the  $d$ -th desired user with respect to the  $j$ -th interfering user. Then,  $T_{d,j} = (\tau_d - \tau_j) - \lfloor (\tau_d - \tau_j)/T_S \rfloor \cdot T_S$ , where  $\lfloor x \rfloor = \text{first integer} < x$ . As discussed in [10][11], the sum of MAI caused by the symbol timing misalignment



is negligible when  $0 \leq T_{d,j} < T_g$ . So, the MAI term over  $T_g \leq T_{d,j} < T_S$  is calculated by

$$\begin{aligned} M_l^j(p) = & \frac{1}{T} \sum_{\substack{d=1 \\ d \neq j}}^K \sum_{\substack{k \in \mathcal{B}_d \\ k \neq p}} H_{l-\tau_{d,j}-1}^d(k) X_{l-\tau_{d,j}-1}^d(k) \Delta_{d,j,1}(k; p) \\ & + \frac{1}{T} \sum_{\substack{d=1 \\ d \neq j}}^K \sum_{\substack{k \in \mathcal{B}_d \\ k \neq p}} H_{l-\tau_{d,j}}^d(k) X_{l-\tau_{d,j}}^d(k) \Delta_{d,j,2}(k; p) \end{aligned} \quad (4)$$

with

$$\Delta_{d,j,1}(k; p) = \frac{e^{j2\pi k(T_g - T_{d,j})/T} [e^{j2\pi(k-p)(T_{d,j} - T_g)/T} - 1]}{j2\pi(k-p)/T} \quad (5)$$

and

$$\Delta_{d,j,2}(k; p) = \frac{e^{j2\pi k(-T_{d,j})/T} [1 - e^{j2\pi(k-p)(T_{d,j} - T_g)/T}]}{j2\pi(k-p)/T} \quad (6)$$

where  $\tau_{d,j} = \lfloor (\tau_d - \tau_j)/T_S \rfloor$  and  $\Delta_{d,j,i}(k; p)$  is the partial integral term.

### 3 MAI-Robust Transmission for OFDMA Uplink System

In this section, we provide a modified OFDMA uplink system equipping with CTS to mitigate the MAI incurred by early-arriving symbols without loss of the throughput. To do this, the  $(l+1)$ -th OFDM symbol  $x_{l+1}^d(n)$  is formulated from the  $l$ -th frequency-domain OFDM symbol  $X_l^d(k)$  as follows

$$\begin{aligned} x_{l+1}^d(n) &= \sum_{k \in \mathcal{B}_d} X_l^d(k) e^{j2\pi k(n+N_g)/N} \\ &= x_l^d(n + N_g), \quad 0 \leq n < N \end{aligned} \quad (7)$$

which results in  $(N + N_g)$  times cyclic shift of  $x_l^d(n)$  and is similar to multiband OFDM using time-domain spreading developed in [12]. With this provision, the CP of the  $(l+1)$ -th OFDM symbol can be regarded as a cyclic suffix (CS) of the  $l$ -th OFDM symbol. Then, the  $(l+1)$ -th transmitted symbol in frequency domain is given by

$$X_{l+1}^d(k) = X_l^d(k) e^{j2\pi k N_g/N}, \quad k \in \mathcal{B}_d. \quad (8)$$

When a perfect channel equalization is assumed and a pre-advancement of DFT window timing by  $N_g$  samples is used for  $(l+1)$ -th OFDM symbol, the decision variable for the  $j$ -th user's  $p$ -th subcarrier is expressed as

$$\begin{aligned} \hat{X}_l^j(p) &= G_l^j(p) \left( Y_l^j(p) H_l^{j*}(p) + Y_{l+1}^j(p) H_{l+1}^{j*}(p) \right) \\ &= \hat{G}_l^j(p) X_l^j(p) + \hat{M}_l^j(p) + \hat{W}_l(p) \end{aligned} \quad (9)$$

with

$$\hat{M}_l^j(p) = G_l^j(p)(M_l^j(p)H_l^{j*}(p) + M_{l+1}^j(p)H_{l+1}^{j*}(p)) \quad (10)$$

and

$$\hat{W}_l(p) = G_l^j(p)(W_l(p)H_l^{j*}(p) + W_{l+1}(p)H_{l+1}^{j*}(p)) \quad (11)$$

where  $p \in \mathcal{B}_j$  and  $(\cdot)^*$  denotes the complex conjugate. In Eqns. (10) and (11), the gain factor  $G_l^j(p) = (|H_l^j(p)|^2 + |H_{l+1}^j(p)|^2)^{-1/2}$  is chosen such that  $E[\hat{W}_l(p)\hat{W}_l^*(p)]$  is  $\sigma_W^2$ . If we assume that  $H_l^j(p) = H_{l+1}^j(p)$ , the decision variable becomes

$$\hat{X}_l^j(p) = \sqrt{2}|H_l^j(p)|X_l^j(p) + \hat{M}_l^j(p) + \hat{W}_l(p) \quad (12)$$

with  $\hat{G}_l^j = \sqrt{2}|H_l^j(p)|$ .

## 4 Performance Analysis

We first consider  $M_l^j(p)$  when the uplink signals of other interfering users arrive within a range of  $|T_{d,j}| < T_g/4$ , as specified in [6]. For the notational convenience, let a set of index of late-arriving users be  $\mathcal{K}_L$  and a set of index of early-arriving users be  $\mathcal{K}_E$ . By using the results done in [10][11], the timing misalignment introduced by late-arriving symbol of other users ( $d \in \mathcal{K}_L$ ) can be mitigated by the CP (i.e., MAI = 0). On the other hand, MAI term of the remaining early-arriving users for  $d \in \mathcal{K}_E$  can be represented as

$$\begin{aligned} M_l^j(p) = & \frac{1}{T} \sum_{\substack{d \in \mathcal{K}_E \\ d \neq j}} \sum_{\substack{k \in \mathcal{B}_d \\ k \neq p}} H_l^d(k) X_l^d(k) \Delta_{d,j,1}(k; p) \\ & + \frac{1}{T} \sum_{\substack{d \in \mathcal{K}_E \\ d \neq j}} \sum_{\substack{k \in \mathcal{B}_d \\ k \neq p}} H_{l+1}^d(k) X_{l+1}^d(k) \Delta_{d,j,2}(k; p) \end{aligned} \quad (13)$$

where  $\tau_{d,j} = -1$  is used for late-arriving case and two partial correlation terms have a relation as follows

$$\Delta_{d,j,1}(k; p) = -e^{j2\pi k T_g/T} \cdot \Delta_{d,j,2}(k; p). \quad (14)$$

Then, recalling from Eqn. (12) that  $H_l^j(k) = H_{l+1}^j(k)$ , and from Eqn. (8) that  $X_{l+1}^d(k) = X_l^d(k)e^{j2\pi k N_g/N}$ , we can effortlessly find that MAI term of the remaining early-arriving users expressed in Eqn. (13) becomes zero, i.e.  $M_l^j(p) = 0$ . Similarly, we can obtain the same result for  $M_{l+1}^j(p)$  with the aid of a pre-advancement of DFT window timing with the length of  $N_g$  samples in demodulating the  $(l+1)$ -th OFDM symbol. If we assume that the timing synchronization can provide a portion of the samples that is influenced by one transmitted symbol only within the samples indexes denoted by

$$\Xi = \{\max[-N_g + \tau_{\max}, -3N_g/4], \dots, -1, 0, 1, \dots, N_g/4\}, \quad (15)$$

the orthogonality among the subcarrier will not be destroyed and MAI term in Eqn. (12) may be disappeared, where  $\tau_{\max}$  is the maximum channel delay spread.

Using the notations  $\bar{\Omega} = E[|H_l^j(p)|^2] = E[\Omega]$  in Eqn. (12), the SNR gain per bit can be expressed as  $\bar{\gamma}_R = 2E_b\bar{\Omega}/N_0 = 2 \cdot \bar{\gamma}$ , where  $\bar{\gamma}$  is the average SNR of plain OFDMA system without symbol repetition. For 16 square QAM with Gray coding [13], the BER for instantaneous SNR defined as  $\gamma = E_b\Omega/N_0$  is found to be exactly given by

$$P_{b,16QAM}(\gamma) = \frac{3}{8}\text{erfc}\left(\sqrt{\frac{4}{5}\gamma}\right) + \frac{1}{4}\text{erfc}\left(\sqrt{\frac{36}{5}\gamma}\right) - \frac{1}{8}\text{erfc}\left(\sqrt{20\gamma}\right) \quad (16)$$

where  $\text{erfc}(x) \triangleq 2 \int_x^\infty e^{-t^2} dt / \sqrt{\pi}$ . Because the probability density function of  $\gamma$  under Rayleigh fading is given by  $f(\gamma) = (1/\bar{\gamma})e^{-\gamma/\bar{\gamma}}$ , the average BER of 16 square QAM becomes

$$P_{b,16QAM} = \frac{1}{2} - \frac{3}{8} \left(1 + \frac{5}{4\bar{\gamma}}\right)^{-1/2} - \frac{1}{4} \left(1 + \frac{5}{36\bar{\gamma}}\right)^{-1/2} + \frac{1}{8} \left(1 + \frac{1}{20\bar{\gamma}}\right)^{-1/2}. \quad (17)$$

When there are  $K$  users transmitting simultaneously, the normalized throughput is defined as

$$T_n = \frac{1}{N} \sum_{d=1}^K [1 - P_e(d)] \cdot M \cdot N_b \cdot r_c \quad (18)$$

where  $P_e(d)$  is the packet error rate (PER) of the  $d$ -th user,  $N_b$  is defined as the average number of effective bits over two consecutive modulated symbols, and  $r_c$  is the coding rate. By assuming that  $P_e = P_e(d)$  is identical for all users, the normalized throughput can be simplified as  $T_n = [1 - P_e] \cdot N_b \cdot r_c$ , which has a unit in the number of bits per subcarrier. As discussed above, the CTS-OFDMA system is modulated by 16QAM while ordinary OFDMA system is modulated by QPSK. With this provision,  $N_b = 2$  for both OFDMA systems. In this case, the PER of the CTS-OFDMA system is defined as

$$P_e = 1 - (1 - P_{b,16QAM})^L \quad (19)$$

where  $L$  is the length of a packet per user.

## 5 Results and Discussions

The entire BW of 20MHz is divided into 256 subchannels and each user occupies 16 successive subcarrier (i.e.  $N = 256$ ,  $K = 16$ , and  $M = 16$ ). And,  $N_g = 32$  and the 8-th user is considered as the desired user. To guarantee the same throughput, QPSK is used for ordinary OFDMA signal, while 16QAM is used for the new one. The multipath channel is comprised of 16 paths, each of which is an independent Rayleigh fading channel having an exponential power delay

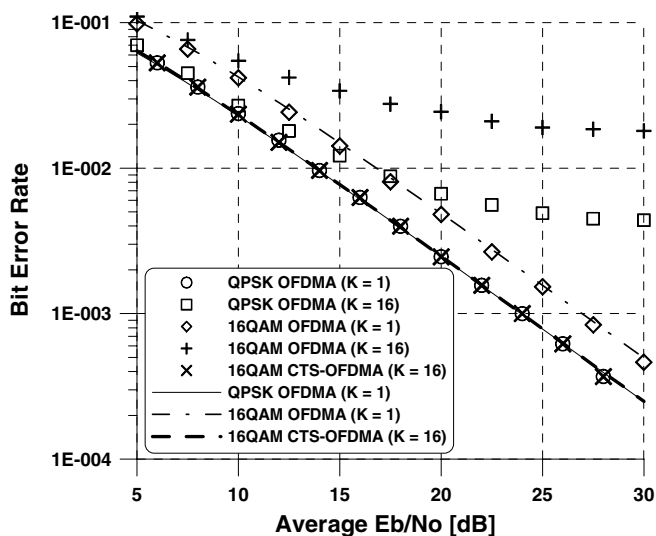


**Fig. 1.** BER performance versus  $E_b/N_0$  for both ordinary OFDMA and new CTS-OFDMA uplink systems (symbols - simulation, lines - analysis)

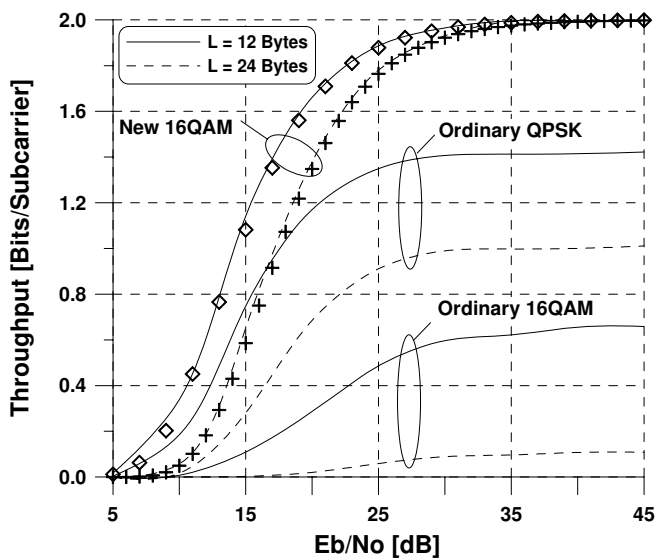


**Fig. 2.** Normalized throughput performance versus  $E_b/N_0$  for both ordinary OFDMA and CTS-OFDMA uplink systems (symbols - analysis, lines - simulation)

profile. In the following examples, the timing requirement is that that all uplink OFDM symbols should arrive at the base station within an accuracy of  $\pm 25\%$  of the minimum guard-interval [6] and perfect channel equalization is assumed at the receiver.

Figure 1 shows the BER comparison between ordinary OFDMA and new CTS-OFDMA systems. It is clear that the performance of the new multiuser OFDMA with 16QAM is parallel to that of ordinary single-user OFDMA with QPSK. At the same time, the analytic curve according to Eqn. (17) is very close to the simulated curve, and they are identical to the analytical results done in [14].

Figure 2 shows the normalized throughput performance for both ordinary and new CTS-OFDMA uplink systems when  $K = 16$  is assumed. In this example, the coding rate  $r_c = 1$  is used because channel coding is not employed. In terms of normalized throughput, CTS-OFDMA with 16QAM, OFDMA with QPSK, and OFDMA with 16QAM achieve 2.0, 1.4, and 0.7 bits/subcarrier respectively when  $\text{SNR} \rightarrow \infty$  and the length of packet/user is 12bytes. This stems primarily from the fact that the BER of new multiuser OFDMA system comes close to that of a single-user case, while ordinary multiuser OFDMA system experiences the error floor caused by MAI from other interfering users, as confirmed by Fig. 1. To improve the throughput of plain OFDMA system, a channel coding is inevitable. We can also find that the analytic throughput performance of the new OFDMA system according to Eqn. (18) is very close to the simulated throughput performance.

## 6 Conclusion

One of the main disadvantages of OFDMA uplink systems is the symbol timing misalignment caused by multiple user access. In order to resolve this problem, this paper has provided a new OFDMA uplink system using a symbol repetition equipped with a cyclic time shifting. We can see that the timing misalignments caused by early-arriving symbol and late-arriving symbol of the interfering users are mitigated by using the proposed symbol repetition and the inherent CP, respectively.

## References

1. R. van Nee, G. Awater, M. Morikura, H. Takanashi, M. Webster, and K. W. Halford, "New high-rate wireless LAN standards," *IEEE Commun. Mag.*, vol. 37, no. 12, pp. 82-88, December 1999.
2. A. Doufexi, S. Armour, M. Butler, A. Nix, D. Bull, J. McGeehan, and P. Karlsson, "A comparison of the HIPERLAN/2 and IEEE802.11a wireless LAN standards," *IEEE Commun. Mag.*, vol. 40, no. 5, pp. 172-180, May 2002.
3. ETSI ETS 300 401, "Radio broadcasting systems: digital audio broadcasting (DAB) to mobile, portable and fixed receivers," ETSI, Tech. Rep., February 1995.
4. ETSI ETS 300 744, "Digital video broadcasting (DVB): frame structure, channel coding and modulation for digital terrestrial television (DVB-T)," ETSI, Tech. Rep., March 1997.

5. I. Koffman and V. Roman, "Broadband wireless access solutions based on OFDM access in IEEE 802.16," *IEEE Commun. Mag.*, vol. 40, pp. 96-103, April 2002.
6. "IEEE draft standard for local and metropolitan area network-part 16: Air interface for fixed broadband wireless access systems - medium access control modifications and additional physical layer specifications for 2-11 GHz," IEEE LAN MAN Standards Committee, 2002.
7. ETSI ETS 301 958, "Digital Video Broadcasting (DVB); interaction channel for digital terrestrial television (RCT) incorporating multiple access OFDM," ETSI, Tech. Rep., March 2002.
8. R. Nogueroles, M. Bossert, A. Donder, and V. Zyablov, "Performance of a random OFDMA system for mobile communications," *Proc. of International Zurich Seminar on Broadband Communications*, pp. 37-43, February 1998.
9. A. M. Tonello, N. Laurenti, S. Pupolin, "Analysis of the uplink of an asynchronous multi-user DMT OFDMA system impaired by time offsets, frequency offsets, and multi-path fading," in *Proc. of VTC2000*, pp. 1094-1099, September 2000.
10. M. S. El-Tanany, Yiyang Wu, and László Háy, "OFDM uplink for interactive broadband wireless: analysis and simulation in the presence of carrier, clock and timing errors," *IEEE Trans. on Broadcasting*, vol. 47, no. 1, pp. 3-19, March 2001.
11. M. Park, K. Ko, H. Yoo, and D. Hong, "Performance analysis of OFDMA uplink systems with symbol timing misalignment," *IEEE Commun. Letters*, vol. 7, no. 8, pp. 376-378, August 2003.
12. A. Batra, *MultiBand OFDM physical layer specification*, MultiBand OFDM Alliance Special Interest Group, 2005.
13. K. Cho and D. Yoon, "On the general BER expression of one- and two-dimensional amplitude modulations," *IEEE Trans. Commun.*, vol. 50, pp. 1074-1080, July 2002.
14. F. Adachi, "BER analysis of 2PSK, 4PSK, and 16QAM with decision feedback channel estimation in frequency-selective slow Rayleigh fading," *IEEE Trans. Vehicular Technology*, vol. 48, no. 5, pp. 1563-1572, September 1999.

# An Efficient Heterogeneous Wireless Access Based on Region and Time Partitioning in D-TDD OFDM Systems

Nak-Myeong Kim<sup>1</sup>, Hee-Jeong Chung<sup>1</sup>, Mee-Ran Kim<sup>2</sup>, and Hye-In Yu<sup>1</sup>

<sup>1</sup>Dept. of Information Electronics Engineering, Ewha Womans University, Seoul, Korea  
nmkim@ewha.ac.kr, {hjchung, hyeinyu}@ewhain.net

<sup>2</sup>Electronics and Telecommunications Research Institute, Daejeon, Korea  
newface82@etri.re.kr

**Abstract.** Under dynamic time division duplexing (D-TDD) architecture in OFDM cellular systems, the inter-BS and inter-MS interference is inevitable during the cross time slot (CTS) period, and this interference seriously degrades the wireless access system performance. To mitigate such interference, we propose a region and time partitioning D-TDD architecture for OFDM systems. Each time slot in the CTS period is split into several minislots, and then each cell is divided into as many regions as the number of minislots per time slot. We then assign each minislot only to the users in its corresponding region. On top of such architecture, which inherently separates interfering entities farther from each other, we design a robust time slot allocation scheme considering the channel status of each mobile station. By computer simulation, it has been verified that the proposed scheme outperforms conventional time slot allocation methods in both the outage probability and the bandwidth efficiency.

**Keywords:** Orthogonal frequency division multiplexing, dynamic time division duplexing, time slot allocation, cross time slot.

## 1 Introduction

As the demands for multimedia services in the next generation wireless communication systems increase, the amount of traffic in downlink (DL) tends to exceed the amount of traffic in uplink (UL). To support asymmetric traffic efficiently, time division duplexing (TDD) scheme became a preferred duplexing method in the systems such as IEEE 802.16, WiBro, and UTRA-TDD. Especially in the dynamic TDD (D-TDD) mode, the switching point, which is the boundary between DL slots and UL slots, is adaptively changed according to the DL/UL traffic variation.

In the D-TDD mode, however, the switching points of the cells depend on the dynamic nature of the traffic in each cell, so the cross time slot (CTS) period among neighboring cells is inevitable. The CTS period seen by a reference cell is defined as the set of time slots in which at least one of the neighboring cells is transmitting in the opposite direction. In the CTS period, the inter-MS co-channel interference (CCI),  $I_{MM}$ , and the inter-BS CCI,  $I_{BB}$ , become serious, and they limit the performance of the D-TDD mode significantly. When the UL users allocated in the CTS period in the

neighboring cells are close to a DL user allocated in the CTS period in the reference cell,  $I_{MM}$  towards the DL user increases. On the other hand, when any of the neighboring cells are transmitting in the DL, the reference cell base station (BS) receives  $I_{BB}$  from those cells and this limits the performance of UL reception [1].

Recently, there have been several researches on time slot allocation (TSA) in the D-TDD mode. In [2], the  $\max\{\text{SIR}\}$  and  $\max\{\text{SIR}\}$  TSA algorithms are proposed to suppress the CCI. The  $\max\{\text{SIR}\}$  algorithm is to maximize the minimum signal to interference ratio (SIR) by introducing extra UL time slot region, while the  $\max\{\text{SIR}\}$  algorithm is a simpler version of it. Since these algorithms utilize the estimated CCI for time slot allocation, the outage performance has been improved.

In [3], the relative distances from a mobile station (MS) in the reference cell to the neighboring BSs are considered in allocating time slots. If the propagation path gains from the MS to the BSs in the neighboring cells are less than a given threshold, the resources in the CTS period can be assigned to the MS. This method, however, only considers the CCI from the BSs, excluding those from the MSs in the neighboring cells. Also in [4], the TSA algorithm based on location has been proposed, which allocates the CTS period closest to the switching point to the nearest user from the BS. Among these algorithms, the performance of the TSA based on path gain and the performance of the TSA based on location are better than the others.

In this paper, we propose an efficient heterogeneous wireless access based on region and time partitioning (RTP) D-TDD architecture for OFDM systems, in which the resource allocation is performed jointly in frequency, time and spatial domains. We introduce the concept of minislots in the CTS period. We divide each time slot in the CTS period into several minislots, and then define as many regions in each cell. Minislots are exclusively allocated to the users in the corresponding region, so that the CCI in the CTS period is reduced. On top of such architecture, which inherently separates the interfering entities farther from each other, we propose a robust time slot allocation (TSA) scheme to minimize the outage probability resulting from the inter-cell interference. By utilizing link gains as a function of the estimated carrier to interference plus noise ratio (CINR) for the MSs, the proposed TSA algorithm can minimize the inter-cell interference by allocating the users having lower link gains to the minislots farther from the switching point.

The rest of the paper is organized as follows. In section 2, the proposed system structure considered in this paper is described and section 3 introduces the proposed robust time slot allocation scheme in detail. In section 4 the simulation results are discussed and the conclusion is made in section 5.

## 2 Proposed System Structure

A TDD frame is composed of a group of time slots for DL traffic and another group of time slots for UL traffic in sequence. In the D-TDD mode, the size of each group is dynamically changed according to the variation in the UL/DL traffic pattern.

Because of the difference in traffic demand in each cell, the CTS period is inevitably generated in any cell. In the CTS period, the multiple access interference from adjacent cells drastically increases, since the directions of transmissions are different from each other.



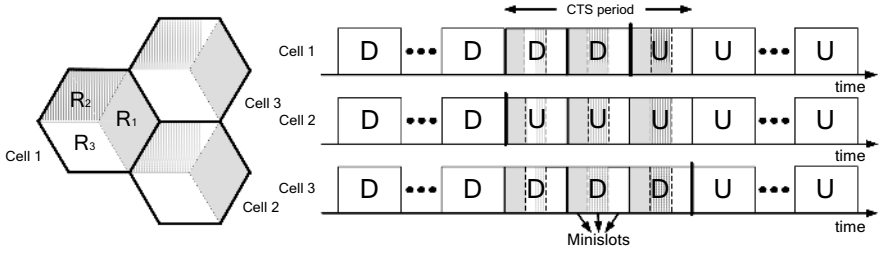


Fig. 1. The proposed RTP D-TDD architecture for the 3-minislot case

In this paper, we propose the RTP D-TDD architecture to mitigate the CCI in the CTS period. We divide each time slot in the CTS period into several minislots, and then define several regions in each cell in the same number as the number of the minislots per time slot. The regions are assumed to have equal degrees of angle. Each minislot is then exclusively mapped to its corresponding region in the cell. Fig. 1 shows an example of the proposed RTP D-TDD architecture when the number of minislots per time slot in the CTS period is three. Under the proposed architecture, users in each region transmit or receive data only in its corresponding minislots if any portion of their traffic has been assigned in the CTS period. For example, in fig. 1, the first minislot of each time slot in the CTS period is allocated to the users in the region of  $R_1$ .

By applying the RTP D-TDD architecture, regions in each cell are mutually exclusively assigned to different minislots in each time slot in the CTS period. So, each region is free from the  $I_{MM}$  from the adjacent regions since the only CCI affecting one region in the reference cell is the CCI from the regions which are located at the same position in the neighboring cells and thus assigned to the same minislots. Therefore, the distances between the receivers in the reference cell and the interfering transmitters in the neighboring cells become longer, and this results in the reduction of the  $I_{MM}$  in the CTS period in downlink. Because of the segregation in time, the proposed architecture can operate along with the sectored antenna system, maintaining the frequency reuse factor as 1. So, if we adopt the sectored antenna system, the interference such as  $I_{BB}$  or  $I_{BM}$  from neighboring BSs can be reduced considerably.

### 3 Robust Time Slot Allocation Based on Region and Time Partitioning

Along with the proposed RTP D-TDD architecture, we propose a TSA scheme which minimizes the inter-cell interference. The locations and the channel quality information of users in the reference cell are assumed to be known to the serving BS in the cell. We also assume that the BSs share their time slot state information among the neighboring BSs so that the reference cell BS can dynamically analyze the characteristics of the CTS period of the cell.

Once, the BS gathers the time slot state information from the neighboring BSs, it verifies the existence of the CTS period, and run the proposed TSA as follows. The

proposed TSA based on RTP D-TDD architecture is composed of four separate steps; time partitioning step, region partitioning step, calculation of link gains, and allocating time slots.

### 1. Step1: Time partitioning

Let  $S$  and  $J$  denote the number of time slots in a frame and the number of neighboring cells, respectively. The time slot state information for the  $s$ th time slot of the reference cell is defined as  $TSSI_{ref,s}$ , and the time slot state information for the  $s$ th time slot of the  $j$ th neighboring cell is defined as  $TSSI_{j,s}$ . Time slot state information is encoded by -1, 1 or 0, according to DL, UL, and unassigned, respectively. Then the CTS period in the reference cell can be found by the following procedure.

```

for (  $s = 1 : S$  ), do
     $flag = 0$ 
    for (  $j = 1 : J$  ), do
        if  $TSSI_{ref,s} \cdot TSSI_{j,s} == -1$ ,
             $flag = 1$ 
        end if
    end for
    if  $flag \neq 0$ ,
        the  $s$ th time slot is included in the CTS period.
    end if
end for

```

We then divide each timeslot included in the CTS period into a certain number of minislots. Since the CTS period in the reference cell includes both the DL and the UL timeslots, the  $s$ th timeslot included in the CTS period is then defined according to the type of link as

$$T_s^D = \{M_{s,1}^D, M_{s,2}^D, \dots, M_{s,R}^D\}, \quad s \in \text{CTS period}, \quad (1)$$

or

$$T_s^U = \{M_{s,1}^U, M_{s,2}^U, \dots, M_{s,R}^U\}, \quad s \in \text{CTS period}, \quad (2)$$

where  $M_{s,r}^D$  is the  $r$ th minislot in the  $s$ th DL time slot,  $R$  denotes the number of minislots defined in CTS timeslot. The notations with superscript U imply the corresponding parameters for the UL.

### 2. Step2: Region partitioning

Let  $(x_k, y_k)$  define the location of the  $k$ th user. Then, the users in the reference cell are partitioned according to their locations into one of  $R$  regions in the cell.

Note that  $R$  is the same as the number of minislots in the CTS timeslot. The set of users located in the  $r$ th region is then represented as

$$U_r = \{k \mid (x_k, y_k) \in \text{region } r\}, \quad r = 1, 2, \dots, R. \quad (3)$$

### 3. Step3: Link gain calculation

The next step in the proposed TSA is to evaluate the communication channel for each user according to the CINR estimate. We define a link gain function of the communication channel for the each user as

$$G_{r,k} \square H_{r,k}^2 - PL_{r,k} - \hat{\sigma}_{I,r,k}^2 - \sigma_{N,r,k}^2 \text{ (dB)}, \quad k = 1, 2, \dots, |U_r| \text{ and } r = 1, 2, \dots, R, \quad (4)$$

where  $PL_{r,k}$  represents the pathloss of the  $k$ th user in the  $r$ th region and  $\hat{\sigma}_{I,r,k}^2$  denotes the estimated power of the interference to the  $k$ th user in the  $r$ th region,  $|U_r|$  denotes the number of users in region  $r$ , and

$$H_{r,k} = \sqrt{\frac{1}{N} \cdot \sum_{n=1}^N |H_{r,k,n}|^2}, \quad (5)$$

where  $H_{r,k,n}$  represents the frequency response of the  $n$ th subcarrier for the  $k$ th user in the  $r$ th region, and  $N$  is the total number of subcarriers. Here the pathloss,  $PL_{r,k}$  is defined by

$$PL_{r,k} = PL(d_0) + 10\gamma \log_{10} \left( \frac{d_{r,k}}{d_0} \right) + X_\sigma \text{ (dB)}, \quad (6)$$

where  $d_0$  represents a reference distance, and  $d_{r,k}$  is the distance between the BS and the  $k$ th user,  $\gamma$  is the pathloss exponent, and  $X_\sigma$  represents a lognormal distributed random variable for shadow fading with standard deviation,  $\sigma$  [7]. The interference power for the  $k$ th user in the  $r$ th region is estimated as

$$\hat{\sigma}_{I,r,k}^2 = (1 - \varepsilon) \cdot \sum_{j=1}^J \delta_j \cdot \sigma_{(Ij),r,k}^2, \quad \text{where } \delta_j = \begin{cases} 1, & \text{if } TSSI_j \neq TSSI_{ref} \\ 0, & \text{otherwise} \end{cases}, \quad (7)$$

and where  $\sigma_{(Ij),r,k}^2$  denotes the interference power from the  $j$ th neighboring cell, and  $\varepsilon$  represents the estimation error.

When the above procedure is repeated for all the users in each region, we finally get the lists of link gains for the DL and those for the UL as follows;

$$G_r^D = \{G_{r,k}^D\} = \{G_{r,1}^D, G_{r,2}^D, \dots, G_{r,|U_r|}^D\}, \quad (8)$$

and

$$G_r^U = \{G_{r,k}^U\} = \{G_{r,1}^U, G_{r,2}^U, \dots, G_{r,|U_r|}^U\}. \quad (9)$$

#### 4. Step4: TSA based on RTP D-TDD architecture

The final step is to allocate time slots to the users in each region so that the inter-cell interference can be minimized. It is assumed that the BS has an estimate of the total amount of DL traffic, so the indices of the starting time slots for the actual traffic loading for the DL time slots and the UL time slots,  $s_f^D$  and  $s_f^U$ , have been calculated. We also define  $S^D$  and  $S^U$  to denote the number of DL slots in the CTS period and the number of UL slots in the CTS period, respectively.

*Downlink TSA:*

```

initialization :  $U_r = \{1, 2, \dots, |U_r|\}$ 
for  $r = 1 : R$ , do
  while ( $\{G_{r,k}\} \neq null$ , and minislots remain), do
     $s = s_f^D$ 
     $\tilde{k} = \arg \max_k \{G_{r,k}\}$ 
    while ( $\tilde{k}$  not assigned to any minislot), do
      if  $s < (s_f^D - S^D + 1)$ , break;
      else if  $M_{s,r}^D$  is unassigned,
         $\tilde{k}$  is assigned to  $M_{s,r}^D$ .
        update that  $M_{s,r}^D$  is assigned.
         $U_r = U_r - \{\tilde{k}\}$ 
      else,  $s = s - 1$ 
    end while
     $\{G_{r,k}\} = \{G_{r,k}\} - G_{r,\tilde{k}}$ 
  end while
  if ( $U_r \neq null$ , and non-CTS time slots remain),
    Assign the non-CTS time slots randomly to the users in  $U_r$ .
    update the time slot assignment.
  end if
end for

```

*Uplink TSA:*

The UL TSA follows basically same procedure as the DL TSA except that the parameters are taken from the UL estimates for the  $k$ th user in the  $r$ th region.

As an example, we consider the proposed TSA based on RTP D-TDD architecture in two cell environment in fig. 2. Since 3 minislots are defined in each time slot in the

CTS period, the cell is divided into 3 mutually exclusive regions. In the figure, the numbers in each region represent the user locations. The shade for the background of the numbers represents the level of link gains, from white as the highest, to dark gray as the lowest level of link gains. As is shown in figure, the numbers with white background are allocated to the minislots near the switching point due to higher link gains. On the other hand, the numbers with gray background are allocated to the minislots farther from the switching point.

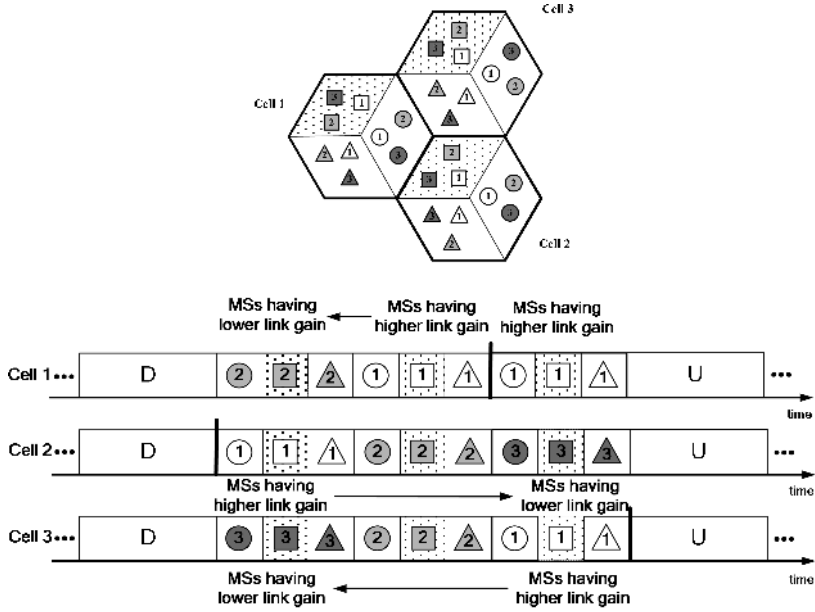


**Fig. 2.** An example of the proposed minislot allocation based on RTP D-TDD architecture

Now, for the performance evaluation, we utilize the CINR outage probability. To define the average CINR outage probability, we first define the CINR for the  $n$ th subcarrier of the  $k$ th user as

$$CINR_{k,n} = \frac{C_{k,n}}{\sum_{j=1}^J I_{j,k,n} + N_{k,n}}, \quad (10)$$

where  $C_{k,n}$  denotes the power of the  $n$ th subcarrier in the received signal for the  $k$ th user,  $I_{j,k,n}$  denotes the interference from the  $j$ th neighboring cell, and  $N_{k,n}$  represents the noise power. Then, the CINR outage probability of the  $n$ th subcarrier for the  $k$ th user,  $P_{OUT,k,n}(Th)$ , is defined as the probability that the CINR of a certain subcarrier is less than a given threshold,  $Th$ , is defined as

$$P_{OUT,k,n}(Th) = \frac{1}{L} \cdot \sum_{l=1}^L \delta_{k,n,l}, \quad \text{where } \delta_{k,n,l} = \begin{cases} 1, & \text{if } CINR_{k,n,l} < Th \\ 0, & \text{otherwise} \end{cases}, \quad (11)$$

and where  $L$  denotes the number of time samples to take the average.

Now the average CINR outage probability,  $P_{OUT}(Th)$ , is defined as

$$P_{OUT}(Th) = \frac{1}{|S \cdot k|} \sum_{k=1}^K \sum_{n \in S} P_{OUT,k,n}(Th), \quad (12)$$

where  $S$  denotes the set of subcarriers allocated to the  $k$ th user.

In this paper, the average CINR outage probability for each user is calculated over the CTS period, including both the DL timeslots and the UL timeslots. So the set  $S$  in this paper is defined as the set of time slots in the CTS period.

## 4 Performance Evaluation

The performance of the proposed TSA based on the RTP D-TDD architecture is evaluated using MATLAB under Pentium-IV personal computers. A random frequency hopping OFDMA is considered as the multiple access scheme and the transmission power for each subcarrier is assumed to be equal. The simulation assumes a 7-cell environment with 150 users in each cell. The center cell is designated as the reference cell. User locations in each cell are assumed to be uniformly distributed, but for the reference cell, both uniform and non-uniform distributions of user locations are investigated. The simulation was performed under the multipath fading channel described in Pedestrian A model in [6]. It is also assumed that a time slot consists of 6 symbols, so that 3 or 6 separate regions can be defined in each cell. Table 1 summarizes the system parameters in detail. In order to investigate the possibility of varying conditions in CTS regions, the evaluation has been done in two different scenarios, which are as described in fig. 3. In the figures, the CTS periods of the reference cell are represented as the shaded area.

**Table 1.** System parameters

|                                   |              |
|-----------------------------------|--------------|
| Number of cells                   | 7            |
| Cell radius                       | 1000m        |
| Transmission power (BS, MS)       | 43dBm, 23dBm |
| Number of time slots in a frame   | 14           |
| Number of minislots per time slot | 3/6          |
| Number of subcarriers in OFDM     | 128          |
| Bandwidth of a subcarrier         | 10kHz        |
| Pathloss exponent                 | 4            |
| Shadowing effect                  | 8dB          |

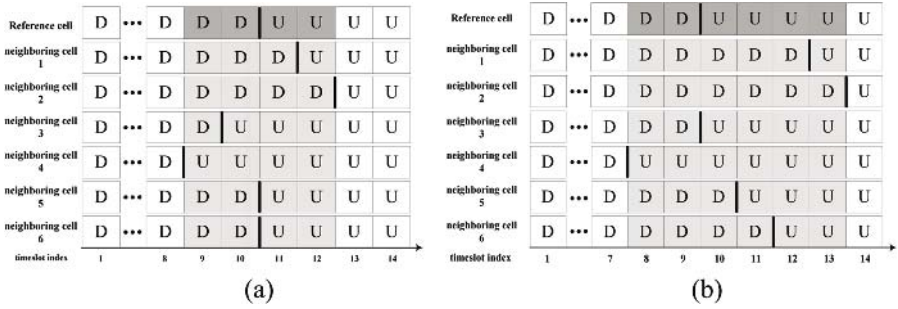


**Fig. 3.** Two different scenarios; Time slot state information for (a) scenario1, and (b) scenario 2 under 7-cell environment

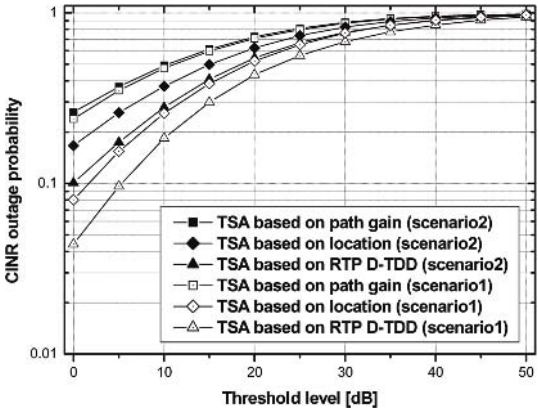


**Fig. 4.** Comparison of the CINR outage probabilities under the two scenarios

In the simulation, the proposed TSA based on RTP D - TDD architecture is compared with two conventional TSA schemes, the TSA based on path gain [3] and the TSA based on location [4].

In fig. 4, the CINR outage probabilities of the proposed algorithm have been compared with those of two conventional algorithms. Using either scenario1 or scenario2, the proposed TSA based on RTP D-TDD outperforms the conventional algorithms. As shown in the figure, the threshold level of the proposed TSA is about 4dB higher than that of the TSA based on location scheme at the same CINR outage probability. This results from the extended distances between the receiver in the reference cell and the interfering transmitters in the neighboring cells by applying the RTP D-TDD system. Besides, the influence of the CCI is further optimized by assigning minislots nearer the switching point to the users having higher link gains in the proposed TSA.

Note that for all the TSA schemes, the performance for scenario1 is better than the performance for scenario2, which shows the influence from the characteristics of CTSs. That is, since the scenario2 has 2 more time slots in the CTS period than the scenario1, it gets more CCI from neighboring BSs, so ends up with higher CINR outage probability. In fact, the length of the CTS period can be as long as the full

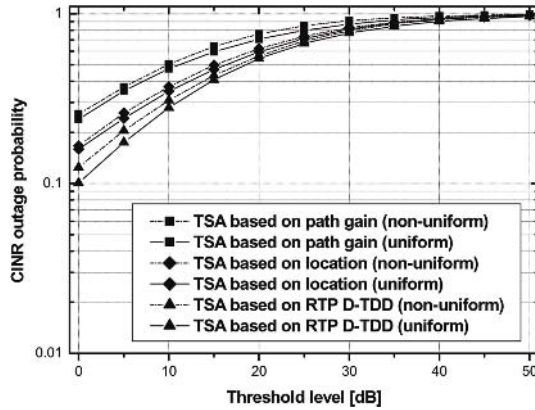


**Fig. 5.** Comparison of the CINR outage probabilities under various traffic conditions for scenario2

length of TDD time frame, which implies that better performance in CTS period is critical to the overall system performance.

Fig.5 describes the CINR outage probabilities under various TSA schemes under different traffic conditions. Here, scenario2 has been considered for the performance evaluation. The 'non-uniform' traffic model represents the traffic condition when 50% of the total traffic is concentrated in one specific region in the reference cell and the remain traffic is uniformly distributed to the rest of the regions. Even under the non-uniform traffic distribution, the proposed TSA based on RTP D-TDD architecture outperforms the conventional TSA schemes.

In table 2, the average throughput and the bandwidth efficiency in CTS period are evaluated under different traffic conditions with scenario2. For the performance evaluation, we have applied an adaptive modulation using M-quadrature amplitude modulation (QAM) according to [5]. The non-uniform traffic is as defined in Fig.6, and it is assumed that the BS operates with the omnidirectional antenna. Under uniform traffic condition, the bandwidth efficiency of the TSA based on RTP D-TDD is 1.14bps/Hz higher than that of the TSA based on path gain, and 0.43bps/Hz higher than that of the TSA based on location with uniform traffic condition. And under non-uniform traffic condition, the proposed TSA has 1.15bps/Hz and 0.37bps/Hz higher bandwidth efficiency than the TSA based on path gain and the TSA based on the location, respectively. Although the average throughput and the bandwidth efficiency for three algorithms are all reduced for non-uniform traffic condition, the proposed TSA based on RTP D-TDD still outperforms the conventional schemes by effectively reducing the CCI.

Fig. 6 shows the CINR outage probability of the proposed TSA according to the number of minislots per CTS time slot under scenario1 and scenario2. With more minislots per CTS time slot, the users allocated in the CTS period gets more separation from the interfering sources, since such separation is achieved by dividing the cells into more regions, which implies restricting the positions of interfering sources in certain regions. If the sectorized antenna system is also adopted, each region can be



**Table 2.** Comparisons in throughput and bandwidth efficiency for scenario2

|                           | Uniform              |                           | Non-uniform          |                           |
|---------------------------|----------------------|---------------------------|----------------------|---------------------------|
|                           | throughput<br>[Mbps] | BW efficiency<br>[bps/Hz] | throughput<br>[Mbps] | BW efficiency<br>[bps/Hz] |
| TSA based on<br>path gain | 1.36                 | 2.11                      | 1.23                 | 1.94                      |
| TSA based on<br>location  | 1.80                 | 2.82                      | 1.74                 | 2.72                      |
| TSA based on<br>RTP D-TDD | 2.08                 | 3.25                      | 2.01                 | 3.09                      |

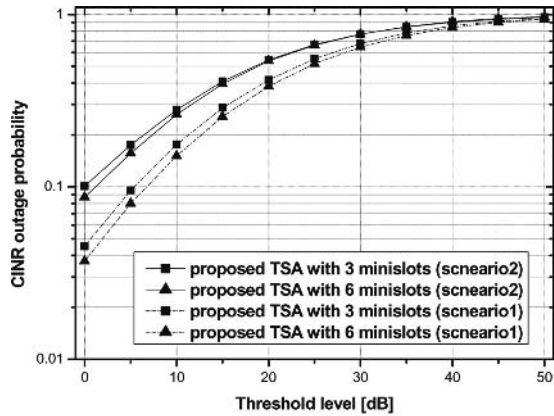


**Fig. 6.** Comparison of the CINR outage probabilities according to the number of minislots per CTS time slot with omnidirectional antenna

defined to correspond to a mutually exclusive sector. Then, since each region in the proposed RTP D-TDD architecture is segregated in time, the cellular system can be operated with frequency reuse factor of 1, whereas the subcarriers in the OFDM structure should be exclusively shared among the sectors in conventional scheme. With the sectored antenna, the CINR outage probability of the proposed RTP D-TDD architecture can be further enhanced by reducing the interference from neighboring BSs considerably.

**5 Conclusion**

In this paper, we have proposed an efficient heterogeneous wireless access scheme based on RTP in dynamic TDD OFDM systems. In the RTP D-TDD architecture, a time slot in the CTS period is divided into several minislots, and each cell is accordingly divided into as many regions. With the principle that the minislots in the CTS period is exclusively allocated to the users in the each corresponding region, the CCI in the CTS period is mitigated. On top of such architecture, a robust TSA scheme has been proposed in order to minimize that CINR outage probability in the CTS period, and thus increasing the throughput. The system performance has improved by the

extended distances among the interfering sources. By the computer simulation, the proposed TSA based on the RTP D-TDD architecture has been proved to outperform other conventional TSA schemes with regards to the CINR outage probability and the bandwidth efficiency.

## References

1. J. C.-I. Chuang.: Performance limitations of TDD wireless personal communications with asynchronous radio ports: *Electron. Letters*, vol.28, (1992) 532-534
2. Harri Holma, and Mohsen Kavehrad.: Dynamic-TDD and Optimum Slot-Allocation in Fixed Cellular Systems: *VTC 2001 Fall*, vol. 1, (2001) 37-41
3. Nasreddine, J., and Lagrange, X.: Time Slot Allocation Based on a Path Gain Division Scheme for TD-CDMA TDD Systems: *IEEE VTC-2003 Spring*, vol. 2, (2003) 1410-1414
4. Angeliki Alexiou, Sangboh Yun, et al.: Duplexing, Resource Allocation and Inter-cell Coordination-Design Recommendations for Next Generation Systems: *WWRF White Paper*, (2004)
5. A. J. Godlsmith, and S. G. Shua.: Variable-rate Variable-power MQAM for Fading Channels: *IEEE Trans. Commun.*, vol. 45 (1997)
6. Guidelines for Evaluation of Radio Transmission Technologies for IMT-2000, Recommendation ITU-R M.1225 (1997)
7. Hee-Jeong Chung, Mee-ran Kim, Nak-Myeong Kim, and Sangboh Yun.: Time Slot Allocation Based on Region and Time Partitioning for Dynamic TDD-OFDM Systems: *IEEE VTC-spring 2006*, vol.5, (2006) 2459-2463
8. Xing Zhang, et al.: Adaptive multiuser radio resource allocation for OFDMA systems: *IEEE Globecom 2005*, vol.6, (2005) 3846-3850

# Cognitive Radio Based Bandwidth Sharing Among Heterogeneous OFDM Access Systems

Mee-Ran Kim<sup>1</sup>, Eun-Ju Kim<sup>2</sup>, Yeon-Joo Kang<sup>2</sup>, and Nak-Myeong Kim<sup>2</sup>

<sup>1</sup> Electronics and Telecommunications Research Institute, Daejeon, Korea  
newface82@etri.re.kr

<sup>2</sup> Dept. of Information Electronics Engineering, Ewha Womans University, Seoul, Korea  
{ejkim99, myzhu}@ewhain.net, nmkim@ewha.ac.kr

**Abstract.** Cognitive radio (CR) has been proposed as an effective technology for flexible use of the radio spectrum. The interference among primary users and CR users, however, becomes a critical problem when they are operating nearby using adjacent frequency channels with different transmission powers. In this paper, a CR-based bandwidth sharing architecture is proposed, which can effectively suppress adjacent channel interference (ACI) between them. This new approach is characterized by the adaptive frequency spreading, and the adaptive time spreading with power control for the subcarriers near the borders of the CR user's spectrum. The frequency spreading scheme provides with extra power gain against ACI, whereas the time spreading with power control guarantees the minimal interference to primary users from CR users. By computer simulation, it has been proved that the proposed system outperforms the conventional OFDM systems in both throughput and BER performance.

**Keywords:** Cognitive Radio, Frequency Spreading, Time Spreading, Orthogonal Frequency Division Multiplexing.

## 1 Introduction

In future wireless communication systems, more spectral resources are required to provide ubiquitous multimedia wireless services. Considering the limited available frequencies, however, innovative technologies for the efficient spectrum utilization are needed. Sometimes the spectrum scarcity results from unsuitable spectrum management rather than from fundamental lack of spectrum. The cognitive radio (CR) technology offers a possible solution based on spectrum sharing or leases in terms of time, frequency or space diversity. The CR system is defined as a communication system which has the ability to detect unused spectrum dynamically, and to adjust its power, frequency, modulation, coding and other parameters to efficiently utilize the vacant spectrum [1].

Recently, there have been several researches on the interference for the CR system. In [2], the orthogonal frequency division multiplexing (OFDM) is considered as one of the promising candidates for the CR system since it is possible to provide a flexible spectral shape that fills the spectral gaps. Here, the concept of adaptive guard bands

has been proposed for the mitigation of mutual interference between the CR system and the primary system. However, some decrease in the bandwidth (BW) efficiency and throughput is inevitable because of additional deactivation of the subcarriers near the borders of the OFDM spectrum. In [3], the power scaling to CR users is proposed to maintain the guaranteed quality of the primary service. This scheme, however, only focuses on the mitigation of interference from the CR system to the primary system.

For realizable operation of the CR system, however, it is required to mitigate the interference in both ways between primary users and CR users. When a CR user uses a frequency channel in the vicinity of an active primary user's transmission channel, the OFDM subcarriers nearer to the border get severer adjacent channel interference (ACI) from primary users, and on the other hand, give more interference to primary users. In this paper, we propose a CR-based BW sharing architecture for OFDM systems, which includes adaptive frequency spreading, and adaptive time spreading with power control capabilities against ACI between primary users and CR users. The frequency spreading scheme provides with extra power gain against ACI, whereas the time spreading with power control guarantees the minimal interference from CR users to primary users.

The rest of the paper is organized as follows. In section 2, the architecture of CR OFDM is briefly introduced, and in section 3, the proposed adaptive frequency spreading and time spreading with power control are described. The simulation results are described in section 4, and we conclude in section 5.

## 2 A Cognitive Radio OFDM Architecture

In CR environment, the primary users operate with its own base station (BS), called primary BS, while CR users operate with its own BS, called CR BS. Both types of BSs may coexist within the same area. Primary users use most of the spectrum, but they can provide their unused spectrum to CR users. Some process for the negotiation to share the opportunistic spectrum is needed between primary users and CR users. Allowing the CR users to share the spectrum on the condition that the CR users do not degrade the performance of primary users is one of the most critical concerns in the CR system. Here, for the primary user, an upper limit of the power spectral density of the interference from the CR users may have been previously defined for determining whether the CR users can use the channels or not. On the other hand, reducing the influence of the interference from the primary user to CR users is also important in order to maximize the throughput of CR users.

Fig. 1(a) describes the ACI from a CR user to primary users. In this case, since the ACI from the CR user is higher than the interference limit of the primary users, the performance of the primary users is degraded. For this reason, the CR system has to control the transmission power somehow to mitigate the interference to the primary system. On the other hand, due to relatively higher transmission power of the primary system, the CR user operating close to primary user's transmitter receives a considerable influence of the ACI. The ACI from primary users to the CR user can be modeled like in fig. 1(b).

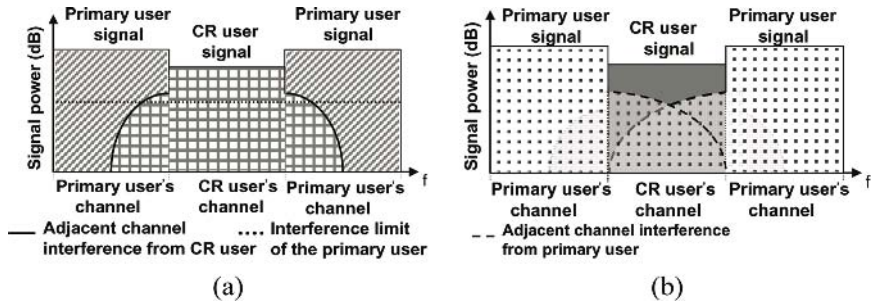


**Fig. 1.** The ACI (a) from CR user to primary users, and (b) from primary users to CR user in the CR environment

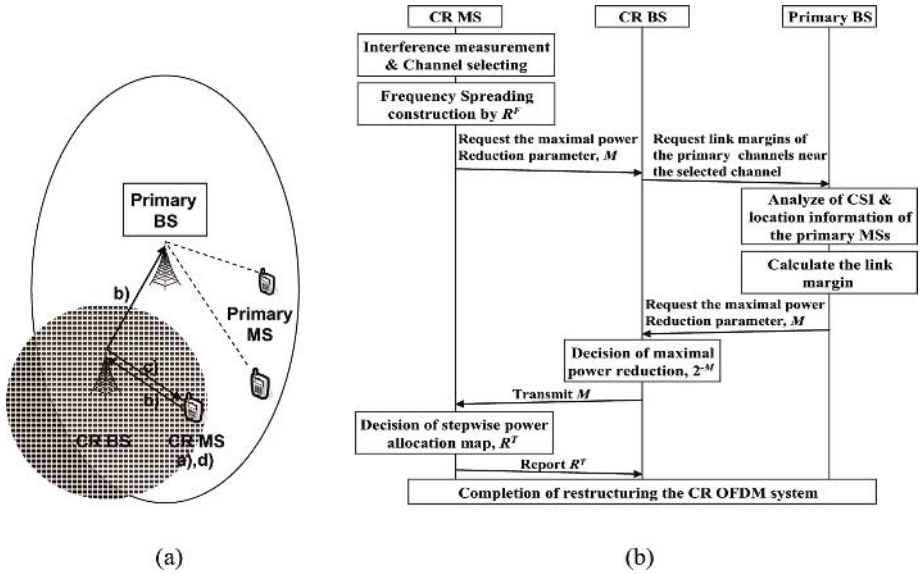


**Fig. 2.** (a) CR OFDM system architecture and (b) the proposed operational protocol for the CR OFDM system

Applying OFDM modulation to the CR system is known to have an advantage of moderating the ACI between the two systems [2] flexibly. Since the subcarriers can be controlled separately in the OFDM modulation scheme, the OFDM-based CR architecture may feed certain subcarriers with zeros, resulting in no emission of radio power. Fig. 2(a) describes the architecture of the proposed CR OFDM system in the CR environment, based on adaptive frequency spreading, and adaptive time spreading with power control. The operational protocol for the CR OFDM system is shown in fig. 2(b) and can be described as follows;

a) The CR MS measures ambient interference and selects an available channel. According to the measured interference, the CR MS determines the structure of frequency spreading scheme to overcome any interference from the primary system.

b) The CR MS inquires the status of nearby primary users to the CR BS. The CR BS then requests to the primary BS, the link margins of the primary channels neighboring to the selected channel. The primary BS then analyzes the channel state information (CSI) and location information of the primary MSs near the CR MS, and responds with the calculated link margin information.

c) The CR BS decides the amount of required transmission power reduction of the CR MS and then transmits the information to the CR MS.

d) The CR MS decides the time spreading structure based on the provided information. It finally reports the OFDM symbol structure to the CR BS, and then both of them are restructured with the designated parameters.

### 3 Adaptive Frequency/Time Spreading with Power Control

When a primary user is operating at either end of the CR user's channel frequency band, the signal to interference plus noise ratio (SINR) of the subcarriers nearer to the boundary of the CR user's spectrum is much degraded. The SINR's of such subcarriers may easily decrease below the required SINR, denoted as  $SINR_{th\_0}$ .

In order to improve the performance using such subcarriers, we propose to spread the transmission data along adjacent subcarriers, considering relative separation from the primary user's frequency band. The amount of the spreading depends on the current ACI and the required SINR. Here, we assume several SINR threshold values,  $SINR_{th\_k}$ , according to the level of degradation, which is defined as

$$SINR_{th\_k} = SINR_{th\_k} - 3 \cdot k \text{ [dB]}, \quad (1)$$

so that the thresholds are separated by 3dB from each other. Let  $n_k^F$  be the number of subcarriers which satisfy

$$\gamma_i < SINR_{th\_k}, \quad (2)$$

where  $\gamma_i$  is the SINR of the  $i$ th subcarrier. We then spread the transmission data by repeating  $2^k$  times and assign them to the subcarriers whose SINR is between  $SINR_{th\_k}$  and  $SINR_{th\_k-1}$ . To accommodate the case in which the interval is not a multiple of 2, we recursively define  $b_k^F$ , the number of data symbols which is spread by  $2^k$  times, as follows;

$$b_k^F = \left\lceil \frac{n_{k-1}^F - \sum_{j=0}^{K-k} (b_{(K-j+1)}^F \cdot 2^{(K-j+1)})}{2^k} \right\rceil, \quad (3)$$

for  $k = K, K-1, \dots, 1$ . Here,  $b_k^F$ ,  $k > K$ , is assumed to be zero. Then, the map for adaptive frequency spreading for the CR OFDM system is defined by

$$\mathbf{R}^F = \{(2^k, b_k^F)\}, k = 1, 2, \dots, K. \quad (4)$$

Such spread data is further multiplied by a pseudo random binary code,  $C_k^F$ , in order to randomize nearby data sequences, where

$$C_k^F = [c_k^F[0], c_k^F[1], \dots, c_k^F[2^k - 1]], k = 1, 2, \dots, K. \quad (5)$$

Then, the index of the modulated data symbol,  $u$ , assigned to the  $i$ th subcarrier for  $i \leq N/2$  is expressed as

$$u = \sum_{k=k^*+1}^K b_k^F - \left\lfloor \frac{i - n_K^F - \sum_{k=k^*+1}^K 2^k \cdot b_k^F}{2^{k-1}} \right\rfloor. \quad (6)$$

The index of the modulated data symbol assigned to the  $i$ th subcarrier for  $i > N/2$  can be similarly obtained.

On the other hand, in the CR environment, the primary users far from the primary BS receive lots of interference from the nearby CR users, unless appropriate protection mechanism is provided. The power of subcarriers from the CR users nearer to the boundary of the frequency band causes more interference. To mitigate such interference, we propose to taper the transmission power of the subcarriers nearer to the edges of the CR user's frequency band. Instead, the tapered portion of data from the CR user is protected by an adaptive time spreading structure. That is, the tapered portion of data is repeated in the subsequent symbols in time so that the total amount of power can be recovered.

Fig. 3 describes the proposed structure of time spreading and stepwise power reduction, which is described in three dimensions; power, time and frequency. The powers of subcarriers near the edges of frequency band are gradually reduced, while the powers of the center portion of the subcarriers are unchanged. The specific set of subcarriers to apply the power reduction is controlled adaptively according to the possible interference towards primary users.

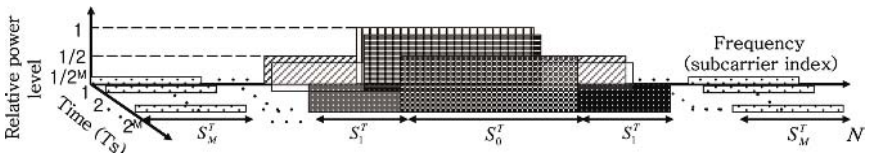


**Fig. 3.** Time spreading and stepwise power reduction structure

Let  $s_m^T$  be the number of subcarriers to apply a reduced power of  $2^{-m}$  for  $m = 1, 2, \dots, M$ . Then, the overall subcarrier power allocation map for the current OFDM symbol can be expressed as

$$\mathbf{R}^T = \{(2^{-m}, s_m^T)\}, \quad m=1, 2, \dots, M, \quad (7)$$

where  $2^{-M}$  represents the amount of the maximal power reduction.

In the time spreading process, the data whose power have been reduced by  $2^{-m}$  is spread in the subsequent  $2^m$  OFDM symbols. A pseudo random binary code,  $c_m^T[t]$  is multiplied to the sequence of repeated data symbols to get a randomization effect. Since the sequence of data symbols can be combined in the receiver at the end of the  $2^m$ -th OFDM symbol, the reduced data symbol can recover its original power. Therefore, the performance of the CR users can be maintained while the channel condition of the primary users is protected.

### 3.1 Transmitter of the Proposed CR OFDM System

The transmitter of the proposed CR OFDM system is shown in fig. 4(a). The output of data modulator enters the serial-to parallel (S/P) converter which adaptively links the data symbols to the CR OFDM modulator. The adaptive frequency/time spreading with power control is carried out by the spreading controller. At the output of the S/P converter, each data symbol,  $d_u$ , is adaptively spread in the frequency domain using  $R^F$ . After being multiplied by a pseudo random binary code,  $c_m^T[t]$ , the resultant data symbols are loaded in  $2^m$  consecutive OFDM symbols. The power of the spread symbols is then reduced by  $2^{-m}$ .

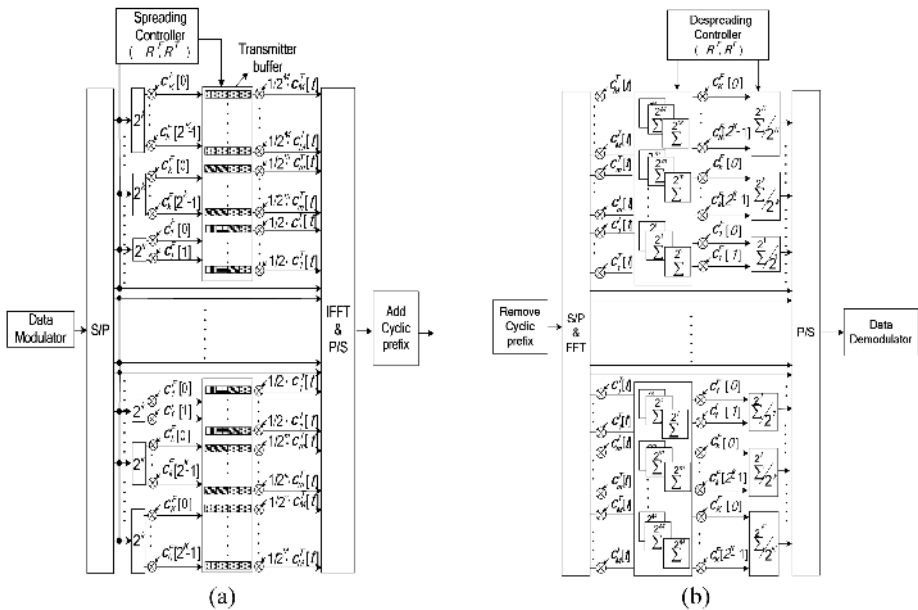


Fig. 4. (a) Transmitter and (b) Receiver of the proposed CR OFDM system



Now, the  $i$ th subcarrier before the inverse fast Fourier transform (IFFT) is described as

$$X_i = \begin{cases} d_u[i], & \text{if } SINR(i) > SINR_{th\_0} \\ d_u[i] \cdot c_k^F[i], & \text{if frequency spreading only,} \\ \frac{1}{2^m} \cdot d_u[i] \cdot c_k^F[i] \cdot c_m^T[i], & \text{if frequency/time spreading} \end{cases} \quad \text{for } i=0,1,\dots,N-1, \quad (8)$$

where  $SINR(i)$  is the measured SINR of the  $i$ th subcarrier, and  $N$  is the FFT length. Here,  $k$  and  $m$  are the spreading parameters evaluated by the resolution process for the frequency and time spreading.

### 3.2 Receiver of the Proposed CR OFDM System

The receiver of the CR OFDM system can be represented as in fig. 4(b). At the receiver, the source data is recovered in the reverse order of the transmitter process. The  $n$ th symbol of the received signal is described as

$$y(n) = \hat{x}(n) * h(n) + \ell(n) + w(n), \quad (9)$$

where  $\hat{x}(n)$  is the received OFDM symbol,  $h(n)$  is channel impulse response,  $\ell(n)$  is the interference from primary users, and  $w(n)$  is additive white Gaussian noise. We assume that the synchronization is perfect so that no ISI and ICI occur within the primary system alone or the CR system alone. Then the FFT output for the  $i$ th subcarrier is given by

$$Y_i = \hat{X}_i \cdot H_i + I_i + W_i, \quad (10)$$

where  $\hat{X}_i$ ,  $H_i$ ,  $I_i$  and  $W_i$  are the corresponding frequency domain expressions of  $\hat{x}(n)$ ,  $h(n)$ ,  $\ell(n)$ , and  $w(n)$  respectively.

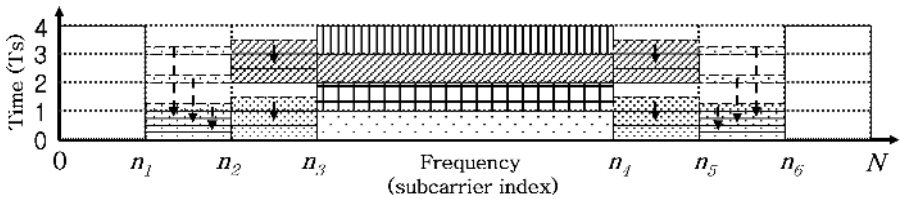


**Fig. 5.** An example of the combined data structure for the adaptive time despreading

Now, the CR OFDM system performs the despreading process controlled by a despreading controller. First, for the time despreading, a pseudo random binary code,  $c_m^T[t]$ , is multiplied to each data symbol spread in time. And the data symbols in each subcarrier are added up as many times as the length of each time spreading, so that all the data can be combined by the end of  $2^m$  OFDM symbol

periods. Fig. 5 shows an example of combining data symbols adaptively. Each color in the background represents different set of subcarriers combined within each specified time spreading interval. In the given example, the subcarriers ranged from  $n_3$  to  $n_4$  is the subcarriers transmitted without spreading, and can be demodulated at every time period in the receiver, while the subcarriers spread by 2 times, from  $n_2$  to  $n_3$  and from  $n_4$  to  $n_5$ , should be combined at the end of 2 time periods for demodulation. Considering the time of data arrival at the transmitter, the combined data symbol is demodulated at the end of the spreading period.

After completing the time despreading process, a frequency despreading is performed. For the frequency despreading, a pseudo random binary code,  $C_k^F$ , is first multiplied to the subcarrier symbols which are assigned to the subcarriers spread in frequency. Then,  $2^k$  consecutive subcarrier symbols are added up according to the corresponding spreading structure, and normalized by  $2^k$ . Then, the  $i$ th subcarrier symbol after the time and frequency despreading process is obtained as

$$S_i = \begin{cases} Y_i, & \text{if } SINR(i) > SINR_{th\_0} \\ \frac{\sum_{k=1}^{2^k} Y_i \cdot c_k^F[i]}{2^k}, & \text{if frequency spreading only, for } i = 0, 1, \dots, N-1. \\ \frac{\sum_{k=1}^{2^k} \sum_{m=1}^{2^m} \{Y_i \cdot c_m^T[i] \cdot c_k^F[i]\}}{2^k}, & \text{if frequency/time spreading} \end{cases} \quad (11)$$

Now, the SINR for the  $i$ th subcarrier can be obtained as

$$\gamma_i = \frac{P_i}{\eta_i + \sigma_i^2} |\hat{H}_i|^2, \quad (12)$$

where  $P_i$  and  $|\hat{H}_i|^2$  are the transmitted signal power and the estimated channel gain of the  $i$ th subcarrier, respectively. Here,  $\eta_i$  and  $\sigma_i^2$  represent the interference power and the received noise power at the  $i$ th subcarrier, respectively.

## 4 Computer Simulation

In this section, we evaluate the performance of the proposed CR OFDM system in the presence of ACI from the primary systems. We adopt as the primary system the digital television (DTV) stations operating in the vicinity of the CR user's frequency band. The ACI from the DTV is modeled as the adjacent channel emission mask defined in the FCC [4]. The performance of the proposed CR OFDM system has been evaluated using MATLAB under Pentium-IV personal computers. The system parameters of the CR OFDM system are summarized in table 1. For the comparison of the system performance, the conventional CR OFDM system with guard band is considered.

In fig. 6, we have evaluated the average BER of the proposed CR OFDM system in comparison with the conventional CR OFDM system. The performance of

the OFDM system assuming no ACI is also included. Note that the proposed CR-OFDM outperforms the conventional OFDM with guard band in BER when the Eb/No is greater than 5dB. In this range, the proposed CR OFDM system has a gain of more than 3dB at the BER of  $10^{-4}$ . When the Eb/No becomes less than 5 dB, however, the ACI from the primary system severely affects the proposed system, while it is avoided by extended guard bands in the conventional CR-OFDM at the sacrifice of throughput. The performance of the proposed CR OFDM system gets closer to that of the conventional OFDM system without ACI, even if the proposed system operates among severe ACI.

Table 1. System Parameters

| Parameter             | Value       |
|-----------------------|-------------|
| Center frequency      | 500MHz      |
| BW                    | 6MHz        |
| Number of subcarriers | 512         |
| Subcarrier spacing    | 11.72kHz    |
| Modulation            | 16QAM, QPSK |
| Channel               | AWGN        |

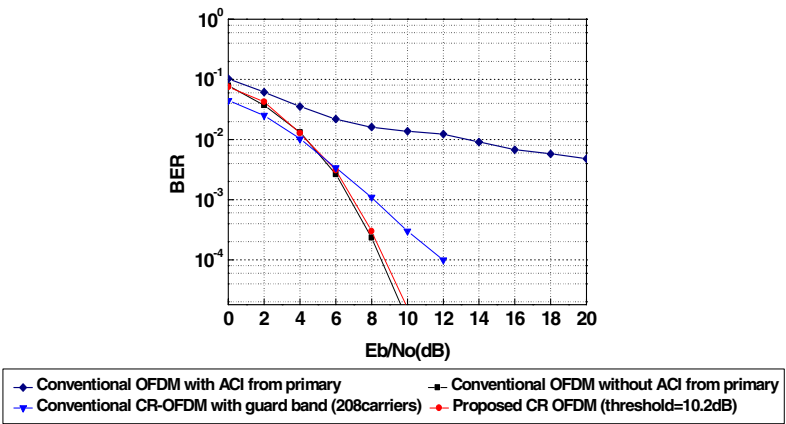


Fig. 6. BER performance of the proposed CR OFDM system in the presence of ACI from primary users

In table 2, the average throughput of CR OFDM systems is evaluated at Eb/No of 8dB. As shown in the table, the average throughput of the proposed CR OFDM system is 1.64Mbps higher than the conventional CR OFDM system with guard band. This is because the subcarriers having SINR less than the target SINR threshold are still utilized in the proposed CR OFDM system, while they are unused in the conventional CR OFDM system. Such utilization of barren subcarriers for data transmission increases available BW at the receiver, so results in the increase of throughput.

Fig. 7 shows the average throughput of the proposed CR OFDM system with various target SINR thresholds,  $SINR_{th\_0}$ . As shown in the figure, the average throughput decreases as the target SINR threshold increases, since the target threshold determines the acceptable span of adaptive modulations. An abrupt change in performance is encountered around the target SINR threshold of 10.2 dB, above which the utilization of BPSK modulation begins to be limited. Fig. 8 shows the BER performance of the proposed CR OFDM system with various target SINR thresholds. The BER performance improves as threshold level increases. Therefore, there exists a kind of tradeoff between the throughput and the BER in the design of the target SINR threshold in the proposed CR OFDM system. Note that, when the required SINR threshold is 10.2dB, the BER performance is only about 0.2dB away from the performance of the OFDM without ACI at the BER of  $10^{-3}$ . If we reduce the required SINR threshold by 2dB, the BER performance is about 1dB away, but the increase in the throughput is about 0.7Mbps. Further reduction of the threshold, for example, by 6dB, is not acceptable since the BER performance is degraded by about 6dB at BER of  $10^{-3}$ . Therefore, the proposed CR OFDM system can be adaptively configured considering the performance in BER and throughput.

**Table 2.** Throughput Comparison (Eb/No=8dB)

|                                               | Throughput (Mbps) |
|-----------------------------------------------|-------------------|
| Conventional CR OFDM system with a guard band | 7.15              |
| Proposed CR OFDM system (10.2dB threshold)    | 8.79              |

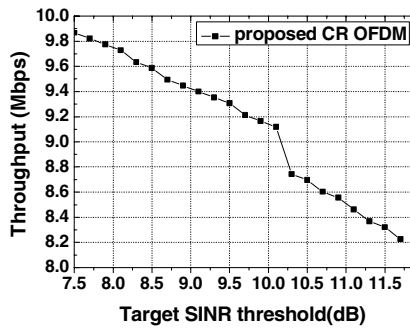


**Fig. 7.** Throughput vs. target SINR threshold in the proposed CR OFDM

Fig. 9 compares the average throughput of the proposed CR OFDM system according to the relative location of the CR user within the influence of the primary system. As shown in the figure, the average throughput of the proposed CR OFDM system is about 2Mbps higher than that of the conventional CR OFDM system using guard band. When the CR user gets closer to the primary BS, the throughput of the CR user decreases. This results from the increase of ACI. However, the throughput can still be kept to more than 8Mbps when the threshold values are lower than 10.2dB

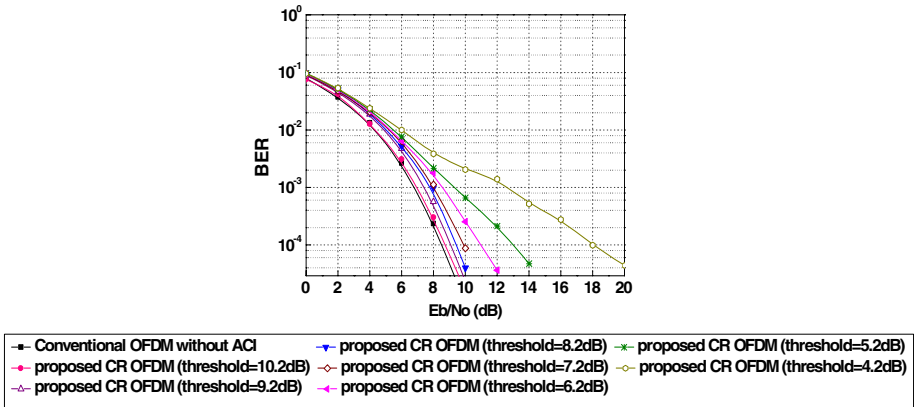


**Fig. 8.** BER performance according to the SINR threshold levels in the CR OFDM systems

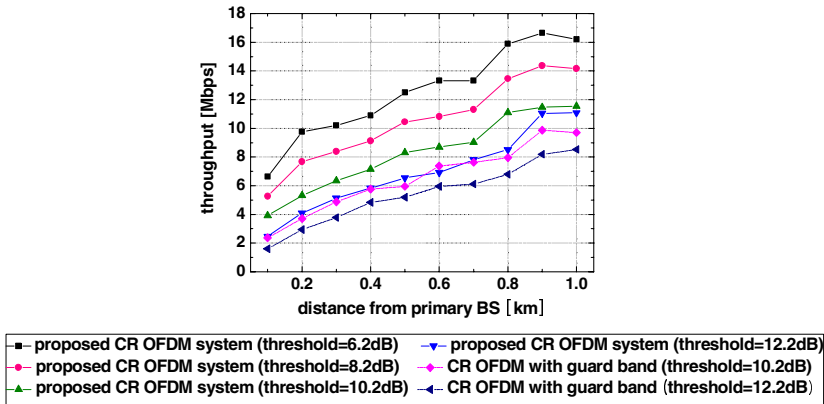


**Fig. 9.** Comparison of throughput according to the distance between primary BS and CR MS under different threshold values

in a distance of a 500m from the primary BS. When the CR user is located in 1km from the primary BS, the throughput of the CR user increases up to 10.5Mbps at the target SINR of 10.2dB. With lower SINR thresholds, the throughput can be further enhanced. Therefore, selecting an optimum target SINR threshold according to the system requirements is important in operating the proposed CR OFDM system.

## 5 Conclusion

We have proposed a BW sharing scheme based on the cognitive radio among heterogeneous OFDM access systems. An adaptive frequency spreading scheme has been applied according to the instantaneous SINR, which improves the power gain of data

to overcome the interference from primary users. On the other hand, the stepwise power reduction and time spreading in both edges of the frequency band prevent the performance degradation of the primary user. Computer simulations have shown that the proposed CR OFDM system outperforms the conventional OFDM system with adaptive guard band in both the throughput and the BER. We have also confirmed that the performance of the primary user is maintained without degradation. The proposed CR OFDM system can be adaptively configured according to the target BER and throughput.

## References

1. Joseph Mitola.: Cognitive Radio for Flexible Mobile Multimedia Communications: IEEE Int. Workshop on Mobile Multimedia Comm, 3-10 (1999)
2. Weiss, T.A. Jondral, F.K.: Spectrum pooling: An Innovative Strategy for the Enhancement of Spectrum Efficiency: IEEE Communications Magazine, vol. 42, 8-14 (2004)
3. A. Sahai, N. Hoven. : Power scaling for Cognitive Radio: IEEE WirelessCom (2005)
4. G. Sgrignoli. : DTV Repeater Emission Mask Analysis: Broadcasting, IEEE Transaction, vol. 48, 32-80 (2003)
5. A. J. Godlsmith, and S. -G. Shua. : Variable-rate Variable-power MQAM for Fading Channels: IEEE Trans. Commun., vol. 45 (1997)
6. Faith Capar and Friedrich Jondral. : Resource allocation in a Spectrum Pooling System for Packet Radio Networks Using OFDM/TDMA: IST Mobile and Wireless Telecom (2002)

# CDMA Power Control Using Channel Prediction in Mobile Fading Channels

Sangho Choe

School of Information, Communication & Electronic Engineering  
The Catholic University of Korea, Gyeonggi-do, Korea  
Tel.: +82-2-2164-4870; Fax: +82-2-2164-4777  
schoe@catholic.ac.kr

**Abstract.** This paper presents a CDMA closed-loop power control (CLPC) using channel prediction in time-varying mobile fading channels. The proposed CLPC that relies on least square (LS) or least mean square (LMS) based autoregressive (AR) low-rate prediction provides a particularly robust performance for large loop delays, especially in the presence of channel estimation errors. As the mobility increases the proposed CLPC reduces mean-square error (MSE) further than conventional CLPCs. The Monte-Carlo simulation results demonstrate the superiority of the proposed CLPC over conventional nonpredictive and predictive CLPCs.

**Keywords:** Predictive closed-loop power control (CLPC), channel prediction, robustness, time-varying mobile fading.

## 1 Introduction

By closed-loop power control (CLPC), the base station (BS) controls the transmit power levels of all MSs within its control area (a cell or a sector) through feedback channels. While open-loop power control (OLPC) mainly combats the slow variation of link loss due to path loss and shadowing, CLPC focuses on the compensation of the fast variation of link loss due to Rayleigh fading in cellular networks. First, the BS measures the received signal-to-interference power ratio (SIR) for each MS over a short time interval, which is usually defined as the power control group (PCG); one PCG indicates a period of power control cycle ( $1/T_p = 1/f_p =$  the PCG rate). The BS then compares each measured value to a preset SIR threshold  $\Gamma_{TH}$  and sends a corresponding power control bit (PCB) per PCG to its managing MSs via feedback. Besides the basic delay of one PCG for every power control update, the PCB typically experiences the additional delay ( $d_A$ ) within power control loop due to the processing delay, the round-trip delay, and the frame delay (assuming coded PCB is used). The overall loop delay is therefore given by  $d (= d_A + 1)$  whose unit is also PCG. In current cellular systems, this is typically greater than 2-3 PCG units [1], [2]. Traditional nonpredictive CLPC methods may not trace well the fast fading channel fluctuations in the presence of such loop delays resulting in excessive power control errors. Thus,

predictive CLPCs have been proposed to effectively compensate the loop delay [2]-[6]. Single-step predictive CLPCs, such as [3], are able to compensate one PCG delay, but have a limited compensation capability for multiple PCG delays.

Therefore several predictive CLPCs that have multi-step linear predictors to effectively compensate the multiple PCG delays are recently presented. The multi-step<sup>1</sup> Smith predictor [6] and time delay compensation (TDC) method [2] focuses on slow (shadowing) fading compensation due to the loop instability and the external signal delay. multi-step predictive CLPC based on minimum-variance method [5] that is applicable to wireless local area network compensates fast fading. Recently, minimum mean square error (MMSE) based multi-step AR predictive power control [4] that is applicable to adaptive transmission compensates fast fading. Those conventional multi-step predictive power control schemes employ a high-rate predictor that predicts the future samples  $\hat{a}_{n+p}$  based on a high-rate sample vector that consists of previous  $L$  contiguous channel samples:  $\mathbf{a} = \{a_n, a_{n-1}, \dots, a_{n-L+1}\}$ , where  $L$  is the filter order and the sampling rate  $f_s$  to obtain the vector is equal to the PCG rate  $f_p$ .

This paper presents a multi-step AR predictive power control scheme that has a low-rate predictor which predicts  $\hat{a}_{n+p}$  based on a low-rate sample vector  $\mathbf{a} = \{a_n, a_{n-p}, \dots, a_{n-(L-1)p}\}$ , whose sampling rate  $f_s$  is  $p$  times lower than the PCG rate, i.e.,  $f_s = f_p/p$ . Thus, assuming the prediction step size  $p = d$ , the sample span of this low-rate prediction is  $d$  times greater than high-rate prediction. The increase of the sample span that correlation matrix is defined makes low-rate prediction more robust to the increase of delay  $d$  than high-rate prediction, especially in noisy Rayleigh fading channels [4]. As an AR prediction filter this paper selects LS and LMS instead of MMSE whose correlation matrix is difficult to get over time-varying multi-path fading [13]. Numerical results show that the proposed low-rate sample based predictive CLPC reduces power control error and improves bit-error rate (BER) performance compared to conventional high-rate sample based predictive CLPC in the presence of delay and channel estimation error.

The organization of the rest of the paper is as follows: Section II explains the details on the low-rate sample based AR prediction. Section III presents the proposed predictive CLPC which has a linear AR predictor. Section IV demonstrates Monte-Carlo simulation results for the proposed CLPC and compare them to the results of nonpredictive and predictive CLPCs. Finally, we conclude this paper in Section V.

## 2 Fading Channel Prediction

Assume the time instance is  $n$  (i.e., the  $n$ th PCG). Conventional high-rate sample based multi-step AR prediction (see Figure 1.a) [4] predicts the future channel samples  $\hat{a}_{n+p}$  ( $p \geq 2$ ) based on a high-rate sample vector that consist of previous  $L$  contiguous channel samples:  $\mathbf{a} = \{a_n, a_{n-1}, a_{n-2}, \dots, a_{n-(L-1)}\}^T$  where  $T$  denotes the transpose operator. The sampling rate  $f_s$  to get the vector  $\mathbf{a}$  is

<sup>1</sup> The term 'multi-step' in this paper means multiple prediction steps but does not indicate multiple power control steps.



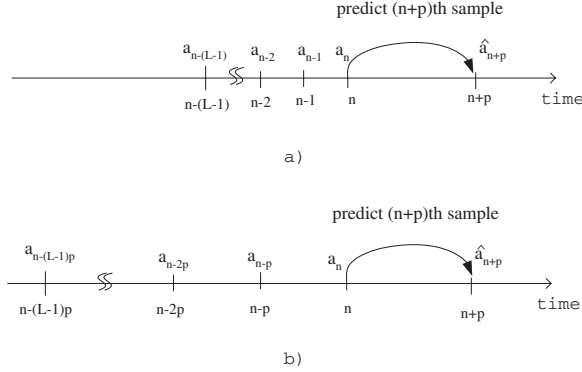


**Fig. 1.** Comparison of the two prediction schemes: a) high-rate sample based conventional prediction scheme b) low-rate sample based proposed prediction scheme

equal to the PCG rate:  $f_s = f_p$ . That is, conventional high-rate prediction has a fixed sample span of  $L$  (= the order of prediction filter) so its MSE increases sharply as the delay  $d$  increases [4].

In this paper, we propose a new low-rate sample based multi-step AR prediction. The proposed scheme predicts the future channel samples  $\hat{a}_{n+p}$  based on  $p$  low-rate sample vectors  $\mathbf{a}_i$ , where the vector index  $i = 0, \dots, p-1$ . Each sample vector  $\mathbf{a}_i$  consists of  $L$  previous channel samples whose spacing is  $p$ , i.e.,  $\mathbf{a}_i = \{a_k, a_{k-p}, \dots, a_{k-(L-1)p}\}^T$ , where  $k$  is the start time index to the sample vector. Note that the sampling rate to get those sample vectors in the proposed AR prediction is  $f_s = f_p/p$  ( $p \geq 2$ ), which is  $p$ -times lower than conventional AR prediction. That is, when assuming  $p = d$ , proposed low-rate prediction which has a proportionally extended sample span ( $d \times L$ ) to the delay  $d$  improves the MSE performance compared to conventional high-rate prediction.

The proposed prediction procedure using the  $p$  low-rate sample vectors is as follows. Assume all the sample vectors  $\mathbf{a}_i$  are initialized at the time  $n = 0$  as follows:  $\mathbf{a}_i = \{a_{i-(p-1)-lp} | l = 0, \dots, L-1\}^T$ . For the  $n(\geq 1)$ th time instance, one of those  $p$  sample vectors is rotationally chosen by a modular operation  $v = (n-1) \bmod p$ . Then we shift right once the selected sample vector  $\mathbf{a}_v$  and renew its first element with the current channel sample estimate  $a_n$  yielding  $\mathbf{a}_v = \{a_n, a_{n-p}, \dots, a_{n-(L-1)p}\}^T$ . Finally we use the sample vector to predict the  $p$ -step future channel sample  $\hat{a}_{n+p}$  as follows:

$$\begin{aligned} \hat{a}_{n+p} &= \sum_{l=1}^L c_l a_{n-(l-1)p} \\ &= \mathbf{c}^T \mathbf{a}_v, \end{aligned} \quad (1)$$

where  $\mathbf{c} = [c_1, c_2, \dots, c_L]^T$  is the tap coefficient column vector and the detail procedures to get the coefficient vector for AR filters are omitted here and referred to [13].  $\mathbf{c}$  may be periodically updated through the use of a training (pilot)

symbol sequence. Note that in our scheme the previously-predicted channel samples  $\hat{a}_{n+p-m}$ ,  $m = 1, \dots, p-1$  are not reused to get the  $p$ th future samples  $\hat{a}_{n+p}$ . Therefore, the proposed scheme avoids the problem of error propagation for multi-step prediction [7]. Assuming the same filter order  $L$ , the size of prediction window (buffer) in the proposed scheme is  $p$  times larger than that of single-step scheme, but such a complexity increase is marginal since  $p$  is typically 2 or 3 for cellular systems. As an AR prediction filter, this paper employs LS and LMS instead of MMSE whose correlation matrix over time-varying fading is not easy to get.

### 3 CLPC for CDMA Systems

For an uplink of CDMA system with  $K$  users, the BS receiver SIR for user  $i$  (mobile station  $i$ ) is given as

$$\gamma^i = \frac{g^{ii}x^i}{\theta(\sum_{j \neq i} g^{ij}x^j + \eta^i)} = \frac{g^{ii}x^i}{\xi^i + \eta'^i},$$

$$i, j \in \{1, 2, \dots, K\}, \quad (2)$$

where  $g^{ij} > 0$  is the link loss from the transmitter of user  $j$  to the access point (BS) of user  $i$ ,  $x^i$  is the transmit power of user  $i$ , and  $\theta$  is the inverse of CDMA processing gain [8]. From now on, the user index  $i$  is dropped for simplicity. Furthermore, we assume that the interference power  $\xi$  is stationary by using Gaussian approximation (based on the central-limit theorem) for the sum of all other users' signal and the noise power is negligible, i.e.,  $\eta' \ll \xi$ . In general, the link loss  $g$  consists of the path loss, the shadowing effect, and the short-term fading. Assuming that OLPC perfectly compensates the path loss and the shadowing effect,  $g$  in (2) can be attributed only to short-term fading. In our work, we assume a Rayleigh fading channel with Bessel autocorrelation following the Jakes model [12]. If the measured SIR is higher than the preset threshold ( $\Gamma_{TH}$ ), the BS issues a command bit to the MS to lower its transmission power for the next PCG and vice versa. In the following, we first provide a brief review of nonpredictive CLPC, then present the details of our proposed algorithm.

#### 3.1 Review of Nonpredictive CLPC

A nonpredictive CLPC has the same block diagram as in Figure 2 excluding linear predictor. The capital characters in Figure 2 denote logarithmic versions (unit dB) of the corresponding lower-case ones, i.e.,  $X \text{ dB} = 10 \log_{10} x$ . For the  $n$ th PCG, the nonpredictive CLPC relies on the following state equation [8]:

$$\begin{aligned} X_n &= X_{n-1} + \Delta\Psi(\Gamma_{TH} - X_{n-d} - G_{n-d} + \Xi) \\ &= X_{n-1} + \Delta\Psi(\Gamma_{TH} - \Gamma_{n-d}) \\ &= X_{n-1} + \Delta\Psi(Z_{n-d}), \end{aligned} \quad (3)$$

where  $\Xi$  denotes the interference term and the link loss is given by  $G_m = 10 \log_{10} g_m = 20 \log_{10} |a_m|$ . Here,  $a_m$  is the  $m$ th complex fading signal sample.

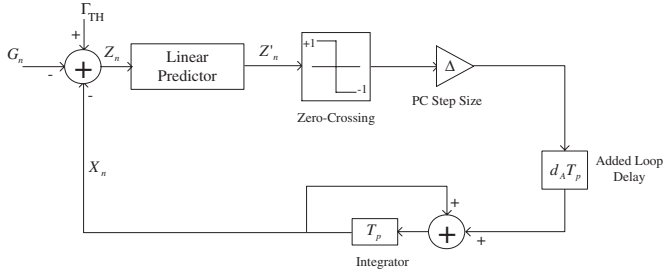


**Fig. 2.** Block diagram of predictive CLPC with a linear predictor

In (3),  $\Delta$  is the fixed power control step size in dB and  $\Psi$  denotes the the signum function given by

$$\Psi(x) = \begin{cases} 1 & \text{if } x \geq 0, \\ -1 & \text{otherwise.} \end{cases}$$

Based on the loop error statistics  $Z_n$ , the CLPC adjusts the PCB by  $\pm\Delta$  dB. The PC loop delay  $d$  incurs a delayed fading channel estimate,  $G_{n-d}$ , and therefore a delayed SIR estimate,  $\Gamma_{n-d}$ , which, in return, increases the power control error standard deviation  $\sigma_Z$ . The more the loop delay increases, the greater  $\sigma_Z$  becomes [9].

### 3.2 Proposed Predictive CLPC Algorithm

In order to reduce the power control error due to the loop delay  $d$  of multiple PCGs, we propose a predictive power control scheme with a  $p$ -step linear predictor. The block diagram of the proposed CLPC is shown in Figure 2. After the linear predictor,  $Z_n$  converts its corresponding output into

$$Z'_n = \Gamma_{TH} - \Gamma'_n, \quad (4)$$

where  $\Gamma'_n$  is the predicted estimate of receiver SIR for the  $(n+p)$ th PCG which is defined as

$$\Gamma'_n = \hat{X}_{n+p-1} + \hat{G}_{n+p} - \Xi. \quad (5)$$

$\Gamma'_n$  is used to obtain the  $p$ -step ahead predicted MS transmit power  $\hat{X}_{n+p}$  by the following state equation

$$\hat{X}_{n+p} = \hat{X}_{n+p-1} + \Delta\Psi(\Gamma_{TH} - \Gamma'_n), \quad (6)$$

which is recursively used in (5) to generate the receive SIR  $\Gamma'_{n+1}$  for the next PCG. Here,  $\hat{X}_{n+p-1}$  is the  $(p-1)$ -step ahead predicted MS transmit power.  $\hat{G}_{n+p}$  is the  $p$ -step ahead predicted fading channel loss given by  $\hat{G}_{n+p} = 20 \log_{10} |\hat{a}_{n+p}|$  where  $\hat{a}_{n+p}$  denotes the predicted complex fading sample.

In (3), by replacing  $Z_k$  with  $Z'_k$  (where  $k = n - d$ ) of (4), the state equation of the proposed predictive CLPC becomes

$$X_n = X_{n-1} + \Delta \Psi(\Gamma_{TH} - \hat{X}_{n-d+p-1} - \hat{G}_{n-d+p} + \Xi), \quad (7)$$

where the prediction step size  $p$  is chosen as  $d$  to properly compensate the loop delay.

The proposed power control algorithm can be then summarized as follows:

- Step 1:** Initialize  $\hat{X}_{p-1} = \Gamma_{TH} + \Xi$ ,  $\mathbf{a}_i$ ,  $i = 0, \dots, p-1$ , and  $\mathbf{c}$ . Start the time index  $n$  with 0.
- Step 2:** Update the channel sample vector and get the  $(n+p)$ th predicted fading channel sample  $\hat{a}_{n+p}$  by (1).
- Step 3:** Obtain the predicted estimate of receiver SIR  $\Gamma'_n$  by (5) and generate (and transmit to MS) the PCB decided by (4) to adjust the MS transmit power.
- Step 4:** Obtain the  $(n+p)$ th predicted MS transmit power  $\hat{X}_{n+p}$  by (6).
- Step 5:** Set  $n = n+1$ . Repeat Step 2 to 5 unless training sequence is received or 'release mode' (i.e. algorithm termination) is called. When the training sequence is received, go to Step 1.

## 4 Numerical Results

In this section, we provide Monte-Carlo simulation results to demonstrate the performance of the proposed predictive CLPC. In the simulation study, we consider the Jakes fading model [12] and assume a carrier frequency  $f_c$  of 2 GHz, a data rate  $R_b$  of 10 Kbps, and a PCG rate  $f_p$  of 2 KHz. The CLPC employs one of the two AR prediction filters, LS or LMS, to predict noisy Rayleigh fading channels. Considering a typical sample-average channel estimator, the noisy channel samples can be assumed to form a zero-mean complex Gaussian random variable whose variance is  $\sigma^2$  [10]. The channel estimation error variance  $\sigma^2$  over fading channels typically ranges from  $10^{-1}$  to  $10^{-4}$  in practical scenarios [11]. For computer simulation, the power control step size  $\Delta$  is chosen as 1 dB, and the employed linear prediction filter has an order  $L$  of 50.

The MSE for conventional high-rate (sample based) LS or LMS prediction and proposed low-rate (sample based) LS or LMS prediction is compared in Figure 3. Assume the mobile speed is 60km/h and the delay  $d$  is 3 or 5 and the prediction step size  $p$  is equal to  $d$ . We can observe that the proposed schemes have a better MSE performance than conventional schemes over the range of channel estimation error variance  $\sigma^2$ . The increase of delay  $d$  from 3 to 5 increases MSE although the MSE difference between  $d = 3$  and  $d = 5$  reduces as  $\sigma^2$  increases above  $10^{-4}$ . Proposed LMS has the better MSE performance than proposed LS as well as conventional LMS by using a filter coefficient vector adaptive to fading channels. Although the overall performance of LMS is better than LS, LS also stands as an attractive alternative with its lower complexity and comparable performance to LMS [13].

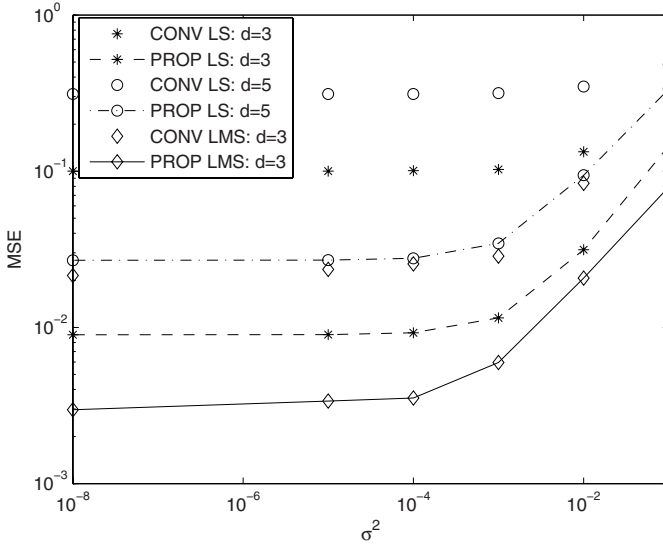


**Fig. 3.** MSE comparison of conventional and proposed prediction schemes based on LS or LMS filters over noisy Rayleigh fading channels, when assuming a mobile speed of 60km/h and a delay  $d$  of 3 or 5

In Figure 4, we demonstrate the power control error standard deviation  $\sigma_Z$  for conventional and proposed prediction using LS or LMS. We assume  $\sigma^2 = 10^{-1}$  and  $p = d = 3$ . For low mobile speed ( $v < 35$ km/h), there is no big difference between proposed schemes and conventional schemes. However for high mobile speed ( $v \geq 35$  km/h) proposed schemes are better than conventional schemes. Furthermore, the proposed LMS reduces  $\sigma_Z$  a little further (less than 0.2dB) than the proposed LS.

Figure 5 illustrates the bit error rate (BER) performance versus  $E_b/I_0$  (bit energy/ interference power spectral density) of the proposed CLPC at  $v = 60$  km/h ( $f_D = 111$  Hz) for a convolutional coded<sup>2</sup> BPSK transmission. We assume various combinations of  $d$  and  $\sigma^2$ . As a prediction filter, LMS is assumed. We compare proposed predictive CLPC and conventional predictive CLPC and nonpredictive CLPC (denoted 'NONPRED' in Figure 5). For  $\sigma^2 = 10^{-3}$ , both conventional CLPC and proposed CLPC have a similar coding gain of 0.8 dB (for  $10^{-3}$  BER criterion) over nonpredictive CLPC. However, for  $\sigma^2 = 10^{-1}$ , conventional predictive CLPC is highly sensitive to the loop delay and yields even worse BER performance than nonpredictive CLPC. For the same error variance  $\sigma^2 = 10^{-1}$ , our proposed scheme still enjoys an improved gain over the nonpredictive scheme. Specifically, 0.7 dB and 1.5dB performance gains of the proposed CLPC over nonpredictive CLPC are observed for  $d = 3$  and  $d = 5$ , respectively. This clearly illustrates that the proposed CLPC is more robust to delay and error than conventional predictive or nonpredictive CLPCs.

<sup>2</sup> Rate=1/2, 8-state code with generator matrix  $[\mathbf{g}_1=15, \mathbf{g}_2=17]$ .

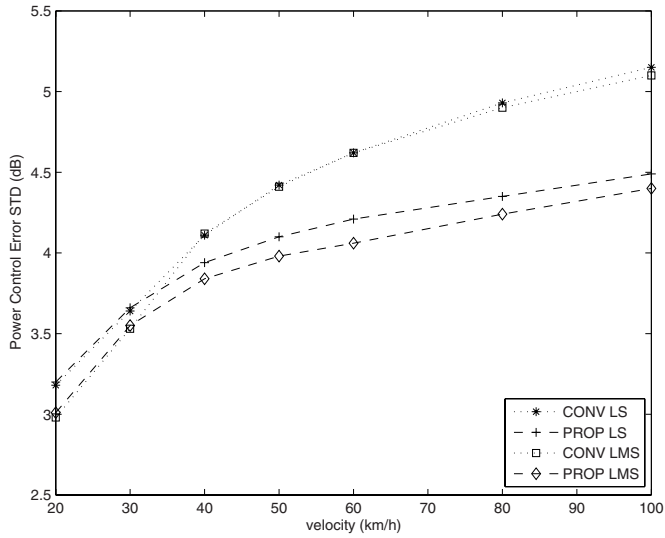


**Fig. 4.** Power control error standard deviation  $\sigma_Z$  (dB) versus mobile speed  $v$  for conventional and proposed predictive CLPCs over noisy Rayleigh fading channels

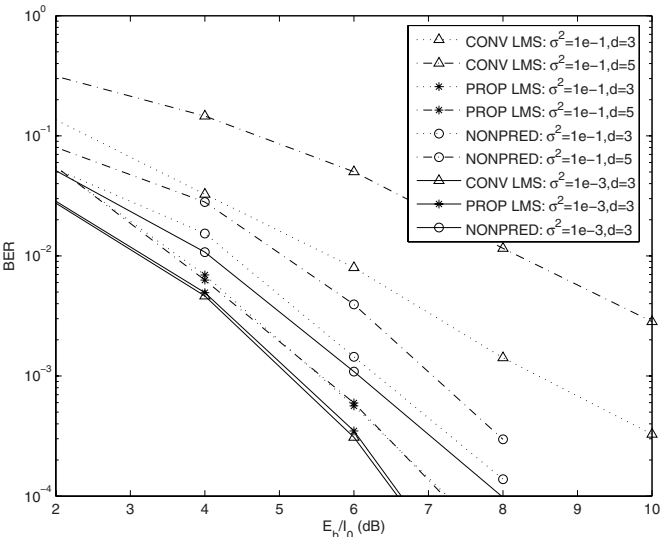


**Fig. 5.** BER performance of proposed predictive CLPC and conventional predictive and nonpredictive CLPCs over noisy Rayleigh fading channels when assuming coded BPSK transmission

## 5 Conclusion

In this paper, we have proposed a novel predictive CLPC with a low-rate multi-step linear AR predictor which effectively compensates multiple PCG delays over noisy mobile fading channels. For the imperfect channel estimation conditions, the low-rate prediction is more robust to the multiples PCG delays than the high-rate prediction, especially in the presence of imperfect channel estimation. As the mobility increases proposed CLPC reduces MSE further than conventional CLPC. The Monte-Carlo simulation results demonstrate the superiority of the proposed CLPC over conventional nonpredictive or predictive CLPCs.

**Acknowledgments.** “This research was supported by the MIC, Korea, under the ITRC support program supervised by the IITA” (IITA-2006-(C1090-0602-0011)).

## References

1. K. Leibnitz, P. Tran-Gia, and J. E. Miller, “Analysis of the dynamics of CDMA reverse link power control,” in *Proc. IEEE GLOBECOM*, Sydney, Australia, pp. 2109-2114, Nov. 1998.
2. F. Gunnarsson, F. Gustafson, and J. Blom, “Dynamical effects of time delays and time delay compensation in power controlled DS-CDMA,” *IEEE J. on Select. Areas in Commun.*, vol. 19, no. 1, pp. 141-151, Jan. 2001.
3. D. M. Ionescu and A. Boariu, “Predictability of uplink channels from downlink channels in frequency division duplex systems: a power control application,” *IEEE Trans. Wireless Commun.*, vol. 3, no. 5, pp. 1781-1788, Sept. 2004.
4. A. Duel-Hallen, S. Hu, and H. Hallen, “Long-range prediction of fading signals enabling adapting transmission for mobile radio channels,” *IEEE Signal Processing Magazine*, vol. 17, no. 3, pp. 62-75, may 2000.
5. M. Rintamaki, H. Koivo, and I. Hartimo, “Adaptive closed-loop power control algorithms for CDMA cellular communications systems,” *IEEE Trans. Veh. Technol.*, vol. 53, no. 6, pp. 1756-1768, Nov. 2004.
6. B-K Lee, H-W Chen, and B-S Chen, “Power control of cellular radio systems via robust Smith prediction filter,” *IEEE Trans. Wireless Commun.*, vol. 3, no. 5, pp. 1822-1831, Sept. 2004.
7. T. Eyceoz, A. Duel-Hallen, and H. Hallen, “Deterministic channel modeling and long range prediction of fast fading mobile radio channels,” *IEEE Commun. Lett.*, vol. 2, no. 9, pp. 254-256, Sept. 1998.
8. L. Song, N. B. Mandayam, and Z. Gajic, “Analysis of an up/down power control algorithm for the CDMA reverse link under fading,” *IEEE J. on Select. Areas in Commun.*, vol. 19, no. 2, pp. 227-286, Feb. 2001.
9. S. Choe, H. M. Kwon, and M. Uysal, “Performance analysis of imperfect closed-loop power control over Rayleigh fading,” *IEE lElectron. Lett.*, vol. 41, no. 19, pp. 1071-1073, Sept. 2005.

10. H.-J. Su and E. Geraniotis, "Adaptive closed-loop power control with quantized feedback and loop filtering," *IEEE Trans. Wireless Commun.*, vol. 1, no. 1, pp. 76-86, Jan. 2002.
11. D. N. Kalofonos, M. Stojanovic, and J. G. Proakis, "Performance of adaptive MC-CDMA detectors in rapidly fading Rayleigh channels," *IEEE Trans. Wireless Commun.*, vol. 2, no. 2, pp. 229-239, Mar. 2003.
12. W. C. Jakes, *Microwave Mobile Communications*, New York: Wiley, 1974.
13. S. Haykin, *Adaptive filter theory*, Prentice-Hall, Inc., 1996.



# Author Index

- Avrachenkov, Konstantin 30  
Azcorra, Arturo 91, 129
- Bagnulo, Marcelo 129  
Baiocchi, Andrea 223  
Bang, Ki-Seok 67  
Braun, Torsten 247  
Brouwer, Frank 177  
Bueno-Delgado, M.V. 279
- Cerqueira, Eduardo 79  
Cetin, Bilge 201  
Chai, Wei Koong 17  
Choe, Sangho 370  
Chung, Hee-Jeong 346  
Conway, Eoghan 43  
Cortez, Paulo 141
- Dai, Tran Thanh 235  
de Bruin, Irene 177  
De Cicco, Luca 153  
Dunaytsev, Roman 30
- Egea-López, E. 279  
Ekeröth, Ted 315  
Erten, Y. Murat 103
- Fallon, Enda 43  
Feeney, Laura Marie 201  
Fu, Yusun 177
- García, Jaime 91  
García-Haro, J. 279  
García-Martínez, Alberto 129  
Georgoulas, Stylianos 115  
Gonzalez, Oscar F. 302
- Hamadani, Ehsan 1  
Hanley, Austin 43  
Harju, Jarmo 30  
Hayes, Gregory 43  
Ho, Kin-Hon 115  
Hollos, Daniel 201  
Hong, Choong Seon 235  
Howarth, Michael 302
- Javaheri, Hooman 259  
Joung, Jinoo 189
- Kang, Yeon-Joo 358  
Karaliopoulos, Merkourios 17  
Karl, Holger 201  
Kihl, Maria 315  
Kim, Eun-Ju 358  
Kim, Hyun-Ki 326  
Kim, Mee-Ran 346, 358  
Kim, Nak-Myeong 346, 358  
Koucheryavy, Yevgeni 30  
Kubisch, Martin 201  
Kwon, Wook Hyun 213
- Laouti, Anis 290  
Lee, Ki-Chang 338  
Lee, SuKyoung 271  
Lee, Tae-Jin 326  
Li, Mingxiu 55  
Lim, Jongin 67  
Lim, Sun-Hee 67  
Liu, Yi 55
- Martínez-Sala, A.S. 279  
Mascolo, Saverio 153  
Matthews, Adrian 43  
Meer, Michael 247  
Mendes, Paulo 79  
Mengesha, Seble 201  
Monteiro, Edmundo 79  
Moon, Tae-Hyun 338  
Murphy, John 43  
Murphy, Liam 43
- Naimi, Amina Meraihi 290  
Noubir, Guevara 259
- Palmisano, Vittorio 153  
Papadimitriou, Panagiotis 165  
Park, Hong Seong 213  
Park, Jae Hee 213  
Pavlou, George 17, 115, 302  
Pinto, Vitor 91
- Qian, Depei 55  
Qiao, Yuansong 43

- Rakocevic, Veselin 1  
Ribeiro, Vitor 91  
Rio, Miguel 141  
Rocha, Miguel 141  
Rodolakis, Georgios 290  
Rozenberg, Michael 315
- Seo, SungHoon 271  
Shin, Soo Young 213  
Sichitiu, Mihail 315  
Skoczylas, Piotr 247  
Song, JooSeok 271  
Sousa, Pedro 141
- Tomur, Emrah 103  
Trimintzios, Panos 115  
Tsaoussidis, Vassilis 165
- Vacirca, Francesco 223  
Valera, Francisco 91  
Vales-Alonso, J. 279  
Veloso, Luis 79  
Vidal, Iván 91
- Wälchli, Markus 247  
Wang, Yin 259  
Whillans, Neill 177  
Wu, Weiguo 55
- Xiao, Youqian 177
- Yang, Bo 55  
Yi, Okyeon 67  
Yoon, Dae Gil 213  
Yu, Hye-In 346
- Zhu, Xiaosong 43