Lecture Notes in Computer Science 2811

Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

Springer
Berlin
Heidelberg
New York
Hong Kong
London
Milan
Paris
Tokyo

Gunnar Karlsson   Michael I. Smirnov (Eds.)

Quality for All

4th COST 263 International Workshop on
Quality of Future Internet Services, QoFIS 2003
Stockholm, Sweden, October 1-2, 2003
Proceedings

Springer

Volume Editors

Gunnar Karlsson
Royal Institute of Technology
Department of Microelectronics and Information Technology
Electrum 229, 16440 Kista, Sweden
E-mail: gk@imit.kth.se

Michael I. Smirnov
FHI FOKUS
Kaiserin-Augusta Allee 31, 10589 Berlin, Germany
E-mail: smirnov@fokus.fraunhofer.de

Cataloging-in-Publication Data applied for

A catalog record for this book is available from the Library of Congress.

Bibliographic information published by Die Deutsche Bibliothek
Die Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliografie;
detailed bibliographic data is available in the Internet at <http://dnb.ddb.de>.

CR Subject Classification (1998): C.2, D.2, H.3, H.4, H.5

ISSN 0302-9743
ISBN 3-540-20192-0 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

Typesetting: Camera-ready by author, data conversion by Olgun Computergrafik
Printed on acid-free paper SPIN: 10949979 06/3142 5 4 3 2 1 0

Preface

The Internet has nearly a ten year history as a global, public communication infrastructure. The two applications that have created the demand from private and business users have been the World-Wide Web and electronic mail. We have in the last five years seen the rapidly emerging popularity of peer-to-peer sharing of files, mostly for music, and to a more limited extent also the introduction of Internet telephony, television, and radio. These services place demands on the infrastructure that are higher with respect to quality and connectivity than web surfing and e-mail.

Mobile (cellular) telephony has rivaled the Internet with respect to growth during the last decade. The hitherto separate networks are now set to merge into a mobile Internet that will give wireless access to all Internet services. The ambition behind the Internet's continuing development is that it should serve as a general-purpose infrastructure and provide adequate support for all types of applications in terms of quality, connectivity, and cost. Thus the demands made on all Internet services must also be met by wireless access, and the circuit quality of a voice connection for mobile telephony must also be provided in the wired IP networks.

This volume of the Lecture Notes in Computer Science series contains 22 research papers that address in particular the problems associtated with providing quality of service to communication applications. The contributions pertain to traffic engineering and quality-of-service routing, performance evaluation, explicit mechanisms and methods for the provision of quality, and to quality of service in wireless access networks. The goal of the research community is to ensure sufficient quality from network services and end systems so that the communication applications appear natural to use and the intermediating systems do not interfere in the information exchange between the persons or the machines. We wish to support all applications for every user over any network; in short, to provide *quality for all!*

The papers in this volume were accepted for the Fourth COST Action 263 International Workshop on Quality of Future Internet Services, QoFIS 2003. It took place on October 1–2, 2003 at the Royal Academy of Engineering Sciences in Stockholm, Sweden, and was arranged by the Laboratory for Communication Networks of KTH, the Royal Institute of Technology. QoFIS 2003 followed the highly successful workshops in Zurich in 2002, Coimbra in 2001, and Berlin in 2000. It was the last workshop for the COST Action 263; it continues under the auspices of the European Union Network of Excellence E-NEXT. The workshop was organized in seven sessions and featured two invited talks by Dr. James Roberts of France Telecom R&D and Prof. Jon Crowcroft of the University of Cambridge. In addition to the main technical program, the third day of the workshop on October 3 was dedicated to QoS in wireless networks. The program consisted of invited presentations and was organized by Prof. Jens Zander of the KTH Center for Wireless Systems and Dr. Bengt Ahlgren of the Swedish Institute of Computer Science.

The workshop received 73 submissions, which underwent strict peer review by members of the program committee or reviewers assigned by them; each member provided on average six reviews. It is our pleasure to acknowledge the excellent work of the program committee in helping to select the papers for the program from the submissions. This work was done in addition to the daily business and we were fortunate to have such a committed and careful group of experts to help us.

The arrangements for the workshop were handled by a wonderfully dedicated local organizations committee, by Prof. Peter Sjdin. They managed the ConfMan paper handling system, the QoFIS web site (now with its own registered domain), the preparation of the camera-ready papers for the proceedings in this volume, and much more. We would also like to thank Mrs Barbro Redin, the secretary for LCN at KTH, who did most of the work for the conference regarding accommodations, registrations, and social events. We also extend our gratitude to our sponsors, in particular Vinnova, the Swedish Agency for Innovation Systems.

We learned about the untimely death of our dear colleage Prof. Olga Casals on June 11 and would like to dedicate this volume to her memory.

October 2003

Gunnar Karlsson
Michael Smirnov

QoFIS

Organization

QoFIS 2003 was organized by the Laboratory for Communication Networks (LCN) of KTH, the Royal Institute of Technology (Sweden).

Executive Committee

General Chair:	Gunnar Karlsson	KTH, Sweden
Technical Program Co-chair:	Michael Smirnov	FhG FOKUS, Germany
Technical Program Co-chair:	Gunnar Karlsson	KTH, Sweden
Local Organization Chair:	Peter Sjödin	KTH, Sweden

Steering Committee

Fernando Boavida	University of Coimbra, Portugal
Jon Crowcroft	University of Cambridge, UK
Gunnar Karlsson	KTH, Sweden
James Roberts	France Telecom R&D, France
Michael Smirnov	FhG FOKUS, Germany
Burkhard Stiller	UniBw Munich, Germany and ETH Zürich, Switzerland

Program Committee

Bengt Ahlgren	SICS, Sweden
Arturo Azcorra	UC3M, Spain
Hans van den Berg	TNO Telecom, The Netherlands
Chris Blondia	Univ. Antwerp, Belgium
Torsten Braun	Univ. of Bern, Switzerland
Fernando Boavida	Univ. of Coimbra, Portugal
Olivier Bonaventure	UCL, Belgium
Georg Carle	Univ. Tübingen, Germany
Olga Casals	UPC, Spain
Jon Crowcroft	Univ. of Cambridge, UK
Michel Diaz	LAAS, France
Jordi Domingo-Pascual	UPC, Spain
Peder Emstad	NTNU, Norway
Gísli Hjalmtýsson	Reykjavik Univ., Iceland
David Hutchison	Univ. Lancaster, UK
Yevgeni Koucheryavy	TUT, Finland
Marko Luoma	HUT, Finland
Hermann de Meer	Univ. College London, UK
Edmundo Monteiro	Univ. of Coimbra, Portugal
Rob van der Mei	TNO Telecom, The Netherlands
Giovanni Pacifici	IBM TJ Watson, USA
George Pavlou	Univ. of Surrey, UK
Guido Petit	Alcatel, Belgium
Thomas Plagemann	Univ. of Oslo, Norway
Mihai Popa	Procetel, Romania
George Polyzos	AUEB, Greece
Dimitris Serpanos	Univ. of Patras, Greece
Vasilios Siris	ICS-FORTH and Univ. of Crete, Greece
Michael Smirnov	FhG FOKUS, Germany
Josep Solé-Pareta	UPC, Spain
Ioannis Stavrakakis	Univ. of Athens, Greece
Burkhard Stiller	UniBw Munich, Germany and ETH Zürich, Switzerland
Piet Van Mieghem	Delft Univ. of Tech., The Netherlands
Giorgio Ventre	Univ. of Napoli, Italy
Jorma Virtamo	HUT, Finland
Lars Wolf	TU Braunschweig, Germany
Adam Wolisz	TU Berlin, Germany

Local Organization Committee

Juan Alonso
György Dán
Henrik Lundqvist
Ian Marsh
Ignacio Más
Evgeny Ossipov
Peter Sjödin
Héctor Velayos

Reviewers

S. Aalto
A. Acharya
J. Albrecht
J. Alonso
P. Antoniadis
G. Auriol
M. Bagnulo
P. Barlet-Ros
M. Bechler
O. Brun
B. Brynjúlfsson
D. Careglio
Ll. Cerdà
S. Chan
C. Chassot
M. Curado
J. Cushnie
J. Diederich
M. Dramitinos
E. Efstathiou
A. Garcia
S. Georgouls
J. Gerke
C. Griwodz
J. Harju
Hasan
D. Hausheer
T. Heinis
M. Heissenbüttel
Ó. Helgason
A. Houyou
M. Howarth
N. Huu Thanh
E. Hyytiä
J. Karvo
M. Klinkowski
T. Koulouris
P. Kurtansky
P. Kuusela
V. Laatu
N. Larrieu
D. Larrabeiti
P. Lassila
A. Makki
J. Mangues-Bafalluy
I. Más
X. Masip-Bruin
J. Mischke
D. Moltchanov
J. Orvalho
P. Owezarski
A. Papaioannou
C. Pelsser
P. Pinto
P. Racz
C. Reichert
P. Reviriego
J. Da Silva
P. Salvione
S. Sanchez-Lopez
A. Santos
S. Sivasubramanian
I. Soto
S. Soursos
H. Sverrisson
A. Tantawi
S. Uhlig
H. Velayos
J. Walpole
N. Wang

Sponsoring Institutions

VINNOVA
VERKET FÖR INNOVATIONSSYSTEM

Emerging NETwork technologies

Wireless@KTH

Center for Wireless Systems

KTH
VETENSKAP OCH KONST
ROYAL INSTITUTE OF TECHNOLOGY

Table of Contents

Performance Analysis

Quality of Service Provisioning

Traffic Engineering and Routing

Local Area and Multi-hop Wireless Networks

Cellular Networks

On the Impacts of Traffic Shaping on End-to-End Delay Bounds in Aggregate Scheduling Networks

Markus Fidler

Department of Computer Science, Aachen University
Ahornstr. 55, 52074 Aachen, Germany
fidler@i4.informatik.rwth-aachen.de

Abstract. The Differentiated Services architecture allows for the provision of scalable Quality of Service by means of aggregating flows to a small number of traffic classes. Among these classes a Premium Service is defined, for which end-to-end delay guarantees are of particular interest. However, in aggregate scheduling networks the derivation of such worst case delays is significantly complicated and the derived bounds are weakened by the multiplexing of flows to aggregates.

A means to minimize the impacts of interfering flows is to shape incoming traffic, so that bursts are smoothed. Doing so reduces the queuing delay within the core of the domain, whereas an additional shaping delay at the edge is introduced. In this paper we address the issue of traffic shaping analytically. We derive a form that allows to quantify the impacts of shaping and we show simulation results on the derivation of end-to-end delay bounds under different shaping options.

1 Introduction

Differentiated Services (DS) [2] addresses the scalability problems of the former Integrated Services approach by an aggregation of flows to a small number of traffic classes. Packets are identified by simple markings that indicate the respective class. In the core of the network, routers do not need to determine to which flow a packet belongs, only which aggregate behavior has to be applied. Edge routers mark packets and indicate whether they are within profile or, if they are out of profile, in which case they might even be discarded at the edge router. A particular marking on a packet indicates a so-called Per Hop Behavior (PHB) that has to be applied for forwarding of the packet. The Expedited Forwarding (EF) PHB [8] is intended for building a service that offers low loss and low delay, namely a Premium Service. For this purpose delay bounds are derived for a general topology and a defined load in [4]. However, these bounds can be improved, when additional information concerning the current load and the special topology of a certain DS domain is available.

In [12] a resource manager for DS domains called Bandwidth Broker (BB) is conceptualized. The task of a BB in a DS domain is to perform a careful admission control and to set up the appropriate configuration of the domain's edge routers. While doing so, the BB knows about all requests for resources of

G. Karlsson and M.I. Smirnov (Eds.): QofIS 2003, LNCS 2811, pp. 1–10, 2003.

certain Quality of Service (QoS) classes. Besides it can learn about the domain's topology by implementing a routing protocol listener. Thus, a BB can have access to all information that is required, to base the admission control on delay bounds that are derived for individual flows, for the current load, and for the actual mapping of flows onto the topology of the administrated domain, applying the mathematical methodology of Network Calculus [15].

In this paper we investigate the impacts of traffic shaping on end-to-end delay bounds. The remainder of this paper is organized as follows: In section 2 the required background on Network Calculus is given. Section 3 addresses the impacts of traffic shaping. Related simulation results are given in section 4. Section 5 concludes the paper. Proofs are given in the appendix.

2 Network Calculus

Network Calculus is a theory of deterministic queuing systems based on the calculus for network delay presented in [5,6] and on Generalized Processor Sharing in [13,14]. Extensions, and a comprehensive overview on current Network Calculus are given in [3,11]. Here only a few concepts are covered briefly. In the sequel upper indices j indicate links and lower indices i indicate flows.

Flows can be described by arrival functions $F(t)$ that are given as the cumulated number of bits seen in an interval $[0, t]$. Arrival curves $\alpha(t)$ are defined to give an upper bound on the arrival functions with $\alpha(t_2 - t_1) \geq F(t_2) - F(t_1)$ for all $t_2 \geq t_1 \geq 0$. In DS networks, a typical constraint for incoming flows can be given by the affine arrival curve $\alpha_{r,b}(t) = b + r \cdot t$. Usually the ingress router of a DS domain meters incoming flows against a leaky bucket algorithm, which allows for bursts b and a rate r. Non-conforming traffic is either shaped or dropped.

The service that is offered by the scheduler on an outgoing link can be characterized by a minimum service curve, denoted by $\beta(t)$. A special characteristic of a service curve is the rate-latency type that is given by $\beta_{R,T}(t) = R \cdot [t - T]^+$, with a rate R and a latency T. The term $[x]^+$ is equal to x, if $x \geq 0$, and zero otherwise. Service curves of the rate-latency type are implemented for example by Priority Queuing (PQ) or Weighted Fair Queuing (WFQ).

Based on the above concepts, bounds for the backlog, the delay, and for the output flow can be derived. If a flow i that is constrained by α_i^j is input to a link j that offers a service curve β^j, the output arrival curve α_i^{j+1} of flow i can be given by (1).

$$\alpha_i^{j+1}(t) = \sup_{s \geq 0}[\alpha_i^j(t+s) - \beta^j(s)] \tag{1}$$

Multiplexing of flows can simply be described by addition of the belonging arrival functions, respective arrival curves. For aggregate scheduling networks with FIFO service elements, families of per-flow service curves $\beta_\theta^j(t)$ according to (2), with an arbitrary parameter $\theta \geq 0$ are derived in [7,11]. $\beta_\theta^j(t)$ gives a family of service curves for a flow 1 that is scheduled in an aggregate manner with a flow, or a sub-aggregate 2 on a link j. The term $1_{t>\theta}$ is zero for $t \leq \theta$.

$$\beta_\theta^j(t) = [\beta^j(t) - \alpha_2(t-\theta)]^+ 1_{t>\theta} \tag{2}$$

3 Traffic Shaping

A means to reduce the impacts of interfering bursts on network performance is to shape incoming traffic at the edge of a domain [12]. Queuing of the initial bursts is in this case performed at the edge, with the aim to minimize the delay within the core. Especially, if heterogeneous aggregates have to be scheduled, shaping allows to reduce impacts of different types of flows on each other [15]. However, to our knowledge the work on shapers in [10] has not been extended to aggregate scheduling networks and an analysis of the effects of shaping on the derivation of end-to-end delay bounds is missing in current literature.

Theorem 1 (Bound for Output, General Case) *Consider two flows* 1, *and* 2 *that are* α_1^j, *respective* α_2^j *upper constrained. Assume these flows are served in FIFO order and in an aggregate manner by a node* j *that is characterized by a minimum service curve of the rate-latency type* $\beta_{R,T}^j$. *Then, the output of flow* 1 *is* α_1^{j+1} *upper constrained according to (3), where* θ *is a function of* t *and has to comply with (4).*

$$\alpha_1^{j+1}(t) = \alpha_1^j(t+\theta) \tag{3}$$

$$\theta(t) = \frac{\sup_{v>0}[\alpha_1^j(v+t+\theta(t)) - \alpha_1^j(t+\theta(t)) + \alpha_2^j(v) - R^j \cdot v]}{R^j} + T^j \tag{4}$$

Corollary 1 (Bound for Output, Single Leaky Bucket) *In case of a leaky bucket constrained flow* 1, *with rate* r_1 *and burst size* b_1^j, *(4) can be simplified applying* $\alpha_1^j(v+t+\theta(t))-\alpha_1^j(t+\theta(t)) = r_1 \cdot v$. *As an immediate consequence,* θ *becomes independent of* t. *With (3) we find that the output flow* 1 *is leaky bucket constrained with* r_1 *and* $b_1^{j+1} = \alpha_1(\theta)$. *Further on, if the flow or sub-aggregate* 2 *is leaky bucket constrained with rate* r_2 *and burst size* b_2^j, *the* $\sup[\dots]$ *in (4) is found for* $v \to 0$ *resulting in* $\theta = b_2^j/R^j + T^j$ *and* $b_1^{j+1} = b_1^j + r_1 \cdot (T^j + b_2^j/R^j)$.

Definition 1 (Sustainable Rate Shaping) *Assume a flow that is leaky bucket constrained with the parameters* $(r_1, \bar{b}_1)$, *where* r_1 *is called the sustainable rate. If this flow is input to a traffic shaper that consists of a bit-by-bit system with a shaping rate* r_1, *and a packetizer with a maximum packet size* $l_{\max}$, *the output flow is constrained by* $(r_1, b_1 = l_{\max})$ *[10]. Further on, the shaper adds a worst-case delay of* $\bar{b}_1/r_1$.

In [11] it is shown that shaping at the sustainable rate does not worsen the end-to-end delay bounds in Integrated Services networks, if the reserved rate matches the shaping rate, and in turn the rate of the flow. However, this assumption does not hold true for DS domains. DS Premium resources are intended to be reserved statically and PQ is a likely means of implementation. Thus, the allocated Premium capacity usually exceeds the capacity that is requested by Premium traffic sources. In this scenario shaping at the sustainable rate can significantly worsen delay bounds, whereas Premium bursts that are not shaped can result in unwanted interference and increase queuing delays in the core of the domain. Hence, adaptivity when setting the shaping rates is required.

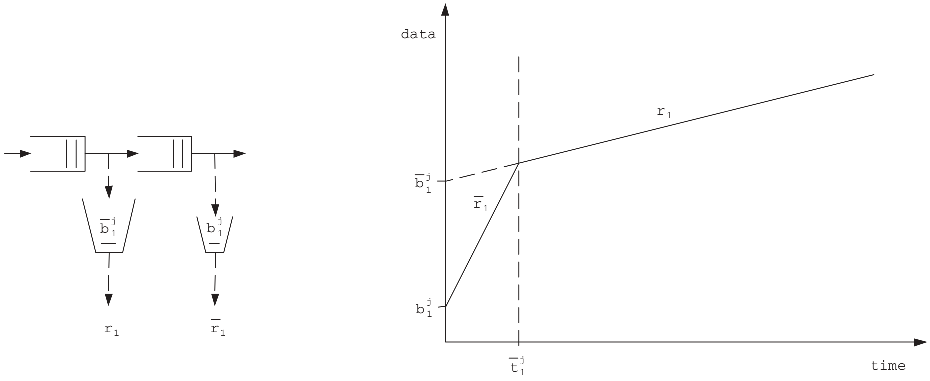


Fig. 1. Two Leaky Bucket Constraint.

Definition 2 (Two Leaky Bucket Constraint) *Consider a two leaky bucket configuration according to figure 1. Define* $(r_1, \overline{b}_1^j)$*, and* $(\overline{r}_1, b_1^j)$ *to be the parameters of the first, respective second leaky bucket, with* $\overline{r}_1 > r_1$ *and* $\overline{b}_1^j > b_1^j$*. The resulting arrival curve is defined in (5). It allows for bursts of size* b_1^j*, then it ascends by* $\overline{r}_1$ *until* $\overline{t}_1^j = (\overline{b}_1^j - b_1^j)/(\overline{r}_1 - r_1)$*, and finally it increases with rate* r_1*.*

$$\alpha_1^j(t) = \min[b_1^j + \overline{r}_1 \cdot t, \overline{b}_1^j + r_1 \cdot t] = \begin{cases} b_1^j + \overline{r}_1 \cdot t & , \textit{if } t \leq \overline{t}_1^j = \frac{\overline{b}_1^j - b_1^j}{\overline{r}_1 - r_1} \\ \overline{b}_1^j + r_1 \cdot t & , \textit{else} \end{cases} \tag{5}$$

An arrival curve of the type in (5) can be given, if a leaky bucket constrained flow with the arrival curve $\alpha_1^j(t) = \overline{b}_1^j + r_1 \cdot t$ traverses a combination of a bit-by-bit traffic shaper with rate $\overline{r}_1$ and a packetizer with a maximum packet size $l_{\max}$ [10]. Then, the output arrival curve is two leaky bucket constrained with the parameters $(r_1, \overline{b}_1^j)$ and $(\overline{r}_1, l_{\max})$. The shaper adds a worst-case delay of $\overline{b}_1^j/\overline{r}_1$.

Theorem 2 (Bound for Output, Two Leaky Bucket) *Consider two flows* 1 *and* 2 *that are* α_1^j*, respective* α_2^j *upper constrained. Assume that these flows are served in FIFO order and in an aggregate manner by a node* j *that is characterized by a minimum service curve of the rate-latency type* β_{R^j,T^j}^j*. If the input flow* 1 *is constrained by two leaky buckets with* $(r_1, \overline{b}_1^j)$*,* $(\overline{r}_1, b_1^j)$*, and* $\overline{t}_1^j = (\overline{b}_1^j - b_1^j)/(\overline{r}_1 - r_1)$*, the output flow is two leaky bucket constrained with* $(r_1, \overline{b}_1^{j+1})$*, and* $(\overline{r}_1, b_1^{j+1})$*, where* $b_1^{j+1} = \alpha_1^j(\theta(0))$ *and* $\overline{b}_1^{j+1} = \overline{b}_1^j + r_1 \cdot \theta(\overline{t}_1^j)$*.*

Definition 3 (Minimum Interference Shaping) *We define minimum interference shaping to be a configuration, where all flows* i *with* $(r_i, \overline{b}_i^j)$ *of the set of flows* $\mathbb{I}$ *that form an aggregate are shaped with a rate* $\overline{r}_i$*, so that* $\sum_{i=\mathbb{I}_j} \overline{r}_i \leq R^j$ *holds on all links* j *of the set of links* $\mathbb{J}$ *of the domain, where* $\mathbb{I}_j$ *is the set of flows* i *that traverse a link* j*. Thus, flows are constrained by* $(r_i, \overline{b}_i)$ *and* $(\overline{r}_i, b_i = l_{\max})$*.*

For the settings given in definition 3 the sup[...] in (4) is found on all links $j \in \mathbb{J}$ for any $t \geq 0$ with $v \rightarrow 0$, whereby θ is constant over time with $\theta = b_2^j/R^j + T^j$. Hence, the impact of interfering flows is reduced to the impact of their effective burst size after shaping. The output constraint of the flow of interest 1 that is scheduled in an aggregate manner with a flow, or a sub-aggregate 2 on a link j is given by the parameters $(\overline{r}_1, b_1^{j+1} = b_1^j + \overline{r}_1 \cdot (b_2^j/R^j + T^j))$, and $(r_1, \overline{b}_1^{j+1} = \overline{b}_1^j + r_1 \cdot (b_2^j/R^j + T^j))$, with $\overline{t}_1^{j+1} = (\overline{b}_1^j - b_1^j)/(\overline{r}_1 - r_1) - (b_2^j/R^j + T^j)$. If $\overline{t}_1^{j+1} \leq 0$, the output constraint is reduced to a single leaky bucket constraint.

For over-provisioned links minimum interference shaping allows for a variety of settings of the per-flow shaping rates $\overline{r}_i$. However, the use of high priority traffic classes, such as a PQ-based Premium class, can lead to starvation of other services including the Best-Effort (BE) Service. Thus, it is reasonable to limit the Premium burst size by shaping and to restrict the overall Premium rate, as is supported by various router implementations. Here, we define a parameter d_q to give an upper bound on the allowed queuing delay of Premium traffic within the core of the network, from which an upper bound of the Premium burst size can be derived. To set up corresponding shaping rates $\overline{r}_i$, we apply a two step approach. The maximum allowed shaping delay d_{s_i} is computed as $d_{s_i} = d_{r_i} - (d_{t_i} + d_q)$ with d_{r_i} denoting the requested maximum delay for flow i, and d_{t_i} giving the end-to-end propagation delay on the corresponding path. If $d_{s_i} > 0$, the corresponding shaping rate follows as $\overline{r}_i = \max[r_i, \overline{b}_i/d_{s_i}]$, otherwise the target delay cannot be guaranteed. Then, if still all of the conditions in definition 3 hold, and, if the queuing delay within the core can be derived by Network Calculus to be smaller than d_q for all flows $i \in \mathbb{I}$, a solution is found. Note that the configuration of the shapers does not have to be updated during the lifetime of the corresponding flows, since all shaping rates $\overline{r}_i$ are set to account for queuing delay of at most d_q. Thus, the approach scales similar to sustainable rate shaping. If the rate of the Premium traffic shall be restricted in addition, the conditions in definition 3 have to be replaced by stricter ones.

4 Evaluation Results

We implemented an admission control for an application within the framework of a Bandwidth Broker [15]. The admission control currently knows about the topology of the domain statically, whereas a routing protocol listener can be added. Requests for Premium capacity are signalled in a Remote Procedure Call (RPC) style. The requests consist of a Committed Information Rate (CIR), a Committed Burst Size (CBS), and a target maximum delay. Whenever the admission control receives a new request, it computes the end-to-end delay for all requests that are active concurrently. If none of the target maximum per-flow delays is violated, the new request is accepted, which otherwise is rejected. Note that requests are usually made for aggregated traffic flows that use the same path from the ingress to the egress router to allow for scalability.

For performance evaluation a simulator that generates such Premium resource requests is used. Sources and sinks are chosen uniformly from a predefined set. Start and end times are modelled as negative exponentially distributed with a mean inter-arrival time $1/\lambda$ and a mean service time $1/\mu$. Thus, $\rho = \lambda/\mu$ can be defined as the network load, that is the mean number of concurrently active requests. The target delay, CIR, and CBS are either used as parameters or as uniformly distributed random variables for the following simulations.

Different topologies have been used [9], whereby the results that are included in this paper have been obtained for the G-WiN topology of the German research network (DFN) [1]. The level one nodes of this topology are core nodes. End systems are connected to the level two nodes that are edge nodes. Links are either Synchronous Transfer Mode (STM) 4, STM 16, or STM 64 connections. The link propagation delay is assumed to be 2 ms. Shortest Path First (SPF) routing is applied to minimize the number of hops along the paths. Further on, Turn Prohibition (TP) is used to ensure the feed-forward property of the network that is required for a direct application of Network Calculus [16]. For the G-WiN topology the combination of SPF and TP increases the length of only one path by one hop compared to SPF routing. Simulation results of a Guaranteed Rate Service, which only considers capacity constraints, have shown that the impacts of TP on SPF routing are comparably small [9]. Further on, the TP algorithm can be configured to prohibit turns that include links with a comparably low capacity with priority [16].

The emulated Premium Service is assumed to be based on PQ. Thus, service curves are of the rate-latency type. The latency is set to the time it takes to transmit 4 Maximum Transmission Units (MTU) of 9.6 kB, to account for non-preemptive scheduling, packetization, and a router internal buffer for 2 MTU.

Simulation results that compare the different shaping options are shown in figure 2. The performance measure that we apply is the ratio of accepted requests divided by the overall number of requests. Simulations have been run, until the 0.95 confidence interval of the acceptance ratio was smaller than 0.01. Requests for Premium capacity are generated by the simulator with random parameters. The CIR is chosen uniformly from 10 Mb/s to 80 Mb/s, the CBS from 1 Mb to 8 Mb, and the target worst case delay from 40 ms to 80 ms. The CIR and CBS are comparably large to model service requests for aggregated traffic trunks. In case of sustainable rate shaping, we find that the acceptance ratio drops to less than 0.2, independent of the actual load ρ with $0 \leq \rho \leq 400$. This is due to the static shaping configuration, which can result in comparably large shaping delays, independent of the requested delay bound. We address this shortcoming by minimum interference shaping, where shaping rates are adaptive. An end-to-end queuing delay of $d_q = 4$ ms respective $d_q = 8$ ms has been applied, to quantify the influence of the setting of d_q. However, we find only minor impacts of d_q in the investigated scenario. Minimum interference shaping allows to increase the acceptance ratio significantly compared to the option without shaping as well as compared to sustainable rate shaping, as can be seen from figure 2.

For illustrative purposes the cumulative density functions of the respective delay bounds are shown in figure 3 for a load of $\rho = 50$. The requested delay bounds are set to infinity to achieve an acceptance ratio of 1.0 for all of the

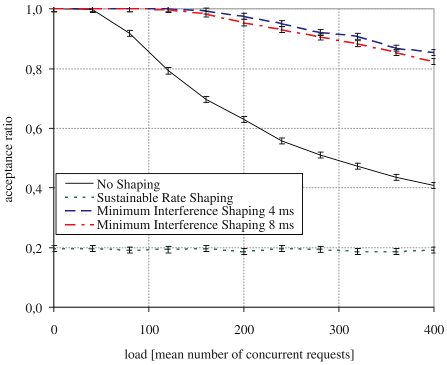


Fig. 2. Acceptance Ratio.

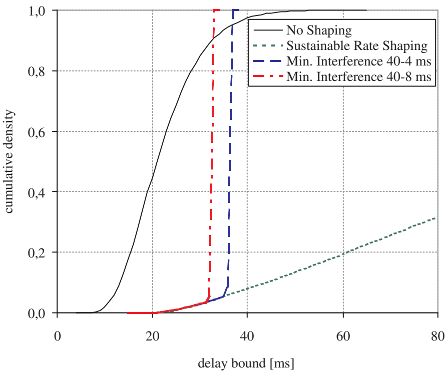


Fig. 3. Delay Bounds.

investigated shaping options, to allow for comparison. Here we find the reason for the bad performance of sustainable rate shaping. The delays that are introduced by shaping frequently exceed the range of 40 to 80 ms that is applied for figure 2. For a delay bound of infinity, minimum interference shaping applies the smallest possible shaping rate and becomes the same as sustainable rate shaping. Therefore, results are added for minimum interference shaping for a target delay of 40 ms. Figure 3 clearly shows the impacts of the parameter d_q. In case of $d_q = 4$ ms at most 4 ms of queuing delay are allowed to occur in the core of the network. Thus, shapers are configured so that the propagation delay and the shaping delay sum up to 36 ms for a target delay bound of 40 ms. In case of $d_q = 8$ ms, the propagation delay and shaping delay sum up to 32 ms, leaving room for up to 8 ms of queuing delay in the core of the network which, however, are not required for a load of $\rho = 50$.

Apart from the measured performance increase, traffic shaping is of particular interest, if a Guaranteed Rate Service and a Premium Service are implemented based on the same PHB. In this case a traffic mix with significantly heterogeneous traffic properties and service requirements results. For example Guaranteed Rate Transmission Control Protocol (TCP) streams that can have a large burstiness but no strict delay requirements can share the same PHB with extremely time critical Premium voice or video traffic. In this scenario traffic shaping can be applied to control the impacts of bursty Guaranteed Rate traffic on the Premium Service [15].

As a further benefit, traffic shaping reduces the impacts of EF on the BE class. The starvation of the BE class that can be due to priority scheduling of EF traffic is bound by the maximum EF burst size at the respective outgoing interface. Shaping EF bursts at the network's ingress nodes, reduces this burst size significantly, resulting in less impacts on the BE class. For the experiment in figure 3 and a load of $\rho = 50$ we find that the aggregated Premium burst size within the core of the network stays below 1 Mbit on all links, resulting in less than 0.5 ms BE starvation on a 2.4 Gb/s STM 16 link.

5 Conclusions

In this paper we have addressed the impacts of traffic shaping in aggregate scheduling networks. For this purpose the notation of two leaky bucket constrained arrival curves was introduced. A general per-flow-based service curve has been derived for a FIFO aggregate scheduling rate-latency service element. This form has been solved for the special case of a two leaky bucket constrained flow of interest and a two leaky bucket output constraint has been obtained.

We found that the shaping rate has to be chosen carefully in aggregate scheduling networks, wherefore we evolved an adaptive shaping scheme. Our scheme allows to configure shaping rates individually for a wide variety of heterogenous flows. It minimizes the interference within an aggregate scheduling domain, while it allows to support individual application-specific delay bounds. Thus, it can be applied to adapt end-to-end delay bounds to support heterogenous aggregates, while still allowing for scalability.

References

1. Adler, H.-M., *10 Gigabit/s Plattform für das G-WiN betriebsbereit*, DFN Mitteilungen, 60, 2002.
2. Blake, S., et al., *An Architecture for Differentiated Services*, RFC 2475, 1998.
3. Chang, C.-S., *Performance Guarantees in Communication Networks*, Springer, TNCS, 2000.
4. Charny, A., and Le Boudec, J.-Y., *Delay Bounds in a Network with Aggregate Scheduling*, Proceedings of QofIS, Springer, LNCS 1922, 2000.
5. Cruz, R. L., *A Calculus for Network Delay, Part I: Network Elements in Isolation*, IEEE Transactions on Information Theory, vol. 37, no. 1, pp 114-131, 1991.
6. Cruz, R. L., *A Calculus for Network Delay, Part II: Network Analysis*, IEEE Transactions on Information Theory, vol. 37, no. 1, pp 132-141, 1991.
7. Cruz, R. L., *SCED+: Efficient Management of Quality of Service Guarantees*, Proceedings of IEEE Infocom, 1998.
8. Davie, B., et al., *An Expedited Forwarding PHB*, RFC 3246, 2002.
9. Einhoff, G., *Quality of Service Routing for an IP Premium Service based on MPLS Traffic Engineering*, Master's Thesis, Aachen University, 2003.
10. Le Boudec, J.-Y., *Some Properties Of Variable Length Packet Shapers*, Proceedings of ACM Sigmetrics, 2001.
11. Le Boudec, J.-Y., and Thiran, P., *Network Calculus A Theory of Deterministic Queuing Systems for the Internet*, Springer, LNCS 2050, 2002.
12. Nichols, K., Jacobson, V., and Zhang, L., *A Two-bit Differentiated Services Architecture for the Internet*, RFC 2638, 1999.
13. Pareck, A. K., and Gallager, R. G., *A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case*, IEEE/ACM Transactions on Networking, vol. 1, no. 3, pp. 344-357, 1993.
14. Pareck, A. K., and Gallager, R. G., *A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Multiple-Node Case*, IEEE/ACM Transactions on Networking, vol. 2, no. 2, pp. 137-150, 1994.
15. Sander, V., *Design and Evaluation of a Bandwidth Broker that Provides Network Quality of Service for Grid Applications*, PhD Thesis, Aachen University, 2002.
16. Starobinski, D., Karpovsky, M., and Zakrevski, L., *Application of Network Calculus to General Topologies using Turn-Prohibition*, Proceedings of IEEE Infocom, 2002.

Appendix

Proof 1 (Proof of Theorem 1) Substitution of (2) in (1) and application of $\inf_{\theta \geq 0}[\dots]$ yields (6) for the output arrival curve α_1^{j+1} of flow 1.

$$\alpha_1^{j+1}(t) = \inf_{\theta \geq 0}\Big[\sup_{s \geq 0}\big[\alpha_1^j(t+s) - [\beta^j(s) - \alpha_2^j(s-\theta)]^+ 1_{s>\theta}\big]\Big] \tag{6}$$

With $\sup_{0 \leq s \leq \theta}[\alpha_1^j(t+s) - [\beta^j(s) - \alpha_2^j(s-\theta)]^+ 1_{s>\theta}] = \alpha_1^j(t+\theta)$, (7) follows.

$$\alpha_1^{j+1}(t) = \inf_{\theta \geq 0}\Big[\sup\big[\alpha_1^j(t+\theta), \sup_{s>\theta}[\alpha_1^j(t+s) - [\beta^j(s) - \alpha_2^j(s-\theta)]^+]\big]\Big] \tag{7}$$

Then, service curves of the rate-latency type $\beta^j_{R^j,T^j} = R^j \cdot [t - T^j]^+$ are assumed. The condition $R^j \cdot (s - T^j) - \alpha_2^j(s-\theta) \geq 0$ can be found to hold for $\theta \geq \theta'$ with $\theta' = \sup_{s>0}[\alpha_2^j(s) - R^j \cdot s]/R^j + T^j$ [11], whereby $\theta' \geq T^j$. For $\theta \geq \theta'$ (8) and (9) follow.

$$\alpha_1^{j+1}(t) = \inf_{\theta \geq \theta'}\Big[\sup\big[\alpha_1^j(t+\theta), \sup_{s>\theta}[\alpha_1^j(t+s) - R^j \cdot (s - T^j) + \alpha_2^j(s-\theta)]\big]\Big] \tag{8}$$

$$\alpha_1^{j+1}(t) = \inf_{\theta \geq \theta'}\Big[\sup\big[\alpha_1^j(t+\theta), \sup_{v>0}[\alpha_1^j(t+v+\theta) - R^j \cdot (v+\theta-T^j) + \alpha_2^j(v)]\big]\Big] \tag{9}$$

For different settings of θ a θ^* is defined as a function of $(t+\theta)$ in (10). With $\theta^* \geq \theta'$ (11) can be given.

$$\theta^*(t+\theta) = \frac{\sup_{v>0}[\alpha_1^j(t+v+\theta) - \alpha_1^j(t+\theta) + \alpha_2^j(v) - R^j \cdot v]}{R^j} + T^j \tag{10}$$

$$\sup_{v>0}[\alpha_1^j(t+v+\theta) - \alpha_1^j(t+\theta) + \alpha_2(v) - R^j \cdot v] - R^j \cdot (\theta - T^j) \lesseqgtr 0, \text{ if } \theta \gtreqless \theta^* \tag{11}$$

With (10), and (11) the outer $\sup[\dots]$ in (9) is solved in (12).

$$\begin{aligned}\alpha_1^{j+1}(t) = \inf\Big[&\inf_{\theta>\theta^*}\big[\alpha_1^j(t+\theta)\big], \inf_{\theta' \leq \theta \leq \theta^*}\big[\alpha_1^j(t+\theta) + \\ &\sup_{v>0}[\alpha_1^j(t+v+\theta) - \alpha_1^j(t+\theta) + \alpha_2^j(v) - R^j \cdot v] - R^j \cdot (\theta - T^j)\big]\Big]\end{aligned} \tag{12}$$

The $\inf[\dots]$ in (12) is found for $\theta = \theta^*$, which proofs theorem 1. Here, $\theta < \theta'$ is not investigated. Instead, it can be shown that the bound in theorem 1 is attained in the same way as for the special case of a single leaky bucket constrained flow 1 in [11]. Thus, we cannot find a better form for $\theta < \theta'$. □

Proof 2 (Proof of Theorem 2) Based on (4), $\theta(t)$ is derived here for a two leaky bucket constrained flow 1. For the flow 2 arrival curve sub-additivity is assumed without loss of generality.
Case 1 ($t = 0$) With $\alpha_1^{j+1}(t) = \alpha_1^j(t + \theta(t))$ according to (3) we find the output burst size $b_1^{j+1} = \alpha_1^j(\theta(0))$. Equation (4) is applied at $t = 0$ to find $\theta(0)$.

Case 2 ($0 < t < \overline{t}_1^j - v(t) - \theta(t)$) For this case (13) can be derived from (4), where $v(t)$ is the v for which the $\sup_v[\dots]$ in (4) is found.

$$\theta = \frac{\sup_{0<v\leq\overline{t}_1^j-t-\theta}[\overline{r}_1 \cdot v + \alpha_2^j(v) - R^j \cdot v]}{R^j} + T^j \tag{13}$$

Thus, θ is independent of t for $0 < t < \overline{t}_1^j - v - \theta$. With $\alpha_1^{j+1}(t) = \alpha_1^j(t+\theta)$ according to (3) the output arrival curve of flow 1 increases with $\overline{r}_1$.

Case 3 ($\overline{t}_1^j - v(t) - \theta(t) \leq t < \overline{t}_1^j - \theta(t)$) Equation (4) yields (14) for this case. Note that $b_1^j + \overline{r}_1 \cdot \overline{t}_1^j = \overline{b}_1^j + r_1 \cdot \overline{t}_1^j$.

$$\theta(t) = \frac{(\overline{r}_1 - r_1) \cdot (\overline{t}_1^j - t - \theta) + \sup_{v\geq\overline{t}_1^j-t-\theta}[r_1 \cdot v + \alpha_2^j(v) - R^j \cdot v]}{R^j} + T^j \tag{14}$$

$$= \frac{(\overline{r}_1 - r_1) \cdot (\overline{t}_1^j - t) + \sup_{v\geq\overline{t}_1^j-t-\theta}[r_1 \cdot v + \alpha_2^j(v) - R^j \cdot v] + T^j \cdot R^j}{R^j + \overline{r}_1 - r_1} \tag{15}$$

For $t > \overline{t}_1^j - v(t) - \theta(t)$ it can be immediately seen from (15) that any increase of t results in a corresponding decrease of θ by $(\overline{r}_1 - r_1)/(R^j + \overline{r}_1 - r_1)$. With $\alpha_1^{j+1}(t) = \alpha_1^j(t+\theta)$ according to (3) the output arrival curve of flow 1 increases with less than $\overline{r}_1$. Applying the leaky bucket parameters $(\overline{r}_1, b_1^{j+1})$ in theorem 2 overestimates the output arrival curve, which is allowed, since arrival curves are defined to give an upper bound on the respective arrival functions. However, as long as an increase of t results in a comparably smaller decrease of θ, smaller values v that fulfill $t \geq \overline{t}_1^j - v(t) - \theta(t)$ can be applied in (15). As a consequence, if the sup[...] in (15) is found for $t = \overline{t}_1^j - v(t) - \theta(t)$, it can occur that $t = \overline{t}_1^j - v(t) - \theta(t)$ also holds if t is increased by an infinitesimal Δt, resulting in a dependance of the sup[...] in (15) on t. For sub-additive flow 2 arrival curves, it can be shown that if t is increased, θ decreases slower than t increases. Here, for simplicity differentiable flow 2 arrival curves are assumed. Then, $\partial\alpha_2(t)/\partial t \geq R^j - \overline{r}_1$ at $t = \overline{t}_1^j - v(t) - \theta(t)$ holds, because otherwise case 2 would apply. By substitution of this condition in (15) it follows that θ decreases, if t increases. Further on, $\partial\alpha_2(t)/\partial t \leq R^j - r_1$ at $t = \overline{t}_1^j - v(t) - \theta(t)$ holds, wherefrom it can be found that θ decreases slower than t increases. Following the same argumentation as above, the leaky bucket parameters $(\overline{r}_1, b_1^{j+1})$ are applied.

Case 4 ($t \geq \overline{t}_1^j - \theta(t)$) In this case, (16) can immediately be derived from (4).

$$\theta = \frac{\sup_{v>0}[r_1 \cdot v + \alpha_2^j(v) - R^j \cdot v]}{R^j} + T^j \tag{16}$$

Note that $\theta(t)$ according to (16) is constant for $t \geq \overline{t}_1^j - \theta(t)$. With (3), the output arrival curve of flow 1 is given as $\alpha_1^{j+1}(t) = \alpha_1^j(t + \theta(t))$. The conditions $t + \theta(t) \geq \overline{t}_1^j$, and thus $\alpha_1^j(t+\theta(t)) = \overline{b}_1^j + r_1 \cdot (t + \theta(t))$ hold for $t \geq \overline{t}_1^j - \theta(t)$. Resulting, the output arrival curve of flow 1 increases with rate r_1 for $t \geq \overline{t}_1^j - \theta(t)$. The output burst size can be derived as $\overline{b}_i^{j+1} = \alpha_1^j(t+\theta(t)) - r_1 \cdot t = \overline{b}_1^j + r_1 \cdot \theta(t)$ for any $t \geq \overline{t}_1^j - \theta(t)$, so that $\overline{b}_i^{j+1} = \overline{b}_1^j + r_1 \cdot \theta(\overline{t}_1^j)$ holds. □

An Adaptive RIO (A-RIO) Queue Management Algorithm*

Julio Orozco[1,2] and David Ros[1]

[1] GET/ENST Bretagne
Rue de la Châtaigneraie, CS 17607
35576 Cesson Sévigné Cedex, France
Julio.Orozco@irisa.fr
[2] IRISA/INRIA Rennes
Campus Universitaire de Beaulieu
35042 Rennes Cedex, France
David.Ros@enst-bretagne.fr

Abstract. In the context of the DiffServ architecture, active queue management (AQM) algorithms are used for the differentiated forwarding of packets. However, correctly setting the parameters of an AQM algorithm may prove difficult and error-prone. Besides, many studies have shown that the performance of AQM mechanisms is very sensitive to network conditions. In this paper we present an adaptive AQM algorithm, which we call *Adaptive RIO* (A-RIO), addressing both of these problems. Our simulation results show that A-RIO outperforms RIO in terms of stabilizing the queue occupation (and, hence, queuing delay), while maintaining a high throughput and a good protection of high-priority packets; A-RIO could then be used for building controlled-delay, AF-based services. These results also provide some engineering rules that may be applied to improve the behaviour of the classical, non-adaptive RIO.

1 Introduction

Active queue management (AQM) is the name given to a type of router mechanisms used in congestion control. AQM mechanisms manage queue lengths by dropping (or marking) packets when congestion is building up [1], that is, before the queue is full; end-systems can then react to such losses by reducing their packet rate, hence avoiding severe congestion. Random Early Detection (RED) [2] is one of the first AQM mechanisms to have been proposed, and the one that has been most studied. RED intends to avoid congestion by randomly discarding packets based on the comparison of the average queue size with two thresholds.

AQM mechanisms are also relevant in the context of DiffServ. The DiffServ architecture has been defined by the IETF to provide IP networks with scalable QoS processing of traffic aggregates, based on a special field in the IP header.

* This work was partially supported by the Mexican National Council for Science and Technology (CONACyT), as part of the 26th SFERE/CONACyT joint program.

G. Karlsson and M.I. Smirnov (Eds.): QofIS 2003, LNCS 2811, pp. 11–20, 2003.

The value of this field tells a router what particular treatment, the per-hop behaviour (PHB), should be applied to the packet.

One of the standard PHBs is Assured Forwarding (AF) [3]. AF defines the differentiated forwarding of packets classified in up to four classes. Packets from different classes are processed in different physical queues, managed by a scheduling mechanism. Within each class (or queue), there can be up to three drop precedences. Under congestion, packets marked with the highest priority—that is, the lowest drop precedence—should be the last to be discarded, and vice versa. Such differentiated discarding inside a single queue can be achieved by means of special AQM mechanisms, which are usually extensions of RED.

In this paper we describe an adaptive AQM algorithm, which we call *Adaptive RIO* (A-RIO), suitable for building an AF per-hop behaviour. A-RIO is a straightforward combination of the Adaptive RED (A-RED) algorithm described by Floyd et al. [4] and the RIO algorithm of Clark and Fang [5]. The goal of A-RIO is threefold: (1) to simplify the configuration of DiffServ-enabled routers, by alleviating the parameter settings problem most AQM algorithms present; (2) to automatically translate a quality-of-service parameter (that is, delay) into a set of router parameters; (3) to try to stabilize queue occupation around a target value under heavy network load, irrespective of traffic profile.

This paper is organised as follows. In Section 2, we briefly discuss the AQM and DiffServ issues that define the context, motivation and basis of our proposal. In Section 3, the A-RIO algorithm is described in detail. In Section 4, we report on the process and results of a simulation study of A-RIO. Conclusions and perspectives are provided in Section 5.

2 Active Queue Management in DiffServ Networks

The goals of AQM in DiffServ are somewhat different from those in a best-effort context. While the objective of AQM mechanisms in best-effort networks is congestion avoidance, in a DiffServ environment they are used mainly for prioritized discard.

RED with In and Out (RIO) [5] is the basic AQM mechanism suitable for the setup of the AF PHB. RIO is a direct extension of RED that uses two sets of parameters to differentiate the discard of *In* (in-profile) and *Out* (out-of-profile) packets. For deciding whether to discard *Out* packets, RIO uses the average size of the total queue, formed by *In* and *Out* packets. For *In* packets, it uses the average size of a virtual queue formed only by *In* packets. RIO has been extended to handle $n > 2$ precedences following the same principles[1]. The discard probability for packets of precedence $1 \leq j < n$ depends on the average size of a virtual queue containing only the packets with precedences 1 to j. For packets with precedence n (i.e., the lowest priority), the discard probability is a function of the average occupation of the “physical” (total) queue. This original method was eventually called RIO-C (*Coupled*) to differentiate it from methods

[1] In the $n = 3$ case, packet drop precedences are usually identified by colors: *green* for the lowest precedence, *yellow* for the middle one and *red* for the highest one.

proposed later. For example, Weighted RED (WRED) [6] uses the total average queue size for all precedences, whereas RIO-DC (*Decoupled*) [7] calculates the drop probability for packets of precedence j as a function of the average number of packets of the same precedence.

RIO-C discriminates packets of different precedences in three ways. The first one is the use of different thresholds for different precedences, so that discard begins "earlier" for packets of higher precedences. The second way is the use of drop probabilities that increase at different rates for different priorities. The third way lies in the coupled calculation of the discard probability; the fact that the discard probability for precedence j uses the average number of packets of *all* lower precedences yields a significant discrimination. Note that the first two ways of achieving differentiation are based simply on different parameter settings, and that they are not mutually exclusive.

To the best of our knowledge, there are no precise rules for tuning RED's four parameters (two thresholds min_{th} and max_{th}, a maximum drop probability max_p for early discard and an averaging weight w_q); on the contrary, most published results point at the difficulty of finding a robust RED configuration (see for instance [8] and [9]). This problem gets magnified with RIO: for a n-precedence RIO, in principle $3n + 1$ parameters should be set[2]. The problem of parameter setting obviously becomes more complex and, therefore, has also become a subject of research. Studies such as [10] and [11] illustrate the difficulty of tuning RIO in order to achieve a predictable performance. This issue is very relevant, since a key idea behind DiffServ is to allow the provisioning of enhanced services. The operator should know how to set up services with some (loose) guarantees in rate or delay.

3 An Adaptive RIO (A-RIO) Algorithm

We will now describe our proposal for an AQM mechanism called Adaptive RIO (A-RIO). A-RIO is a direct extension of both the A-RED [4] and RIO-C algorithms. A-RIO follows the approach of the former, performing an on-line automatic adaptation of the mechanism to get a more predictable performance, together with a more straighforward parameter tuning. Several approaches have been proposed in the literature for dealing with the tuning of RED; we have chosen A-RED because of its simplicity (both in concept and in implementation).

With this proposal, we look forward to alleviate the problem of parameter tuning in the context of DiffServ AF networks. Like A-RED, A-RIO needs a single input parameter, the *target delay*, which is translated into the required set of router parameters. This feature could be very interesting for providers of differentiated services: configuring routers in terms of delay—a QoS metric directly related to service specifications and customer requirements—should be much easier than in terms of more "abstract" parameters like queue thresholds, discard probabilities or averaging weights.

[2] That is, $2n$ thresholds and n maximum drop probabilities, plus the weight w_q used for averaging queue occupation—assuming the same w_q is used for all virtual queues.

From a performance point of view, A-RIO seeks to achieve a maximum throughput while keeping delay in a bounded, predictable interval when the queue load is high. In the context of DiffServ, we also intend to assure the correct discrimination of packets marked with different precedences.

3.1 Definition

The A-RIO algorithm is based on two main principles. The first one is the use of *a full Adaptive RED instance for each precedence level* in the AF class (physical queue). The second principle is the use of *completely overlapped thresholds for all precedences.* The pseudo-code of A-RIO is shown in Fig. 1 while the concept, for a three-precedence queue, is depicted in Fig. 2. The following salient points of A-RED [4] are kept unchanged in A-RIO:

- The adaptive parameters, $max_p^{(i)}$, are varied between 0.01 and 0.5.
- The lower threshold min_{th} is calculated as a function of the target delay d_t and the link capacity C (in packets/s), with a lower bound of 5 packets. Thus, $min_{th} = \max(5, d_t \cdot C/2)$. The upper threshold max_{th} is fixed at $3 \cdot min_{th}$.
- w_q is also calculated in terms of the link bandwidth as $w_q = 1 - \exp(-1/C)$.
- The *gentle* variant [12] of RED is used throughout. This corresponds, in Fig. 2, to the interval $max_{th} \leq avg^{(i)} \leq 2 \cdot max_{th}$.
- The adaptation function of the $max_p^{(i)}$ uses an AIMD rule (*Additive Increase Multiplicative Decrease*). This rule aims at avoiding brutal changes in $max_p^{(i)}$ that could lead to heavy oscillations of the queue size.
- If the load changes abruptly, the average queue size could find itself out of the target interval. The increase/decrease factors α and β are fixed so that average queue size regains the target interval in no more than 25 seconds.

Design decisions for A-RIO have been made with a guideline in mind: the target delay, which should prevail for every traffic mix *as long as the load is non negligible.* The goal of the algorithm is to keep average queue size in the interval (q_{low}, q_{high}), where $q_{low} = min_{th} + 0.4 \cdot (max_{th} - min_{th})$ and $q_{high} = min_{th} + 0.6 \cdot (max_{th} - min_{th})$.

To better explain this, let us discuss the scenario of a RIO-C queue with non-overlapping thresholds. The precedences are 1 (*In*) and 2 (*Out*). If the total *In* rate is low compared to the link capacity, under a heavy load the *Out* thresholds and discard probability will be active, (hopefully) keeping the average queue size between min_2 and max_2. However, if most of the traffic is *In*, the average queue size will be somewhere between min_1 and max_1. Thus, staggered thresholds may yield different average queue sizes for different traffic mixes. This is a drawback if we want a predictable delay for the queue as a whole and under any congestion scenario. Therefore, we propose the use of the same thresholds for all precedences. This way, the adaptation mechanism of A-RED should pull the average queue size to the same bounded interval, regardless of the traffic mix.

However, the use of overlapped thresholds has implications in differentiation. As stated in Section 2, RIO-C differentiates discard in three basic ways: different

```
for every incoming packet of drop precedence $i$,
    for every drop precedence $j = i, i+1, \ldots, n$
        update $avg^{(j)}$ as: $avg^{(j)} \leftarrow avg^{(j)} \cdot (1 - w_q) + q^{(j)} \cdot w_q$
        every $interval$ time units update $max_p^{(j)}$:
            if $avg^{(j)} > q_{high}$ and $max_p^{(j)} < 0.5$
                compute increase factor: $\alpha \leftarrow \min(0.01, max_p^{(j)}/4)$
                increase $max_p^{(j)}$ as: $max_p^{(j)} \leftarrow max_p^{(j)} + \alpha$
                if $j < n$ then: $max_p^{(j)} \leftarrow \min\left(max_p^{(j)}, max_p^{(j+1)}\right)$
            else if $avg^{(j)} < q_{low}$ and $max_p^{(j)} > 0.01$
                decrease $max_p^{(j)}$ as: $max_p^{(j)} \leftarrow max_p^{(j)} * \beta$
                if $j > 0$ then: $max_p^{(j)} \leftarrow \max\left(max_p^{(j)}, max_p^{(j-1)}\right)$
    if $min_{th} < avg^{(i)} \leq max_{th}$
        calculate $p^{(i)}$ as in A-RED
        discard this packet with probability $p^{(i)}$
    else if $max_{th} < avg^{(i)} \leq 2 * max_{th}$
        calculate $p^{(i)}_{gentle}$ as in A-RED
        discard this packet with probability $p^{(i)}_{gentle}$
    else if $avg^{(i)} > 2 * max_{th}$
        discard this packet
Variables and fixed parameters:
    $avg^{(i)}$:      average queue size for precedence $i$ (coupled:
                counts number of packets with precedence from 1 to $i$)
    $max_p^{(i)}$:    drop probability for precedence $i$ when $avg^{(i)} = max_{th}$
    $p^{(i)}$:        discard probability for precedence $i$
    $p^{(i)}_{gentle}$: discard probability for precedence $i$ in $gentle$ zone
    $interval$:   0.5 s;      $\beta$ (decrease factor): 0.9
```

Fig. 1. A-RIO algorithm.

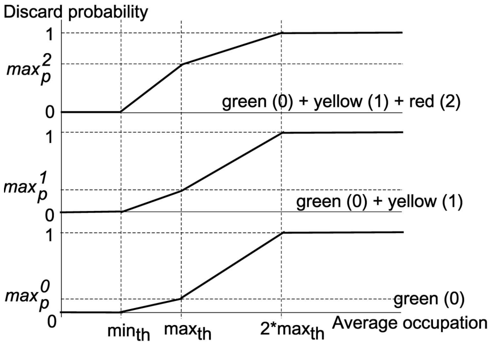


Fig. 2. A-RIO for three precedences.

thresholds, different discard functions and coupled virtual queues. In A-RIO, with the choice of overlapped thresholds, we have excluded the first method. On the other hand, A-RED is based on adapting max_p, and using different $max_p^{(i)}$'s

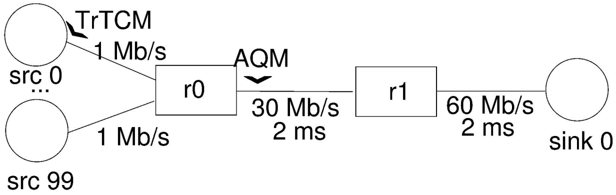


Fig. 3. Dumbbell simulation topology.

is another method of discrimination. Note that the adaptation algorithm is such that: $max_p^{(i)} \leq max_p^{(i+1)}, \forall i \in \{1, \ldots, n-1\}$; together with the coupled virtual queues, this restriction should provide with enough assurance of discrimination.

Concerning implementation issues, note that the complexity of A-RIO is roughly equivalent to that of A-RED and RIO-C combined. Moreover, remark that A-RIO does not store or compute per-flow information, so scalability is (in principle) not an issue.

4 Performance Evaluation of A-RIO

In order to evaluate the performance of the proposed A-RIO algorithm, an extensive series of simulations was carried out using the ns-2 network simulator [13], using diverse network topologies and loads. For space reasons we briefly present a subset of our results; a comprehensive description of simulation scenarios, parameters and results can be found in [14]. The code[3] for the algorithm is based on the DiffServ module included in the 2.1b9a release of ns-2.

The main goal of the evaluation is to verify that A-RIO effectively keeps the average queue size of an AF router at the desired interval under a variety of traffic types and loads. A-RIO should accomplish this goal while providing with protection of high priority packets and fairness of bandwidth allocation.

We check A-RIO's performance by comparing it to that of the original, static RIO Coupled and of a modified version that we call Gentle-RIO or G-RIO. As its name suggests, G-RIO implements the *gentle* mode of RED. Additionally, the values for G-RIO's parameters (thresholds and averaging weights) are set using *the same rules* originally proposed for A-RED and adapted to A-RIO. Hence, G-RIO corresponds to an A-RIO *without* on-line adaptation of the $max_p^{(i)}$.

Figure 3 shows the simplest topology used for the simulations. An AF dumbbell backbone is set up with two routers, r0 and r1. Test traffic is generated by 100 individual TCP sources (src 0 – src 99) connected to ingress router r0 via links of 1 Mb/s each. Each one of the source nodes is also an edge router where traffic is marked by means of a trTCM (Two-Rate Three Color Marker) [15]. All traffic is destined to a single sink node (sink 0) and is marked to be forwarded in a single AF class (physical queue) in the network's core. The backbone link has

[3] Both the code and the simulation scripts can be downloaded from `http://www.rennes.enst-bretagne.fr/~jorozco/aqm.htm`.

Table 1. Simulation cases.

Case	AQM	No. of Sources			RTTs	Assured Rates
		Total	FTP	Pareto On/Off		(% of link capacity)
1	A-RIO, RIO, G-RIO	100	100	0	$=$	25, 50, 75, 100, 125
2	A-RIO, RIO, G-RIO	100	100	0	$\neq$	25, 50, 75, 100, 125
3	A-RIO, RIO, G-RIO	100	0	100	$=$	25, 50, 75, 100, 125
4	A-RIO, RIO, G-RIO	100	0	100	$\neq$	25, 50, 75, 100, 125
5	A-RIO, RIO, G-RIO	100	20	80	$=$	25, 50, 75, 100, 125
6	A-RIO, RIO, G-RIO	100	20	80	$\neq$	25, 50, 75, 100, 125

a rate of 30 Mb/s (i.e., 30% of aggregated access capacity). The AQM algorithm of interest in each scenario is activated in a single physical queue at the output interfaces of backbone routers.

Table 1 summarizes the test scenarios. Simulations were grouped in six cases, depending on the type of input traffic and RTTs. The number of TCP sources remains constant at 100. RTTs of access links are the same (50 ms) in cases 1, 3 and 5 and different (from 40 to 535 ms in steps of 5 ms) in cases 2, 4 and 6. Hence, the complexity of the traffic mix increases from pure long-lived FTP sources and equal RTTs in case 1 to mixed FTP and On/Off with heterogeneous RTTs in case 6. Simulations run for 120 s; sources start at random times between $t = 2$ s and $t = 12$ s.

For each one of the six cases, five different marking profiles were tested. The marking schemes were defined by the total assured rate as a percentage of bottleneck link capacity. This rate is then divided between the 100 individual sources. Given the assured (or committed) rate, the remaining parameters of each trTCM (committed bucket size, excess rate and excess bucket size), were set up following the proportional rules defined in [16].

As seen in Section 3, when configuring an A-RIO queue, only a single parameter is required: the target delay. Values for thresholds and averaging weight w_q are automatically calculated based on that delay. These values are then manually assigned to parameters for the simulations with G-RIO. Table 2 shows the AQM parameter values used for the simulations. Values for A-RIO and G-RIO are calculated for a target delay of 50 ms, a bandwidth of 30 Mb/s and a packet size of 1000 bytes. In the case of A-RIO, the values of $max_p^{(i)}$ are initialisation values, since they change throughout the simulation. The size of the physical queue is 1000 packets. In the case of RIO, a model of partially overlapped thresholds is used. This model is taken from [17] and sets thresholds for three precedences in terms of the queue size; similar settings can be found in e.g. [18].

4.1 Results

The plots in Fig. 4 summarise the results of average queue sizes per AQM for the six scenarios. Averages are calculated after a warm-up time of 20 s, when all sources are active. According to the thresholds shown in Table 2, the target average queue size for A-RIO should be around 187.5 packets. It can be seen

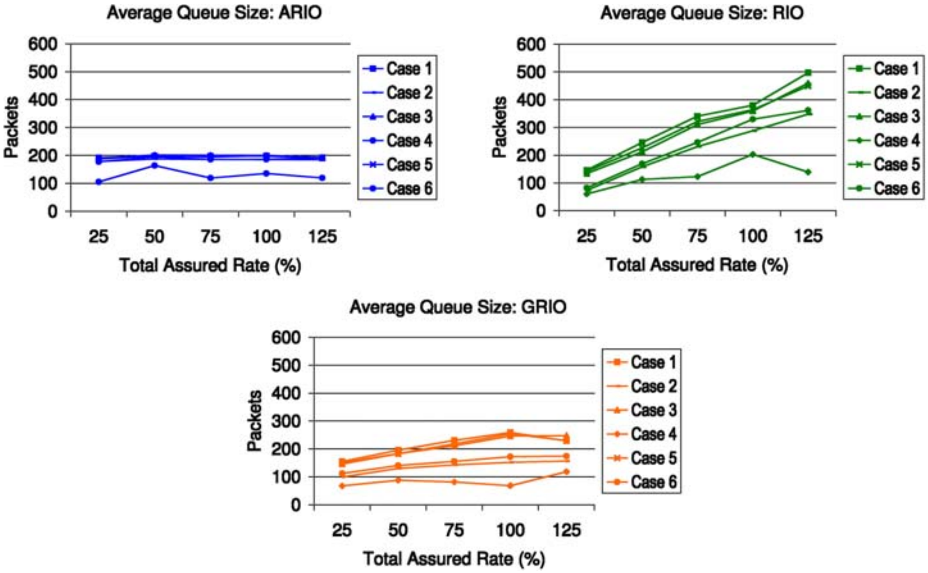


Fig. 4. Average queue sizes per AQM, case and marking profile.

Table 2. AQM parameters.

Color	A-RIO/G-RIO				RIO			
	th_{min}	th_{max}	max_p	w_q	th_{min}	th_{max}	max_p	w_q
Red	94	281	0.2	0.0003	100	250	0.2	0.02
Yellow	94	281	0.1	0.0003	200	400	0.1	0.02
Green	94	281	0.02	0.0003	350	600	0.02	0.02

that A-RIO effectively keeps the queue size close to the target for a wide variety of traffic loads and marking profiles. Only in case number 4 is the average occupation lower for all assured rates. This is due to the fact that the queue remains lightly loaded (because of the traffic characteristics), at a level which is below A-RIO's minimum threshold for adaptation. We could say that, in such a case, A-RIO (like A-RED), lets things follow their natural course. A more detailed discussion can be found in [14].

By contrast, the average queue size for RIO varies in a much wider interval for the different scenarios. This means that delay could vary in a fairly broad interval, making it difficult for a service provider to offer a service with consistent (upper) delay bounds. Finally, note that the results with G-RIO are not as consistent as those with A-RIO, but the average queue size varies in a smaller interval when compared to RIO. Hence, *even without adaptation*, the A-RED rules for setting parameters and the use of overlapped thresholds can be used in static RIO to obtain a more predictable performance in terms of delay.

We mentioned previously that A-RIO should achieve its main goal (keeping average queue close to a target size) while still providing protection for priority

Table 3. Percentages of overall link utilization and green packet discard.

Case	RIO				A-RIO				G-RIO			
	% Overall Utilization		% Discarded Green Packets		% Overall Utilization		% Discarded Green Packets		% Overall Utilization		% Discarded Green Packets	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
1	93.24	0.24	0.26	0.49	93.18	0.30	0.97	1.23	93.00	0.32	0.98	1.07
2	91.26	1.09	0.01	0.03	91.47	0.45	0.19	0.19	90.90	2.04	0.19	0.22
3	91.04	0.29	0.17	0.35	90.10	0.27	0.80	1.07	90.76	0.62	0.73	0.80
4	89.53	1.13	0.00	0.00	89.52	0.30	0.05	0.06	89.10	0.81	0.04	0.09
5	91.39	1.52	0.17	0.32	92.04	0.36	0.79	0.68	91.60	0.68	0.84	0.94
6	90.94	1.21	0.03	0.06	91.65	0.22	0.25	0.29	90.74	1.21	0.26	0.30

packets and fairness of bandwidth allocation. Table 3 shows the summary of results for utilization and protection of priority packets in the bottleneck link. The table includes, for each AQM mechanism, the mean and standard deviation for the overall link utilization (total throughput/capacity) and discarded green packets. Utilization is fairly the same for every case and AQM. Percentages of discarded green packets are slightly higher for A-RIO and G-RIO than for RIO. This was expected, due to the fact that the former two use overlapped thresholds. Nevertheless, these percentages are reasonably low (with means always $< 1\%$).

Comparable results were observed when testing the robustness of A-RIO with respect to the number of TCP connections, as well as its performance in a more complex topology with multiple bottlenecks. A comprehensive description of these scenarios, simulation parameters and results can be found in [14].

5 Conclusions and Future Work

In this paper we have presented an adaptive AQM algorithm, A-RIO, suitable for the AF per-hop behaviour of the DiffServ Architecture. A-RIO draws directly from the original RIO proposal and the Adaptive RED (A-RED) algorithm. A-RIO may be helpful in alleviating the tuning problem most AQM algorithms show, as well as in translating a quality-of-service metric into a set of router parameters. Our simulation results suggest that A-RIO could be used as a building block for controlled-delay, AF-based services. A -RIO stabilises queue occupation, without having an adverse effect on other performance parameters like packet discrimination, throughput and fairness in bandwidth sharing. Our results also suggest that the parameter setting rules introduced in [4], combined with overlapped thresholds and the *gentle* option of RED, may be useful in obtaining a more predictable behaviour of non-adaptive RIO, in terms of queuing delay.

There are a few open issues of A-RIO that are worth exploring, for instance: (1) the impact of using a per-precedence averaging weigth $w_q^{(i)}$ on the performance of A-RIO, with respect to the level of differentiation among virtual queues; (2) what happens to fairness in a mixed assured-rate and heterogeneous RTT scenario; (3) a comparison of A-RIO and G-RIO with WRED [6], as well as with other self-tuning mechanisms proposed in the literature. Since A-RIO might be used to build services with a loose bound on delay, it may be interesting to evaluate its performance when UDP traffic is used, regarding delay *and* jitter, as well as the influence of packet size on these quantities. We would also like to look at implementation issues in a real platform. Finally, we intend to work on the analytical modelling of the algorithm.

References

1. Braden, B., et al.: Recommendations on Queue Management and Congestion Avoidance in the Internet. Internet Standards Track RFC 2309, IETF (1998)
2. Floyd, S., Jacobson, V.: Random Early Detection Gateways for Congestion Avoidance. IEEE/ACM Transactions on Networking **1** (1993) 397–413
3. Heinanen, J., Baker, F., Weiss, W., Wroclawski, J.: Assured Forwarding PHB Group. Internet Standards Track RFC 2597, IETF (1999)
4. Floyd, S., Gummadi, R., Shenker, S.: Adaptive RED: An Algorithm for Increasing the Robustness of RED. Technical Report, to appear (2001)
5. Clark, D., Fang, W.: Explicit Allocation of Best-Effort Packet Delivery Service. IEEE/ACM Transactions on Networking **6** (1998) 362–373
6. Cisco: IOS Release 12.1 Quality of Service Solutions Configuration Guide—Congestion Avoidance Overview. (2002)
7. Seddigh, N., Nandy, B., Pieda, P., Hadi Salim, J., Chapman, A.: An Experimental Study of Assured Services in a DiffServ IP QoS Network. In: Proceedings of SPIE Symposium on Voice, Video and Data Communications. (1998) 217–219
8. Ziegler, T., Fdida, S., Brandauer, C., Hechenleitner, B.: Stability of RED with Two-Way TCP Traffic. In: Proceedings of IEEE ICCCN 2000 (3). (2000)
9. Firoiu, V., Borden, M.: A Study of Active Queue Management for Congestion Control. In: Proceedings of IEEE INFOCOM (3). (2000) 1435–1444
10. Park, W.H., Bahk, S., Kim, H.: A Modified RIO Algorithm that Alleviates the Bandwidth Skew Problem in Internet Differentiated Service. In: Proceedings of IEEE ICC 2000. Volume 1., New Orleans (2000) 1599–1603
11. Malouch, N., Liu, Z.: Performance Analysis of TCP with RIO Routers. Research Report RR-4469, INRIA (2002)
12. Floyd, S.: Recommendations on the use of the gentle variant of RED. `http://www.icir.org/floyd/red/gentle.html`.
13. McCanne, S., Floyd, S.: `ns` Network Simulator. `http://www.isi.edu/nsnam/ns/`.
14. Orozco, J., Ros, D.: An Adaptive RIO (A-RIO) Queue Management Algorithm. Research Report PI-1526, IRISA (2003)
15. Heinanen, J., Guerin, R.: A Two Rate Three Color Marker. Informational RFC 2698, IETF (1999)
16. Medina, O., Orozco, J., Ros, D.: Bandwidth Sharing under the Assured Forwarding Per-Hop Behavior. Research Report PI-1478, IRISA (2002)
17. Medina, O.: Étude des algorithmes d'attribution de priorités dans un Internet à Différentiation de Services. Ph.D. dissertation, Université de Rennes 1 (2001)
18. Hori, Y., Ikenaga, T., Oie, Y.: Queue Management of RIO to Achieve High Throughput and Low Delay. IEICE Transactions on Communications **E85-B** (2002) 63–69

Deterministic End-to-End Delay Guarantees in a Heterogeneous Route Interference Environment*

Florian-Daniel Oţel[1] and Jean-Yves Le Boudec[2]

[1] Dept. of Computer Engineering
Chalmers University of Technology
Gothenburg, Sweden
[2] Laboratory for Computer Communications and Applications (LCA)
École Politechnique Fédérale de Lausanne (EFPL), Switzerland

Abstract. Some of the known results for delivering deterministic bounds on end-to-end queuing delay in networks with constant packet sizes and constant link rates rely on the concept of Route Interference. Namely, it is required to know the number of flows joining on any output link in the whole network. In this paper we extend the existing results for the more generic cases of connection-oriented networks consisting of links with different capacities, carrying different traffic classes and packets of different sizes.

1 Introduction and Related Work

In the quest for delivering deterministic end-to-end delay guarantees in general networks it has been shown [2] that it is feasible to deliver deterministic bounds for queuing delay in networks using FIFO queuing but the bound is dependent on complex network conditions. Specifically, by strictly controlling the number of times flow paths join on output links and by performing ingress traffic shaping in accordance with these metrics, it is possible to compute tight bounds on queuing delay and required buffer capacities. However, the results presented in [2] are limited to very specific network setups i.e. to connection-oriented networks carrying packets of fixed size (ATM networks), with all links having same capacity and where time is considered to be divided in equal slots that are synchronized, network-wide. Moreover, nodes were assumed to be globally FIFO and have zero internal propagation and processing delays.

Later results [3,4] relaxed the requirement for time slot synchronization and improved the bounds on required buffer capacities and end-to-end queuing delay. However, they maintained the limiting requirement of equal packet sizes and equal capacity links.

In this paper we present an extension of the proofs in [1] that relaxes these requirements and generalizes the results to generic connection-oriented networks

* The unabridged version of this paper is available online at `http://www.ce.chalmers.se/staff/otel`

G. Karlsson and M.I. Smirnov (Eds.): QofIS 2003, LNCS 2811, pp. 21–30, 2003.

with links of different speeds and carrying different types of traffic with different packet sizes.

The outline of this paper is as follows: In the next subsection we state the assumed traffic, network, and time models. In Section 2 we introduce the concept of route interference, the source rate condition and we state the main result of this paper – the theorem for bounded buffer and delay. The proof of the theorem is given separately in Section 3 as it involves a detailed analysis of the mechanics of flow aggregation and queue busy periods along a flow path. Section 4 outlines how to compute the required buffer capacity (i.e. maximum amount of queue backlog) and the queuing delay.

1.1 Assumed Network, Traffic and Time Models

Traffic model: For the purpose of this paper we assume that transmission of delay-sensitive data is performed in a connection-oriented manner, with traffic organized in flows whose routes are pre-established before data transmission. Inside each flow data is transmitted in packets having a finite set of possible packet sizes.

Network model: We consider that the network consists of nodes which offer service guarantees in the form of generic rate-latency service curves [1]. However, for the sake of clarity we will consider only the special case of non-preemptive schedulers performing strict priority FIFO queuing. Specifically, we will consider that nodes have a single FIFO queue per traffic class and that delay sensitive traffic has the highest priority in the network. Under these assumptions the node serves the delay-sensitive traffic with a rate-latency service curve $\beta_{r,\tau}$ with rate $r =$ the physical link rate and latency $\tau = \frac{MTU_L}{r}$, where MTU_L is the maximum packet size for lower priority traffic classes. Also, we will denote by MTU_H the maximum packet size for the high priority, delay-sensitive traffic.

Also, we will assume that network links are unidirectional, with variable rates and propagation delays. Without the loss of generality we will consider that the (bounded) internal processing and transmission delays at network nodes are included in the upstream link propagation delays. As such for the rest of this paper we will assume that node internal delays are negligible.

Time model: Time is assumed to be continuous and relevant network events have a time index sequentially numbered starting from time 0, when the network was in an idle state. In other words we assume that all packet receptions and transmissions time ordered, network-wide.

2 The Source Rate Condition and the Theorem for Bounded Buffer and Queuing Delay

In order to present the main result of this paper – the theorem for bounded buffer and queuing delay – we first define the concept of flow joins and introduce the source rate condition as a requirement for ingress traffic shaping.

Definition 1 (Flow joins, Interference Event): Two flows F and G are said to join on link[1] j *if both flows share link* j *but do not share the link upstream from* j *in their respective paths.*

An interference event *is defined as a pair (*j,$\{F,G\}$*) where* j *is a link and F and G are two flows joining at link* j. *As such the number of flows joining F on link* j *is given by the number of interference events that contain* j *and F.*

Definition 2 (Source Rate Condition): We say that a flow F satisfies the Source Rate Condition *if the inter-packet emission time* T_F *satisfies the inequality:*

$$T_F \geq \frac{MTU_H}{r_F^*} \sum_{j=src}^{dst} I_j + MTU_H \sum_{j=src}^{dst} S_j \left(\frac{1}{r_j} - \frac{1}{r_{prev_F(j)}} \right)^+ + \frac{MTU_H}{r_F^*} + \sum_{j=src}^{dst} \frac{MTU_L}{r_j}$$

where:

j *– node along the path of flow F, starting from source node and ending with destination node.*

r_j *– the rate of the link outgoing from node* j *along the flow path. If* j *is the last node then* r_j *is assumed to be infinite.*

$prev_F\,(j)$*– The link previous to* j *along the path of flow F. If node* j *is the first node along the flow path then* $r_{prev_F(j)}$ *is infinite.*

r_F^* *– The minimum capacity link along the path of flow F i.e.* $r_F^* = \min_j\, r_j$.

I_j *– number of flows that join flow F at link* j *i.e. the number of interference events that contain both* j *and F.*

S_j *– number of all flows except F that share both link j and* $prev_F\,(j)$.

In the above formula – as well as for the rest of this paper – the $expression^+$ notation is a shorthand for $\max(expression, 0)$.

Theorem 1 (Theorem for bounded buffer and delay): Provided that the source rate condition holds for all flows, then:

- *The network is stable i.e. the maximum amount of backlog at any queue and the corresponding required buffer capacity are bounded.*
- *The queuing delay at any node is bounded.*

The proof of the theorem – presented in the next section – involves a complex analysis of the queue busy periods and the relations between queue backlogs and interference events. The net result of the theorem are the bounds for the maximum backlog and maximum queuing delay, given both as a description of the algorithmic steps necessary to compute them and as closed-form approximation formulas.

[1] We will alternatively use the expression "node" or "link" as meaning the same thing i.e. the corresponding network event occurring at the named/implied outgoing link of the named/implied node immediately upstream of that link. Also, unless explicitly noted otherwise, when referring to "flows", "traffic", "packets" or "interfering segments" we implicitly refer to the high-priority traffic for which delay guarantees must be delivered.

3 Proof for Bounded Buffer and Delay Theorem

Before delving into the theorem proof proper we introduce some technical definitions that express the concept of chained busy periods i.e. queue busy periods at successive nodes along a flow path.

Definition 3 (Delay operation): For two packets p *and* q *and for some link* j *we say that $p \prec_j q$ if* p *and* q *are in the same busy period of the queue for high-priority traffic at* j *and* p *is transmitted on* j *before* q*. Also by $p \preceq_j q$ we say that* p *leaves on* j *no later than* q *(or, alternatively,* q *leaves on* j *no earlier than* p*).*

It must be noted that, since we refer to packets belonging to highest priority traffic class and since nodes perform priority queuing, the delay relationship between two packets implicitly states that there are no low-priority packets being transmitted between them on the output link.

Definition 4 (Super-Chain, Super-chain path): Consider a sequence of packets $\underline{p} = (p_0......p_i....p_k)$ and a sequence of nodes $\underline{f} = (f_1.......f_k)$. We say that $(\underline{p},\underline{f})$ is a super-chain *if:*

- *$f_1, ..., f_k$ are all on* P *- the path of packet*[2] *p_0, not necessarily consecutive but distinct.*
- *$p_{i-1} \prec_{f_i} p_i$ for $i = 1$ to k.*
- *The path of packet p_i from f_i to f_{i+1} is a sub-path of* P*.*

The path of the super-chain *is defined as the sub-path of p_0 that spans from f_1 to f_k.*

Definition 5 (Relevant network events, arrival and departure time): For the purpose of this paper we define a *relevant network event* as the enqueuing or dequeuing of a packet at a link/node. Also, we will denote the time index of these events as a_j^k for the arrival of packet number k at link j (defined as the time index when the last bit of packet k is received) and, respectively, d_j^k for the corresponding departure time (defined as the time index when the last bit of packet k is transmitted).

It is to be noted that since the set of possible packet sizes was assumed to be finite we cannot have an infinite number of network events occurring in a finite time interval.

Definition 6 (Segment interfering with a super-chain): For a given super-chain we call segment *an ordered pair (*s,P*) where* P *is a sub-path of the path of the super-chain,* s *is a packet whose path has* P *as a sub-path and* P *is maximal (namely we cannot extend* P *to be a common sub-path of both* s *and the super-chain).*

[2] For simplicity we refer to “the path of a packet” as meaning the network route of the flow the said packet belongs to.

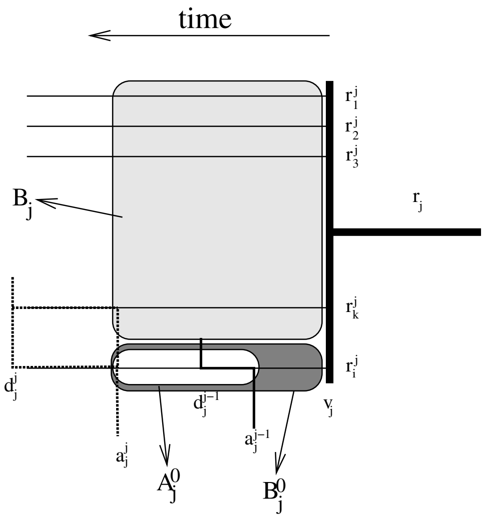


Fig. 1. Node f_j on super-chain path and notation used in section 2.

We say that the segment *(*s*,P) is* interfering with the super-chain *$(\underline{p},\underline{f})$ if there is some node f_i on* P *such that $s \prec_{f_i} p_i$.*

The proof of the theorem is organized as follows: First, based on the number of segments interfering with a super-chain, we will derive an expression for the delay experienced along a super-chain path. Second, we will prove the non intra-flow property i.e. show that, if the source rate condition is imposed for all flows, then there cannot be two packets from the same flow in a super-chain (i.e. there cannot be two packets from the same flow in the same busy period at any queue in the network). Finally, using the formula for the delay along a super-chain path and the non intra-flow interference property, we show that buffer requirements and queuing delay at any node in the network are bounded, thus concluding the proof.

3.1 Delay along a Super-chain Path

Let $(\underline{p},\underline{f})$ be a super-chain and consider the f_j node on the super-chain path (see Fig. 1). Let v_j be the beginning of the busy period (for the queue for high-priority traffic) that a_j^{j-1} is in, i.e. $v_j = a_j^n$ for some packet number n with $n \leq j-1$.

Assume without the loss of generality that packet p_{j-1} arrives on input link i (by necessity belonging to the super-chain path) and packet p_j arrives on input link k, not necessarily distinct from i. Define $\mathcal{B}_j$ as the set interference segments (s,P) such that s is arriving at the node no earlier than time v_j on a link other than input link i (i.e. on a link incident to the super-chain path), $s \preceq_{f_j} p_j$ and P is the maximal common sub-path for s and the path of the super-chain. Define $\mathcal{B}_j^0$ in the same manner but with packet s arriving on the same input link as packet p_{j-1} i.e. on the path of the super-chain. Also define $\mathcal{A}_j^0$ as the subset of

$\mathcal{B}_j^0$ that contains only the packets that depart no earlier than p_{j-1} i.e. $p_{j-1} \preceq_{f_j} s$. Let B_j (resp. B_j^0, A_j^0) be the number of elements in $\mathcal{B}_j$ (resp. $\mathcal{B}_j^0$, $\mathcal{A}_j^0$). Please note that by definition $p_j \notin \mathcal{B}_j \bigcup \mathcal{B}_j^0$ and[3] $p_{j-1} \in \mathcal{A}_j^0$.

With r_i^j and r_j being the rates of input link i and, respectively, link j – the link on the super-chain path outgoing from f_j, we have:

$$a_j^{j-1} - v_j \geq \frac{1}{r_i^j} \sum_{n\in\mathcal{B}_j^0\setminus\mathcal{A}_j^0} l^n$$

$$d_j^j - v_j \leq \frac{1}{r_j} \sum_{n\in\mathcal{B}_j\cup\mathcal{B}_j^0} l^n + \frac{l^j}{r_j} + \frac{MTU_L}{r_j}$$

where l^n is the length of packet n, l^j is the length of packet p_j, l^j/r_j is the transmission time for packet p_j and MTU_L/r_j is the maximum node latency due to non-priority cross-traffic. Subtracting the two we obtain:

$$d_j^j - a_j^{j-1} \leq \frac{1}{r_j} \sum_{n\in\mathcal{B}_j\cup\mathcal{B}_j^0} l^n - \frac{1}{r_i^j} \sum_{n\in\mathcal{B}_j^0\setminus\mathcal{A}_j^0} l^n + \frac{l^j}{r_j} + \frac{MTU_L}{r_j}$$

or, since $\mathcal{B}_j$ and $\mathcal{B}_j^0$ are disjoint and $\mathcal{A}_j^0 \subseteq \mathcal{B}_j^0$:

$$d_j^j - a_j^{j-1} \leq \frac{1}{r_j} \left(\sum_{n\in\mathcal{B}_j} l^n + \sum_{n\in\mathcal{B}_j^0\setminus\mathcal{A}_j^0} l^n + \sum_{n\in\mathcal{A}_j^0} l^n \right) - \frac{1}{r_i^j} \sum_{n\in\mathcal{B}_j^0\setminus\mathcal{A}_j^0} l^n + \frac{l^j}{r_j} + \frac{MTU_L}{r_j}$$

$$d_j^j - a_j^{j-1} \leq \frac{1}{r_j} \left(\sum_{n\in\mathcal{B}_j} l^n + \sum_{n\in\mathcal{A}_j^0} l^n \right) + \sum_{n\in\mathcal{B}_j^0\setminus\mathcal{A}_j^0} l^n \left(\frac{1}{r_j} - \frac{1}{r_i^j} \right) + \frac{1}{r_j} (l^j + MTU_L)$$

Since $d_j^j - a_j^{j-1} \geq \frac{l^j + l^{j-1}}{r_j}$, the right hand side of the inequality must be equal or greater than this quantity for any combination of rates r_j and r_i^j. As $p_{j-1} \in \mathcal{A}_j^0$, the inequality above becomes:

$$d_j^j - a_j^{j-1} \leq \frac{1}{r_j} \left(\sum_{n\in\mathcal{B}_j} l^n + \sum_{n\in\mathcal{A}_j^0} l^n \right) + \sum_{n\in\mathcal{B}_j^0\setminus\mathcal{A}_j^0} l^n \left(\frac{1}{r_j} - \frac{1}{r_i^j} \right)^+ + \frac{1}{r_j} (l^j + MTU_L)$$

As $l^n \leq MTU_H$ for any packet n (including p_j) and $\sum_{n\in\mathcal{B}_j} l^n \leq MTU_H \, B_j$ (with the corresponding inequality also holding for, respectively, $\mathcal{B}_j^0$, $\mathcal{A}_j^0$ and $\mathcal{B}_j^0\setminus\mathcal{A}_j^0$) we obtain:

$$d_j^j - a_j^{j-1} \leq MTU_H \frac{B_j + A_j^0}{r_j} + MTU_H (B_j^0 - A_j^0) \left(\frac{1}{r_j} - \frac{1}{r_i^j} \right)^+ + \frac{1}{r_j} (MTU_H + MTU_L) \qquad (1)$$

By iterative use of relation (1) along the super-chain path (i.e. along the subscripts) and packet numbers (superscripts) we obtain:

$$d_k^k - a_1^0 \leq MTU_H \sum_{j=f_1}^{f_k} \frac{B_j + A_j^0}{r_j} + MTU_H \sum_{j=f_1}^{f_k} (B_j^0 - A_j^0) \left(\frac{1}{r_j} - \frac{1}{r_i^j} \right)^+ +$$

$$+ (MTU_H + MTU_L) \sum_{j=f_1}^{f_k} \frac{1}{r_j} + \tau_{1,k}$$

[3] By an abuse of notation we will write packet $p_j \notin \mathcal{B}_j$ as meaning segment $(p_j, P) \notin \mathcal{B}_j$ for any path P and, respectively, packet $p_j \in \mathcal{B}_j$ as meaning segment $(p_j, P) \in \mathcal{B}_j$ for some path P.

$$d_k^k - a_1^0 \leq \frac{MTU_H}{r_f^*} \sum_{j=f_1}^{f_k} (B_j + A_j^0) + MTU_H \sum_{j=f_1}^{f_k} (B_j^0 - A_j^0) \left(\frac{1}{r_j} - \frac{1}{r_i^j} \right)^+ +$$
$$+ (MTU_H + MTU_L) \sum_{j=f_1}^{f_k} \frac{1}{r_j} + \tau_{1,k} \qquad (2)$$

where $r_f^* = \min_j \, r_j$ corresponds to the smallest capacity link along the super-chain path, $\tau_{1,k}$ is the propagation time along the links in the super-chain path and the penultimate term denotes the transmission times and node latencies, cumulated along the path.

Since the sets in the collection $\{\mathcal{B}_j \bigcup \mathcal{A}_j^0\}_{j=1\,to\,k}$ are two-by-two disjoint (see [1], lemma 6.4.2) and since by definition every element in $\{\mathcal{B}_j \bigcup \mathcal{A}_j^0\}$ is an interfering segment, we have that $\sum_{j=f_1}^{f_k} (B_j + A_j^0) \leq I_{1,k}$ where $I_{1,k}$ is the number of interfering segments along the super-chain path. Thus relation (2) becomes:

$$d_k^k - a_1^0 \leq \frac{MTU_H}{r_f^*} \sum_{j=f_1}^{f_k} I_{1,k} + MTU_H \sum_{j=f_1}^{f_k} (B_j^0 - A_j^0) \left(\frac{1}{r_j} - \frac{1}{r_i^j} \right)^+ +$$
$$+ (MTU_H + MTU_L) \sum_{j=f_1}^{f_k} \frac{1}{r_j} + \tau_{1,k} \qquad (3)$$

3.2 The Non-intra-flow Interference Property

Assume that the source rate condition holds. Let $(\underline{p},\underline{f})$ be a super-chain.

1. For every interference event of packet p_0 there is at most one segment interfering with the super-chain.
2. B_j^0 is upper bounded by the number of flows that share the same input link as packet p_{j-1} and same output link as packet p_j.
3. p_k does not belong to the same flow as packet p_0.

Proof: Define the *time of the super-chain* as the time index for the exit of packet p_k from the last node f_k.We use a recursion[4] on time t.

At time index $t = 1$ the proposition is true because any flow has transmitted at most one packet. Assume now that the proposition holds for *any* super-chain with time index $\leq t - 1$ and consider a super-chain with time index t.

The proof for item 1 is identical to the proof of item 1 of proposition 6.4.2 in [1].

For item 2, consider an interfering packet $s \in \mathcal{B}_j^0$. Assume there exists another interfering packet $s' \in \mathcal{B}_j^0$, with s' belonging to the same flow as s. Consider without the loss of generality that s was emitted before s'. Since by definition $s \prec_{f_j} p_j$ and $s' \prec_{f_j} p_j$, then we must have that $s \prec_{f_j} s'$ and $((s, s'), (f_j))$ is a super-chain with exit time $\leq t - 1$, which contradicts item 3. As such we cannot have two packets in the same flow in $\mathcal{B}_j^0$, which proves item 2.

For item 3, let us compute a bound on maximum queuing delay for packet p_0. Consider u_0 its emission time, P_0 the sub-path of p_0 from its source up to, but excluding, node f_1, and T the total propagation *and transmission* time for

[4] Please note that this is permissible in this case since we have no accumulation point along time indexes.

p_0 along P_0 and the super-chain path. Consider that the component of T along the super-chain path is $T_{1,k}$. Applying relation (3) along P_0 and separating the summation terms for packet transmission times from node latencies, we have:

$$a_1^0 \le d_1^0 \le u_0 + (T - T_{1,k}) + \frac{MTU_H}{r_F^*} \sum\nolimits_{j=src}^{prev_F\,(f_1)} I_{0,1} + MTU_H \sum\nolimits_{j=src}^{prev_F\,(f_1)} (B_j^0 - A_j^0) \left(\frac{1}{r_j} - \frac{1}{r_i^j} \right)^+ +$$
$$+ \sum\nolimits_{j=src}^{prev_F\,(f_1)} \frac{MTU_L}{r_j}$$

where F is the flow the packet p_0 belongs to, r_i^j is infinite for j=source node and $I_{0,1}$ is the number of interference events for F along P_0. From item 2 above $B_j^0 \le$ number of flows sharing both link j and $prev_F\,(j)$. Since $p_{j-1} \in \mathcal{A}_j^0$ we have that $A_j^0 \ge 1$ and thus $B_j^0 - A_j^0 \le S_j$, where S_j is the number of flows sharing link j, minus 1 (i.e. the number of flows sharing links $prev_F(j)$ and j, except the flow p_{j-1}belongs to). As such we can re-write the above expression as:

$$a_1^0 \le u_0 + (T - T_{1,k}) + \frac{MTU_H}{r_F^*} \sum\nolimits_{j=src}^{prev_F\,(f_1)} I_{0,1} + MTU_H \sum\nolimits_{j=src}^{prev_F\,(f_1)} S_j \left(\frac{1}{r_j} - \frac{1}{r_{prev_F\,(j)}} \right)^+ +$$
$$+ \sum\nolimits_{j=src}^{prev_F\,(f_1)} \frac{MTU_L}{r_j}$$

Using the same reasoning, along the super-chain path we have that:

$$d_k^k \le a_1^0 + T_{1,k} + \frac{MTU_H}{r_F^*} \sum\nolimits_{j=f_1}^{f_k} I_{1,k} + MTU_H \sum\nolimits_{j=f_1}^{f_k} S_j \left(\frac{1}{r_j} - \frac{1}{r_{prev_F\,(j)}} \right)^+ +$$
$$+ \sum\nolimits_{j=f_1}^{f_k} \frac{MTU_L}{r_j}$$

Combining the last two inequalities we obtain:

$$d_k^k \le u_0 + T + \frac{MTU_H}{r_F^*} \sum\nolimits_{j=src}^{f_k} I_j + MTU_H \sum\nolimits_{j=src}^{f_k} S_j \left(\frac{1}{r_j} - \frac{1}{r_{prev_F\,(j)}} \right)^+ + \sum\nolimits_{j=src}^{f_k} \frac{MTU_L}{r_j}$$

Assuming that p_k and p_0 belong to the same flow and u_k is the emission time of packet p_k , since the source rate condition holds (including for the sub-path of F from the source node to node f_k), by applying the Source Rate Condition we have that (with $k > 0$):

$$u_k \ge u_0 + \frac{MTU_H}{r_F^*} \sum\nolimits_{j=src}^{f_k} I_j + MTU_H \sum\nolimits_{j=src}^{f_k} S_j \left(\frac{1}{r_j} - \frac{1}{r_{prev_F\,(j)}} \right)^+ + k \left(\frac{MTU_H}{r_F^*} + \sum\nolimits_{j=src}^{f_k} \frac{MTU_L}{r_j} \right)$$

which – by adding on both sides T, the transmission and propagation times for packet p_k from its source to node f_k – translates at node f_k into (since $d_k^k \ge u_k + T$):

$$d_k^k \ge u_0 + T + \frac{MTU_H}{r_F^*} \sum\nolimits_{j=src}^{f_k} I_j + MTU_H \sum\nolimits_{j=src}^{f_k} S_j \left(\frac{1}{r_j} - \frac{1}{r_{prev_F\,(j)}} \right)^+ + \sum\nolimits_{j=src}^{f_k} \frac{MTU_L}{r_j} +$$
$$+ k \left(\frac{MTU_H}{r_F^*} + \sum\nolimits_{j=src}^{f_k} \frac{MTU_L}{r_j} \right)$$

which contradicts the relation above. As such p_k and p_0 cannot belong to the same flow, which proves item 3 of the non intra-flow interference property.

3.3 Bounded Buffer Requirements and Queuing Delay

Given the non intra-flow interference property it follows immediately that, if the all flows rates satisfy the source rate condition, at any output queue for delay-sensitive traffic there can be at most one packet from each flow during any busy period. As such the total amount of traffic that transiently shares the queue during any period is upper bounded by the number of flows sharing that link multiplied the maximum packet size. Consequently, at any node the amount of queue backlog at any instant is bounded and the network is stable. Since the nodes perform FIFO queuing and offer service guarantees in the form of rate-latency curves to the delay-sensitive traffic, the amount of queuing delay at any node is bounded. This completes the proof for the theorem of bounded buffer and delay.

4 Buffer Requirements and Queuing Delay Computation

Without the loss of generality, consider a single-node with I fan-in links to the same output link i.e. for a given output link consider the input links carrying flows which join on that output link. Let r_i be the rate of fan-in link i and N_i be the number of flows that share both input link i and the output link. Let $\theta_i = \frac{N_i - 1}{r_i} MTU_H \geq 0$ (since $N_i \geq 1$) and assume, without the loss of generality, that input links are numbered in the increasing order of θ_i, from 1 to I.

Since during any busy period there can be at most one packet from each flow in the queue, due to packetization effects [1] the envelope for the input link i is $\alpha_i(t) = \min(N_i MTU_H,\, r_i t + MTU_H)$. As a result the envelope for the traffic aggregate at the output link queue is:

$$\begin{aligned}\alpha(t) &= \sum\nolimits_{i=1}^{I} \alpha_i(t) {=} \sum\nolimits_{i=1}^{I} \min\nolimits_t(N_i MTU_H, r_i t{+}MTU_H)\\ &= \sum\nolimits_{i=1}^{I} (MTU_H {+} \min\nolimits_t((N_i{-}1)MTU_H, r_i t))\\ \alpha(t) &= I\, MTU_H {+} \sum\nolimits_{i=1}^{I} r_i \min\nolimits_t(\theta_i, t) {=} I\, MTU_H {+} \alpha'(t)\end{aligned}$$

where $\alpha'(t) = \sum_{i=1}^{I} r_i \min(\theta_i, t)$. The $\alpha'(t)$ function – illustrated in Fig. 2 – is piece-wise linear, with the linear segment with $t \in [\theta_{k-1}, \theta_k]$ having a slope $R_k = r_k + r_{k+1} + ... + r_I$, for $k = 1$ to I, with $\theta_0 = 0$, $R_I = r_I$ and $R_{I+1} = 0$. The ordinates at discontinuity points θ_k are $MTU_H \sum_{i=1}^{k}(N_i - 1) + R_{k+1}\theta_k$, with a maximum value of $MTU_H \sum_{i=1}^{I}(N_i - 1)$ at θ_I.

Since the node offers service guarantees in the shape of a rate-latency curve $\beta_{r\,,\tau}(t)$ (with r being the rate of the output link and $\tau = \frac{MTU_L}{r}$) the maximum backlog and maximum jitter are given by maximum vertical (resp. horizontal) distance between envelope $\alpha(t)$ and the service curve $\beta_{r\,,\tau}(t)$ (see [1]).

Instead of numerically computing these quantities we can use a closed form approximation formula by assuming that the node has as strict service curve the rate-latency curve $\beta_{r^*\,,\tau}$, with $r^* = \min_i(r, r_i)$. In this case the maximum backlog occurs at θ_I and has a value of $MTU_H\, N{-}r\,(\theta_I - \tau)^+$ and the maximum jitter is upper bounded by $MTU_H \left(\frac{N}{r} - \frac{N_I - 1}{r_I}\right) + \frac{MTU_L}{r}$.

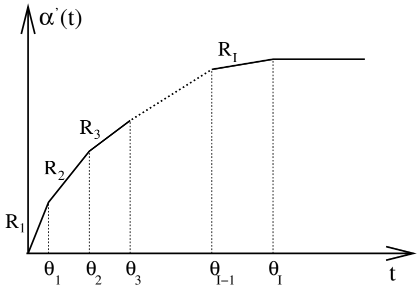


Fig. 2. The $\alpha'(t)$ function.

Along a flow path an upper bound for the end-to-end queuing delay is the sum of the per-node queuing delay bounds along the path. For example in the case of the closed form approximation the end-to-end queuing delay is upper bounded by $MTU_H \sum_{j=src}^{dst} \left(\frac{N^j}{r_j^*} - \frac{N_i^j - 1}{r_i^j} \right) + \sum_{j=src}^{dst} \frac{MTU_L}{r_j}$ where N^j is the number of flows sharing link j, N_i^j is the number of flows sharing both output link j and fan-in link i, and $r_j^* = \min_i \left(r_j, r_i^j \right)$ is the smallest capacity among output link j and all its fan-in links.

References

1. Jean-Yves LeBoudec, Patrick Thiran, "Network calculus: A theory of deterministic queuing systems for the Internet". Online version, July 2002.
2. I. Chlamtac, A. Farago, H. Zhang, A. Fumagalli, "A deterministic approach to the end-to-end analysis of packet flows in Connection-oriented networks" in *IEEE/ACM Transactions on Networking, Vol. 6, No. 4, Aug 1998.*
3. J.Y. LeBoudec, G. Hebutrene, "Comments on A Deterministic Approach to the End-to-End Analysis of Packet Flows in Connection Oriented Networks" in *IEEE/ACM Transactions on Networking, Vol. 8, No. 1, Feb 2000.*
4. H. Zhang, "A note on deterministic end-to-end delay analysis in connection oriented networks" in *Proc. of IEEE ICC'99, Vancouver, p. 1223-1227, 1999.*

Delay Bounds for FIFO Aggregates: A Case Study

Luciano Lenzini, Enzo Mingozzi, and Giovanni Stea

Dipartimento di Ingegneria dell'Informazione
University of Pisa
Via Diotisalvi 2, I-56122 Pisa, Italy
{l.lenzini,e.mingozzi,g.stea}@iet.unipi.it

Abstract. In a Diffserv architecture, packets with the same marking are treated as an aggregate at core routers, independently of the flow they belong to. Nevertheless, for the purpose of QoS provisioning, derivation of upper bounds on the delay of individual flows is of great importance. In this paper, we consider a case study network, composed by a tandem of rate-latency servers that is traversed by a tagged flow. At each different node, the tagged flow is multiplexed into a FIFO buffer with a different interfering flow. For the case study network, we derive an end-to-end delay bound for tagged flow traffic that, to the best of our knowledge, is better than any other applicable result available from the literature.

1 Introduction

Aggregate scheduling has been proposed as a solution for scaling complexity when providing Quality of Service (QoS) in the Internet. A notable example is provided by the Differentiated Services (DS) architecture proposed within the IETF [10]. According to DS, packets are marked at the DS domain ingress as belonging to a small number of different QoS classes, each one receiving a differentiated service within the network. Packets are then treated at core routers according to a specified per-hop behavior (PHB), independently of the flow they belong to. Currently, the Expedited Forwarding (EF) PHB is specified [12,13] for providing a guaranteed delay service. Practical implementations of the EF PHB assume that all EF traffic is shaped and policed at the DS domain ingress, and then shares a single FIFO guaranteed rate queue at each core router. For scalability issues, capacity is statically reserved to the aggregate traffic at core routers, whereas appropriate admission control is performed at the DS domain edges, to provide specific QoS guarantees to the aggregate. Within this context, analytical derivation of delay bounds for individual flows is of great importance, since they can be used as the base for call admission control. In [6], such delay bounds were derived for a generic network as a function of the utilization factor, the maximum hop count, and the parameters of the ingress shapers, without any assumption on the topology. However, in a network domain employing centralized resource management, it is reasonable to assume that the management entity is aware of all the requests at the domain edge, as well as of the domain topology: this is the case, for example, of the Bandwidth Broker (BB) service proposed in [11]. However, how the knowledge of the network topology and traffic load can be exploited in order to derive delay bounds useful for performing careful admission control is an open issue.

G. Karlsson and M.I. Smirnov (Eds.): QofIS 2003, LNCS 2811, pp. 31–40, 2003.

In this paper, we consider a simple case study network, composed by a tandem of rate-latency servers that is traversed by a tagged flow that is leaky-bucket constrained. The tagged flow is multiplexed into a FIFO buffer with another leaky-bucket constrained flow at each node. Based on well-known results on FIFO multiplexing, we derive an end-to-end delay bound for the traffic of the tagged flow. Although limited to the scope of the case study, the interpretation of the obtained result is, in our opinion, of particular importance for at least two reasons. First, it can be shown that the so-called "pay bursts only once" property no longer holds for the tagged flow when aggregate scheduling is in place, reflecting the fact that sometimes paying the burst more than once, but at a higher rate, is better than paying it only once at a lower rate. Second, to the best of our knowledge, the derived delay bound is better than any other applicable result from the literature, thus proving that a tight delay bound for a general feed-forward FIFO aggregate scheduling network is still lacking.

2 Network Calculus Fundamentals

Network calculus is a theory for deterministic network analysis [2,3,5]. The concept of service curve is introduced in network calculus as a general means to model a network element in terms of input and output flow relationships, i.e., how the element transforms an arriving stream of packets into a departing stream. To this aim, data flows are described by means of the cumulative function $R(t)$, defined as the number of bits seen on the flow in time interval $[0,t]$. Function $R(t)$ is wide-sense increasing, that is $R(s) \leq R(t)$ if and only if $s \leq t$. Specifically, let $A(t)$ and $D(t)$ be the cumulative functions characterizing the same data flow before entering a network element, and after having departed, respectively. Then, the network element is modeled by the service curve $\beta(t)$ if

$$D(t) \geq \inf_{0 \leq s \leq t} \{A(t-s) + \beta(s)\} \tag{1}$$

for any $t \geq 0$. The flow is said to be guaranteed the minimum service curve β. The infimum on the right side of (1), as a function of t, is called the min-plus convolution of A and β, and is denoted by $(A \otimes \beta)(t)$. It has been shown that many schedulers proposed for ATM or the Internet integrated services can be modeled by a family of simple service curves called the rate-latency service curves, defined as

$$\beta_{\rho,\theta}(t) = \rho[t-\theta]^{+} \tag{2}$$

for some rate $\rho \geq 0$ and latency $\theta \geq 0$. Notation $[x]^{+}$ denotes $\max\{0,x\}$.

Differentiated and integrated services assume that traffic flows are constrained. In network calculus this feature is modeled by introducing the concept of arrival curve. A wide-sense increasing function α is said to be an arrival curve (or, equivalently, an envelope) for a flow characterized by a cumulative function R if it is $R(t) - R(\tau) \leq \alpha(t-\tau)$, for $\tau \leq t$. As an example, a flow regulated by a *leaky-bucket* shaper, with rate ρ and burst size σ, is constrained by the arrival curve

$$\gamma_{\rho,\sigma}(t) = (\sigma + \rho t)1_{\{t>0\}}. \tag{3}$$

The indicator function $1_{\{expr\}}$ is equal to 1 if *expr* is true, and 0 otherwise.

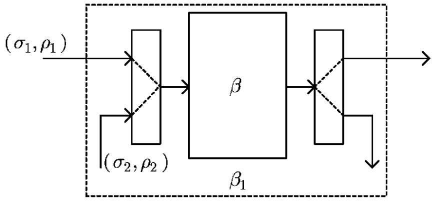


Fig. 1. Two flows multiplexed into the same node.

By combining together arrival and service curve characterizations of data traffic and network elements, respectively, it is possible to derive relevant performance bounds. Specifically, end-to-end delay bounds can be derived. In fact, assume that an element (or network of elements) is characterized by a service curve β and that a flow traversing the node is constrained by the arrival curve α. Then, if the node serves the bits of this flow in FIFO order, the delay is bounded by the horizontal deviation

$$h(\alpha,\beta) \triangleq \sup_{t\geq 0}[\inf\{d \geq 0 : \alpha(t-d) \leq \beta(t)\}] \tag{4}$$

Intuitively, h is the amount of time the curve α must be shifted forward in time so that it lies below β. A very well-known result related to a tandem of rate-latency nodes β_{ρ^i,θ^i} traversed by a $\gamma_{\rho,\sigma}$ constrained flow follows from (4), i.e., the end-to-end delay bound is given by

$$d = \sum\nolimits_i \theta^i + \frac{\sigma}{\bigwedge_i \{\rho^i\}} \tag{5}$$

provided that $\rho \leq \rho^i$ for any i. Notation $\wedge$ denotes the minimum operation.

3 Motivation

In [2], a fundamental result for FIFO multiplexing is given. Assume that two flows are FIFO multiplexed into the same network element, characterized by a service curve β. Figure 1 represents the model under consideration. Assume that α_2 is an arrival curve for flow 2. Then, the service received by flow 1 can be determined by characterizing the element in terms of an *equivalent* service curve $\beta_1(t)$, as follows.

Theorem 1 (FIFO Minimum Service Curves [2]). *Let us define the family of functions*

$$\beta_1(t,\tau) = [\beta(t) - \alpha_2(t-\tau)]^+ 1_{\{t>\tau\}} \tag{6}$$

For any $\tau \geq 0$ such that $\beta_1(t,\tau)$ is wide-sense increasing, then flow 1 is guaranteed the service curve $\beta_1(t,\tau)$.

Assuming that $\alpha_2(t) = \gamma_{\rho_2,\sigma_2}(t)$ and $\beta(t) = \beta_{\rho,\theta}(t)$, it follows from Theorem 1 that an equivalent service curve for flow 1 is

$$\beta_1(t) = \beta_{\rho-\rho_2,\theta+\frac{\sigma_2}{\rho}}(t), \tag{7}$$

i.e., flow 1 is guaranteed a rate-latency service curve, with rate $\rho - \rho_2$ and latency

$\theta + \sigma_2 / \rho$. Assuming further that flow 1 is $\alpha_1(t) = \gamma_{\rho_1,\sigma_1}(t)$ constrained, a delay bound for flow 1 data packets only can be derived as a special case of (5) as follows

$$d_1 = \theta + \frac{\sigma_2}{\rho} + \frac{\sigma_1}{\rho - \rho_2}. \tag{8}$$

On the other hand, this bound is not tight. In fact, the aggregate flow, resulting from multiplexing flows 1 and 2 before service, is also $\alpha = \alpha_1 + \alpha_2$ constrained. Hence, again from (5), a delay bound for the aggregate traffic is

$$d = \theta + \frac{\sigma_1 + \sigma_2}{\rho} \tag{9}$$

Now, (9) holds for any packet in the aggregate flow, hence it holds for any packet belonging to flow 1. Delay d from (9) is then an upper bound also for flow 1 packets only, and d is always lower than d_1 in (8), except for the degenerate case when there is no traffic from flow 2. We can conclude that making use of an equivalent *rate-latency* service curve as obtained from Theorem 1 can lead to overestimate delay bounds for single flows in a scheduled aggregate. It is worth noting that when rate ρ_2 gets closer to the overall available rate ρ, the bound in (8) approaches infinity and hence infinitely diverges from that given by (9), which stays finite.

On the other hand, suppose that for each service curve given by (6), characterized by a value of the parameter τ, we determine a delay bound by calculating the corresponding horizontal deviation according to (4). Then, we take the minimum amongst the set of delay bounds that we have derived as a function of τ. We illustrate this way of proceeding for the example of Figure 1. To this aim, we first apply Theorem 1 to obtain the following result.

Corollary 1. *Consider a node serving two flows, 1 and 2, in FIFO order. Assume that flow 2 is* $\gamma_{\rho_2,\sigma_2}(t)$ *constrained, and the node guarantees a service curve* $\beta_{\rho,\theta}(t)$ *to the aggregate of the two flows. Then, flow 1 is guaranteed the service curve*

$$\beta_1(t,\tau) = (\rho - \rho_2)\left[t - \left(\theta + \frac{\sigma_2 + \rho_2(\theta - \tau)}{\rho - \rho_2}\right)\right]^+ 1_{\{t>\tau\}}, \ \tau \geq 0 \tag{10}$$

The family of service curves given by (10) is represented in Figure 2 for different values of the parameter τ. Assuming now that flow 1 is $\alpha_1(t) = \gamma_{\rho_1,\sigma_1}(t)$ constrained, we can derive the horizontal deviation $h(\alpha_1, \beta_1, \tau)$ as a function of τ, and then find the value of τ for which the lowest delay bound is obtained. Figure 2 can give an intuition of the result: rate-latency curves of the family are excluded, and the best choice among the remaining ones is the leaky-bucket shaper whose burst size exactly matches the burst size σ_1 of flow 1, i.e., that one corresponding to $\tau = \theta + (\sigma_2 + \sigma_1)/\rho$. This result can also be proved formally, and the corresponding lowest delay bound equals the bound derived above and given by (9).

4 The Case Study

Let us consider the case study represented in Figure 3, in which a tandem of rate-latency servers, with latency θ^i and service rate ρ^i, respectively, is traversed by a tagged flow that is (σ, ρ)-constrained. At generic node i, the tagged flow shares a

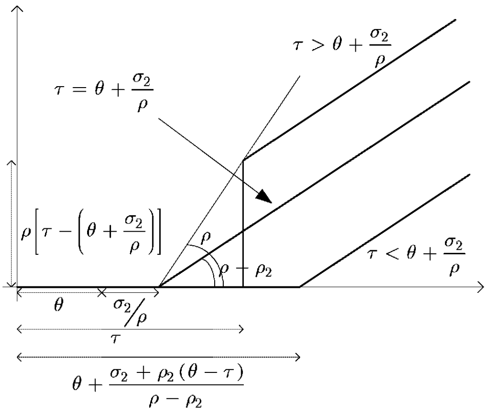

$\tau > \theta + \frac{\sigma_2}{\rho}$
$\tau = \theta + \frac{\sigma_2}{\rho}$
$\rho\left[\tau - \left(\theta + \frac{\sigma_2}{\rho}\right)\right]$
ρ
$\rho - \rho_2$
$\tau < \theta + \frac{\sigma_2}{\rho}$
θ
σ_2/ρ
τ
$\theta + \frac{\sigma_2 + \rho_2(\theta - \tau)}{\rho - \rho_2}$

Fig. 2. The family of equivalent service curves for the tagged flow.

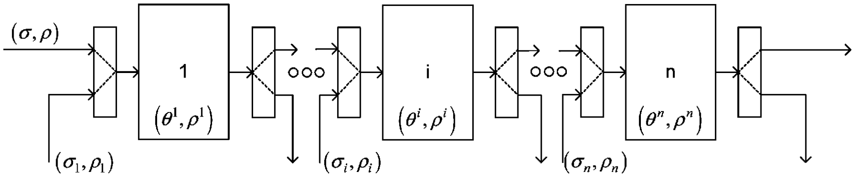


Fig. 3. The case study network.

FIFO buffer with flow i, that is (σ_i, ρ_i)-constrained[1]. We further assume that $\rho^i \geq \rho + \rho_i$ for any i.

Let $\beta_i(t, \tau_i)$ represent a family of service curves for the tagged flow at node i, parameterized by $\tau_i \geq 0$. Furthermore, let β be the end-to-end service curve for the tagged flow. The end-to-end service curve can be computed as the min-plus convolution of the equivalent service curves for each node. Indeed, it is possible to derive a family of end-to-end service curves $\beta(t, \tau_1, \ldots, \tau_n)$, parameterized by $\tau_i \geq 0$, $1 \leq i \leq n$, as follows

$$\beta(t, \tau_1, \ldots, \tau_n) = \bigotimes_i \beta_i(t, \tau_i) \tag{11}$$

We are interested in finding a bound for the end-to-end latency experienced by tagged flow traffic in the case study. According to the procedure described in Section 3, the lowest delay bound is derived by first deriving a different delay bound for any service curve out of the n-dimension infinity set given by (11). By applying (4), the latency bound for tagged flow traffic is given by

$$h = \sup_{t \geq 0} \left\{ \inf \left\{ d \geq 0 : \gamma_{\rho,\sigma}(t - d) \leq \beta(t, \tau_1, \ldots, \tau_n) \right\} \right\} \tag{12}$$

It is then $h = h(\tau_1, \ldots, \tau_n)$, and the lowest upper bound can be derived by minimizing h as a function of parameters τ_i, i.e.

[1] Flow *i* could be also representing an aggregate of background traffic with an effective arrival curve.

$$h_{\text{lub}} = \inf_{\tau_i \geq 0} \{ h(\tau_1,\ldots,\tau_n) \} \tag{13}$$

Let $I(l)$ denote the set of nodes $I(l) \triangleq \{ i : (\rho^i - \rho_i) \leq (\rho^l - \rho_l) \}$. $I(l)$ includes all nodes i whose residual rate, i.e. the difference between the rate guaranteed to the aggregate of flows and the arrival rate of the interfering flow, is not greater than the residual rate of node l. It is obviously $l \in I(l)$. The lowest upper bound h_{lub} defined by (13) is given for the case study by the following Theorem.

Theorem 2. *If* $\sum_i (1 - \rho_i / \rho^i) \leq 1$, *then*

$$h_{\text{lub}} = \sum_i \theta^i + \sum_i \frac{\sigma_i}{\rho^i} + \sum_i \frac{\sigma}{\rho^i} \tag{14}$$

Otherwise, if $\sum_i (1 - \rho_i / \rho^i) > 1$, *then there exists a node* k *such that* $\sum_{i \in I(k)} (1 - \rho_i / \rho^i) > 1$, *and* $\sum_{i \in I(l)} (1 - \rho_i / \rho^i) \leq 1$ *for any other set* $I(l)$, *possibly empty, strictly included in* $I(k)$. *Then*

$$h_{\text{lub}} = \sum_i \theta^i + \sum_i \frac{\sigma_i}{\rho^i} + \left[\sum_{i \in I(k)} \frac{\sigma}{\rho^i} - \frac{\sigma}{\rho^k - \rho_k} \left(\sum_{i \in I(k)} \frac{\rho^i - \rho_i}{\rho^i} - 1 \right) \right] \tag{15}$$

Proof. For space reasons, the proof is omitted. A detailed proof can be found in [14].

We note that, if there is only one node, then necessarily (14) applies, and we get the expected result as given by (9) in Section 3. On the other hand, in the general case when many nodes are considered, the lowest upper delay bound is given by the couple of equations (14) and (15). The overall delay bound given by (14) and (15) can be interpreted as the sum of three different contributions. The first contribution is due to node latencies, which must be all summed up in the worst-case. The second contribution is the one taking into consideration all of the interfering flows: each interfering flow contributes with an additional latency that equals the amount of time needed at each node to serve a maximum burst from that flow. Finally, the last contribution takes into account the additional latency that the tagged flow traffic may experience due to its own maximum burst. This last contribution is given as the amount of time needed to serve a maximum burst from the tagged flow computed independently either on each node of the tandem (equation (14)), or on a subset of nodes (equation (15)), depending on the value of the sum of the residual rates available to the tagged flow, weighted by their respective rate guaranteed to the aggregate. This is a quite surprising result, since it implies that for the case study the "pay bursts only once" principle [4] does not hold when aggregate scheduling is considered, at least for the case under consideration, except for when all nodes belonging to $I(k)$ share the same residual rate which equals the minimum residual rate. Otherwise, the lowest delay bound is obtained instead by "paying" the tagged flow burst on a subset of nodes. An immediate confirmation of this observation is obtained if we consider the special case when the scheduler at the output link of each node is configured so that each aggregate is guaranteed exactly the same rate it is constrained to at network ingress, i.e., $\rho^i = \rho + \rho_i$ for any i. The following result derives from direct application of Theorem 2.

Corollary 2. *For the case study in Figure 3, assume that* $\rho^i = \rho + \rho_i$ *for any* i.

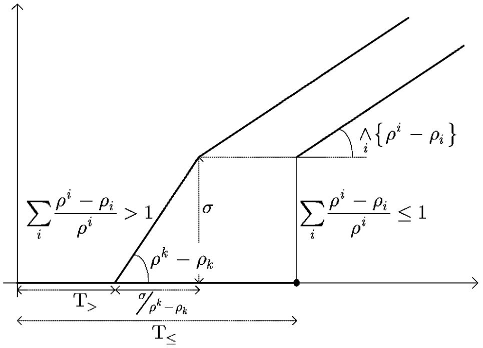

$\bigwedge_i \{\rho^i - \rho_i\}$
$\sum_i \frac{\rho^i - \rho_i}{\rho^i} > 1$
σ
$\sum_i \frac{\rho^i - \rho_i}{\rho^i} \leq 1$
$\rho^k - \rho_k$
$\mathrm{T}_>$
$\sigma / \rho^k - \rho_k$
$\mathrm{T}_\leq$

Fig. 4. The best equivalent service curve for the tagged flow.

Then

$$h_{\text{lub}} = \sum_i \theta^i + \sum_i \frac{\sigma_i}{\rho^i} + \left(\frac{\sigma}{\rho}\right) \wedge \left(\sum_i \frac{\sigma}{\rho^i}\right) \tag{16}$$

According to (16), the delay bound is given by the minimum of the two cases, i.e., either paying the burst only once at the lowest residual rate, or paying it at each node, but at the aggregate service rate.

In order to get a better insight into the result formalized by Theorem 2, we consider the equivalent service curve, out of the set defined by (11), for which the lowest delay bound is reached. To this aim, let define

$$\mathrm{T}_\leq \triangleq \textstyle\sum_i \left\{\theta^i + \frac{\sigma_i}{\rho^i} + \frac{\sigma}{\rho^i}\right\}, \text{ and } \mathrm{T}_> \triangleq \displaystyle\sum_i \left(\theta^i + \frac{\sigma_i}{\rho^i}\right) + \sum_{i \in I(k)} \frac{\sigma}{\rho^i}\left[1 - \frac{\rho^i - \rho_i}{\rho^k - \rho_k}\right]$$

provided, in the last case, that there exists node k as defined in Theorem 2. The following corollary then follows.

Corollary 3. *Under the hypotheses of Theorem 2, the best equivalent service curve, i.e., the one giving the lowest delay bound, out of the family given by (11) is*

$$\beta(t) = \gamma_{\bigwedge_i \{\rho^i - \rho_i\}, \sigma}(t - \mathrm{T}_\leq), \textit{ if } \sum_i (1 - \rho_i / \rho^i) \leq 1, \textit{ and} \tag{17}$$

$$\beta(t) = \beta_{\rho^k - \rho_k, \mathrm{T}_>}(t) \wedge \gamma_{\bigwedge_i \{\rho^i - \rho_i\}, \sigma\left[1 - \frac{\bigwedge_i \{\rho^i - \rho_i\}}{\rho^k - \rho_k}\right]}(t - \mathrm{T}_>), \textit{ if } \sum_i (1 - \rho_i / \rho^i) > 1 \tag{18}$$

Figure 4 shows the shape of the best service curves for the two cases. We note that the best equivalent service curve is not necessarily a straight rate-latency service curve, but rather a piece-wise linear function characterized by two minimum service rates, a low one and a high one, and a latency. It is useful to think of the curve given by (17) as a special case of (18), where the high rate has gone to infinite. By looking at Figure 4, we can observe that, in any case, the low service rate is a long term guaranteed rate matching the minimum available residual rate. On the other hand, the high service rate is a short term guaranteed rate which is related to how fast a burst of tagged flow traffic is served in the worst-case. We note that the fact that a burst can be guaranteed a higher rate than the average one, even in the worst case, follows straightforwardly from the FIFO assumption. In fact, bursts of traffic are assumed to arrive instantaneously, hence, according to FIFO multiplexing, packets in one burst will be served by the node back to back, i.e., at the same guaranteed rate of the flow aggregate. On the other hand, each flow will experience additional latency while waiting for the other flows' bursts to be served.

5 Comparison with Related Work

Despite the research efforts, the amount of results related to deriving end-to-end delay bounds for FIFO aggregate multiplexing is quite poor. A survey on the subject can be found in [8]. A closed form delay bound for a generic network configuration has been derived in [6], under the fluid model assumption, and extended in [7] to take packetization effects into consideration. In both cases, when a generic network configuration is considered, a bound can be derived only for small utilization factors: let H be a bound on the number of nodes traversed by any flow, and ν a bound on the utilization at any link, then the delay bound holds only if $\nu < \nu_{\max} = 1/(H-1)$. Furthermore, the bound is inversely proportional to $1-\nu(H-1)$, that is, the bound approaches infinity when the utilization level ν gets closer to $\nu_{\max}$.

In a more recent paper [4], a tandem of rate-latency servers traversed by a tagged flow as in Figure 3 has been studied. However, an arbitrary network configuration is considered, since any number of interfering flows is allowed, and each flow can interfere with the tagged flow for any number of subsequent nodes. An equivalent rate-latency service curve is derived for such a network. It follows from [4] that a delay bound for the case study represented in Figure 3 is

$$\hat{d} = \sum\nolimits_i \theta^i + \sum\nolimits_i \frac{\sigma_i}{\rho^i} + \frac{\sigma}{\bigwedge_i \left\{ \rho^i - \rho_i \right\}} \tag{19}$$

We compare (19) to the bounds given by (14) and (15). To this aim, we note that, if $\sum_i \left(1-\rho_i/\rho^i\right) > 1$ holds, and $I(k)$ includes only the subset of nodes l (at least one) sharing the same minimum residual rate, i.e., $\rho^l - \rho_l = \bigwedge_i \left\{\rho^i - \rho_i\right\}$ for any $l \in I(k)$, then (15) reduces to (19). On the other hand, in all the other cases, i.e., if either (14) applies, or (15) applies, and $\rho^k - \rho_k \neq \bigwedge_i \left\{\rho^i - \rho_i\right\}$, then h_{lub} is lower than $\hat{d}$. Hence, at least for the case study, the bound we found is better than that derivable from [4]. Furthermore, it is worth noting that when the residual rate at the bottleneck gets closer to zero, then $\hat{d}$ increases indefinitely and approaches infinity. On the other hand, this is never the case for h_{lub}, since either (14) applies, or (15) applies, but then $I(k)$ can not include only the subset of nodes sharing the same minimum residual rate, because otherwise it would not be $\sum_{i\in I(k)} \left(1-\rho_i/\rho^i\right) > 1$. In both cases, h_{lub} is independent of the residual rate at the bottleneck and hence stays finite.

To further detail the comparison between the delay bounds obtained by applying (19) to the case study and the bounds given by Theorem 2, we compute both expressions in the following scenario: the configuration of each node is the same, i.e. $\rho^i = \rho_i + \rho = \rho^0$, $\theta^i = \theta^0$. As far as interfering flows are concerned, we assume $\sigma_i = r \cdot \sigma$ and $\rho_i = r \cdot \rho$, with $r \geq 0$ for simplicity. We plot the delay bound ratio

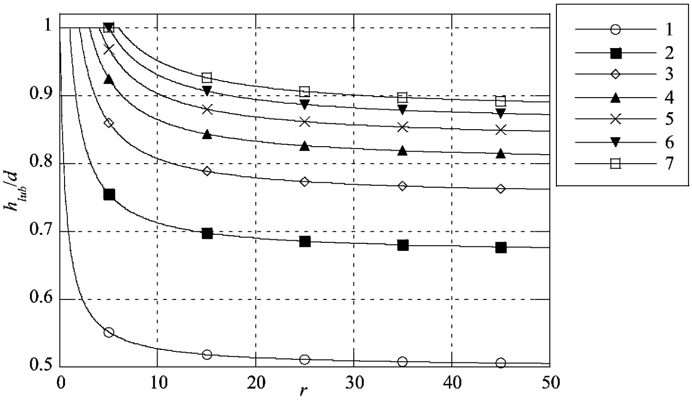


Fig. 5. Delay bound comparison between Theorem 2 and [4] for the case study.

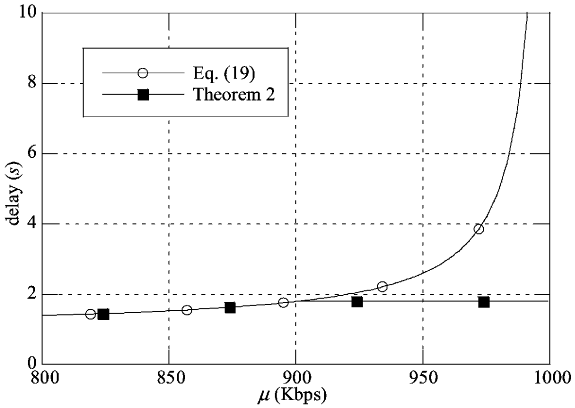


Fig. 6. Delay bound comparison as a function of interfering flows rate.

$h_{\text{lub}} / \hat{d}$ against r for various values of the number of nodes n in Figure 5. As the figure shows, the delay bound ratio is always not greater than 1, it is decreasing with r and approaches $n/(n+1)$ as $r \to \infty$. This shows that, when the tagged flow has a small rate compared to that of the aggregate it belongs to (e.g. when many identical flows are aggregated with the tagged flow at each node), using (19) might lead to gross overrating of the delay bounds.

In order to further detail how employing (19) might lead to inexact conclusions, we report a numerical example. We assume that, for $1 \le i \le n$, $\rho^i = \rho^0 = 1Mbps$, $\theta^i = \theta^0 = 20ms$, $\sigma_i = \sigma = 10Kb$, $\rho_i = \mu$. Furthermore, we set $n = 10$. We compute the delay bounds under (19) and under Theorem 2 when $0 \le \mu \le \rho^0$, and report them in Figure 6. As the figure clearly shows, when $\mu \to \rho^0$ the delay bound computed according to (19) approaches infinite, whereas the one computed according to Theorem 2 stays finite. This has an intuitive explanation. In fact, $\mu \to \rho^0$ implies $\rho \to 0$, which corresponds to the case of a bursty tagged flow requiring an arbitrarily small rate. When scheduled in a per-flow context, the delay bound of such a flow would approach infinite as $\rho \to 0$. However, under FIFO aggregation, the burst of the tagged flow will nevertheless be cleared at the *aggregate* rate ρ^0. Therefore, the delay of the tagged flow stays finite in that case.

6 Conclusions

In this paper, we have derived an end-to-end delay bound for a tagged flow that is FIFO multiplexed with a different interfering flow at each node of a tandem of rate-latency servers. To this aim, we have used the concept of equivalent service curve from network calculus, and derived an infinity of equivalent service curves for the tagged flow, from which we have chosen the one giving the lowest upper bound to the end-to-end delay. The obtained result, summarized by Theorem 2, is not very intuitive but is proved to be better than any other applicable result available in the literature, at least to the best of our knowledge. This paper lays the basis for future work in at least two different directions. Firstly, it should be determined whether the bound given by Theorem 2 is tight or not. Secondly, Theorem 2 should be extended to a general network configuration, where interfering flows are allowed to share the same path of the tagged flow for any number of consecutive nodes. Regarding this second issue, the methodology we have introduced is not directly applicable.

References

1. Firoiu, V., Le Boudec, J.Y., Towsley, D., Zhi-Li Zhang; "Theories and models for Internet quality of service," *Proceedings of the IEEE*, Vol. 90 N. 9 , Sept. 2002, pp. 1565-1591.
2. J.-Y. Le Boudec, and P. Thiran, Network Calculus, Springer-Verlag LNCS vol. 2050, 2001.
3. R. Agrawal, R. L. Cruz, C. Okino, and R. Rajan, "Performance Bounds for Flow Control Protocols," IEEE/ACM Trans. Network., Vol. 7, No. 3, June 1999.
4. M. Fidler, "Extending the Network Calculus Pay Bursts Only Once Principle to Aggregate Scheduling," *Second Int. Work. QoS-IP'03*, Milan, Italy, Feb. 2003.
5. C. S. Chang, Performance Guarantees in Communication Networks, Springer-Verlag, New York, 2000.
6. A. Charny, and J.-Y. Le Boudec, "Delay Bounds in a Network with Aggregate Scheduling," *First Int. Work. QoFIS'00*, Berlin, Germany, Sept. 2000.
7. Y. Yiang, "Delay Bounds for a Network of Guaranteed Rate Servers with FIFO Aggregation," Computer Networks, Vol. 40, 2002, pp. 683-694.
8. J. C. R. Bennett, K. Benson, A. Charny, W. F. Courtney, and J.-Y. Le Boudec, "Delay Jitter Bounds and Packet Scale Rate Guarantee for Expedited Forwarding," IEEE/ACM Trans. on Networking, Vol. 10, N. 4, Aug. 2002, pp. 529-540.
9. P. Goyal, and H. M. Vin, "Generalized guaranteed rate scheduling algorithms: A framework," IEEE/ACM Trans. on Networking, Vol. 5, N. 4, 1997, pp. 561–571.
10. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," IETF RFC 2475, 1998.
11. K. Nichols, V. Jacobson, and L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet," RFC 2638, 1999.
12. B. Davie, et al., "An expedited forwarding PHB (Per-Hop Behavior)", IETF RFC 3246, March 2002.
13. A. Charny, et al., "Supplemental Information for the New Definition of the EF PHB (Expedited Forwarding Per-Hop Behavior), IETF RFC 3247, March 2002.
14. L. Lenzini, E. Mingozzi, G.Stea, "Delay Bounds For FIFO Aggregates: A Case Study," Technical Report, Dip. Ingegneria della Informazione, Pisa, Italy, 2002.

Comparative Performance Analysis of RSVP and RMD

András Császár[1,2] and Attila Takács[1,2]

[1] High Speed Networks Laboratory
Department of Telecommunications and Media Informatics
Budapest University of Technology and Economics
Magyar Tudósok Körútja 2., Budapest, Hungary, h-1117
{Andras.Csaszar,takacs}@tmit.bme.hu
[2] TrafficLab, Ericsson Telecommunication Hungary
Laborc utca 1., Budapest, Hungary, H-1037
{Andras.Csaszar,Attila.Takacs}@eth.ericsson.se

Abstract. Service evolution towards QoS capable applications requires efficient resource reservation protocols. Currently, RSVP is a widely known protocol for this functionality in IntServ networks. Unfortunately, the processing power needs of RSVP make it to a less favoured candidate in high-speed environments. In these scenarios low-complexity DiffServ solutions have a clear advantage. A currently studied and industrially supported DiffServ conform resource management protocol is RMD. In this paper, we certify RMD as a simple and efficient protocol for unicast traffic by comparing the performance of RMD with RSVP, and also verify that the reduction of complexity does not entail loss in performance.

1 Introduction

With the gaining popularity of applications with quality of service requirements the need for sophisticated and scalable resource reservation protocols is increasing. To define the requirements posed against the next generation of resource reservation protocols, a working group has been established within the IETF: Next Steps in Signalling (NSIS). They have defined a signalling framework [1] to build a generic layer structure which can accommodate current and future protocols. Both RSVP and RMD are best suited for the NSIS concept.

Currently, RSVP is a widely accepted solution that was developed within the Integrated Services (IntServ) framework. Unfortunately, as an IntServ conform protocol, RSVP maintains reservation states in a per-flow basis, but with the increasing demand for resource reservation, per-flow information cannot be managed efficiently, especially in high-speed backbone networks with many hundred thousands of flows. To relax the scaling difficulties of RSVP a decreased state-space variant has been proposed [2] that achieves state aggregation by merging the reservations of the same source and destination edge router pairs. This solution is helpful in certain situations but does not solve the overall problem. For example Radio Access Networks (RANs) have a huge number of nodes within a

G. Karlsson and M.I. Smirnov (Eds.): QofIS 2003, LNCS 2811, pp. 41–51, 2003.

single domain, and every base station can be the source or destination of, e.g., a phone call. In this scenario the state aggregation of RSVP is not applicable.

These are just a few reasons, why new protocols must be deployed in the near future. Top candidates are Differentiated Services (DiffServ) conform solutions. A novel framework within IETF is Resource Management in DiffServ (RMD) [3]. RMD is the evolution of Load Control described in [4]. Its development started when the "All-IP" concept evolved making it clear that RANs will be pushed towards IP. It handles reservations at a traffic class or aggregate level, so its scaling property is independent of the number of flows in the network. Although because of its lack of scalability, RSVP and IntServ is not meant to be used in high-speed environments with many flows, it still makes sense to use it as a reference since it provides excellent network utilisation and QoS, and we wish to awake the interest in RMD by outlining that a much simpler protocol can provide similar performance and network efficiency like RSVP.

The paper is organised as follows. In Sec. 2 we shortly describe the protocols and in Sec. 3 we present simulation results about several operational aspects.

2 Protocol Overview

2.1 RSVP

Resource reSerVation Protocol (RSVP) is described in detail in [5], here we only summarise its main characteristics. The most important aspect of RSVP is that it is receiver initiated. This means that the sender only advertises the data stream to be sent with a `Path` message, and the receiver decides whether it is interested in receiving that traffic. If yes, the receiver sends back a `Resv` message to the sender specifying its resource needs. Since the `Resv` message travels upstream but the reservation is required downstream, the `Resv` message must follow the exact same path backward as the Path message did forward. Consequently, per-flow state is required to store the in-bound and out-bound ports for every flow, which are installed when interior routers process the `Path` message.

Throughout the article, we suppose that the application is IP telephony: at dialling, the sender is just establishing the path towards the receiver but already requires the reservation of resources. If the reservation is successful on all links (see Fig. 1), the `Resv` message gets back to the sender node that signals admission to the receiver with a `ResvConf` confirm message. If reservation fails on a link, the corresponding node sends a `ResvError` message back to the receiver application, which decides what happens next, e.g., to retry later or abort. Fig. 2 shows a case where the receiver application decides to release the resources that were reserved on part of the path with the `ResvTear` message. The sender application learns that the connection is refused either by timeout or by application level notification. At the end of the connection resources can be released from either side. The sender may send a `PathTear` message that besides releasing the reservation also deletes all path states of the session. The receiver may initiate a release with a `ResvTear` message that only releases the reservation states but does not delete path states because of multicast considerations.

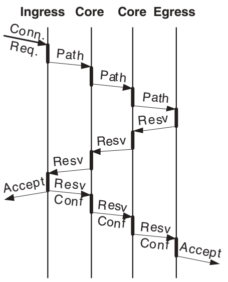


Fig. 1. RSVP: Admitted

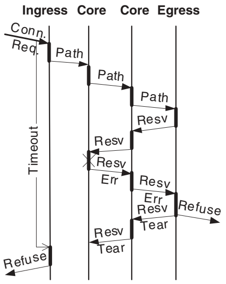


Fig. 2. RSVP: Refused

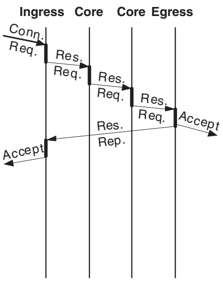


Fig. 3. RMD: Admitted

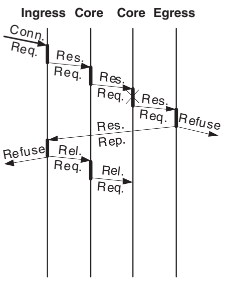


Fig. 4. RMD: Refused

For resource reservation protocols it is imperative that reservations clear even if neither side releases, e.g., due to failure. RSVP solves this with the help of soft states: if not refreshed, reservation and path states time out and are deleted. Path states are refreshed by the directly connected upstream node, while reservation states are refreshed by the immediate downstream node. Upon receiving a `Path` message, the node schedules in itself the sending of a `Path` message to the downstream neighbour. The refresh interval is randomly chosen from an $[0.5R; 1.5R]$ interval, where R is constant for all nodes, typically 30 seconds. Reservation states are similarly refreshed. From this it follow that if the sender or receiver application does not trigger the refreshing of states, sooner or later the states on the whole path will be deleted.

2.2 RMD

The problem with RSVP is that since it is receiver initiated, it needs to send back the `Resv` message on the same path as the `Path` message from which follows that per-flow state information is required in core nodes. This posed no problem to the IntServ environment where the protocol was invented. Of course, RSVP cannot be applied in a high speed unicast environment where simplicity and scalability is of topmost importance. Resource Management in DiffServ (RMD) [3] is a simple and lightweight DiffServ conform resource reservation and admission control protocol. It is sender initiated, meaning that the sender is the one who probes the path towards the receiver. When the probe packet reaches the receiver, it contains the admission decision and the resources have already been reserved over the whole path. Hence, the receiver can notify the sender side with a signalling packet that does not need to traverse the same path as the probe packet followed.

In RMD both inter- and intra-domain operation is specified. In this paper we focus on intra-domain operations where reservations are maintained by stored states. This is called RMD on DemAnd (RODA) [6]. Basically, the protocol operates as follows. After classifying an incoming flow to a DiffServ class, the sender begins the reservation procedure by sending a `Resource_Request` packet towards the destination in which it indicates the resource needs. Core nodes have integer variables for each class to store the reserved bandwidth in bandwidth units, and they also have a pre-configured threshold for each class that determines the maximum reservable bandwidth. The admission decision is a simple comparison to see whether the sum of the reservation counter and the new request is smaller or equal to the reservation threshold. If the inequality evaluates to true then the reservation counter is increased with the value of the new request and the `Resource_Request` packet is forwarded. If the inequality evaluates to false, then the M ("marked") bit in the message is set before passing the packet on, and the TTL field from the IP header may be copied to the signalling message to store the number of hops where the reservation failed. Successive nodes that receive a marked `Resource_Request` will know that the flow was refused on an earlier link, so they will just pass on the message without doing anything.

When the egress receives a `Resource_Request` message, it checks the M bit to see whether the reservation succeeded. Anyway, it creates a `Reservation_Report` packet, in which it sets the M bit according to the value of the received message. If it is 1, it copies the stored TTL value to the report message as well. The report packet is not processed in any intermediate node on its way back to the ingress, so the upstream path is independent of the downstream path. The ingress node learns the admission decision from the report packet. If refused, it generates a `Resource_Release` packet which has the opposite task as the reservation packet: it decreases the reservation counter at each passing link. The ingress node sets the TTL value in accordance with the received report packet so that release happens only down to the rejection point. For our telephony example Fig. 3 shows the protocol messages for a successful reservation and Fig. 4 shows a refused connection. At connection end, the ingress sends a `Resource_Release` packet towards the egress to release the reserved resources on all links.

To avoid infinitely existing reservations, RODA also uses the soft-state timeout principle. Every flow has to send a `Refresh_Update` packet in every R interval. To reduce the time needed to clear "forgotten" reservations from $2 \cdot R$, the refresh period in RODA is divided into equally long cells. Each cell has its own counter for refreshes and new reservations. The current reservation state is calculated based on the sum of the counters of every cell. The reservation in a cells is cleared based on the sliding window principle, for detailed overview see [7]. With this, not-refreshed reservations are detected after $R + \frac{R}{c}$ time, where c is the number of cells in the refresh window.

3 Performance Evaluation and Comparison

Protocol operation can be divided into normal functions and failure recovery. The examination of regular operation is straightforward, the objectives are resource efficiency, fast connection establishment and low protocol overhead. Resource efficiency means that the protocol should allow applications to utilise the network to its full extent. A problem is that the maximum cannot be reached in practice because of non-zero transmission, queuing and propagation delays of the signalling messages. Therefore, we will examine the connection blocking of both protocols to see whether RMD, without per-flow information, is able to

Table 1. Notification times

	RMD		**RSVP**	
	Admitted	**Refused**	**Admitted**	**Refused**
Sender	1 RTT	1 RTT	1 RTT	1 RTT (or timeout)
Receiver	0.5 RTT	0.5 RTT	1.5 RTT	between $(0.5\text{RTT} + d^1 + d_1)$ and 1.5 RTT

provide similar network utilisation as RSVP. Considering the failure tolerance of the protocols, two main issues must be examined. First, we examine how fast the soft states of both protocols handle connections that terminate without proper signalling. Second, we investigate what happens after node or link failures.

3.1 Transient Times

First we analyse the differences in connection establishment and notification times, measured from the arrival of a new flow request at the sender. A successful connection establishment needs 1 RTT to be learnt by the sender with both protocols. With RMD, the receiver side learns the admission decision after 1 downstream trip time. With RSVP the receiver learns the admittance later than the sender by a `ResvConf` message. However, the receiver is the first to know about reservation failure by the `ResvErr` message. After a downstream trip time of the `Path` message, the `Resv` packet travels at least one link upstream. If the reservation is refused on this link, the `ResvError` message needs to travel one link back to the receiver. This way, if d^1 denotes the sum of the queuing-, processing-, transmission- and propagation delays on this first link for the `Resv` packet upstream, while d_1 denotes the same for the `ResvErr` packet downstream, the minimal notification time is $(0.5\text{RTT} + d^1 + d_1)$. The maximal time is 1.5 RTT because in worst case only the last link at the sender node refuses the new flow. Table 1 summarises the notification times for both protocols. We can state that RMD notifies the edges at least as fast as RSVP does.

While regular connection release needs the same time with both protocols, the release of refused connections on the successfully reserved part of the path shows differences. With RMD, a superfluous link reservation always needs 1 RTT to be released. With RSVP, the closer the connection is refused to the receiver, the smaller this partial release time is. Though, as we learn next, this advantage is negligible in real scenarios.

3.2 Connection Blocking

We used the packet-level simulator, ns-2, for our simulation study. We examined the protocols on the COST266 European "large" backbone topology shown on Fig. 5. Traffic was generated between the south-east and north-west parts and between the south-west and north-east parts of the topology with a Poisson-process with 90s mean call holding interval. Since we investigated the behaviour

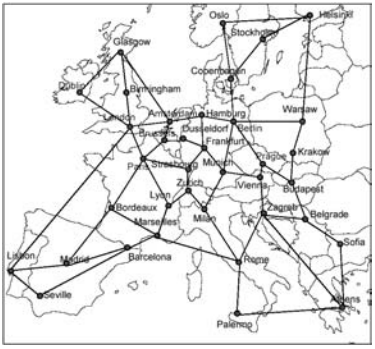


Fig. 5. COST266 "large" topology

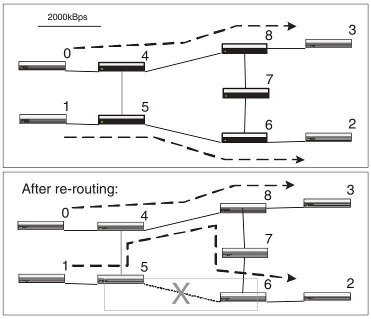


Fig. 6. Severe congestion test-bed

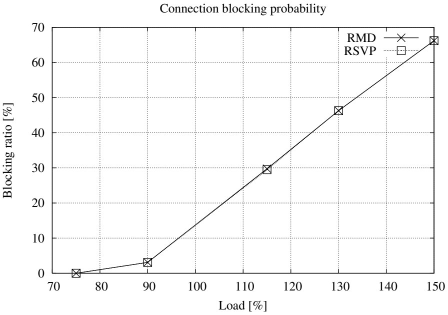


Fig. 7. Connection Blocking

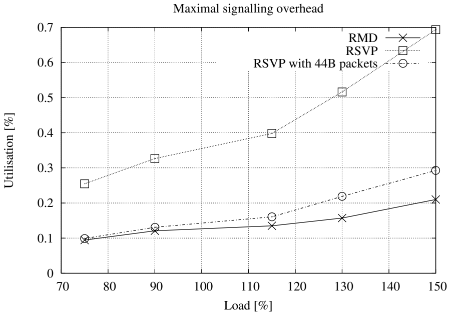


Fig. 8. Signalling overhead

of the protocols as a function of load, the mean session arrival time of the Poisson-process was set to result in the appropriate load. Arriving flows requested 2, 4, 8 or 16 kBps rates so that the aggregation of each flow type means the same load. This was achieved with proportionally increasing session arrival time.

Fig. 7 shows the achieved blocking probabilities. In every case, RMD practically performed identical to RSVP: it did not drop more calls although it did not store per-flow information. This means that a simple protocol can achieve the same network utilisation as the complex reference protocol. Network utilisation is an important performance measure for network efficiency because the task of resource management protocols is to handle the resources so that as many QoS flows are admitted as possible.

3.3 Overheads

Protocol overhead can be divided to computational, storage and bandwidth overhead. Computational and storage overheads are strongly correlated through the complexity of searching algorithms. The memory needs are shown in Table 2, where f means the number of flows and c means the number of cells in case of RMD. In high-speed networks, where the number of flows can be many thousands, DiffServ and RMD have a clear advantage over IntServ and RSVP, since $c \ll f$. Moreover, RMD does not need to create a new signalling message, and so a new IP packet on each passing core node like RSVP, which again brings an additional performance advantage to RMD. Indeed, it has been shown in [8] that the processing delay of RSVP `Resv` messages is more than 1330 times higher than that of RMD `Resource_Request` messages with a 95% confidence interval.

Considering bandwidth overhead, we traced the bandwidth consumption of signalling messages on all links. While RSVP packet sizes followed [5], the packet sizes with RMD were over-estimated with 44 bytes each because [6] describes in detail only part of the IP header option to be used. We calculated the maximal signalling utilisation on any of the links in the network, the values are shown on Fig. 8. We can ascertain that the signalling overhead is not significant with either protocol. Though, RSVP occupies around twice more bandwidth for signalling messages. This is due to the fact that average signalling packet size of RSVP is bigger than 44 bytes. To see the pure difference in the number of signalling messages, we ran the RSVP simulations again with fixed 44 bytes simulated packet sizes. The results show that RSVP still consumes a little higher signalling bandwidth because it involves more packets for refusing a connection than RMD.

3.4 Soft-State Timeout

Both RMD-RODA and RSVP use the soft-state timeout principle to guarantee that reservations are cleared even if the reservation is not explicitly released because of a failure in the client software or in the network. Let us consider an example in which an edge router is shut down because of a power failure, and we would like to know the time after which all bandwidth is soft released.

With RSVP every session times out independently. If R denotes the mean refresh interval, the timeout interval is at least $1.5R$ as the maximum refresh interval for a flow is $1.5R$. In worst case, some sessions were refreshed just before terminating, so these will be soft released after $1.5R$. In the best case, each session times out just after the power outage. We note that RSVP allows higher timeout periods than $1.5R$ to disregard temporary failures and packet losses.

In RODA, the refresh window is divided to c cells, where c is a positive integer. In worst case, there was a refresh just before the flows terminated which is in effect for at least an R long interval (c cells) and one following cell (when it is expected to refresh). At the end of the $(c+1)$th cell its resources will be released. This means a worst-case soft release interval of $(R + \frac{R}{c})$. In the best case, every session is due to refresh in the remaining part of the cell that is active during the failure and this remaining part is very short, ultimately close to zero.

Table 2. Memory requirements

	Edge node	**Core node**
RMD	$O(f)$	$O(c)$
RSVP	$O(f)$	$O(f)$

Table 3. Soft release intervals

	Min	**Avg**	**Max**
RMD	0	$0.5 \cdot (R + R/c)$	$(R + R/c)$
RSVP	0	$0.75 \cdot R$	$1.5 \cdot R$

3.5 Re-routing and Severe Congestion

When link or node failure is detected by dynamic routing protocols, they propagate this information in the network so that other routers can bypass the failed part. At first such a failure seems to be a routing issue, but it has influence onto resource reservation because some flows will traverse a (partially) new path after re-routing, where they did not reserve bandwidth. If the new path has already been highly utilised, the quality of both the original flows of that path and the new ones will degrade, even packet drops may occur. We identify this undesirable situation as "severe congestion". The situation should be detected and handled to reduce the performance degradation introduced by the excess traffic.

Severe congestion and re-routing experiments were carried out on the topology shown on Fig. 6. The traffic scenario was such that node 0 sent traffic towards node 3 and node 1 towards node 2. In other aspects, traffic generation was similar to that we used for the COST266 network, with the arrival process setup so that the result is a 50% load of the network. The reservation threshold was set to 40% on every link. We simulated a failure of link $5-6$ at 200.0 seconds simulation time that forced the Distance-Vector routing protocol to re-route the original $1-5-6-2$ path to the $1-5-4-8-7-6-2$ path which causes severe congestion on the $4-8$ link since it was already utilised before the failure. With the admission threshold set to 40% of the 2000kBps links, before the link failure the two source-destination pairs could together reserve 1600kBps rate on the two distinct paths. After the failure only a single path remains (800kBps), so half of the sessions should be terminated.

With both RMD and RSVP, re-routed flows will sooner or later try to refresh. In both protocols, refreshes operate similarly to reservation messages, they can also be rejected and when the edges are notified about this, the flow can terminate. This means that maximally after an R interval in RMD and an $1.5R$ interval in RSVP, all re-routed flows will either terminate or legally reserve bandwidth and allowed to remain connected. This situation is plotted on Fig. 9. The figure shows the sum of the reservations of the active flows on both source nodes. With $R = 30$sec, RMD needs 30 seconds to retreat below the admission control threshold, while RSVP needs 45 seconds. Both protocols only terminate re-routed flows that did not fit on the new path. However depending on network policy, the 30 or 45 seconds may be too long if during this over-utilisation period packets are dropped, even if from other (e.g. best-effort) traffic classes.

For a quicker reaction, RSVP offers a trivial but perfect solution. This local repair feature of RSVP needs interaction with the routing protocol since the solution is about the immediate refresh of the path states of re-routed flows. This is quite easy since the router has information about the next hop and destination addresses that were affected, so RSVP may search the appropriate flows and send `Path` messages for these flows on the new path. In our example, node 5 can do this. According to [5], the first node (in the example node 6) that has a state installed for the flow of the received `Path` message but with a different previous hop, knows that the path has changed so it will send backwards a `Resv` message on the new path trying to reserve the rate. From this point the procedure is the

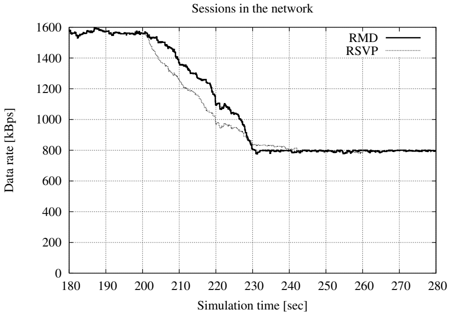


Fig. 9. Severe congestion with refreshes

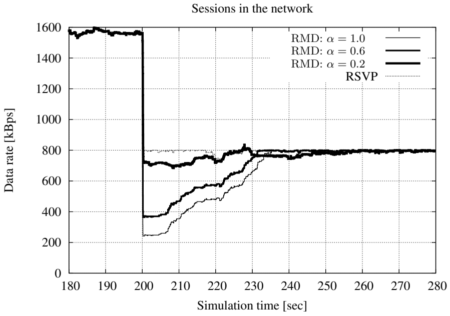


Fig. 10. RSVP and over-reacting RMD

same as regular reservation or refresh: if the reservation is refused, a `ResvErr` message will be sent to the destination. In the example, node 4 will reject these reservations for link $4-8$. This way, RSVP manages to restore normal operation within the order of round-trip times after the failure, see Fig. 10.

In RMD the solution is not so simple because the absence of per-flow states rules out the possibility of forced early refreshes. Therefore, we concentrated onto the effect of re-routing: severe congestion. In [9] we showed that severe congestion can be dissolved in the order of round-trip times. The proposed algorithm was to detect congestion by simple bandwidth measurement or packet drop counts in the core nodes (e.g. node 4), and to mark data packets to indicate congestion towards the egress. More precisely, the core node marks an amount of outgoing packets that is proportional to the overload. This overload is measured against a severe congestion threshold that can be different, typically higher than the reservation threshold. This is called "proportional marking", and marking means the changing of the code-point in the DiffServ field (DSCP) to a value that signals congestion to the egress. This way, the flow agent on the egress node receives marked and not-marked packets from which, after a counting interval, it can itself calculate an overload ratio that is interpreted as a termination probability. Accordingly, the flow agent will decide probabilistically about termination. If it decides to terminate, the egress flow agent will send back a `Congestion_Report` to the ingress to signal that it should stop the data flow.

There is however a problem with this approach that is well known from classical control theory, namely that delay will result in over-reaction of the feedback algorithm and, as a result, too many flows get terminated. Fig. 10 shows this. A simple solution is to dampen the feedback signal by decreasing the probability of flow termination. When an egress flow agent calculates a terminating probability, it is multiplied with $\alpha \in (0;1]$. Fig. 10 also shows the effect of the dampening factor. The over-reaction is effectively reduced but dampening has a natural side-effect of slowing down the reaction.

We have to mention that unlike all previous solutions, this proportional marking – probabilistic termination approach does not guarantee that only re-routed flows get suspended since the congested node has no means to mark packets only from re-routed flows because it does not know which flows were re-routed and which were not, it only knows that there is excess traffic causing overload on the link. Depending on network policy, this can be an advantage. In case of a network error, this approach does not discriminate certain source nodes, rather it lets the burden of failure recovery fall onto a randomly chosen group of flows.

RMD also allows temporal over-allocation as a network policy. This means that, e.g., high-QoS voice flows, for which the users pay a higher price, are allowed to remain in the network even after re-routing at the cost of best-effort traffic. In our example, we could set the severe congestion threshold up to, say, 80% when all voice flows could have been preserved for the time being. Of course, best-effort traffic would be impaired as long as the voice sessions are not released and the voice aggregate does not return below the reservation threshold.

4 Conclusions

In this article we investigated the performance of the RMD-RODA protocol and compared it to RSVP, which can be seen as a reference protocol for performance evaluation purposes. RMD does not store per flow information in core routers, but it is still able to deliver the same connection blocking ratio and meet the same QoS requirements, without so harsh memory and processing requirements like that RSVP poses against the network. Also, connection establishment and transient times of RMD are in the same order as those of RSVP. The soft-state failure handling capabilities of the protocols are similar, however when it comes to re-solve severe congestion in the order of round trip times, RSVP outperforms RMD based on its per-flow information. Though, we showed that with simple heuristic enhancement of the protocol, RMD can achieve similar fault handling performance as RSVP.

Nevertheless, yet further enhancements are possible in this field, so next we will focus on perfecting and refining the severe congestion handling of RMD.

References

1. Hancock, R., Freytsis, I., Karagiannis, G., Loughney, J., den Bosch, S.V.: Next steps in signaling: Framework. Internet draft, IETF (2003) Work in progress.
2. Baker, F., Iturralde, C., Faucheur, F.L., Davie, B.: Aggregation of RSVP for IPv4 and IPv6 reservations. RFC (2001)
3. Westberg, L., Jacobsson, M., Karagiannis, G., Oosthoek, S., Partain, D., Rexhepi, V., Szabó, R., Wallentin, P.: Resource management in diffserv (RMD) framework. Internet Draft draft-westberg-rmd-framework-03, IETF (2002) work in progress.
4. Turányi, Z.R., Westberg, L.: Load control: Congestion notifications for real-time traffic. In: 9th IFIP Working Conference on Performance Modelling and Evaluation of ATM and IP Networks, Budapest, Hungary (2001)
5. Braden, R., Zhang, L., Berson, S., Herzog, S., Jamin, S.: Resource reservation protocol (RSVP) – version 1 functional specification. RFC RFC2205, IETF (1997)
6. Westberg, L., Jacobsson, M., Karagiannis, G., de Kogel, M., Oosthoek, S., Partain, D., Rexhepi, V., Wallentin, P.: Resource management in diffserv on demand (RODA) PHR. Internet draft, IETF (2002) Work in progress.

7. Marquetant, A., Pop, O., Szabó, R., Dinnyés, G., Turányi, Z.: Novel enhancements to load control - a soft-state, lightweight admission control protocol. In: Proceedings of QofIS2001 – 2nd International Workshop on Quality of future Internet Services. Volume 2156 of LNCS., Coimbra, Portugal, COST263, Springer Verlag (2001) 82–96
8. Westberg, L., Császár, A., Karagiannis, G., Marquetant, A., Partain, D., Pop, O., Rexhepi, V., Szabó, R., Takács, A.: Resource management in diffserv (RMD): A functionality and performance behavior overview. In: Proceedings of PfHSN'2002 – Seventh International Workshop on Protocols For High-Speed Networks. Volume 2334 of LNCS., Berlin, Germany, Springer Verlag (2002) 17–34
9. Császár, A., Takcs, A., Szabó, R., Rexhepi, V., Karagiannis, G.: Severe congestion handling with resource management in diffserv on demand. In: Proceedings of Networking 2002 – The Second Intl. IFIP-TC6 Networking Conference. Volume 2345 of LNCS., Pisa, Italy, Springer Verlag (2002) 443–454

Global Time for Interactive Applications over *Global* Packet Networks

Mario Baldi[1,2] and Yoram Ofek[1]

[1] Synchrodyne Networks, Inc., New York, NY
{baldi,ofek}@synchrodyne.com
[2] Torino Polytechnic, Computer Engineering Department, Torino, Italy
mario.baldi@polito.it
www.polito.it/~baldi

Abstract. This work presents that *global time* (a.k.a. time-of-day or coordinated universal time – UTC) is essential in maximizing user perceived quality of service, while eliminating both switching bottlenecks – critical in very high capacity network core – and communications link bottlenecks – at the low speed access (e.g., wireless and DSL). Global time obtained, for example, from GPS (Global Positioning System) or Galileo, is used in the design of all streaming media applications such as toll quality telephony, videotelephony and videoconferencing. The proposed solution, that can be applied to the Internet without changes to any of the existing protocols, provides a guaranteed quality service to each application without requiring nodes to keep state information on microflows.

1 Introduction

The deployment of new high bandwidth multimedia applications will boost network traffic and consequently the deployment of very high capacity transmission technologies, such as Wavelength Division Multiplexing (WDM). On the other side, since multimedia services will have to be widely available, various “low speed” access technologies, such as wireless, DSL, and cable modem will be deployed. In this scenario networks will be characterized by (*i*) *electronic switching* bottlenecks and (*ii*) *communications link* bottlenecks that are created by the bandwidth mismatch between high capacity core technologies and low speed access technologies.

Global time (a.k.a. time-of-day or coordinated universal time – UTC) enables to implement *time-driven priority* (TDP) forwarding which eliminates communications link bottlenecks since it completely avoids congestion, also in bandwidth mismatch points. Moreover, the design of TDP-based packet switches is highly scalable because it is based on simple switching fabrics without any speedup with respect to input link capacity.

Many interactive applications, such as, telephony, videotelephony and videoconferencing, require at the receiver *continuous playing* of the samples captured at the sender, as shown in Figure 1. Continuous playing requires a *constant delay* service to be provided by the application layer, i.e., where samples are acquired and played.

G. Karlsson and M.I. Simirnov (Eds.): QofIS 2003, LNCS 2811, pp. 52–62, 2003.

Figure 1 shows the generic model of an application requiring continuous playing and highlights the components of the end-to-end delay.

- The processing delay (**P**) is introduced on the multimedia desktop side. It encompasses the time needed for analog-to-digital conversion and coding (e.g., voice or video compression).
- The network delay (**N**) is the time needed to move packets across the network; it also includes the shaping and packetization delays.

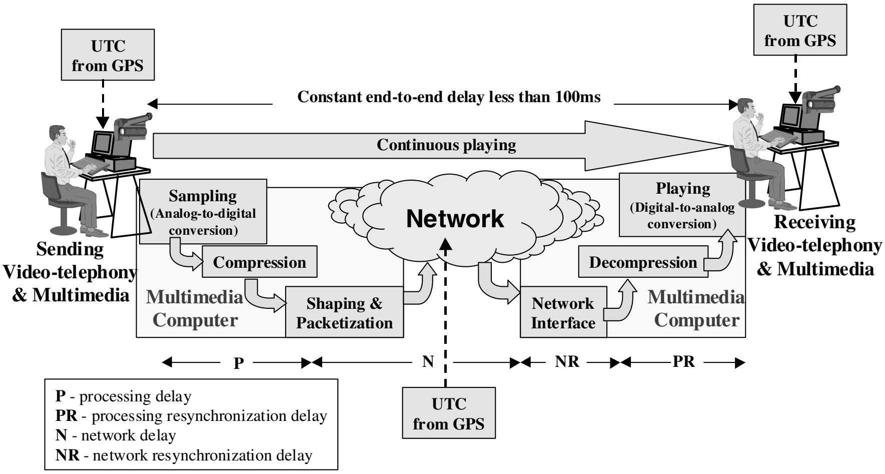

UTC from GPS
UTC from GPS
Constant end-to-end delay less than 100ms
Continuous playing
Sampling (Analog-to-digital conversion)
Network
Playing (Digital-to-analog conversion)
Sending Video-telephony & Multimedia
Receiving Video-telephony & Multimedia
Compression
Decompression
Multimedia Computer
Shaping & Packetization
Network Interface
Multimedia Computer
P
N
NR
PR
P - processing delay
PR - processing resynchronization delay
N - network delay
NR - network resynchronization delay
UTC from GPS

Fig. 1. Generic Model of an Application Requiring Continuous Playing

Since the above delay components can vary during a session, some specific action is required at the receiver to keep constant the end-to-end delay between the application layers, thus enabling continuous playing. Before samples are played, delay variations should be "smoothed out" by buffering the samples that have experienced (in the network and in the decoder) a delay shorter than the maximum. This introduces two *resynchronization delay* components that are typically the time spent in a *replay buffer* [4]. This time is such that all samples on exiting the replay buffer have experienced the same delay since the time they were acquired at the sender side. Such an overall delay is equal to or larger than the *delay bound* the system can guarantee.

- The processing resynchronization delay (**PR**) cancels delay variations introduced by the sample coding process.
- The network resynchronization delay (**NR**) cancels variations of the delay experienced in the network (e.g., the delay jitter due to queuing in the network).

The processing delay (**P)** depends on the signal processing performed, which differs, for example, for voice and video. A delay below 100 ms gives human communicators the feeling of live interaction. Since in a global network the propagation delay alone is about 100 ms, every other delay component should be kept as short as possible. This paper shows that global time enables the other delay components (**N, NR,** and **PR**) to be minimized independent of the packet technology (e.g., ATM or IP) deployed and the rate of sessions.

The two resynchronization components, **PR** (processing resynchronization delay) and **NR** (network resynchronization delay), can be kept small, e.g., 25-125 μs. The network delay (**N**) is propagation delay plus a small additional delay per switch, e.g., 50-250 μs. Thus, global time enables the end-to-end delay bound to be minimized [5].

Global time is used in two ways:

1. To implement time-driven priority (TDP) forwarding of packets in global networks, which
   i. guarantees a maximum queuing delay of a few milliseconds, independent of the flow rate and the network load, also in bandwidth mismatch points;
   ii. enables the implementation of efficient packet switch architectures based on low complexity switching fabrics. This increases the scalability of switches and eliminates the electronic switching bottleneck.
2. To synchronize the acquisition of samples at the sender (e.g., video capture card) and their continuous playing at the receiver (e.g., video display) with one another and with the TDP forwarding.

2 Network Architecture and Deployment

According to various provisioning models, such as, Integrated Services [6] over the Internet and ATM User Network Interface (UNI), applications signal their Quality of Service (QoS) requirements to the network. If the network has enough resources to satisfy the request, they are reserved and packets transmitted by each application are handled in a way that QoS is guaranteed to their flow (usually called *micro-flow*). Most of the queuing algorithms used to implement such packet handling have to maintain status information for each micro-flow, which is recognized not to be scalable. Time-Driven Priority (TDP) forwarding does not require per micro-flow information in intermediate nodes. Thus, TDP has similar provisioning scalability as forwarding equivalent class (FEC) in IP/MPLS traffic engineering (TE).

In accordance to the Differentiated Services [7] model over the Internet, micro-flows should be aggregated in the network core in order to improve scalability by increasing the granularity with which switches handle packet flows.

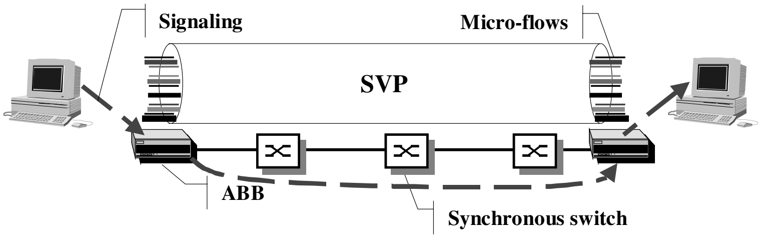


Fig. 2. Synchronous Virtual Pipe (SVP) and Access Bandwidth Brokers (ABBs)

Synchronous Virtual Pipes (SVPs) can be set up over networks deploying global time in order to aggregate multiple micro-flows, thereby relieving core nodes from participating in micro-flow level signaling. An SVP can be regarded as a virtual leased line. In order to deterministically guarantee QoS to single micro-flows, Access

Bandwidth Brokers (ABBs) at the edges of an SVP handle signaling requests from the applications whose packet micro-flows are to traverse the SVP, and determine the availability of resources within the SVP. If a request is accepted, the ABB reserves a fraction of the SVP resources to the corresponding micro-flow. As shown in Figure 2, intermediate switches are not involved in the signaling operation, but the micro-flow will receive deterministic QoS guarantees, even though intermediate switches on the SVP do not have any awareness of the micro-flow.

The Internet (as well as ATM networks) is based today on asynchronous packet (cell) switches which do not feature TDP forwarding. Thus, especially in the initial deployment phase, TDP switches will coexist and interoperate with current asynchronous packet switches. Figure 3 shows a scenario, likely to be common in the early days of TDP deployment, in which end stations connected to asynchronous local area or access networks communicate through a TDP backbone. Synchronous *boundary nodes* control the access to SVPs set up on the synchronous backbone performing both policing and shaping of packets flows– i.e., they *synchronize* packet forwarding. TDP provides the minimum delay bound when deployed end-to-end, but it can be beneficial even when its use is confined to subnetworks. The node at the ingress of a TDP subnetwork, which shares the global time reference, eliminates the delay variation experienced by packets in the asynchronous network; then packets benefit of the controlled delay service provided by the synchronous subnetwork.

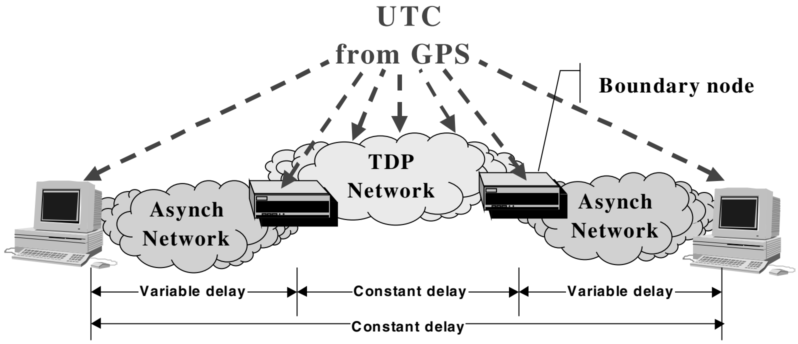


Fig. 3. Interoperation between TDP Networks and Asynchronous Networks

The delay jitter introduced by asynchronous segments can be completely canceled in the end-to-end communication if end stations use the common global time to attach time stamps to the packets they generate. The Real-time Transport Protocol (RTP) [8] can be used to carry the time stamp. The source end system generates the time stamp according to the common global time. The node at the ingress of a TDP subnetwork can take advantage of the time stamp to eliminate the delay variation across the asynchronous network. For example, assuming a known delay bound on the asynchronous subnetwork, say of *k* time units, the ingress TDP switch determines the time the packet should be forwarded by adding *k* time units to the time stamp value.

Packets travel through the TDP cloud with controlled delay and no loss. With reference to Figure 3, when packets arrive to the destination they have experienced a variable delay in the asynchronous access subnetwork to which the destination is connected. By knowing the delay bound, say of *b* time units, on the asynchronous segments traversed by the packet, the end station can completely eliminate the jitter by determining the replay time as the time stamp value plus *b*.

The same approach can also be applied when packets traverse more than one TDP subnetwork and more than two asynchronous networks. SVPs can be set up over multiple synchronous subnetworks interconnected by asynchronous ones. Access devices at the ingress of synchronous segments resynchronize packets that have experienced variable delay across asynchronous subnetworks so that the overall delay throughout the SVP is constant.

3 Global Time and Periodic Forwarding: Time-Driven Priority

All packet switches are synchronized to a global common clock with a basic time period that is called *time frame* (TF). The TF duration is derived from the UTC (coordinated universal time) second received, for example, from a time distribution system such as GPS, GLONASS, TWSTFT (Two-Way Satellite Time and Frequency Transfer), and, in the future, Galileo. For example, by dividing the UTC second by 8,000, the duration of each time frame is T_f = 12.5 to 125 μs; however, the time frame duration can be set as needed[1].

TFs are grouped into a *time cycle*; Figure 4 shows an example of a time cycle that contains 100 TFs, i.e., there are 80 time cycles in a UTC second. Time cycles are further organized in *super cycles*, each of which typically equals one UTC second. This timing structure is useful to perform resource reservation in order to provide guaranteed services.

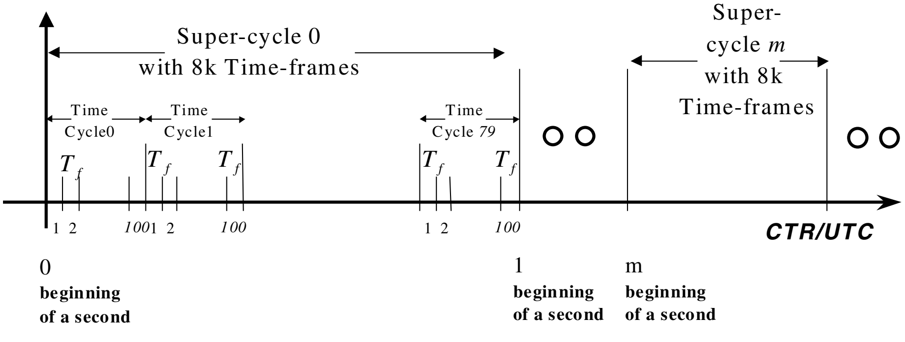


Fig. 4. Global common time reference

Thus, all switches around the globe have an identical time structure that is collectively called a *Common Time Reference* (CTR). The CTR can be used to coordinate the acquisition of samples at the sender with the playing of them at the

[1] UTC receivers from GPS are available from many vendors for a low price (for example, the price of a one PPS (pulse per second) *UTC* clock, with accuracy of 10-20 nanoseconds, is about $200). By combining UTC from GPS with local Rubidium or Cesium clocks it is possible to have a correct UTC (±1 μsecond) without an external time reference from GPS for days (with Rubidium clock) and months (with Cesium clock).

receiver. Moreover, the CTR enables the implementation of *Time-Driven Priority* (TDP) [1] [2] for periodically forwarding real-time packets, for example inside IP and ATM networks, as shown in Figure 5.

Periodic forwarding indicates that the forwarding pattern repeats itself in every time cycle and in every super cycle. TDP guarantees that the end-to-end delay jitter is less than one TF and that reserved real-time traffic is transferred from the sender to one or more receivers with no loss due to congestion.

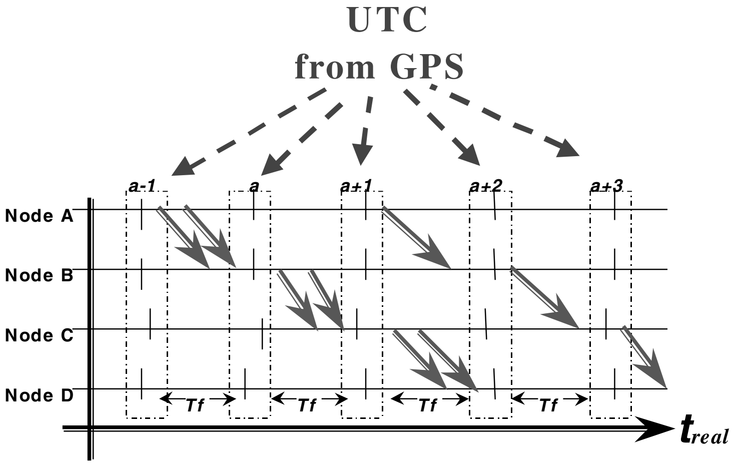


Fig. 5. Periodic forwarding with time-driven priority (TDP)

The simple TDP operation is generalized by adding the following two conditions:

(i) All packets that should be sent in TF *i* by a node are in its output port before the beginning of TF *i*, and

(ii) The delay between an output port of one node and the output port of the next down-stream node is a constant integer number of TFs.

The generalized TDP forwarding, exemplified in Figure 6, is important because of the possibly long lasting software protocol processing in heterogeneous multi-protocol internetworking environments. In this case, a predefined, but fixed, number of TFs will be added in some intermediate switches with an increase in the end-to-end delay; the end-to-end delay jitter will remain constant.

In Table 1 some of the unique properties of TDP is compared with four types of communication networks.

4 Periodic Bursty Services: Videotelephony and Videoconferencing

Videotelephony and videoconferencing, like telephony, rely on continuous playing at the receiver of samples acquired at a fixed rate at the sender. Samples are both voice and video frames captured by a camera and digitized by a frame grabber. Video frames have two main differences from voice samples.

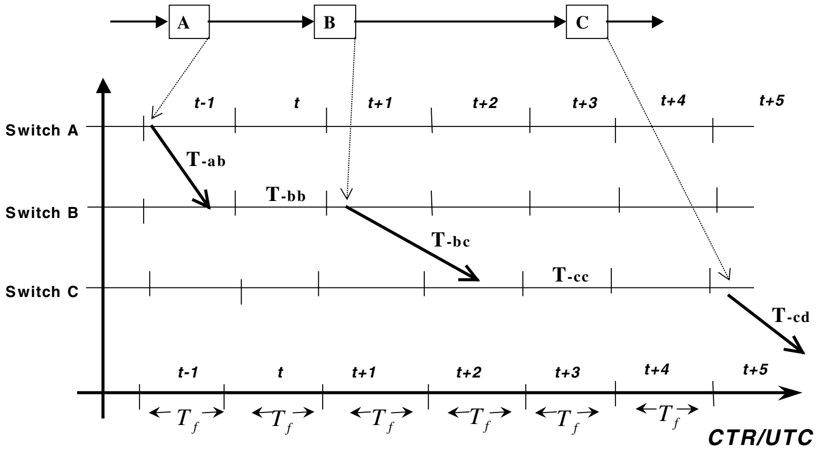


Fig. 6. Generalized TDP with arbitrarily bounded link and switch delays

Table 1. A comparison with other methods

Communication methods Service capabilities		Circuit Switching or PSTN	Single Async. Priority w/no reservation	Multiple Asynchronous Priorities – DiffServ	Time-driven Priority
Data: mail, ftp, etc.		No	Yes	Yes	Yes
Interactive – on a Global Scale	Phone	Yes	No	Not proven: depends on scheme	Yes
	Video-phone	Yes	No	Not proven: depends on scheme	Yes
Utilization vs. Loss: with a mixture of high and low speed links		Full utilization and No loss	either Low utilization or High loss	Utilization can be low, and loss can be high depends on the specific scheme – Requires overprovisioning	Full utilization and No loss Easy to schedule
Experience		100+ years	25+ years	New technology	New technology

1. The sampling rate is usually lower, from a few to 30 frames per second – versus the 8,000 voice samples per second required for voice encoding.
2. The amount of bits required to encode each video frame sample is much larger, at least a few kilobits – versus the 8 bit or less used for a single voice encoding.

When circuit switching is used to transfer video frames, the encoder is operated in such a way that it produces a constant bit rate flow. This is required in order to fully utilize the channel allocated to the session. Consequently, the transmission delay of a single video frame is the time between two successive video frames. This is because the transmission of the current video frame should continue, in a constant rate, until the next video frame is ready. For example, if the sampling rate is ten video frames a second, the transmission delay alone is 100 ms, which is unacceptable in interactive applications.

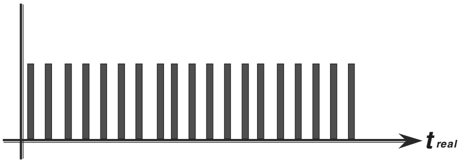


Fig. 7. Periodic Bursty Transmission of a Video Stream

The elimination of such a long transmission delay is achieved by transmitting the captured video frame as a short burst. Packet switching allows burst transmission of video frames in packets, i.e., as shown in Figure 7, a video frame is captured, put into a packet, and then transmitted as a burst into the network. Therefore, the only way to transmit video frames for interactive applications with minimum delay is over packet-switched network.

The next question is how to ensure that each transmission of a video frame will reach its destination with no loss and with minimum delay bound. Since video frames are captured periodically, in order to minimize the delay bound, *periodic resource allocation* with *periodic transmission synchronized with their capture* are required. TDP is the only known way to satisfy those requirements, while guaranteeing no loss with minimum delay bound, as shown in Figure 8.

It is worth noting that even though all the video frames are not encoded with exactly the same amount of bits, the capacity reserved on the links is not wasted since it used to forward "best effort" (i.e., non-reserved) traffic. No loss due to congestion is guaranteed to all the video frames, provided that the amount of bits encoding them does not exceed the reservation.

4.1 Complex Periodicity: MPEG Video

Some video encoding schemes, like MPEG, encode frames with significantly different amounts of bits in a *periodic* fashion. MPEG encodes pictures in one of two different ways[2]:

Intra-frame Coding eliminates spatial redundancy inside pictures and the resulting encoded picture is called *I-frame*.

Predictive Coding eliminates temporal redundancy between a picture and the previous one through motion estimation. The obtained encoded picture is called *P-frame* and it is typically from 2 to 4 times smaller than an I-frame. The more similar two subsequent pictures, the smaller the amount of bits produced for each P-frame. Subsequent pictures are similar if the scene is slow moving, thus not changing much from a video frame period to the other. In summary, predictive coding delivers more compression on slow scenes, such as those captured in videoconferences.

It may be inefficient to transfer such a compressed video stream over a constant bit rate channel, e.g., the one provided by a circuit switched network (see [4] for a detailed discussion). If the encoder is operated in such a way that it produces a

[2] Actually, a third type of encoding, called bi-directional predictive coding exists. Before a picture can be coded, a reference subsequent picture must be captured and coded. This introduces a delay of a multiple video frame periods that is not acceptable given the 100 ms end-to-end delay bound requirement. Thus, this type of compression is not considered here.

constant bit rate flow, it can introduce a delay up to the time between two successive I-frames; such a delay, which can be on the order of 500 ms, is obviously unacceptable for interactive applications.

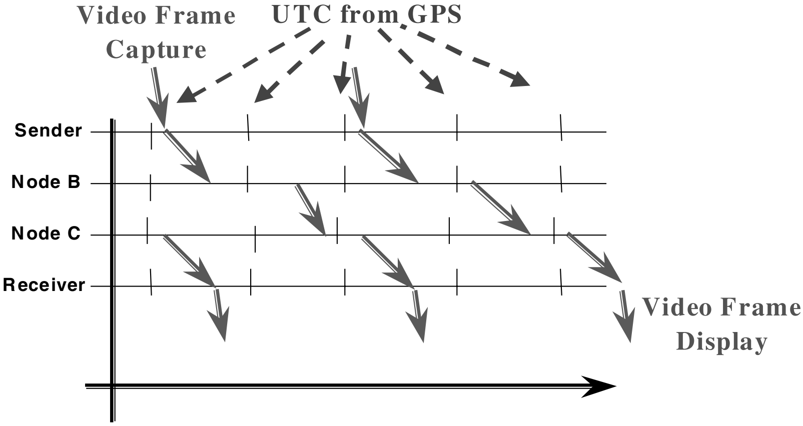


Fig. 8. Periodic Capture, Transmission and Display of Video Frames

Complex periodicity scheduling allows MPEG video frames to be transmitted as soon as they are encoded, analogously to what is described in Section 4 for fixed size video frames. TDP together with global time facilitates the realization of complex periodicity scheduling, which provides deterministic quality of service guarantees to variable bit rate traffic. In complex periodicity scheduling the amount of transmission capacity reserved on the links traversed by a session varies in a repetitive manner. Thus, with complex periodicity scheduling an MPEG video stream can be transmitted as shown in Figure 9.

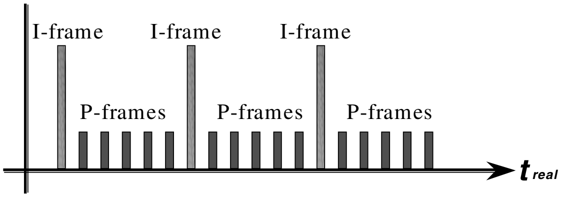


Fig. 9. Complex Periodicity Scheduling of MPEG Video Stream

Thus, TDP with complex periodicity scheduling enables transmission of MPEG encoded video without the need of introducing a shaping delay in the encoder, with effective utilization of reserved network resources, with network delay basically equal to the propagation delay, with virtually no jitter, and without loss due to network congestion for all the video frames, provided that their size does not exceed the allocation in the corresponding TF. In a related study [5] it was shown that an MPEG encoder can be successfully implemented in a way that encoded video frame size never violates the resource allocation.

5 Scalability of Synchronous Switches

The deployment of global time to control packet forwarding has a twofold impact on switch scalability.

TDP allows controlling traffic patterns across each switch since it bounds the maximum number of packets to be moved during each TF to the same output from every input. This can be leveraged in the switch design: optimal input-output switching is obtained with low (x2) speed up in the switching fabric, or even without speed up at all. Instead, asynchronous switches require high speed up in order to achieve high throughput. Thus, given the state of the art aggregate switching fabric capacity, synchronous switches can accommodate higher capacity inputs than asynchronous switches. In other words, since the throughput of a *P* port synchronous packet switch with no fabric speedup is roughly the same as an asynchronous packet switch with a speed up of *P*, the interfaces mounted by the former can be *P* times faster than the latter.

The ability to control traffic patterns across switches can be benefited even further. A non-blocking fabric allows any possible input-output connection *at any time*; a simpler fabric allows only a limited number of simultaneous input-output connections. When TDP switches are deployed, packet arrival can be controlled in a way that incompatible input-output connections are not required during the same TF, thus avoiding unfeasible switching configurations. In other words, thanks to the increased flexibility introduced by the time dimension, synchronous packet switches with blocking fabrics can achieve the same throughput as asynchronous ones with non-blocking fabrics. For example, a non-blocking PxP crossbar requires P^2 crosspoints, while a self routing blocking fabric requires only $2P\log_2 P$ crosspoints. As a consequence, by using the latter fabric, a synchronous switch can accommodate as much as $½P/\log_2 P$ more ports than an asynchronous one that uses a crossbar.

5.1 CTR/UTC accuracy

The requirement of CTR accuracy, and hence UTC accuracy, has a direct impact on cost, stability, and implementation complexity. With a time frame delimiter the UTC accuracy requirement is $\underline{½\cdot T_f}$ (i.e., UTC±1/2·(12.5μs to 125μs)). The reason for such a relaxed requirement is that the UTC is not used for detecting the time frame boundaries, as they are detected by the delimiters (e.g., unused code-words in the serial bit-stream). Consequently, the only function of UTC is enabling the correct mapping of the incoming time frames from the input channel to the CTR time frames. It is easy to show that up to $\underline{½\cdot T_f}$ timing error can be tolerated while maintaining the correct mapping of time frames. (Today, a time card with 1 pps (pulse per second) UTC with accuracy of 10-20 ns is available from multiple vendors. The card is small and costs $100-200.)

6 Conclusions

This work shows how global time can be used to minimize the end-to-end delay for applications that require at the receiver continuous playing of samples captured at the sender. Specifically, global time eliminates the resynchronization delay (see

Figure 1). Deployment of global time as a common time reference among switches to implement time-driven priority (TDP) forwarding of packets enables the provision of a service with the following characteristics:

- Deterministic absence of loss;
- Quality independent of the connection rate;
- Per switch delay of two time frames;
- End-to-end jitter of one time frame.

The service with such characteristics can be provided to any application generating one of the following types of traffic:

- Constant bit rate (e.g., voice telephony),
- Variable bit rate with periodic burstiness (e.g., videotelephony),
- Variable bit rate with complex periodicity (e.g., MPEG).

When a TDP network is deployed to carry voice calls, compression can be fully benefited to reduce the amount of link capacity used by each call. This is not the case with other asynchronous queuing schemes that possibly require overallocation to satisfy end-to-end delay requirements.

When dealing with videotelephony, encoded video frames are transmitted in bursts of packets with controlled delay and no loss. The delay perceived by users is lower than the one obtained by carrying video calls over a circuit switching network which requires delay to be introduced for smoothing out the burstiness of the video source.

Global time and TDP forwarding offer the only solution for the transmission of video frames also when they are encoded with a highly variable amount of bits, such as with MPEG. Each video frame can be transmitted in a burst of packets as soon as it is encoded with no shaping delay and no loss due to congestion. Through complex periodicity scheduling, resource reservation is fitted to the size of encoded video frames thus leading to efficient resource utilization.

References

1. C-S. Li, Y. Ofek, A. Segall and K. Sohraby, “Pseudo-Isochronous Cell Switching in ATM Networks,” *Computer Networks and ISDN Systems*, No. 30, 1998.
2. C-S Li, Y. Ofek and M. Yung, “Time-driven Priority Flow Control for Real-time Heterogeneous Internetworking,” *IEEE INFOCOM'96*, Apr. 1996.
3. M. Baldi and F. Risso, "Efficiency of Packet Voice with Deterministic Delay," *IEEE Communications*, Vol. 38, No. 5, May 2000.
4. M. Baldi and Y. Ofek, “End-to-end Delay Analysis of Videoconferencing over Packet Switched Networks,” *IEEE/ACM Transactions on Networking*, Aug. 2000.
5. M. Baldi, Y. Ofek, “End-to-end Delay of Videoconferencing over Packet Switched Networks,” RC 20669 (91480), IBM - T. J. Watson Research Center, Yorktown Heights, NY, USA, Dec. 1996.
6. R. Braden, D. Clark, and S. Shenker, “Integrated Service in the Internet Architecture: an Overview,” RFC 1633, Jul. 1994.
7. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, “An Architecture for Differentiated Services,” RFC 2475, Dec. 1998.
8. H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications,” RFC 1889, Jan. 1996.

The Performance of Endpoint Admission Control Based on Packet Loss

Ignacio Más*, Viktória Fodor, and Gunnar Karlsson

Department of Microelectronics and Information Technology
KTH, Royal Institute of Technology
S-16440 Kista, Sweden
{nacho,viktoria,gk}@imit.kth.se

Abstract. Endpoint admission control solutions, based on probing a transmission path, have been proposed to meet quality requirements of audio-visual applications with little support from routers. In this paper we present a mathematical analysis of a probe–based admission control solution, where flows are accepted or rejected based on the packet–loss statistics in the probe stream. The analysis relates system performance to design parameters and the experienced probe packet loss probability to the packet loss pro bability of accepted flows.

1 Introduction

Today's new applications on the Internet require a better and more predictable service quality than what is possible with the available best–effort service. Audiovisual applications can handle limited packet loss and delay variation without affecting the perceived quality. Interactive communication, in addition, requires stringent delay requirements. For example, IP telephony requires that a maximum of 150 ms one–way delay should be maintained during the whole call.

In recent years a new family of admission control solutions has been proposed to support applications with quality of service (QoS) requirements by limiting the network load. These proposals aim at providing QoS with very little or no support in the routers. They also share a common idea of endpoint admission control: A host sends probe packets before starting a new flow and decides about the flow admission based on statistics of the probe packet loss [7,5,10], explicit conges tion notification marks [8,9], delay or delay variation [2,1]. The admission decision is thus moved to the edge nodes and is made for the entire path from the source to the destination, rather than on a per–hop basis. Consequently, the service does not require any explicit support from the routers other than one of the various scheduling mechanisms supplied by DiffServ, and the mechanism of dropping or marking packets.

In this paper we provide a mathematical analysis of the probe–based admission control (PBAC) scheme proposed in [7,5,10,11]. In this scheme, the admission control is based on the packet–loss ratio of the probe stream. The aim of

* Corresponding author

G. Karlsson and M.I. Smirnov (Eds.): QofIS 2003, LNCS 2811, pp. 63–72, 2003.

the admission control is to provide a reliable upper bound on the packet loss probability for accepted flows. The end–to–end delay and delay jitter is limited by the use of small buffers inside the network. The goal of the mathematical anal ysis is to relate performance parameters, such as probe and data packet loss probabilities, flow acceptance probability and network utilization to system parameters, such as buffer size, probe length and admission threshold.

The paper is organized as follows: In Section 2 we provide a short description of the PBAC solution, while Section 3 presents an approximate analytical model to calculate probe and data loss probabilities. Section 4 gives a performance evaluation of the system as well as the validation of the analytical model. Finally, we conclude our work in Section 5.

2 Probe–Based Admission Control

QoS provisioning with probe based admission control is based on the following main ideas: i) Network nodes utilize short buffers for data packets of accepted flows to limit end–to–end delay and delay jitter; ii) all end points in the network perform admission control; iii) the admission control is responsible for limiting the packet loss of accepted flows; iv) the admission decision is based on the packet loss ratio in the probe stream, thus flows are accepted if the estimated packet loss probability is b elow the acceptance threshold; and v) probe packets are transmitted with low priority at the routers to ensure that probe streams do not disturb accepted flows.

To provide priority queuing for probe and data packets at the network nodes we consider a double–queue solution in this paper: One buffer is dedicated for high priority data packets and the other for low priority probe packets. The size of the high priority buffer for the data packets is selected to ensure a low maximum queuing delay and an acceptable packet loss probability, i.e., to provide packet scale buffering [12]. The buffer for the probe packets can accommodate one packet at a time, to en sure an over–estimation of the data packet loss. Similar priority queuing can be achieved using a single buffer with a discard threshold for the probe packets. The two solutions are compared in [10]. The mathematical analysis of the threshold–queue is not included in this paper.

The acceptance threshold is fixed for the service class and is the same for all flows. The reason for this is that the QoS experienced by a flow is a function of the load from the flows already accepted in the class. Considering that this load depends on the highest acceptance threshold among all flows, by having different thresholds all flows would be degraded to the QoS required by the one with the less stringent requirements. The class definition has also to state the maximum data rate allowed to limit t he size of the flows that can be set up. Each data flow should not represent more than a small fraction of the service class capacity (in the order of 1%), to ensure that statistical multiplexing works well.

In Fig. 1 we can see the phases of the probing procedure. When a host wishes to set up a new flow, it starts sending a constant bit rate probe to the destination

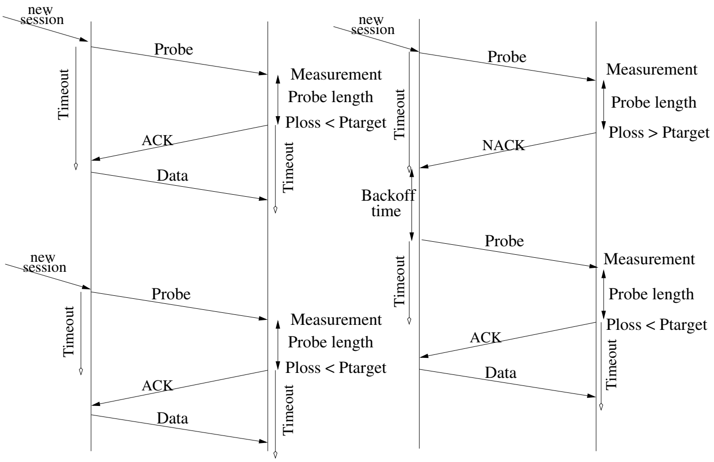


Fig. 1. The probing procedure.

host at the maximum rate the flow will require (r_{pr}). The probe packet size (l_{pr}) should be small to have a high number of packets in the probing period (t_{pr}) to perform the acceptance decision with a sufficient level of confidence. The probe packets contain information about the peak bit rate and length of th e probe, as well as a sequence number. With this information the receiving host can perform a quick rejection, based on the expected number of packets that it should receive in order not to surpass the target loss probability. The probe also needs to contain a flow identifier to allow the end host to distinguish probes for different flows, since one sender could transmit more than one flow simultaneously. The IP address in the probes would consequently not be enough to differentiate them.

To perform the acceptance decision, the end–host measures the empirical probe loss rate, P_{me} in the probe stream. Assuming a normal distribution of the probe loss [10], the flow is accepted, if:

$$P_{me} + z_R \sqrt{\frac{P_{me}\,(1 - P_{me})}{n_{pr}}} \leq Th, \text{ given that } Th\, n_{pr} > 10, \tag{1}$$

where Th is the acceptance threshold, n_{pr} is the number of probe packets sent, R is the required confidence level and z_R is the $1 - (1 - R/2)$–quantile of the normal distribution. This condition ensures a sufficient number of samples for the estimation.

3 Analytical Model

The approximate analytical model presented in this section determines the performance of the PBAC system depending on the main system parameters such

Table 1. Main parameters of the mathematical analysis.

K :	High priority queue size including the server (packets)
Th :	Acceptance packet loss threshold
t_{pr} :	Length of the probe (seconds)
r_{pr} :	Probe rate (bits/second)
l_{pr} :	Probe packet size (bits)
n_{pr} :	Number of probe packets per probing period
n_{min} :	Minimum number of successful probe packets for admission
P_{succ} :	Probability of successful transmission of one probe packet
P_a :	Acceptance probability
P_{loss} :	Data packet loss
U :	Link utilization

as the queue size for the high priority data packets (K), the acceptance packet loss threshold for new flows (Th), the length of the probe (t_{pr}), the probe rate (r_{pr}) and the probe packet size (l_{pr}). Specifically, we determine the probability of the successful transmission of one probe packet (P_{succ}), the acceptance probability of a new flow (P_a), the achievable network utilization (U) and the packet loss probability of accepted flows (P_{loss}). Table 1 summarizes the different parameters used in the analysis.

The analysis focuses on a single link, and on a single probing process, thus it does not consider repeating probes and possible probe thrashing effects [10,3]. Also, the effects of the probe packets on the data packets, due to the fact that we use a non–preemptive priority, is not considered, since the probe packets are small.

Each new flow probes the link with a constant bit rate train of probe packets during a time t_{pr} at the rate r_{pr} equal to the peak rate of the flow, and with probe packets of l_{pr} bits, which gives n_{pr} probe packets per probing period ($n_{pr} = r_{pr}\ t_{pr}/l_{pr}$). Assuming that the packet loss probability for consecutive probe packets is independent due to the small size of the flows, the probability that a new flow is accepted can be expressed as a function of the probability of successful probe packet transmission, as:

$$P_a = \sum_{i=n_{min}}^{n_{pr}} \binom{n_{pr}}{i} {P_{succ}}^{i}\ (1 - P_{succ})^{n_{pr}-i} \tag{2}$$

where n_{min} is the minimum number of probe packets that has to be transmitted successfully for a flow to be accepted, and can be expressed as:

$$n_{min} = n_{pr} - \lfloor P_{me}\ n_{pr} \rfloor$$

where P_{me} is the acceptable measured probe loss probability according to (1).

To calculate the probability of success of one probe packet, P_{succ} in (2), we consider the double–queue scheme with a low priority buffer of one packet for probe packets. The high priority buffer is considered to be infinite to simplify the analysis. We consider the high priority queue as an M/D/1 system, with the assumptions that the multiplexing level of accepted streams is high and packet sizes are constant, which seem to be reasonable assumptions for voice and video communication. The analysis follows the ideas presented in [2].

As the buffer for probe packets has one buffer position, an arriving probe packet is transmitted successfully, i.e. not dropped, if the previous probe packet has already left the buffer. For this, the high priority data queue had to be empty at the previous probe packet arrival, or the residual busy period of the queue f_{rb} has to be less than the probe packet inter-arrival time $T = 1/r_{pr}$. Thus,

$$P_{succ} = (1-\rho) + \rho\, F_{rb}(T) = (1-\rho) + \rho \int_0^T f_{rb}(t)dt, \tag{3}$$

where ρ is the utilization of the high priority queue.

Modeling the high priority queue as an M/D/1 system [4], we can obtain the cumulative probability function of the busy period $F_{bp}(t)$ and the probability density function of the remaining busy period $f_{rb}(t)$ [6] as:

$$F_{bp}(t) = \sum_{j=1}^{n} \frac{(j\rho)^{j-1}}{j!} e^{-j\rho} \text{ where } n = \left\lfloor \frac{t}{E[s]} \right\rfloor \text{ and} \tag{4}$$

$$f_{rb}(t) = \frac{1 - F_{bp}(t)}{E[bp]} \text{ with } E[bp] = \frac{1}{1-\rho}, \tag{5}$$

where $E[s]$ is the average packet service time and $E[bp]$ is the average length of the busy period.

Equations (4) and (5) show that the distribution function of the remaining busy period is a step-wise function, and thus:

$$\int_0^T f_{rb}(t)dt = \sum_{i=0}^{\lfloor T-1 \rfloor} f_{rb}(i) + f_{rb}(\lfloor T \rfloor)(T - \lfloor T \rfloor). \tag{6}$$

Finally, applying (6) to (3), we obtain the probability of success of one probe packet as a function of r_{pr} and ρ, and the probability of flow acceptance in (2) as a function of P_{succ}, P_{me} and n_{pr}.

In order to calculate the link utilization, we can model the number of accepted connections as a birth–death Markov chain, if we assume that we have a Poisson flow arrival process with an average of λ flows per second and the average flow holding time is $1/\mu$. Assuming that the acceptance decisions are independent, then the birth/death coefficients are:

$$\lambda_J = \lambda\, P_a(J) \text{ and } \mu_J = J\, \mu \tag{7}$$

where J represents the number of accepted connections, and $P_a(J)$ is the acceptance probability for an utilization given by the J accepted connections. The steady state probabilities π_J of the birth–death process can be obtained with the transition probabilities in (7), by using the acceptance probability obtained in (2). The link utilization, defined as the fraction of the link used by accepted flows, is thus obtained as:

$$U = \sum_J \frac{J\ \alpha\ r_{pr}}{C}\ \pi_J, \tag{8}$$

where C represents the link capacity and α the activity factor of the sources, i.e. the percentage of time they are in the 'on' state.

Finally, we want to obtain the loss probability of data packets of the accepted flows. As we are interested in the loss values in small buffers providing packet-scale buffering, we follow the corresponding burst-scale loss approximation in [13]. The queuing system operates in slotted time, serving one packet in one time unit, and can hold a maximum of K packets. Ongoing flows are modeled as n independent on–off streams with activity factor α. Within on periods the sources generate pack ets with fixed inter-arrival time D, that corresponds to the peak rate of the sources.

The packet loss probability is approximated by the sum of the packet scale and burst scale loss rates, multiplied by $D/(n\ \alpha)$ as:

$$P_{loss}(k, D, n) = \frac{D}{n\alpha}(R_{packet-scale}(K, D, n) + R_{burst-scale}(K, D, n)) \tag{9}$$

with

$$R_{packet-scale}(K, D, n) = \sum_{m=1}^{\min n, D} \binom{n}{m} \alpha^m (1-\alpha)^{n-m} {Q_D}^m(K) \tag{10}$$

and

$$R_{burst-scale}(K, D, n) = \sum_{m=\min n, D+1}^{n} \binom{n}{m} \alpha^m (1-\alpha)^{n-m} \frac{m-D}{D}, \tag{11}$$

where ${Q_D}^m(K)$ is the exact solution for the tail probability of an $m * D/D/1$ queue (see [13]).

4 Numerical Results

To evaluate the performance of the system we first consider the probability of probe packet loss, the probability of flow acceptance and the network utilization at a given level of offered load, followed by the evaluation of maximum link utilization, data loss probability at a given link utilization and the relation of probe and data packet loss probabilities. The results are based on the analysis presented in Sec. 3. To validate our analytical model we performed simulations with NS–2.

For all scenarios considered, the sources have exponentially distributed on–off holding times with an average of 20 and 35.5 ms. Packet sizes were 64 bytes for the probe packets and 128 bytes for the data packets, while the probe length

Table 2. Link utilization for acceptance probabilities of 5 and 95%.

	r_{pr}=100 kb/s			r_{pr}=1 Mb/s		
P_a	Model	Simulation K=1000	Simulation K=10	Model	Simulation K=1000	Simulation K=10
0.95	0.769	0.732	0.735	0.789	0.753	0.759
0.5	0.825	0.832	0.837	0.808	0.821	0.827

was always 2 seconds. We used a confidence interval for the admission decision of 95%. The simulation time was 2000 seconds and the average holding time for accepted flows $(1/\mu)$ was 50 seconds. Confidence intervals for each simulation are too small to be plotted in the figures. The flow arrival rate (λ) varied in each simulation to increase the offered load to the system.

We considered sources with 100 kb/s and 1 Mb/s peak rates (r_{pr}), multiplexed over 10 Mb/s and 100 Mb/s links respectively. The double–queue system had a low priority buffer of one packet and a high priority buffer of ten packets in the finite buffer case. To validate the analytical models that assumes infinite buffer, simulations with 1000 high priority packet buffers were performed.

Figure 2 shows the probe packet loss probability as a function of network load. The figure compares the analytical results with the assumption of an infinite high-priority buffer to simulation results with 1000 and 10 buffer positions and for the two different peak rate values. Note that the analytical results do not depend on the actual peak rates but only on the ratio of the peak rate over link rate. The figure shows a close matching between the analysis and the simulation. The analys is with the assumption of infinite high-priority buffer gives an upper bound for losses in the finite buffer case. This is due to the fact that more data packets are lost in the high priority queue when we have a buffer size of ten packets, thus reducing slightly the average remaining busy period, which in turn increases the probability of success of one probe packet (see Eq. 3). The different curves suggest that the probe loss increases exponentially as the link utilization increases, which ca n then offer a sharp transition in the acceptance probability.

The flow acceptance probability as a function of the link utilization is plotted in Figs. 3 and 4 for an acceptance threshold of 10^{-2} and for the two considered flow peak rates. In both cases the analytical curve intersects the curves obtained by simulation, due to the assumption of independent probe loss in the analysis. The values of the curves with a finite buffer size of K = 10 are always slightly over the infinite case, for the reason explained in the previous parag raph. The transient period is shorter in the case of high peak rates, as the number of probe packets transmitted is higher and the loss estimation more accurate. Table 2 summarizes the relationship for the three curves for each of the two peak rates. From the values in the table, it can be seen that the model gives an upper bound on the utilization for an acceptance probability of 95% which is approximately 3% higher than the simulation results, while it offers a lower bound for the 5% case with a difference of less than 2% of link utilization.

Figure 5 shows the link utilization achieved as a function of the offered load, for an admission threshold of 10^{-2}. This figure illustrates that the admission control scheme leads to a stable system. The utilization follows the offered load up to a load level of 0.75. After this point, the mathematical analysis slightly overestimates the utilization of the simulated system, due to the sharper change of the flow admission probability. Towards high loads the analysis might underestimate the achievable utilization due to the same reason, with up to 2% of link capacity.

Data–packet loss probability values are shown in Fig. 6 as a function of link utilization for different high priority buffer sizes, considering streams with peak rate of 1Mb/s. The results show that the data packet loss probabilities grow exponentially with increasing link utilization. The mathematical model provides accurate results for the small buffer sizes of interest, and, as expected, does not give correct result for large buffers, since it does not consider burst-scale buffering.

Finally, based on the previous results we evaluate the connection between probe and data loss values in Fig. 7. The figure shows the data packet loss as a function of the probe loss experienced in the probing phase for ongoing flows. The probe loss is always higher than the data loss independently of the utilization level. The actual relationship between probe and data packet loss is in this case over half an order of magnitude.

As stated before the whole admission procedure of PBAC relies on a high degree of multiplexing [5,10], in order to obtain a smooth statistical behavior of the ongoing traffic. For lower levels of multiplexing, the effects of thrashing on the simulation results should be taken into account, thus reducing the acceptance probability and the link utilization (see [10]), as well as the reliability of the measurement of the accepted traffic. Other traffic sources experience the same type of behavior regarding the probe/data packets loss relationship, as shown in [10]. For multiplexing levels below 1%, the various sources used to model real–time communications (i.e. Poisson or exponentially distributed on–off sources, or even real traffic traces) show the same behavior, so we are expecting our model to work well for all these sources. The effect of the acceptance threshold on the model is high, since lower acceptance thresholds require longer probes to be performed, in order to achieve enough accuracy (see (1)).

5 Conclusions

In this paper we have presented an approximate analytical model of the probe based admission control scheme based on the end–to–end measurements of packet loss probabilities in the probe streams. In this solution the admission control is responsible for limiting the end–to–end packet loss probability of accepted flows, while the end–to–end delay and delay jitter requirements are ensured by the use of small buffers in the routers.

Consequently, the analysis focuses on the packet loss probabilities of probe streams and accepted flows, on the relation between these two parameters, and on the flow acceptance probabilities at a given link utilization. The analytical results, verified by simulations, prove that the considered probe–based admission

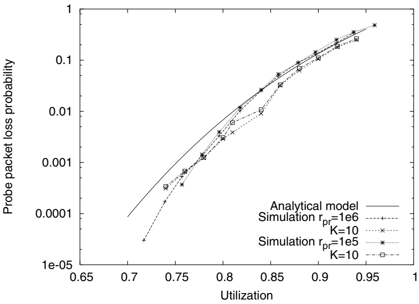


Fig. 2. Probe packet loss probability.

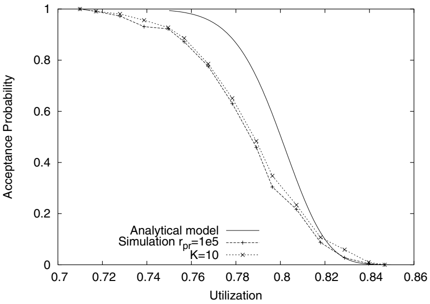


Fig. 3. Acceptance probability for a new flow as a function of the load on the system for 100 kb/s flow peak rate, with an acceptance threshold of 10^{-2}.

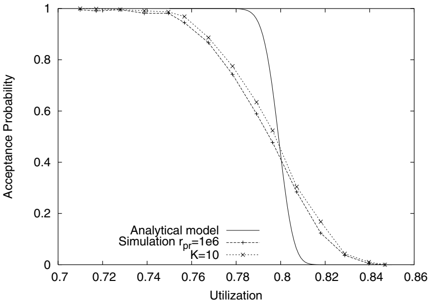


Fig. 4. Acceptance probability for a new flow as a function of the load on the system for 1 Mb/s flow peak rate, with an acceptance threshold of 10^{-2}.

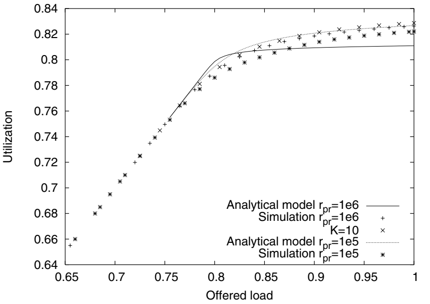


Fig. 5. Accepted versus offered load for a 100 Mb/s link with acceptance threshold of 10^{-2}.

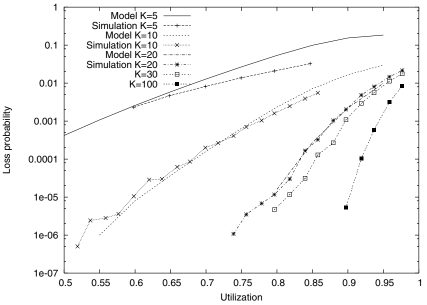


Fig. 6. Packet loss probability for accepted flows, for different high priority queue buffer sizes.

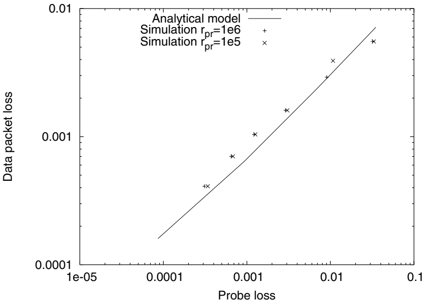


Fig. 7. Probe loss versus data loss for a peak rate $r_{pr} = 1Mb/s$, $Th = 0.01$.

control leads to a stable link utilization which has a clear upper bound on the packet loss probability.

As the acceptance decision is based on the probe loss probability it is important to see how probe loss and data loss probabilities relates at different link utilization. The analysis proved that the probe loss always overestimates the data loss with over half an order of magnitude, independently of the actual link load.

Consequently, the probe–based admission control based on measurements of the packet loss probabilities in the probe stream provides a reliable and efficient solution for QoS provisioning for delay and loss sensitive applications, without extensive support in the routers.

References

1. Giuseppe Bianchi, Flaminio Borgonovo, Antonio Capone, and Chiara Petrioli. Endpoint admission control with delay variation measurements for QoS in IP networks. *ACM Computer Communication Review*, 32(2):61–69, April 2002.
2. Giuseppe Bianchi, Antonio Capone, and Chiara Petrioli. Packet management techniques for measurement based end-to-end admission control in IP networks. *Journal of Communications and Networks*, 2(2):147–156, June 2000.
3. Lee Breslau, Edward W. Knightly, Scott Shenker, Ion Stoica, and Hui Zhang. Endpoint admission control: Architectural issues and performance. In *Computer Communication Review – Proc. of Sigcomm 2000*, volume 30, pages 57–69, Stockholm, Sweden, August/September 2000. ACM.
4. J. Cao and K. Ramanan. A poisson limit for buffer overflow probabilities. In *Proc. of Infocom 2002*, New York, New York, June 2002. IEEE.
5. Viktoria Fodor (née Elek), Gunnar Karlsson, and Robert Rönngren. Admission control based on end–to–end measurements. In *Proc. of the 19th Infocom*, pages 623–630, Tel Aviv, Israel, March 2000. IEEE.
6. R. Jain. *The Art of Computer Systems Performance Analysis.* Wiley Professional Computing. John Wiley & Sons, Inc, 1991.
7. Gunnar Karlsson. Providing quality for internet video services. In *Proc. of CNIT/IEEE ITWoDC 98*, pages 133–146, Ischia, Italy, September 1998.
8. Frank P. Kelly, Peter B. Key, and Stan Zachary. Distributed admission control. *IEEE Journal on Selected Areas in Communications*, 18(12):2617–2628, 2000.
9. Tom Kelly. An ECN probe–based connection acceptance control. *ACM Computer Communication Review*, 31(3):14–25, July 2001.
10. Ignacio Más Ivars and Gunnar Karlsson. PBAC: Probe–based admission control. In *Proc. of QofIS 2001*, volume 2156 of *LNCS*, pages 97–109, Coimbra, Portugal, September 2001. Springer.
11. Ignacio Más, Viktoria Fodor, and Gunnar Karlsson. Probe–based admission control for multicast. In *Proc. of the 10th IWQoS*, pages 99–105, Miami Beach, Florida, May 2002. IEEE.
12. J. W. Roberts, U. Mocci, and J. Virtamo, editors. *Broadband Network Teletraffic – Final Report of Action COST 242*, volume 1155 of *LNCS*. Springer, 1996.
13. J. W. Roberts, editor. *COST 224: Performance evaluation and design of multiservice networks*, volume EUR 14152 EN of *Information technologies and sciences.* Commission of the European Communities, 1992.

TFRC Contribution to Internet QoS Improvement

Nicolas Larrieu and Philippe Owezarski

LAAS-CNRS
7, avenue du Colonel ROCHE
31077 Toulouse Cedex 4, France
{nlarrieu,owe}@laas.fr

Abstract. The Internet is on the way of becoming the universal communication network, and then needs to provide various services and QoS for all kinds of applications. We show in this paper that oscillations that are characteristic of the Internet traffic provokes huge decrease of the QoS that flows can get. After having demonstrated that such oscillations can be characterized by the Hurst (LRD) parameter, we propose an approach for improving Internet flows QoS based on smoothing sending rate of applications. TFRC is a congestion control mechanism that has been issued for this purpose. This paper then proposes an evaluation of TFRC benefits on traffic profile and flows QoS.

Keywords: Internet monitoring, traffic characterization, quality of service, TFRC, congestion control for elephants

1 Introduction

The Internet is on the way of becoming the universal communication network for all kinds of information, from the simple transfer of binary computer data to the transmission of voice, video, or interactive information in real time. It has then to integrate new services suited to new applications. In addition, the Internet is rapidly growing, in size (number of computers connected, number of users, etc.), and in complexity, in particular because of the need of new advanced services, and the necessity to opti mize the use of communication resources to improve the QoS[1] provided to users. In fact, the Internet has to evolve from a single best effort service to a multi-services network.

Since at least a decade, Internet QoS is then, one of the major issues in the Internet. Many proposals have appeared as IntServ, DiffServ, etc., but until now, they have not been deployed (or their deployment has been quite limited). Indeed, Internet community contributions to propose differentiated and guaranteed services did not provide the solutions users and operators (Internet service roviders, carriers, etc.) are expecting. There are always difficulties with the complexity of the Internet and all its network interconnections, with their resource

[1] QoS: Quality of Service.

G. Karlsson and M.I. Smirnov (Eds.): QofIS 2003, LNCS 2811, pp. 73–82, 2003.

heterogeneity in terms of technologies but also in terms of provisioning, and of course with the traffic characteristics. Indeed, because of the growing complexity of the Internet, all new applications with various and changing requirements, introduce in Internet traffic many characteristics that are very far from common beliefs. In fact, models with simple static metrics such as throughput, delay, or loss rate are really not sufficient to model completely and precisely Internet traffic dynamics that are its essential features. The evolution of the Internet is then strongly related to a good knowledge and understanding of traffic characteristics that will indicate the kind of mechanisms to deploy. Consequently, the development of monitoring-based tools and technologies to collect Internet traffics information, and methodologies to analyze their characteristics is currently an important topic for network engineering and research. In particular, the definition and quantification of Internet QoS is still not completely solved. First monitoring results showed that Internet traffic is very far from Poisson or Markovian models, used in telephony, and also reused as the model for Internet traffic as well. These first results showed that models that better represent Internet traffic are models with self-similarity or LRD[2] characteristics.

Given this previous work on traffic monitoring, our work showed also that Internet traffic has very significant oscillatory behaviors, whose peaks are responsible of some instability issues of the Internet QoS, as well as a serious decrease of Internet performances. This is especially true for big flows transporting a huge quantity of data (called "elephants"). That is why section 2 exposes the analysis results on some Internet links traffic characteristics and shows how oscillating phenomena can have such a bad impact on network QoS and performances. This analysis also indicates that TCP congestion control mechanism is largely responsible of such oscillations, what makes us propose some improvements for the Internet. More precisely, section 3 proposes to use a smoother transport protocol, at least for elephants, to separately smooth the flow behaviors (with a less aggressive congestion control mechanism), and explains how this individual optimization for each flow can bring important improvements for the whole network QoS. Some experiments assessing this approach, are presented in section 4. These experiments have been performed with the NS-2 [1] simulator, and allow the evaluation of the TFRC[3] congestion control mechanism. It is shown in this section that TFRC can optimize Internet QoS by smoothing its traffic. Finaly, section 5 concludes this paper.

Note however that this work is achieved in the framework of the METROPOLIS project, a French national project granted and funded by the Frenck Network for Research in Telecommunications. METROPOLIS main goal deals with issuing new network monitoring and analysis methodologies.

2 Traffic Oscillation Issues and Elephant Flows

Current Internet links monitoring results show the presence of very high oscillations in Internet traffic. An example of an Internet link traffic is given on

[2] LRD: Long Range Dependence.

[3] TFRC: TCP-Friendly Rate Control.

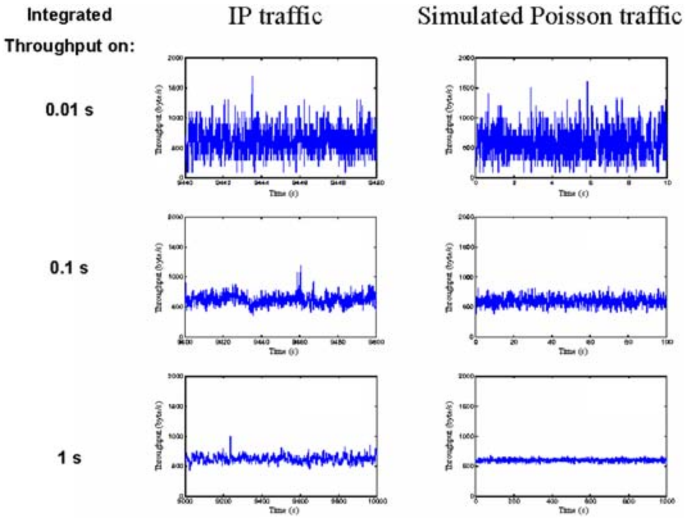


Fig. 1. Comparison between oscillations of Internet and Poisson traffics.

Figure 1. This figure also compares current Internet traffic with a simple model of traffic: the Poisson model that is the model that was supposed to be the one of the Internet several years ago. In fact, traffic curves have to be smoother when the granularity of observation increases. This is what is represented in Figure 1 where for each traffic (actual Internet and simulated Poisson traffic) the amplitude of oscillations is decreasing when the observation granularity is coarser. What also appears on this figure is the difference between the two traffics: with coarse grain analysis, the oscillations amplitude of Internet traffic is much larger than Poisson traffic ones.

Some analysis of Internet traffic performed in recent Internet monitoring projects showed that these oscillations are in fact the results of the presence of LRD and self-similarity in the traffic [2]. These phenomena are due to several causes and in particular to congestion control mechanisms, especially the ones of TCP that is the dominant protocol in the Internet [3]. Among these mechanisms, it is clear that the closed control loop of TCP introduces short scale dependence as the acknowledgment depends on the reception of one packet, and all the following packets of the flow depend on this acknowledgement. In the same way, the two TCP mechanisms – slow start and congestion avoidance – are responsible of introducing dependences between packets of different congestion control windows. And of course, this notion of burstiness in TCP sources plus the LRD explain oscillations in the global traffic. By extending this process, all packets of a flow are dependent from each other. As the increase of capacities in the Internet allows users to transmit larger and larger files (i.e. elephant flows[4]), as music or movies for instance, it is clear that the scale of LRD is increasing,

[4] In this paper, we define an elephant as a flow that contains more than 100 packets exchanged in the same mono-directional connection.

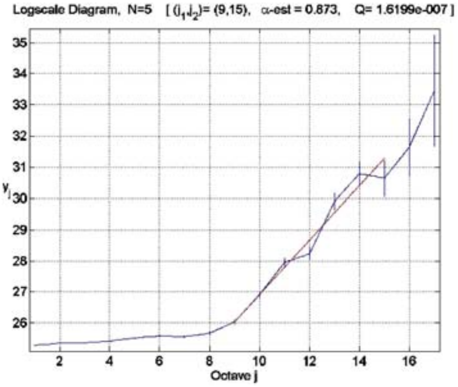


Fig. 2. LRD evaluation for edge network traffic.

explaining why oscillations of Internet traffic, even with a coarse observation granularity, are so high. Of course, oscillations are very damaging for the global use of network resources as the capacity freed by a flow after a loss for example cannot be immediately used by other flows: this corresponds to some resource waste, and of course a decrease of the global QoS of the traffic and network : the higher the oscillations amplitude, the lower the global network performance [4].

It is also clear that elephants introduce oscillations with higher amplitudes than mice (short flows). Indeed, elephants, because of their long life in the network, have time to reach large values of the congestion control window, and thus, any loss event can provoke a huge reduction, followed by a huge increase of the sending rate. This phenomenon is even more important in current Internet compared to what happened few years ago. Few years ago, Internet traffic consisted almost exclusively of web traffic with very short flows. Nowadays, because of the arrival of peer-to-peer applications used most of the time for huge files exchanges (as audio tracks or movies), Internet traffic consists of both web and Peer-to-peer traffic, meaning that there are more and more elephants and that elephants are getting larger and larger (essentially thanks to new high capacity Internet access technologies: ADSL, cable modem, etc.). Our past and current network monitoring results shows that elephants are now reaching more than 5 % of the number of flows in the Internet (it was 2 or 3 % few years ago), and that this 5 % of elephants represent around 60 % of the full Internet traffic.

It is then clear that elephants and the huge oscillations they induce, directly impact traffic profile and also global network performances. Figure 2 represents the LRD evaluation of the network traffic depicted in Figure 1. This figure has been produced using the LDestimate tool designed by Abry and Veitch [5] [6] that estimates the LRD that appears in Internet traffic at all scales. The principle of this tool relies on a fractal decomposition of traffic time series, what then allows users to have a graphical representation of the dependence laws at all time scales. Then small value octaves represent short range dependence, while large value ones represent long range dependence (LRD). In figure 2, we can note a "bi-scaling" phenomenon (cf. the elbow in Figure 2 around octave 8) which shows a difference in the LRD level between short and long time scales for the traffic exchanged. For short scale (octave < 8), representing the dependence between close packets (i.e. packets whose sending time are not very far from each other), the dependence is quite limited. Such dependence is the one that can exist for packets belonging to the same congestion window and that are then very close from each other. On the other side, for long time scales (octave > 8) LRD can be very high. For octaves 8 to 12, that correspond for instance to the dependence between packets of consecutive congestion windows, the dependence is higher. This can be explained by the closed loop structure of TCP congestion control mechanism in which the sending of one packet of a congestion control window depends on the receiving of the acknowledgement of one packet of the previous congestion control window. Of course, this phenomenon exists for consecutive congestion window, but also for all congestion windows of the same flow. This means, that the presence in the traffic of very long flows introduces very long scale dependence phenomenon, as depicted on figure 2 for very large octaves. The consequence of such LRD is one major issue as every oscillation at time t will be repeated at any other time t' that is dependent from t (because of the long range dependence between packets due to protocols – here TCP on long flows). That is why, if we want to both improve traffic profile and QoS, it is mandatory to decrease both LRD and oscillation levels for elephants. A solution for this is proposed in next section.

3 A New Approach for Improving Internet QoS

3.1 Increasing QoS by Smoothing Flow Behaviors

It is clear now that network traffic has complex and high oscillating features. Indeed, it clearly appears the presence of scale laws in the traffic that induce the repetition of an oscillating phenomenon. This is especially visible on Figure 1. From this observation, it appears that the most urgent problem to address deals with reducing oscillations and more precisely with regulating the long term oscillations having such a damaging effect on traffic QoS and performance. Therefore, the main objective is then to bring more stability to elephants flows.

Such an approach is quite different from what can be proposed by classical service differentiation techniques. In general, classification of application flows depends on the QoS level they require, meaning that a web browser and a video streaming application are not in the same class: in this case the web browser is generally assigned a best effort service, while the streaming application get the best existing service (EF, gold or whatever name). This is an application oriented service selection, but that has the disadvantage of not taking into account network requirements. Our approach is basically based on a network centric point of view, and the classification proposed is based on the disturbance that flows induce on the traffic. Based on monitoring results, it appeared that elephants are the ones that introduce the more disturbances. Applications, as videoconferences, video on demand, telephony on IP, etc., that are typical applications requiring high quality services with classical service differentiation approaches, are also typical application generating long flows. In addition, such applications also require smooth services for smooth traffic. Our approach then perfectly fits the requirements of such applications. Of course, with our approach we are also going to smooth FTP or peer-to-peer long flows, that do not have the same requirements. Nevertheless, our approach introduces a big difference with application classes oriented approaches, as here, long and smooth flows that introduce few disturbances in the network are considered as low quality. This means that stream oriented applications are the applications introducing the less disturbances, and are the ones that should pay less. On the other side, applications that have sending rates oscillating a lot and that cannot be shaped or smoothed (as interactive video applications using MPEG[5]), are the ones that are considered as the most constraining and they will be charged more as the most disturbing. Note however that FTP or web traffic that is nowadays sent using the usual best effort service can easily use a smooth service thanks to its elastic nature. In both approaches, elastic traffic is the one that is the more flexible and then the one that is the easier to handle and then the cheaper.

To increase elephant flows regularity (i.e. to suppress observable oscillating behaviors at all scales), the new TFRC congestion control mechanism seems to be able to provide a great contribution. TFRC has been designed to provide a service suited for stream oriented applications requiring smooth throughputs. TFRC, then, tries as much as possible to avoid brutal throughput variations that occur with TCP because of loss recovery. Note however that for both TFRC and TCP, we will estimate the evolution of the oscillating behavior of the traffic by evaluating LRD features (also called the Hurst factor: H) on packet arrival series.

3.2 TFRC Principles

TFRC aims to propose to applications a smooth sending rate with very soft increases and decreases; at least much softer than the ones of TCP. By associating such a congestion control mechanism to elephants, i.e. to the main part of the traffic, we expect to be able to control traffic oscillations, and then to increase global QoS and performance of the network. The sending rate of each TFRC source is made thanks to a receiver oriented computation, that calculates, once by RTT[6], the sending rate according to the loss event rate measured by the receiver [8] [9] according to equation 1:

$$X = \frac{s}{R * \sqrt{2 * b * \frac{p}{3}} + (t_{RTO} * (3 * \sqrt{3 * b * \frac{p}{8}}) * p * (1 + 32 * p^2)))} \tag{1}$$

where:

[5] Because of the dependence induced between frames with this coding (P frames depends on previous frames and B frames on previous and next frames [7]).

[6] RTT: Round Trip Time.

- X is the transmit rate in byte/second,
- s is the packet size in byte,
- R is the round trip time in second,
- p is the loss event rate (between 0 and 1.0), of the number of loss events as a fraction of the number of packets transmitted,
- t_{RTO} is the TCP retransmission timeout value in second,
- b is the number of packets acknowledged by a single TCP acknowledgement.

In TFRC, a loss event is considered if at least one loss appears in a RTT. This means that several losses appearing in the same RTT are considered as a single loss event. Doing so, the loss dependence model of the Internet is broken since most dependent losses are grouped in a same loss event. Thus, the recovery will be eased and more efficient compare to what TCP can do: it is well known that TCP is not very efficient to recover from several losses in sequence. This approach follows the results of [10] that proposes an analysis and a model for the Internet loss process.

4 Evaluation of TFRC Impact on QoS

4.1 Experiment Description

Our experiment aims to provide a comparative evaluation of the global traffic characteristics if elephants use TCP or TFRC as the transmission protocol. This experiment aims to provide values in a realistic environment. For that, of course, the experiment relies on the use of traffic traces grabbed thanks to passive monitoring tools as the DAG [11] equipments. Therefore, traffic flows identified in the original traffic trace are replayed in NS-2 with the same relative starting date and the same others characteristics. Elephant flows are transmitted in the simulator using TFRC while others flows use TCP New Reno[7]. Then in the remainder, the comparative study will focus on the original trace and the simulated one where elephants are generated using TFRC.

In addition of the classical traffic throughput parameter, this study focuses on QoS statistical parameters as the LRD (as justified in section 2) and some parameters related to variability. For that, we used the Stability Coefficient (SC), that is define as the following ratio:

$$\text{Stability Coefficient (SC)} = \frac{\text{exchanged average traffic}}{\text{exchanged traffic standard-deviation } (\sigma)} \qquad (2)$$

[7] TCP New Reno has been selected as it is currently the most used version of TCP in the Internet. To increase again the realism of simulations, it would be interesting to replay short flows with the same TCP version than the one that was used in the original trace, but finding out such information is impossible for most of short flows: only the on es that experiment a huge number of losses can provide enough information to find out the TCP version that was used.

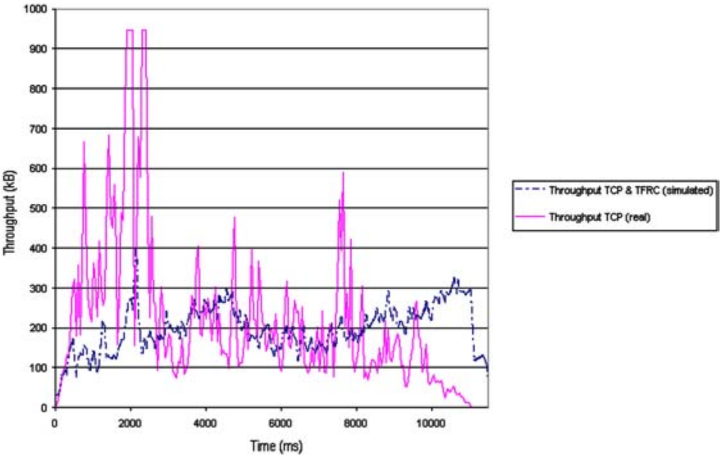


Fig. 3. Throughput evolution during time.

4.2 TFRC Impact on Flow QoS

Figure 3 presents the traffic in both cases, i.e. in the real and simulated cases. It visually clearly appears that using TFRC for sending elephants, instead of TCP, makes global traffic much smoother, avoiding all the huge peaks that can be seen on the real traffic.

Quantitatively speaking, results are indicated in table 1. This confirms that the traffic variability in the case of real traffic (using TCP for transmitting elephants) is much more important compared to the simulated case in which elephants are transmitted using TFRC (for the standard deviation σ it has been calculated that σ(real traffic) = 157.959 ko $\ggg$ σ(simulated traffic) = 102.176 ko). In the same way the stability coefficient is less important in the real case (SC = 0.521) than in the simulated one (SC = 0.761).

Dealing with the global throughput we got for both real and simulated traffic rather equal values (Throughput(real traffic) = 82.335 ko $\approxeq$ Throughput(simulated traffic) = 77.707 ko). This result is quite good as TFRC is not able to consume as many resources as TCP [12], and even if TFRC is less aggressive than TCP, it is able to reach the same performance level as TCP. This confirms the importance of stability for good performances [4].

Speaking about LRD in the simulated case, figure 4 shows that the bi-scaling property of the curve is strongly decreased, and that the curve has a very small slope. This means that all kinds of dependences, especially the long term ones have been drastically reduced. The values for the LRD (Hurst factor are: (H(real traffic) = 0.641 and H(Simulated traffic) = 0.194). Such result confirms two aspects of our proposal:

- TFRC helps to smooth individual flow traffic (thus providing a smoother QoS better suited for stream oriented applications) as well as the global traffic of the link;

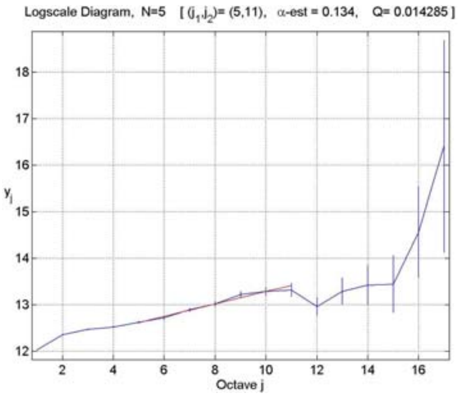


Fig. 4. LRD evaluation for simulated traffic including TFRC elephants.

Table 1. Throughput evolution during time for TCP and TFRC protocols.

Protocol	Average throughput (kB)	Throughput σ (kB)	SC
TCP New Reno (NR): real case	82.335	157.959	0.521
TCP NR & TFRC: simulated case	77.707	102.176	0.761

- LRD is the right parameter to qualify and quantify all scaling laws and dependencies between oscillations.

5 Conclusion

In this paper, we proposed a new approach for improving flow QoS. This approach relies on a preliminary study of Internet traffic characteristics that has been made possible thanks to some passive monitoring tools. This traffic characterization showed that Internet traffic suffers from the number and the amplitude of oscillations, especially important in the case of long flows, called elephants. The first contribution of this paper was then to explain why such oscillations arise, and proposes to use the LRD metric to characterize such feature in addition to the stability coefficient and other well known statistic moments as standard deviation. Therefore, the solution proposed in this paper consists in smoothing the traffic generated by each flow, especially elephants. The main protocol designed for this purpose (and under discussion at the IETF) is TFRC. This paper then proposed a comparative evaluation of real traffic, and the same traffic but this time with elephants running TFRC instead of TCP. The results we got confirmed all our starting hypothesis in relation with oscillations, the LRD metric to characterize them, and the impact of TFRC for their reduction and for getting a smoother traffic, much more easy to handle.

However, it also appears that the global throughput that can be transmitted using TFRC instead of TCP is not higher. This is due, in fact, because TFRC is a less aggressive congestion control mechanism than the one used in TCP. The problem with congestion control is really tricky: on one side, transport protocols and their congestion control mechanisms have to be very aggressive to be able to rapidly consume network resources and being able to exploit the capacity of new networks, and on the other side, protocols have to be not aggressive to limit the oscillation phenomena that are very damaging for flow QoS. These two requirements are c ontradictory, but this is the challenge to enforce for next generation transport protocols, i.e. being able to rapidly consume resources without provoking damaging oscillations.

References

1. NS-2 web site: *http://www.isi.edu/nsnam/ns/.*
2. K. Park and W. Willinger, *"Self-similar network traffic: an overview"*, In Self-similar network traffic and performance evaluation, J.Wiley & Sons, 2000
3. K. Park, G. Kim and M. Crovella, *"On the relationship between file sizes, transport protocols, and self-similar network traffic"*, IEEE ICNP, 1996.
4. K. Park, G. Kim and M. Crovella, *"On the Effect of Traffic Self-similarity on Network Performance"*, SPIE International Conference on Performance and Control of Network Systems, November, 1997.
5. Abry P. and Veitch D., *"Wavelet Analysis of Long Range Dependent Traffic"*, Trans. Info. Theory, Vol.44, No.1 pp.2-15, Jan 1998.
6. Abry P., Veitch V. and Flandrin P., *"Long-Range Dependence: Revisiting Aggregation with Wavelets"*, Journal of Time Series Anal., Vol.19, No.3 pp.253- 266 May 1998.
7. LeGall D., Mitchell J.L., Pennbaker W.B. and Fogg C.E., *"MPEG video compression standard"*, Chapman & Hall, New York, USA, 1996.
8. Floyd S. and Fall K., *"Promoting the use of end-to-end congestion control in the Internet"*, In Proc. IEEE ACM Transactions on Networking, 14 pages, February 1998.
9. Floyd S., Handley Mr., Padhye J. and Widmer J., *"Equation-based congestion control for unicast applications"*, In Proc. ACM SIGCOMM, 14 pages, 2000.
10. Y. Zhang, N. Duffield, V. Paxson, and S. Shenker, *"On the Constancy of Internet Path Properties"*, Proc. ACM SIGCOMM Internet Measurement Workshop (IMW'2001), San Francisco, California, USA, November 2001.
11. S. Donnelly, I. Graham, R. Wilhelm, *"Passive calibration of an active measurement system"*, in Passive and active measurements workshop, Amsterdam, April 2001.
12. P. Owezarski and N. Larrieu, *"Coherent Charging of Differentiated Services in the Internet Depending on Congestion Control Aggressiveness"*, to be published in Computer Communications, special issue on "Internet Pricing and Charging: Algorithms, Technology and Applications", 2003.

Adaptive Bandwidth Provisioning with Explicit Respect to QoS Requirements

Hung Tuan Tran and Thomas Ziegler

Telecommunications Research Center Vienna (ftw.)
Donaucity Strasse 1, 1220, Vienna, Austria
{tran,ziegler}@ftw.at
Phone: +43 1 505 28 30/50, Fax +43 1 505 28 30/99

Abstract. We propose adaptive bandwidth provisioning schemes enabling quality of service (QoS) guarantees. To this end, we exploit periodic measurements and traffic predictions to capture closely traffic dynamics. We make use of the Gaussian traffic model providing available bounds for QoS to derive the associated bandwidth demands. Moreover, special attention is paid for alleviating some typical problems with adaptive provisioning like QoS degradations and signaling overhead. Numerical and simulative investigations using real traffic traces show that the proposed schemes outperform some previous ones.

Keywords: adaptive provisioning, QoS, Gaussian process

1 Introduction

Extensive research has been focused on finding a solid architectural solution for providing QoS over IP networks. Potential alternatives are for example the service differentiation concept [1] eventually coupled with MPLS (Multiple Protocol Label Switched) technology [2] for traffic engineering, or the user-network interaction based concept [3,4] relying on the existing operational features of the current Internet such as active queue management, congestion notification and QoS-aware utilization functions.

In this paper we follow an alternative concept, namely bandwidth provisioning without service differentiation for providing QoS guarantees. The issue we address is how to adjust adaptively the bandwidth of a given link to meet the given requirements on objective QoS parameters like loss and delay probabilities. The main tasks of adaptive provisioning are capturing traffic dynamics to predict the future traffic volume and based on this, deciding the bandwidth to provision. While several traffic predictors have been developed in the literature, most of the work simply uses the link utilization as a basic factor for provisioning without specifying the role of the utilization level on QoS perception. This gives us a motivation to elaborate novel provisioning schemes with explicit respect to statistical QoS requirements. The proposed schemes are tested with real traffic traces and exhibit superiority over some existing schemes. Concerning the applicability of the proposed provisioning schemes, we mention some potential examples, namely *i)* adaptive resizing of high-speed LSPs (Label Switched

G. Karlsson and M.I. Smirnov (Eds.): QofIS 2003, LNCS 2811, pp. 83–92, 2003.

Paths) in MPLS networks; *ii)* adaptive resizing of customer-pipes in VPNs (Virtual Private Networks); and *iii)* adaptive bandwidth allocation of logical links in the Service Overlay Networks architecture [5] for providing end-to-end QoS over inter domains.

The paper is organised as follows. In Section 2 we present the conceptual descriptions of our provisioning schemes. Afterwards, in Section 3 we assess the performability of the proposed schemes. Based on experimental results, we further enhance our schemes by developing specific traffic prediction rules. We demonstrate the achievable gain concerning signaling overhead and QoS achievement. Finally, Section 4 concludes the paper.

2 Conceptual Proposals for Adaptive Provisioning

2.1 Model for the Aggregate Traffic

In this work, we adopt the Gaussian process as a traffic model for the aggregate traffic with high degree of multiplexing. Given a single, discrete-time queue with the aggregate input rate λ_n and service rate c at time n, define a stochastic process X_n as $X_n = \sum_{k=1}^{n} \lambda_k - cn$. For a buffer size x, define the normalized variance $\sigma^2_{x,n}$ of X_n as $\sigma^2_{x,n} := \frac{Var\{X_n\}}{(x-E\{X_n\})^2}$, and let μ_x be the reciprocal of the maximum of $\sigma^2_{x,n}$, i.e. $\mu_x := \frac{1}{\max_{n\geq 1}\sigma^2_{x,n}}$. We note that if the aggregate input rate λ_n is a Gaussian process, so is X_n. Using the Gaussian property of X_n and the so called *dominant time scale* approach, the so called MVA (Maximum Variance Asymptotic) bound on the tail probability of the queue length $P(Q > x))$ is given as $e^{-\mu_x/2}$ [6]. The loss estimation (i.e. the loss probability $P_L(x)$ for buffer size x) is given as $\gamma e^{-\mu_x/2}$ [7]. Here, the term γ is calculated as $\gamma = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{(c-\bar{\lambda})^2}{2\sigma^2}} \int_c^\infty (r-c) e^{\frac{(r-\bar{\lambda})^2}{2\sigma^2}} dr$, where $\bar{\lambda} = \mathbb{E}\{\lambda_n\}$, $\sigma^2 = Var\{\lambda_n\} = C_\lambda(0)$ and $C_\lambda(l)$ is the autocovariance functions of λ_n.

The mean and variance of X_n can be computed from the mean and the autocovariance function of the input rate as

$$\mathbb{E}\{X_n\} = n(c - \bar{\lambda}) \tag{1}$$

$$Var\{X_n\} = nC_\lambda(0) + 2\sum_{l=1}^{n-1}(n-l)C_\lambda(l) \tag{2}$$

The presented MVA and loss bounds can be used to make the mapping between the bandwidth to provision and QoS requirements explicit. In the remainder of this paper we will present MVA bound based (or equivalently, delay-based) provisioning schemes. However, the same concept is straightforwardly applicable to the loss or loss-delay combination based provisioning as well.

2.2 Provisioning Schemes Combining Use of the Gaussian Model, Periodical Measurements, and Traffic Predictions

The target QoS we have to keep is the packet level constraint $P(delay > D) < \epsilon$, where D and ϵ are the given delay bound and violation probability, respectively.

```
INPUT: QoS requirement: $P(delay > D) < \epsilon$
       Traffic description: $m$, $Var\{X_n\}$
       Bandwidth amount of the current resizing window: $l$
BANDWIDTH ASSIGNMENT
       $\epsilon_{MVA} := MVA\_bound\,(m, Var\{X_n\}, D, l)$
       IF ($\epsilon_{MVA} < \epsilon$) THEN
          $l_{upper} := l$
          $l_{lower} := m$
          WHILE ($|\log \epsilon_{MVA} - \log \epsilon| > \epsilon^*$)
                  $l := \frac{l_{upper}+l_{lower}}{2}$
                  $\epsilon_{MVA} := MVA\_bound\,(m, Var\{X_n\}, D, l)$
                  IF ($\epsilon_{MVA} > \epsilon$) THEN $l_{lower} := l$ ELSE $l_{upper} := l$
          END WHILE
       ELSE
          $l_{upper} := ll$ provided that $MVA\_bound\,(m, Var\{X_n\}, D, ll) < \epsilon$
          $l_{lower} := l$
          WHILE ($|\log \epsilon_{MVA} - \log \epsilon| > \epsilon^*$)
                  $l := \frac{l_{upper}+l_{lower}}{2}$
                  $\epsilon_{MVA} := MVA\_bound\,(m, Var\{X_n\}, D, l)$
                  IF ($\epsilon_{MVA} > \epsilon$) THEN $l_{lower} := l$ ELSE $l_{upper} := l$
          END WHILE
OUTPUT: $l$, the needed bandwidth for the next resizing window
```

Fig. 1. A binary search to compute the needed bandwidth amount.

We consider the delay constraint equivalent with the constraint on the queue tail probability $P(Q > Dc) < \epsilon$ that can be estimated by the MVA bound, symbolically by function MVA_bound in Fig. 1.

The incipient point of our provisioning schemes is to collect periodically the aggregate rate of the incoming traffic in consecutive time slots with length τ. Bandwidth provisioning is performed at a larger time scale expressed in *resizing windows* (or shortly, windows). One resizing window consists of N measurement time slots. We compute the mean and covariance functions of traffic over a given window j as $m_j = \frac{\sum_{i=1}^{N} x_i^{(j)}}{N}$, and $C_j(k) = \frac{1}{N-k}\sum_{i=1}^{N-k}(x_i^{(j)} - m_j)(x_{i+k}^{(j)} - m_j)$ for $k = 0, 1, \ldots N-1$. The computed quantities enable us to capture the mean and variance of the accumulated traffic process X_n by using equations (1) and (2) and in turn the MVA bound. At the end of each resizing window we make a decision about the bandwidth amount needed for the next resizing window. We specify two schemes for this task.

PS1 scheme, delay-based, with prediction. In this scheme we propose to perform traffic prediction at the end of each resizing window for the next one. *We predict both the mean rate and the variance of the cumulative process X_n (determined by the correlation structure of the traffic).* We opt the exponential smoothing (ES) technique for the prediction due to its proven stability and suitability on trend prediction [8]. Formally, for the resizing window $j+1$ we predict

$$m_{j+1}^* = w.m_j + (1-w).m_j^*, \tag{3}$$

where w is the weighting parameter ($0 \leq w \leq 1$), m_j^* and m_j are the predicted and measured value of the mean rate for the resizing window j, respectively. Similarly, for the variance of the accumulated traffic, we predict

$$Var^*\{X_{n,j+1}\} = w.Var\{X_{n,j}\} + (1-w).Var^*\{X_{n,j}\} \quad (4)$$

where $Var^*\{X_{n,j}\}$ and $Var\{X_{n,j}\}$ $(n = 0, 1, \ldots N-1)$ are the predicted and measured value of the corresponding accumulated variance for the resizing window j, respectively. We then use the predicted m^*_{j+1} and $Var^*\{X_{n,j+1}\}$ values as the inputs for the binary search presented in Fig. 1 to define the needed bandwidth for the window $j+1$. The output of the search is the bandwidth amount assuring that the achievable QoS is sufficiently close (expressed via the parameter ϵ^*) to the target QoS requirements.

Later, in Section 3 we will consider several aspects to enhance the performance of this provisioning scheme.

PS2 scheme: Delay-based, without prediction. In this scheme, we simply use the computed m_j and $Var\{X_{n,j}\}$ $(n = 0, 1, \ldots N-1)$ as the input parameters of the binary search for the needed bandwidth of the window $j+1$.

3 Performability Investigations

3.1 Scenario Settings and Methodology for Evaluations

Besides PS1 and PS2, we also take two other provisioning schemes into consideration:

PS3 scheme: Utilization-based, without prediction [5]. In this scheme, link bandwidth is adjusted based on the relation between the link utilization threshold and the measured utilization of the last resizing window. The bandwidth amount to be added or released is measured in *quota*. One quota can be set to e.g. $\beta\sqrt{v}$, where v is the variance of the measured traffic rate. In accordance with [5], we set the target link utilization to 0.8 and $\beta = 0.6$.

PS4 scheme: Variance-based, without prediction [9]. In this scheme, the provisioned bandwidth for the next resizing window is chosen to be $m_j + \alpha\sqrt{v_j}$, where m_j and v_j are the mean and variance in the current window j. In accordance with [9], we set $\alpha = 3$.

We use three real traffic traces[1] to produce the aggregate load offered to the link. The MPEG trace is the trace of a *James Bond movie* available from [10]. The *BC-pAug89* trace of Ethernet traffic available from [11]. The WAN trace is a wide-area TCP traffic trace *dec-pkt-1* available also from [11]. To generate the aggregate traffic with high multiplexing degree, we merge 100 individual sources having the above recorded traffic pattern. The starting time of each individual source is randomly chosen to assure independency between the sources.

We examine two scenarios of provisioning as regards the scale of basic measurement time slots and resizing windows. Namely, we consider *resizing at small time scale* when each resizing window contains 100 measurement time slots of length 40ms, i.e. resizing is done after each 4s interval. In this case, the provisioning is tested along 100 resizing windows, meaning that the whole period of

[1] We process the original data traces so that the load over consecutive time intervals with fix length, i.e. the measurement slots, can be obtained.

provisioning is 400s. With *resizing at large time scale*, each time slot is 1200ms, and resizing is done after each 2 minutes. In this case, we test the provisioning schemes along 13 resizing windows, meaning that the whole period of provisioning is approximately half an hour.

Besides performing analytical results, we also resort to trace-driven simulation to verify and evaluate the performability of the provisioning schemes. The basic scenario is that traffic according to the generated aggregate trace is accommodated via a link. The link capacity is adjusted after each resizing window time and a value computed off-line with the specific provisioning scheme is assigned.

In order to evaluate qualitatively the applied provisioning schemes, we introduce the notion of *Average Goodness Factor (AGF).* This is a measure of how fast and closely the provisioned bandwidth follows the real traffic dynamics, while assuring the target QoS. The basic idea is that in an ideal provisioning case, the link utilization should be kept constant at a fix optimal level u_{opt}. The concrete value of u_{opt} is chosen from the experiments gained with using the Gaussian traffic model to deduce the relation between the objective QoS parameters and the link utilization. For a given resizing window j, let us denote the provisioned link capacity by l_j, the real aggregate traffic rate by r_j. We then define the Goodness Factor (GF) as follows[2]:

$$GF_j := \begin{cases} \frac{(l_j - r_j)/r_j}{u_{opt}} & \text{if } l_j \le r_j \\ \frac{r_j/l_j}{u_{opt}} & \text{if } l_j > r_j \text{ and } r_j/l_j \le u_{opt} \\ \frac{u_{opt}}{r_j/l_j} & \text{if } l_j > r_j \text{ and } r_j/l_j > u_{opt} \end{cases} \tag{5}$$

The AGF is then obtained by averaging all individual GF values over the resizing windows, $AGF = \sum_1^M GF_j/M$, where M is the number of resizing windows. The higher the degree of over-provisioning or under-provisioning, the smaller the AGF value. The minimum AGF value is $-1/u_{opt}$, the maximum AGF value is 1. Moreover, the closer the AGF to 1, the better the provisioning scheme.

In the following experiments, if not stated otherwise, we set $w = 0.8$, the initial value of the link bandwidth to 150Mbps. The input parameters for the binary search in Fig. 1 are delay bound $D = 10ms$, violation probability $\epsilon = 10^{-4}$, and $\epsilon^* = 0.1$.

3.2 Comparative Discussions

In Figs. 2-3 we depict the measured aggregate rate (m_j), the predicted aggregate rate (m_j^*) and the bandwidth provisioned by the schemes versus the resizing windows[3] Considering the figures we see that the PS1, PS2, and PS4 schemes capture very well the shape of the aggregate traffic. The three schemes react fast and closely to variation of the aggregate rate. The PS3 scheme does not follow well all the fluctuations, but rather has a smooth shape with linear increase or

[2] We refer to [12] for the detailed discussions of the introduced expressions.

[3] Due to space limitation, we skip the figures related to WAN and Ethernet traffic.

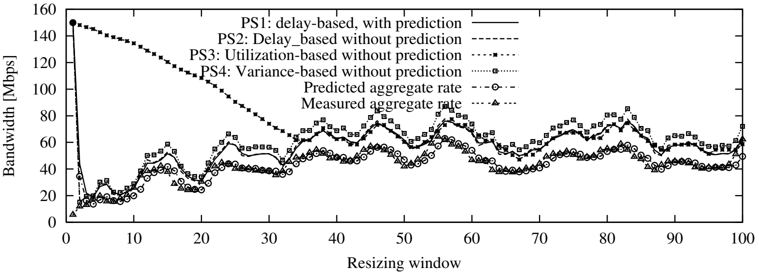


Fig. 2. Small time scale bandwidth provision for MPEG traffic.

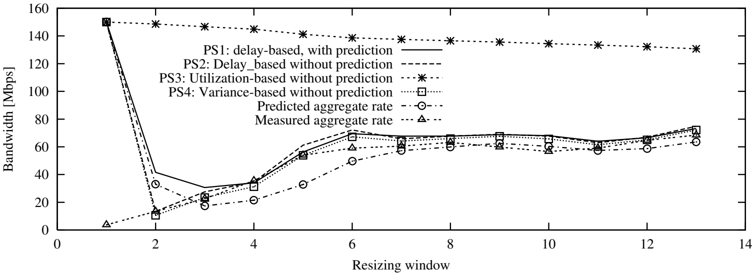


Fig. 3. Large time scale bandwidth provision for MPEG traffic.

decrease. This is in accordance with the original intention of this approach, i.e. *adjusting bandwidth insensitively to small short-time traffic fluctuation* [5].

At a larger time scale when provisioning actions are taken at every 2 minutes, we see that the PS3 scheme works unacceptably (Fig. 3). It is due to the inadequate setting of the quota volume through the parameter β. In fact, the AGF assessment in Table 1 confirms that with $\beta = 0.6$, the PS3 scheme exhibits the poorest performance in such cases. The table also shows that the PS4 scheme, though still captures well the tendency of the traffic rate, is outperformed by the PS1 and PS2 schemes. Intuitively, appropriately tuning α and β probably leads to better performance of PS4 and PS3, respectively. *However, if we do not have a reasonable mapping between the QoS requirements and the needed link utilization then there is no way to choose the right values of α and β. Thus, more sophisticated provisioning schemes with model-based mapping features (like using the Gaussian approximation as in PS1 and PS2 schemes) are definitely preferable.*

In Table 1, we report the AGF values of the provisioning schemes for different traffic scenarios. The value of u_{opt} is set to be the average value of the utilizations achievable over all the resizing windows with the Gaussian traffic model to meet the target QoS requirement. To make the comparative evaluation complete, we also include the AGF of the static provisioning schemes, where bandwidth the is kept constant over the whole time. In the "bad" scheme *Static-1* the bandwidth is fixed at 150Mbps. In scheme *Static-2*, with the rough

Table 1. AGF values of different provisioning schemes.

	Small time scale			Large time scale		
Scheme\Traffic	MPEG	WAN	Ethernet	MPEG	WAN	Ethernet
	$u_{opt} = 0.772$	$u_{opt} = 0.890$	$u_{opt} = 0.804$	$u_{opt} = 0.788$	$u_{opt} = 0.781$	$u_{opt} = 0.737$
Static-1	0.368	0.123	0.422	0.403	-0.095	0.793
Static-2	0.778	0.721	0.751	0.734	0.722	0.683
PS1	0.880	0.898	0.913	0.704	0.721	0.840
PS2	0.916	0.914	0.914	0.683	0.803	0.848
PS3	0.750	0.830	0.841	0.443	-0.133	0.794
PS4	0.889	0.908	0.912	0.556	0.503	0.676

knowledge on the aggregate rate of traces, the bandwidth is fixed at 70Mbps, 250Mbps and 200Mbps for MPEG, WAN and Ethernet traffic, respectively. Table 1 shows that the static schemes are in general outperformed by the rest of the schemes. Moreover, the qualitative relation between the AGF values of the schemes confirm all our previous arguments. For MPEG and WAN traffic, the PS3 scheme works unacceptably at large time scale provisioning. PS1 and PS2 have nearly the same goodness and they are the best among the tested schemes. Despite this fact, there are still questionable issues with PS1 and PS2 concerning signaling overhead and under-provisioning. We deal with these issues in the next subsection.

3.3 Provisioning Enhancements: Signalling Reduction and Under-estimation Avoidance

From the macroscopic point of view, the simplest way to reduce signalling overhead is to skip non-critical bandwidth adjustments. Therefore, we propose the following potential solutions for scalability improvements. For *skipping downward adjustments*, we consider that *over-provisioning is always less detrimental than under provisioning*. Thus, if the relative bandwidth bias remains below a certain value (e.g. 5% of the current bandwidth value), we could keep the current bandwidth amount and do not perform deallocation. By doing this, a certain range of over-provisioning is implicitly involved at the gain of signalling effort.

For *skipping upward adjustments*, we introduce a certain number of bandwidth levels. We refer to the difference between two consecutive bandwidth levels as a bandwidth interval. If both the current and the new bandwidth values stay within the same bandwidth interval, we do not initiate the upgrade process. This is based on the compromise that *within one bandwidth interval, we can tolerate a certain degree of QoS degradation*. The procedure checking the impact of bandwidth interval's size on QoS degradations is done as follows. We compute first the needed bandwidth for the next resizing window taking the required delay violation probability into account. Afterward, we reduce the computed bandwidth by the amount identical to one bandwidth interval size and then recompute the delay violation probability we should get. We use the ratio between the recomputed violation probability and the original violation probability as an QoS degradation index, which is depicted in Figs. 4 and 5. We see that the bandwidth interval should be chosen smaller than 1% of the mean aggregate rate

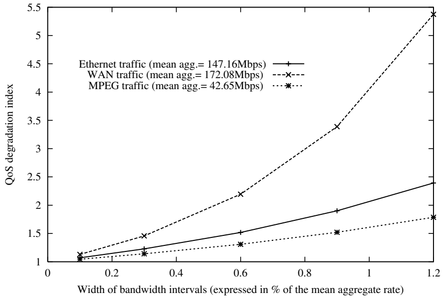


Fig. 4. QoS degradation index vs. bandwidth interval (small time scale).

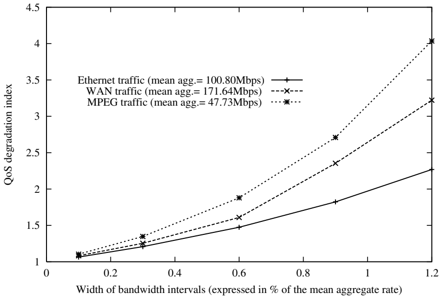


Fig. 5. QoS degradation index vs. bandwidth interval (large time scale).

Table 2. Number of adjustments of different provisioning schemes (small time scale).

Bias\Traffic	MPEG				WAN				Ethernet			
	PS1	PS2	PS3	PS4	PS1	PS2	PS3	PS4	PS1	PS2	PS3	PS4
0.01	93	91	79	90	88	85	92	87	95	95	81	92
0.02	89	88	79	87	85	82	92	85	92	90	81	88
0.03	85	85	79	86	79	78	92	77	83	87	81	84
0.04	76	82	79	79	70	75	92	73	80	83	81	80
0.05	75	75	79	75	69	73	92	65	70	83	81	70

to ensure that the QoS degradation index is below 4-5. Note that the small value of the desired delay violation probability (in a range of 10^{-4} or even smaller) makes the QoS degradation index in a range of 1-4 considered acceptable.

In Table 2, we present the needed adjustment number of the considered provisioning schemes. The bias (applied for skipping downward changes) is varied from 1% to 5%, the bandwidth interval (applied for skipping upward changes) is chosen to be 2Mbps for Ethernet and WAN traffic trace, 0.4Mbps for the MPEG trace. As can be seen in the tables, the implication of the proposed skipping rules indeed yields noticeable signaling gains, which can be up to a range of 20%-30%.

Another problem with the PS1 scheme is the negative effect of under provisioning. As can be observed from Figs. 2-3 (see e.g. Fig. 3, windows 3-7), although use of the ES technique for prediction enables a quite close track on the trend of the actual traffic, there is a certain lag between the real traffic rate and the predicted rate. This induces the fact that the predicted rate underestimates the real one in certain cases. Since the predicted rate is used as an input for the procedure finding the bandwidth amount to provision, we suffer from QoS degradations. To remedy such undesirable situations, we first reveal an important observation. *With ES technique, if we have a predicted load smaller than the current measured load, then this under estimation remains as long as the aggregate load exhibits increasing trend* (see e.g. again Fig. 3 windows 3-7). In fact, it is not difficult to prove mathematically that if $m_{j+1} > m_j > m_{j-1}$ and $m_j > m_j^*$ then $m_{j+1} > m_{j+1}^*$. The effect of this observation should be highly avoided, since it causes under-provisioning, and in turn long-term QoS degradations. Therefore, we develop two enhanced versions of PS1 correcting

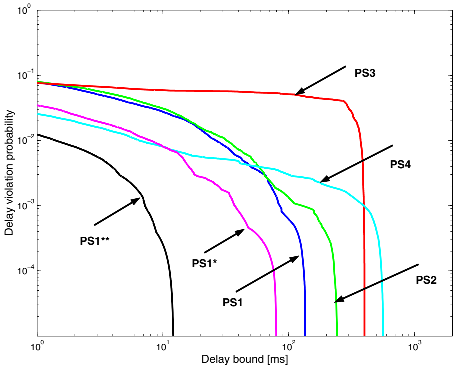


Fig. 6. Delay violation probability (Ethernet traffic, small time scale).

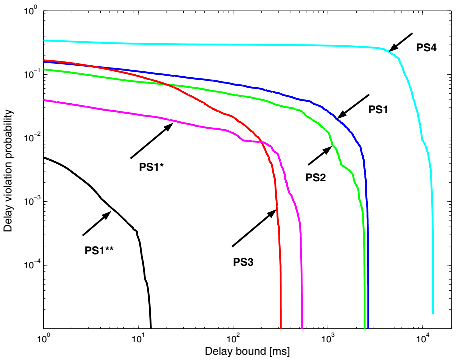


Fig. 7. Delay violation probability (Ethernet traffic, large time scale).

this impact. The basic idea is that we modify the prediction rules as soon as we observe an evidence of increasing traffic trend and underestimation.

PS1*: Modifying prediction rules with linear extrapolation. We change the prediction rule whenever we experience first (let's say at the resizing window j) two facts together: increasing traffic trend and under prediction, i.e. $m_j > m_{j-1}$ and $m_j > m_j^*$. Instead of (3), we predict $m_{j+1}^* = m_j + (m_j - m_{j-1})$, assuming a local linear increasing trend of traffic. This modification is again applied in the window $j+1$ if we still have $m_{j+1} > m_j$ and $m_{j+1} > m_{j+1}^*$, otherwise we switch back to the normal ES rule according to (3), and so on.

The same consideration is employed for the prediction of the variance function $Var\{X_{n,j}\}$ $(n = 0, 1, \dots N-1)$.

PS1**, Modifying prediction rules with predicted increments. Similar to the previous approach, we modify the prediction rule from the first time (i.e. window j) we observe two facts together: increasing traffic trend and under prediction. However, in this approach we use predicted increments between the loads from two successive resizing windows to give a load forecast. Namely, at the first window j, we reset $m_j^* := m_j + (m_{j-1}^* - m_{j-1})$ to ensure that the predicted value is above the measured value with the difference observed earlier in window $j-1$. Then for the following windows k, $k \geq j$, as long as $m_k > m_{k-1}$ (i.e. the increasing trend is still valid), we predict $m_{k+1}^* := m_k^* + \Delta_{k+1}^*$ instead of using (3). Here $\Delta_{k+1} = m_{k+1} - m_k$ and Δ_{k+1}^* is estimated by the ES technique, i.e. $\Delta_{k+1}^* = w\Delta_k + (1-w)\Delta_k^*$. We initially set $\Delta_j^* = 0$. When the increasing trend stops, we switch back to the original ES rule (3).

We plot the delay violation probability obtained with trace driven simulation in Fig. 6 and Fig. 7 for Ethernet traffic. Indeed, significant improvements can be seen in the figures ranking PS_1^* and PS_1^{**} to be the best. The target QoS $P(delay > 10ms) < 10^{-4}$ is indeed fulfilled with PS_1^{**} scheme. Note that the obtained QoS improvements are not at the expense of extensive over-dimensioning. The computed AGF values for PS_1^* and PS_1^{**} are 0.895 and 0.870 in case of small time scale provisioning, and 0.847 and 0.832 in case of large time scale provisioning, respectively, i.e. still in the range of the AGFs for PS1 and PS2 (see Table 1). Also note that the QoS attained with simulation for other schemes is worse than the original QoS target. This is exactly due to the effect of under-provisioning stemming from traffic under-estimation during a certain number of windows which have been investigated above. The same positive effects of PS_1^* and PS_1^{**} are also experienced with other traffic traces.

4 Conclusions

We have presented novel adaptive provisioning schemes derived from the explicit mapping between target QoS requirements and link bandwidth. The operation of the schemes combines measurement, Gaussian traffic model and traffic prediction features. Performance analysis with real traffic traces has indicated that the proposed schemes, particularly $PS1^{**}$, are more suitable than existing schemes for provisioning with explicit respect to statistical QoS requirements.

We are currently working on the insights into the effect of provisioning time scales and performance assessment with longer traffic traces.

References

1. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services. RFC2475, December 1998.
2. E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. RFC3031.
3. R. Gibbens and P. Key. Distributed Control and Resource Marking Using Best-Effort Routers. *IEEE Network*, 15(3):54–59, May/June 2001.
4. B. Wydrowsky and M. Zukerman. QoS in Best-Effort Networks. *IEEE Communications Magazine*, 40(12):44–49, December 2002.
5. Z. Duan, Z-L Zhang, and Y. T. Hou. Service Overlay Networks: SLAs, QoS and Bandwidth Provisioning. In *Proceedings of IEEE 10th International Conference on Network Protocols (ICNP)*, pages 334–343, 2002.
6. J. Choe and N. B. Shroff. A Central Limit Theorem based Approach for Analyzing Queue Behavior in High Speed Networks. *IEEE/ACM Transactions on Networking*, 6(5):659–671, October 1998.
7. H. S. Kim and N. B. Shroff. Loss Probability Calculations and Asymptotic Analysis for Finite Buffer Multiplexers. *IEEE/ACM Transactions on Networking*, 9(6):755–768, December 2001.
8. W. S. Wei. *Time Series Analysis.* Addison Wesley, 1990.
9. N. G. Duffield, P. Goyal, and A. Greenberg. A Flexible Model for Resource Management in Virtual Private Networks. In *Proceedings of SIGCOMM*, pages 95–108, 1999.
10. MPEG traces. ftp-info3.informatik.uni-wuerzburg.de/pub/MPEG/.
11. The Internet traffic archive. http://ita.ee.lbl.gov/index.html.
12. H. T. Tran. Adaptive Bandwidth Provisioning with Explicit Respect to QoS Requirements. Technical Report, http://cntic03.hit.bme.hu/˜hung/Prov-report.pdf.

Wide Area Measurements of Voice over IP Quality

Ian Marsh[1], Fengyi Li[2], and Gunnar Karlsson[2]

[1] SICS AB, Kista S-164 29, Sweden
ianm@sics.se
[2] Department of Microelectronics and Information Technology
KTH, Royal Institute of Technology
S-164 40 Kista, Sweden
d97-fli@nada.kth.se, gk@imit.kth.se

Abstract. Time, day, location and instantaneous network conditions largely dictate the quality of Voice over IP calls. In this paper we present the results of over 18000 VoIP measurements, taken from nine sites connected in a full-mesh configuration. We measure the quality of the routes on a hourly basis by transmitting a pre-recorded call between a pair of sites. We repeat the procedure for all nine sites during the one hour interval. Based on the obtained jitter, delay and loss values as defined in RFC 1889 (RTP) we conclude that the VoIP quality is acceptable for all but one of the nine sites we tested. We also conclude that VoIP quality has improved marginally since we last conducted a similar study in 1998.

1 Introduction

It is well known that the users of real-time voice services are sensitive and susceptible to variable audio quality. If the quality deteriorates below an acceptable level or is too variable, users often abandon their calls and retry later. Since the Internet is increasingly being used to carry real-time voice traffic, the quality provided has become, and will remain an important issue. The aim of this work is therefore to disclose the current quality of voice communication at end-points on the Internet.

It is intended that the results of this work will be useful to many different communities involved with real-time voice communication. Within the next paragraph we list some potential groups to whom this work might have relevance. Firstly end users can determine which destinations are likely to yield sufficient quality. When deemed insufficient they can take preventative measures such as adding robustness, for example in the form of forward error correction to their conversations. Operators can use findings such as these to motivate upgrading links or adding QoS mechanisms where poor quality is being reported. Network regulators can use this kind of work to verify the quality level that was agreed upon, has indeed been deployed. Speech coder designers can utilise the data as input for a new class of codecs, of particular interest are designs which yield good quality in the case of bursty packet loss. Finally, researchers could use the data to investigate questions such as, “Is the quality of real-time audio communication on the Internet improving or deteriorating?”.

The structure of the paper is as follows: Section 2 begins with some background on the quality measures we have used in this work namely, loss, delay and jitter. Following

G. Karlsson and M.I. Smirnov (Eds.): QofIS 2003, LNCS 2811, pp. 93–101, 2003.

on from the quality measures, section 3 gives a description of the methodology used to ascertain the quality. In section 4 the results are presented, and due to space considerations we condense the results into one table showing the delay, loss and jitter values for the paths we measured. In section 5 the related work is given, comparing results obtained in this study with other researchers' work. This is considered important as it indicates whether quality has improved or deteriorated since those studies. Section 6 rounds off with some conclusions and a pointer to the data we have collated.

2 What Do We Mean by Voice over IP Quality?

Ultimately, users judge the quality of voice transmissions. Organisations such as ETSI, ITU, TIA, RCR plus many others have detailed mechanisms to assess voice quality. These organisations are primarily interested in speech coding. Assigning quality 'scores' involves replaying coded voice to both experienced and novice listeners and asking them to adjudge the perceived quality. Measuring the quality of voice data that has been transmitted across a wide area network is more difficult. The network inflicts its own impairment on the quality of the voice stream. By measuring the delay, jitter and loss of the incoming data stream at the receiver, we can provide some indication on how suitable the *network* is for real-time voice communication. The two schemes can be combined as was proposed by the ITU using with the E-model [ITU98]. It is important to point out we did not include the quality contribution of the end systems in this study. This is because the hardware was different at each site (albeit all UNIX systems), however the software was our own VoIP tool, Sics*o*phone [HMH03]. In order to assess the delay contribution of each end system it would be difficult without isolation tests. We did however choose to use simple A-law PCM coding to maintain a theoretically known coding/decoding delay.

The quality of VoIP sessions can be quantified by the network delay, packet loss and packet jitter. We emphasise that these three quantities are the major contributors to the perceived quality as far as the *network* is concerned. The G.114 ITU standard states that the end-to-end one way delay should not exceed 150ms [RG98]. Delays over this value adversely effect the quality of the conversation. An alternative study by Cole and Rosenbluth state that users perceive a linear degradation in the quality up to 177ms [CR02]. Above this figure the degradation is also linear although markedly worse. As far as the packet loss is concerned, using simple speech coding such as A-law or μ-law coding, tests have shown that the mean packet loss should not exceed 10% before glitches due to lost packets seriously affect the perceived quality. Note that a loss rate such as this does not say anything about the distribution of the losses. As far as the authors are aware of, no results exist that state how jitter solely can affect the quality of voice communication. Work on jitter and quality are often combined with loss or delay factors. When de-jittering mechanisms are employed, the network *jitter* is typically transferred into application *delay*. The application must hold back a sufficient number of packets in order to ensure smooth, uninterrupted playback of speech. To summarise, we refer to the quality as a combination of delay, jitter and loss. It is important to mention we explicitly do not state how these values should be combined. The ITU E-

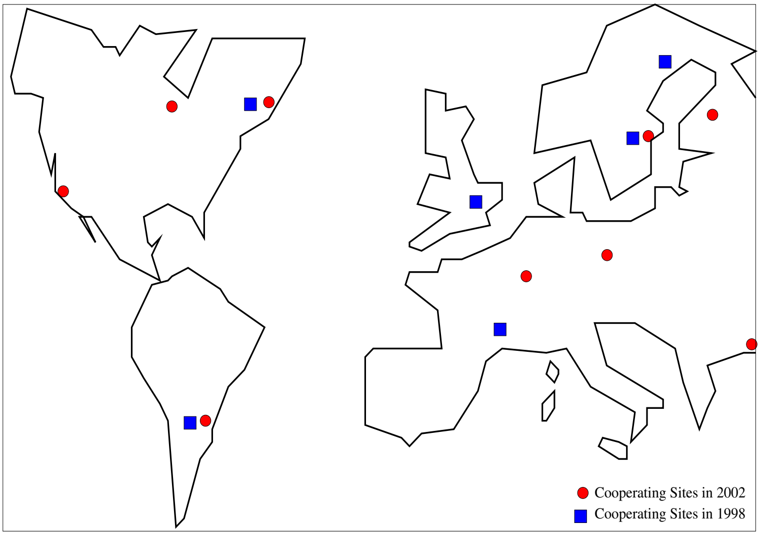


Fig. 1. The nine sites used in the 2002 measurements are shown with circles. The six depicted with squares show those that were available to us in 1998, three remained unchanged during the four years.

model is one approach but others exist, therefore we refer the interested reader to the references given as well as [LE01] and [KKI91].

3 Simulating and Measuring Voice over IP Sessions

Our method to measure VoIP quality is to send pre-recorded calls between globally distributed sites. Through the modification of our own VoIP tool, Sics*o*phone, the intervening network paths are probed by a 70 second pre-recorded 'test signal'. The goal of this work is therefore to report in what state the signal emerges after traversing the network paths. Incidentally, we do not include the signalling phase, i.e. establishing a connection with the remote host, rather we concentrate solely on the quality of the data (or speech) transfer.

Nine sites have been carefully chosen with large variations in hops, geographic distances and time-zones to obtain a diverse selection of distributed sites. One important limitation of the available sites was they were all located at academic institutions, which are typically associated with well provisioned networks. Their locations are shown in the map of Figure 1. The sites were connected as a full mesh allowing us, in theory, to measure the quality of 72 different Internet paths. In practice, some of the combinations were not usable due to certain ports being blocked, thus preventing the audio being sent to some sites. There were four such cases. Bi-directional sessions were scheduled on a hourly basis between any two given end systems. Calls were only transferred once per hour due to load considerations on remote machines.

In Table 1 below we list the characteristics of the call we used to probe the Internet paths between those sites indicated on the map. Their locations, separation in hops and time zones are given in the results section. As stated, the call is essentially a fixed length PCM coded file which can be sent between the sites. The file consisted of a PCM coded recording of a read passage of text. Over a 15 week period we gathered just over 18,000 recorded sessions. The number of sessions between the nine sites is not evenly distributed due to outages at some sites, however we attempted to ensure an even number of measurements per site, in total nearly 33 million individual packets were transmitted during this work.

Table 1. The top half of the table gives details of the call used to measure the quality of links between the sites. The lower half provides information about the data which was gathered.

Test "signal"	
Call duration	70 seconds
Payload size	160 bytes
Packetisation time (ms)	20ms
Data rate	64kbits/sec
With silence suppression	2043 packets
Without silence suppression	3653 packets
Coding	8 bit PCM
Recorded call size	584480 bytes
Obtained data	
Number of hosts used (2003)	9
Number of traces obtained	18054
Number of data packets	32,771,021
Total data size (compressed)	411 Megabytes
Measurement duration	15 weeks

3.1 A Networking Definition of Delay

We refer to the delay as the *one way network* delay. One way delay is important in voice communication, particularly if it is not the same in each direction. Measuring the one way delay of network connections without synchronised clocks is a non-trivial task. Hence many methods rely on round-trip measurements and halve the values, hence estimating the one way delay. We measured the network delay using the RTCP protocol which is part of the RTP standard [SCFJ96]. A brief description follows. At given intervals the sender transmits a so called "report" containing the time the report was sent. On reception of this report the receiver records the current time. Therefore two times are recorded within the report. When returning the report to the sender, the receiver subtracts the time it initially put in the report, therefore accounting for the time it held the report. Using this information the sender can calculate the round-trip delay and importantly, discount the time spent processing the reports at the receiver. This can be done in both directions to see if any significant anomalies exist. We quote the network delay in the results section as they explicitly do not include any contribution from the end hosts. Therefore it is important to state the delay is *not* the end-to-end delay but the network delay. We chose not to include the delay contributed by the end system as it varies widely from operating system to operating system and how the VoIP application itself is implemented. The delay incurred by an end system can vary from 20ms up to 1000ms, irrespective of the stream characteristics.

3.2 Jitter – An IETF Definition

Jitter is the statistical variance of the packet interarrival time. The IETF in RFC 1889 define the jitter to be the mean deviation (the smoothed absolute value) of the packet spacing change between the sender and the receiver [SCFJ96]. Sics*o*phone sends packets of identical size at constant intervals which implies that $S_j - S_i$ (the sending times of two consecutive packets) is constant. The difference of the packet spacing, denoted D, is used to calculate the interarrival jitter. According to the RFC, the interarrival jitter should be calculated continuously as each packet i is received. The interarrival jitter J_i for packet i is calculated using the previous packet J_{i-1} thus:

$$J_i = J_{i-1} + (|D(i-1,i)| - J_{i-1})/16 \ .$$

According to the RFC "the gain parameter 1/16 gives a good noise reduction ratio while maintaining a reasonable rate of convergence". As stated earlier, buffering due to jitter adds to the delay of the application. This delay is accounted for in the results we present. The "real" time needed for de-jittering depends on how the original time spacing of the packets should be restored. For example if a single packet buffer is employed it would result in an extra 20ms (the packetisation time) being added to the total delay. Note that packets arriving with a spacing greater than 20ms should be discarded by the application as being too late for replay. Multiples of 20ms can thus be allocated for every packet held before playout in this simple example. To summarise, the delay due to de-jittering the arriving stream is implementation dependent, thus we do not include it in our results.

3.3 Counting Ones Losses in the Network

We calculate the lost packets as is exactly defined in RFC 1889. It defines the number of lost packets as the expected number of packets subtracted by the number actually received. The loss is calculated using expected values so as to allow more significance for the number of packets received. For example 20 lost packets from 100 packets has a higher significance than 1 lost from 5. For simple measures the percentage of lost packets from the total number of packets expected is stated. As stated the losses in this work *do not* include those incurred by late arrivals, as knowledge of the buffer playout algorithm is needed, therefore our values are only the network loss. Detailed analysis of the loss patterns is not given in the results section, we simply state the percentages of single, double and triplicate losses.

4 Results

The results of 15 weeks of measurements are condensed into figure 2. The table should be interpreted as an 11 by 11 matrix. The locations listed horizontally across the top

receiver sender	Massachusetts	Michigan	California	Belgium	Finland	Sweden	Germany	Turkey	Argentina	Mean
Massachusetts	*	D:38.0 (17.1) J:2.4 (1.7) L:0.1 (0.6) H:14 (+1) T:0	D:54.2 (15.8) J:2.4 (1.8) L:0.1 (0.9) H:19 T:-3	D:67.1 (15.5) J:3.6 (1.5) L:0.1 (0.8) H:11 T:+6	D:97.1 (2.6) J:2.5 (1.5) L:0.1 (0.8) H:15 T:+7	D:99.5 (8.5) J:3.2 (1.7) L:0.04 (0.2) H:21 T:+6	D:58.4 (5.0) J:4.5 (1.4) L:0.0 (0.0) H:17 (+3) T:+6	D:388.2 (43.2) J:10.4 (4.9) L:4.9 (4.7) H:20 T:+7	D:99.7 (4.9) J:19.9 (8.4) L:8.9 (7.2) H:25 T:+1	D:112.8 J:6.1 L:1.2 H:17
Michigan	D:36.4 (15.4) J:4.7 (0.8) L:0.0(0.2) H:14 (+1) T:0	*	D:40.4 (4.5) J:4.4 (0.8) L:0.2 (1.1) H:20 (+1) T:-3	D:63.5 (4.2) J:4.3 (0.7) L:0.0 (0.1) H:11 T:+6	D:88.2 (8.0) J:4.1 (0.7) L:0.1 (1.1) H:17 T:+7	D:86.7 (4.7) J:5.2 (0.6) L:0.1 (2.2) H:23 T:+6	D:63.6 (8.2) J:7.3 (1.9) L:0.2 (0.9) H:16 (+1) T:6	D:358.9 (44.9) J:5.6 (1.7) L:3.0 (1.9) H:20 T:7	D:112.1 (10.6) J:18.7 (7.9) L:6.5 (7.0) H:25 T:+1	D:106.2 J:6.8 L:1.3 H:18
California	D:54.5 (16.7) J:2.0 (1.0) L:0.1 (0.36) H:18 (+1) T:+3	D:40.6 (5.1) J:1.2 (0.6) L:0.1 (1.9) H:21 T:+3	*	D:81.0 (2.2) J:1.6 (0.8) L:0.2 (0.8) H:20 T:+9	D:106.0 (3.0) J:1.4 (0.8) L:0.6 (1.4) H:25 (+1) T:+10	D:108.0 (2.4) J:2.1 (0.9) L:0.2 (0.3) H:30 (+2) T:+9	D:81.5 (1.8) J:4.9 (1.5) L:2.8 (3.0) H:23 T:+9	D:386.9 (60.5) J:5.3 (1.7) L:4.4 (2.4) H:23 T:+10	D:123.9 (12.4) J:18.1 (9.9) L:8.9 (8.2) H:25 T:+4	D:122.2 J:4.6 L:2.2 H:23
Belgium	D:65.2 (10.1) J:1.6 (0.6) L:0.0 (0.0) H:16 T:-6	D:63.4 (3.3) J:0.6 (0.1) L:0.0 (0.0) H:17 T:-6	D:84.0 (1.3) J:0.9 (0.8) L:1.2 (1.0) H:23 T:-9	*	D:31.3 (0.6) J:0.9 (0.5) L:0.0 (0.0) H:17 T:+1	D:33.4 (0.2) J:1.6 (0.9) L:0.0(0.0) H:22 T:0	D:16.6 (10.4) J:3.4 (1.5) L:0.21 (0.7) H:13 T:0	D:341.1 (24.7) J:6.9 (2.0) L:3.8 (2.7) H:16 (+2) T:+1	D:136.5 (7.1) J:NA L:NA H:19 T:-5	D:96.4 J:2.0 L:0.6 H:17
Finland	D:97.8 (4.2) J:1.7 (0.8) L:0.0 (0.1) H:15 (+1) T:-7	D:86.8 (1.9) J:1.1 (0.6) L:0.0 (0.3) H:17 (+1) T:-7	D:109.9 (4.7) J:1.4 (0.8) L:0.7 (1.4) H:24 (+2) T:-10	D:30.7 (0.3) J:1.4 (0.6) L:0.1 (0.3) H:16 T:-1	*	D:13.6 (1.0) J:1.9 (0.9) L:0.0 (0.0) H:20 T:-1	D:26.8 (7.3) J:3.9 (1.1) L:0.0(0.0) H:20 (+1) T:-1	D:321.2 (39.3) J:3.4 (1.7) L:3.2 (1.7) H:17 (+2) T:0	D:161.5 (12.2) J:17.4 (8.2) L:7.5 (6.5) H:19 T:-6	D:106.3 J:4.1 L:1.4 H:18
Sweden	D:99.3 (8.8) J:3.0 (1.9) L:0.0 (0.0) H:22 (+1) T:-6	D:84.9(1.9) J:2.5 (2.0) L:0.03 (0.4) H:25 T:-6	D:105.6 (2.1) J:3.2 (1.96) L:0.1 (0.1) H:30 T:-9	D:33.3 (0.4) J:2.8 (1.6) L:0.1 (0.3) H:24 T:0	D:13.5 (0.5) J:2.4 (1.8) L:0.0 (0.01) H:21 T:+1	*	D:29.8 (12.8) J:4.8 (2.5) L:0.0 (0.0) H:25 T:0	D:322.2 (30.3) J:3.2 (1.49) L:2.9 (1.0) H:26 T:+1	D:165.6 (17.9) J:NA L:NA H:41 T:-5	D:107.8 J:2.8 L:0.4 H:26
Germany	D:63.5 (9.6) J:1.72 (0.7) L:0.0(0.0) H:15 T:-6	D:60.4 (0.5) J:0.7 (0.3) L:0.0 (0.0) H:16 T:-6	D:84.4 (1.0) J:1.8 (0.7) L:2.5 (1.9) H:22 T:-9	D:11.1 (0.2) J:0.8 (0.3) L:0.0 (0.0) H:12 T:0	D:27.8 (7.3) J:1.0 (0.5) L:0.0 (0.0) H:17 T:+1	D:29.2 (7.6) J:1.5 (0.6) L:0.0 (0.0) H:22 T:0	*	D:300.7 (39.7) J:4.8 (2.1) L:3.7 (2.5) H:16 T:+1	D:149.8 (15.6) J:NA L:NA H:18 T:-5	D:90.9 J:1.6 L:0.8 H:17
Turkey	D:379.1 (47.1) J:8.6 (0.7) L:8.1 (2.8) H:18 (+1) T:-7	D:387.9 (35.5) J:8.9 (1.2) L:8.0 (2.9) H:20 T:-7	D:410.9 (43.9) J:8.8 (2.5) L:7.6 (6.8) H:19 T:-10	D:330.2 (28.6) J:9.2 (2.0) L:7.10 (4.0) H:17 T:-1	D:318.9 (42.4) J:8.8 (0.6) L:7.8 (2.7) H:19 T:0	D:311.1 (8.3) J:9.1 (0.7) L:8.4 (3.1) H:25 T:-1	D:378.2 (49.3) J:10.7 (1.2) L:8.0 (3.1) H:16 T:-1	*	D:490.8 (26.0) J:NA L:NA H:18 T:-6	D:375.9 J:8.0 L:6.9 H:19
Argentina	D:117.0 (30.8) J:4.2 (2.0) L:0.5 (1.4) H:NA T:-1	D:146.7 (44.2) J:4.3 (2.3) L:0.5 (1.5) H:NA T:-1	D:152.0 (47.8) J:3.1 (2.4) L:0.6 (1.8) H:NA T:-4	D:NA J:4.2 (2.0) L:0.5 (1.4) H:NA T:+5	D:164.1 (27.2) J:3.9(2.2) L:0.5 (1.4) H:NA T:+6	D:160.9 (47.7) J:2.9 (0.8) L:0.0 (0.1) H:NA T:+5	D:180.5 (50.5) J:4.7 (1.5) L:0.1 (0.1) H:NA T:+5	D:NA J:6.0(1.2) L:5.8 (3.0) H:NA T:+6	*	D:115.2 J:4.2 L:1.1 H:NA
Mean	D:114.1 J:3.4 L:1.1 H:14	D:113.6 J:3.4 L:1.1 H:16	D:115.7 J:3.2 L:1.6 H:19	D:77.1 J:3.5 L:1.0 H:13	D:105.8 J:3.1 L:1.1 H:16	D:105.2 J:3.4 L:1.1 H:20	D:104.4 J:5.5 L:1.4 H:16	D:345.6 J:5.7 L:4.0 H:17	D:180.0 J:9.3 L:4.00 H:23	D:136.2 J:4.1 L:1.8 H:18

Fig. 2. A summary of 18000 VoIP sessions. The delay (D), jitter (J) and loss (L) for the nine sites. The delay and jitter are in milliseconds, the losses are in percentages. The number of hops (H) and time zones (T) in hours are also given. The means for each and all sites are given plus the standard deviations (in parentheses). An NA signifies 'Not Available'.

of the table are the locations used as receivers. Listed vertically they are configured as senders. The values in the rightmost column and bottom row are the statistical means for all the connections *from* the host in the same row and *to* the host in the same column respectively. For example the last column of the first row (directly under "Mean") the average delay to all destinations from Massachusetts is 112.8ms.

Each cell includes the delay, jitter, loss, number of hops and the time difference prefixed by the letters D, J, L, H and T for each of the connections. The units for each quantity are the delay in milliseconds, the jitter in milliseconds, the loss in percentage, the hops as reported by traceroute and time differences in hours. A '+' indicates that the local time from a site is ahead of the one in the corresponding cell and behind for a '-'. The values in parenthesis are the standard deviations. A NA signifies "Not Available" for this particular combination of hosts. The bottom rightmost cell contains the mean for all 18054 calls made, both to and from all the nine hosts involved.

The most general observation is the quality of the paths is generally good. The average delay is just below the ITU's G.114 recommendation for the end-to-end delay. Nevertheless at 136ms it does not leave much time for the end systems to encode/decode and replay the voice stream. A small buffer would absorb the 4.1ms jitter and a loss rate of 1.8% is more than acceptable with PCM coding [LE01].

There are two clear groupings from these results, those within the EU and the US and those outside. The connections in Europe and the United States (and between them) are very good. The average delay between the US/EU hosts is 105ms, the jitter is 3.76ms and the loss 1.16%. Those outside fair less well. The Turkish site suffers from large delays, which is not surprising as the Turkish research network is connected via a satellite link to Belgium (using the Geant network). The jitter and loss figures however are low, 5.7ms and 4% respectively. The Argentinian site suffers from asymmetry problems. The quality when sending data to it is significantly worse than when receiving data from it. The delay is 1/3 higher, the jitter is more than twice it in the opposite direction and the loss is nearly four times higher than when sending to it. Unfortunately we could not perform a traceroute from the host in Buenos Aires, so we cannot say how the route contributed to these values.

We now turn our attention to results which are not related to any particular site. As far as loss is concerned the majority of losses are single losses. 78% of all the losses counted in all trace files were single losses whereas 13% were duplicate losses and 4% triplicate losses. For some connections (22 from 68), some form of packet loss concealment would be useful, as the loss is over 1% but always under 10%.

Generally the jitter is low relative to the delay of the link, approximately 3-4%. This is not totally unexpected as the loss rates are also low. With the exception of the Argentinian site, the sites did not exhibit large differences in asymmetry and were normally within 5% of each other in each direction. It is interesting to note that the number of hops could vary under the 15 week measurement period denoted by () in the hops field. Only very few ($< 0.001\%$) out of sequence packets were observed. Within [Li02] there are details of other tests, such as the effect of using silence suppression, differing payload sizes and daytime effects. In summary no significant differences were observed in these tests. We can attribute this (and the good quality results) to generally well-provisioned academic networks.

5 Related Work

Similar but less extensive measurements were performed in 1998 [HHM99]. Only three of the hosts remain from four years ago so comparisons can only be made for these routes. An improvement, in the order of 5-10% has been observed for these routes. We should point out though, the number of sessions recorded four years ago numbered only tens per host, whereas on this occasion we performed hundreds of calls from each host. Bolot et. al. looked at consecutive loss for designing an FEC scheme [BCG95]. They concluded that the number of consecutive losses is quite low and stated that most losses are one to five losses at 8am and between one to ten at 4pm. This is in broad agreement with the findings in this work, however we did not investigate the times during the day of the losses. Maxemchuk and Lo measured both loss and delay variation for intra-state connections within the USA and international links [ML97]. Their conclusion was the quality depends on the length of the connection and the time of day. We did not try different connection durations but saw much smaller variations (almost negligible) during a 24 hour cycle (see [Li02]). We attribute this to the small 64kbits per second VoIP session on well dimensioned academic networks. It is worthy to point out our loss rates were considerably less than Maxemchuks (3-4%). Dong Lin had similar conclusions [Lin99], stating that in fact even calls within the USA could suffer from large jitter delays. Her results on packet loss also agree with those in [BCG95], which is interesting, as the measurements were taken some four years later.

6 Conclusions

We have presented the results of 15 weeks of voice over IP measurements consisting of over 18000 recorded VoIP sessions. We conclude that the quality of VoIP is good, and in most cases is over the requirements of many speech quality recommendations. Recall that all of the sites were at academic institutions which is an important factor when interpreting these results as most universities have well provisioned links, especially to other academic sites. Therefore this work should not be generalised to the whole Internet. Nevertheless, the loss, delay and jitter values are very low and from previous measurements the quality trend is improving. We can only attribute this to more capacity and better managed networks than those four years ago. However some caution should be expressed as the sample period was only 15 weeks, the bandwidth of the flows very small and only used once per hour. We have a large number of sample sessions so can be confident the findings are representative of the state of the network at this time. One conclusion is that VoIP is obviously dependent on the IP network infra-structure and not only on the geographic distance. This can be clearly seen in the differences between the Argentinian and Turkish hosts. Concerning the actual measurement methodology, we have found performing measurements on this scale is not an easy task. Different access mechanisms, firewalls, NATs and not having permissions on all machines, complicates the work in obtaining (and validating later) the measurements. Since it is not possible to envisage all the possible uses for this data we have made it available for further investigation at http://www.sics.se/˜ianm/COST263/cost263.html.

Future work involves further improvements in collecting and analysing the data. During these measurements we did not save (but sent) the audio samples at the receiver, however future measurements will do so in order to capture the distortion and echo impairment. Extending the measurement infra-structure to non-academic sites is also a natural progression of this work. Performing quality measures that include the end systems should also be considered, although how to include the heterogeneity of the end systems still remains unresolved.

References

[BCG95] J. Bolot, H. Crepin, and A. Garcia. Analysis of audio packet loss in the internet. In *Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, Lecture Notes in Computer Science, pages 163–174, Durham, New Hampshire, April 1995. Springer.

[CR02] R.G Cole and J.H Rosenbluth. Voice over IP Performance Monitoring. *ACM Computer Communication Review*, 2002.

[HHM99] Olof Hagsand, Kjell Hansson, and Ian Marsh. Measuring Internet Telephone Quality: Where are we today ? In *Proceedings of the IEEE Conference on Global Communications (GLOBECOM)*, Rio, Brazil, November 1999. IEEE.

[HMH03] Olof Hagsand, Ian Marsh, and Kjell Hanson. Sics*o*phone: A Low-delay Internet Telephony Tool. In *29th Euromicro Conference*, Belek, Turkey, September 2003. IEEE.

[ITU98] ITU-T Recommendation G.107. The E-Model, a computational model for use in transmission planning, December 1998.

[KKI91] Nobuhiko Kitawaki, Takaaki Kurita, and Kenzo Itoh. Effects of Delay on Speech Quality. *NTT Review*, 3(5):88–94, September 1991.

[LE01] B.M.Lines L.F.Sun, G.Wade and E.C.Ifeachor. Impact of Packet Loss Location on Perceived Speech Quality. In *Proceedings of 2nd IP-Telephony Workshop (IPTEL '01)*, pages 114–122, Columbia University, New York, April 2001.

[Li02] Fengyi Li. Measurements of Voice over IP Quality. Master's thesis, KTH, Royal Institute of Technology, Sweden, 2002.

[Lin99] Dong Lin. Real-time voice transmissions over the Internet. Master's thesis, Univ. of Illinois at Urbana-Champaign, 1999.

[ML97] N. F. Maxemchuk and S. Lo. Measurement and interpretation of voice traffic on the Internet. In *Conference Record of the International Conference on Communications (ICC)*, Montreal, Canada, June 1997.

[RG98] ITU-T Recommendation G.114. General Characteristics of International Telephone Connections and International Telephone Circuits: One-Way Transmission Time, Feb. 1998.

[SCFJ96] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 1889, Internet Engineering Task Force, January 1996. http://www.rfc-editor.org/rfc/rfc1889.txt.

Bi-directional Search in QoS Routing

Fernando A. Kuipers and Piet Van Mieghem

Delft University of Technology
Electrical Engineering, Mathematics and Computer Science
P.O Box 5031, 2600 GA Delft, The Netherlands
{F.A.Kuipers,P.VanMieghem}@ewi.tudelft.nl

Abstract. The "bi-directional search method" used for unicast routing is briefly reviewed. The extension of this method unicast QoS routing is discussed and an exact hybrid QoS algorithm HAMCRA that is partly based on bi-directional search is proposed. HAMCRA uses the speed of a heuristic when the constraints are loose and efficiently maintains exactness where heuristics fail. The performance of HAMCRA is simulated.

1 Introduction

One of the classical shortest path algorithms was proposed by Dijkstra [6]. Ever since, many variations of shortest path algorithms have been proposed in the literature [8], [2], mainly based on different proposals for a priority queue [3]. Nearly all proposed shortest path algorithms are designed to find a shortest paths tree rooted at the source to all other nodes. In practice, these algorithms are often only used to find a path between a single source-destination pair. This particular use may be inefficient. The idea to improve the Dijkstra algorithm for source-destination routing was provided in 1960 by Dantzig [4] and concretized in 1966 by Nicholson [14]. Their *bi-directional search* idea consisted of building two shortest path trees alternating between the source and the destination. Bi-directional search can lead to significant savings in time, but unfortunately seems to have been overshadowed by classical shortest path routing. We briefly revisit the concept of bi-directional search and evaluate its application to Quality of Service (QoS) routing.

One of the key issues in QoS routing is how to determine paths that satisfy multiple QoS constraints such as constraints on bandwidth, delay, jitter, and reliability. We focus on this so-called *multi-constrained path* problem and assume that the network-state information is temporarily static, has been distributed throughout the network and is accurately maintained at each node. Before giving the formal definition of the multi-constrained path problem, we first describe the notation that is used throughout this paper.

Let $G(N, E)$ denote a network topology, where N is the set of nodes and E is the set of links. With a slight abuse of notation, we also use N and E to denote the number of nodes and the number of links, respectively. The number of QoS measures is denoted by m. Each link is characterized by an m-dimensional link weight vector, consisting of m non-negative QoS weights ($w_i(u, v)$, $i = 1, ..., m$,

G. Karlsson and M.I. Smirnov (Eds.): QofIS 2003, LNCS 2811, pp. 102–111, 2003.

$(u, v) \in E$) as components. The vector $\boldsymbol{L}$ represents the set of m QoS constraints. The QoS measure of a path can either be additive (e.g., delay, jitter), in which case the weight of that measure equals the sum of the QoS weights of the links defining that path, or the weight of a QoS measure of a path can be the minimum(maximum) of the QoS weights along the path (e.g., available bandwidth and policy flags). Constraints on min(max) QoS measures can easily be treated by omitting all links (and possibly disconnected nodes) that do not satisfy the requested min(max) QoS constraints. Constraints on additive QoS measures cause more difficulties. Multiplicative QoS measures can be transformed into additive measures by taking their logarithm. Hence, without loss of generality, we assume all QoS measures to be additive. The multi-constrained path problem can be defined as follows:

Definition 1 *Multi-Constrained Path (MCP) problem:*
Consider a network $G(N, E)$. Each link $(u, v) \in E$ is specified by m additive QoS weights $w_i(u, v) \geq 0$, $i = 1, ..., m$. Given m constraints L_i, $i = 1, ..., m$, the problem is to find a path P from a source node A to a destination node B such that $w_i(P) \overset{def}{=} \sum_{(u,v)\in P} w_i(u, v) \leq L_i$, for $i = 1, ..., m$.

A path that satisfies all m constraints is often referred to as a feasible path. There may be multiple different paths in the graph $G(N, E)$ that satisfy the constraints. According to **Definition 1**, any of these paths is a solution to the MCP problem. In some cases it might be desirable to retrieve the path with smallest length $l(P)$ from the set of feasible paths. This more difficult problem is called the *multi-constrained optimal path* (MCOP) problem. In general, MCP, irrespective of path optimization, is known to be an NP-complete problem [9]. This explains why the lion's share of proposed QoS routing algorithms are heuristics [11].

The rest of this paper is structured as follows. In Section 2 we discuss multi-constrained bi-directional search. In Section 3 we propose an exact QoS routing algorithm HAMCRA that is partly based on bi-directional search and discuss its complexity. In Section 4 we present the simulation results of HAMCRA and we end in Section 5 with the conclusions.

2 Multi-constrained Bi-directional Search

The basic idea behind bi-directional search originated after observing that the Dijkstra algorithm examines a number of "unnecessary" nodes. Especially when the shortest (sub)path grows towards the destination, it can make an increasingly number of unnecessary scans. To reduce the number of unnecessary scans, it is better to start scanning from the source node as well as from the destination node[1]. In that case a large part of the topology will not be scanned, resulting in a higher efficiency.

[1] In case of a directed graph, the scan-procedure from destination B towards A should proceed in the reversed direction of the links.

When the shortest path has an odd number of hops, the simple idea of alternating between two directions and meeting in the middle is not enough to find the shortest path. We also need to keep track of the minimum path length found sofar. Since we execute the Dijkstra algorithm from two sides, we need two queues Q_A and Q_B. The bi-directional Dijkstra algorithm extracts a node u by alternating between Q_A and Q_B. If a node u has been extracted from Q_A and from Q_B and if the end-to-end path length is smaller or equal to the shortest discovered (but not extracted) shortest path sofar, then we have found the shortest path and can return it by concatenating the two sub-paths from A to u and u to B.

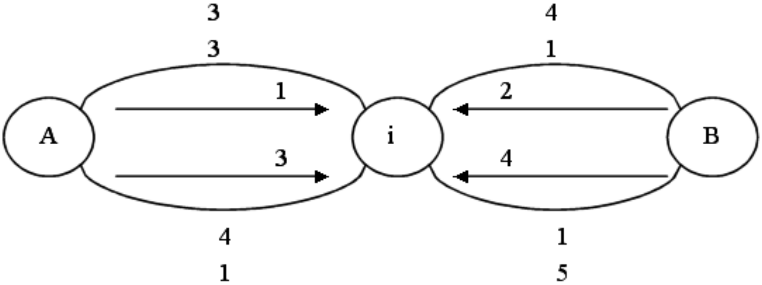


Fig. 1. Example of bi-directional search in two dimensions. The links represent paths, with their corresponding path weight vector.

Extending bi-directional search from $m = 1$ to $m > 1$ dimensions is not a trivial task. The complicating factor is the necessity of a non-linear length function for exact QoS routing [15]. A non-linear length causes that *subsections of shortest paths in $m > 1$ dimensions are not necessarily shortest paths themselves.* If two shortest paths (one originating in source node A and the other in destination node B) in $m > 1$ dimensions meet at an intermediate node, the resulting complete path is not necessarily the shortest path from A to B. We must keep track of the minimum length of the complete paths found sofar. Even if a new complete path exceeds the length of a previously found complete path, that new path cannot be discarded as illustrated in Fig. 1. We use the non-linear path length $l(P) = \max_{i=1,\ldots,m}(\frac{w_i(P)}{L_i})$. The arrows indicate the order of arrival of these sub-paths at node i. Once the first two sub-paths have arrived at node i, we have our first complete path with weight vector $\boldsymbol{w}(P) = (7{,}4)$. If the constraints are $\boldsymbol{L} = (10{,}10)$ then the length of this path equals 0.7. Once the third path arrives at node i, it makes a complete path with the second path, with total length 0.8. However, we cannot remove this sub-path, because combined with path 4, it forms the shortest path with link weight vector (5,6) and length 0.6. This example also indicates that we will have to connect at each intermediate node with $k \geq 1$ paths (even if they violate the constraints), where k can grow exponentially with the number of nodes N.

These problems in multiple dimensions complicate the determination of an efficient stop criterion for the MCOP problem. The bi-directional search in $m > 1$ dimensions has more potential for the MCP problem, where the routing algorithm can stop as soon as a complete path obeys the constraints. Unfortunately, when no feasible path is present in the graph, bi-directional search may require twice the time of uni-directional search. Again, an efficient stop criterion that foresees that there is no feasible path present seems difficult to find.

3 A Hybrid QoS Algorithm

In the previous section we have argued that the use of bi-directional search has a higher potential for MCP than for MCOP. This need not be a disadvantage. When the constraints are loose and many feasible paths exist, then optimization is not very important and precious CPU time could be wasted in computing the optimal path. In a heavily loaded network, the need for optimization increases. Fortunately, under heavy load, the number of feasible paths is expected to be small, in which case the MCP problem approximates the MCOP problem.

Although applying bi-directional search to the MCP problem is possible, finding a clear stop criterion when no feasible paths are present is still problematic. This difficulty suggests to deviate from the alternating bi-directional search to a hybrid algorithm that uses concepts of bi-directional search. We have named our hybrid QoS algorithm HAMCRA, the Hybrid Auguring Multiple Constraints Routing Algorithm. HAMCRA is exact in solving the MCP problem, but is not always exact in solving the MCOP problem. The rest of this section will present HAMCRA, give its worst-case complexity and show that it is indeed exact in solving the MCP problem.

3.1 HAMCRA

HAMCRA is composed of the exact algorithm SAMCRA, the Self-Adaptive Multiple Constraints Routing Algorithm, [15] and its heuristic predecessor TAMCRA, the Tunable Accuracy Multiple Constraints Routing Algorithm, [5]. Both SAMCRA and TAMCRA are based on three fundamental concepts:

1. In order for any QoS algorithm to be exact, we must use a non-linear length function, such as
$$l(P) = \max_{i=1,\ldots,m} \left(\frac{w_i(P)}{L_i} \right) \tag{1}$$
where $w_i(P)$ is the i-th weight of path P and L_i is the i-th constraint. If $l(P) > 1$ then path P is not feasible.
2. If a non-linear length function like (1) is used, then the subsections of shortest paths in multiple dimensions are not necessarily shortest paths themselves. It may therefore be necessary to store in the computation more (sub-)paths then just the shortest. SAMCRA and TAMCRA use the k-shortest path approach [7], where in TAMCRA k is predefined by the user and in SAMCRA k is adapted in the course of the computation to assure that the exact shortest path is found.

3. Both TAMCRA and SAMCRA only consider non-dominated paths, where a path P is called non-dominated if there[2] does not exist a path P^* for which $w_i(P^*) \leq w_i(P)$ for all link weight components i except for at least one j for which $w_j(P^*) < w_j(P)$.

Since HAMCRA is composed of SAMCRA and TAMCRA, it is also based on their three concepts. In HAMCRA, first the TAMCRA algorithm is executed with a queue-size $k = 1$ from the destination node to all other nodes in the graph. This is the similarity with bi-directional search, because we also scan from the destination node. However we do not alternate between the source and the destination. We can use TAMCRA with $k > 1$, which will lead to a better accuracy at the cost of increased complexity (of TAMCRA). The running time of TAMCRA with $k = 1$ is comparable to that of the Dijkstra algorithm. At each node, the path weight vector found by TAMCRA from that node to the destination is stored. These values will later be used to predict the end-to-end path length. If TAMCRA has found a path within the constraints between the source and the destination, HAMCRA can stop and return this path. If TAMCRA does not find a feasible path, HAMCRA continues by executing the SAMCRA algorithm from the source node. The difference between HAMCRA and SAMCRA is that HAMCRA uses the information obtained by TAMCRA and only stores predicted end-to-end lengths in the queue instead of the real lengths of the sub-paths. The idea for using predictions was originally presented in [10]. The predicted end-to-end length is found by summing the real weights of a path from source A to the intermediate node u with the weights of the TAMCRA path from u to the destination B. The algorithm continues searching in this way until a feasible path from A to B is found or until the queue is empty.

HAMCRA uses a fourth concept to reduce the search-space, namely that of lower bounds (LB) [13]. The LB concept uses the property that if $\sum_{i=1}^{m} \alpha_i w_i(P) > \sum_{i=1}^{m} \alpha_i L_i$, $\alpha_i \geq 0$, then path P is not feasible. By computing the shortest, according to the linear length function $l(P) = \sum_{i=1}^{m} \alpha_i w_i(P)$, paths $P^*_{B\to n}$ rooted at the destination B to each node n in the graph $G(N, E)$, we obtain the lower bounds $w_i\left(P^*_{B\to n}\right)$. These lower bounds can be used to check if a path $P_{A\to n}$ from source A to node n can possibly obey the constraints: if $\sum_{i=1}^{m} \alpha_i \left(w_i(P_{A\to n}) + w_i\left(P^*_{B\to n}\right)\right) > \sum_{i=1}^{m} \alpha_i L_i$, then path $P_{A\to n}$ need not be considered further. For the simulations we have chosen to compute via Dijkstra the shortest path tree rooted at the destination to each node n in the graph $G(N, E)$ for each of the m link weights separately. For measure j, the Dijkstra shortest path agrees with $\sum_{i=1}^{m} \alpha_i w_i(P)$, where $\alpha_i = 0$ for $i = 1, ..., m$ except $\alpha_j = 1$. Hence, for each of the m link weight components, the lowest value from the destination to a node $n \in N$ is stored in the queue of that node n.

Note that the LB concept is also based on (lower bound) predictions for the end-to-end path length. We could therefore also use the LB predictions instead

[2] If there are two or more different paths between the same pair of nodes that have an identical weight vector, only one of these paths suffices. We therefore assume one path out of the set of equal weight vector paths as being non-dominated and regard the others as dominated paths.

of the TAMCRA predictions. Because HAMCRA uses TAMCRA to predict the end-to-end path length, this prediction (if erroneous) could be larger than the real end-to-end path length. It may then happen that HAMCRA extracts a non-shortest path first. Therefore HAMCRA using TAMCRA cannot guarantee a solution to the MCOP problem. If $l_{predicted}(P) \leq l_{actual}(P)$ as is the case with LB predictions, a solution to MCOP can be guaranteed. Unfortunately, simulations (see Fig. 3) have shown that such predictions are usually not as good as the TAMCRA predictions, leading to an increased running time.

3.2 Worst-Case Complexity of HAMCRA

The total worst-case complexity of HAMCRA is constructed as follows. Executing heap-optimized Dijkstra m times leads to $mO(N \log N + E)$ and m times computing a length of a path leads to $mO(mN)$. Executing TAMCRA with $k = 1$ requires $O(N \log N + mE)$. The search from the destination adds $O(mN \log N + mE + m^2N)$, which is polynomial in its input. The "SAMCRA" search from the source adds $O(kN \log(kN) + k^2mE)$ [15]. Combining these contributions yields a total worst-case complexity of HAMCRA with $k = k_{\max}$ of $O(mN \log N + mE + m^2N + kN \log(kN) + k^2mE)$ or

$$C_{HAMCRA} = O(kN \log(kN) + k^2mE) \qquad (2)$$

where m is fixed and $m \leq k = k_{\max}$ and $k_{\max}$ is an upper bound on the number of paths in the search-space. For a single constraint ($m = 1$ and $k = 1$), this complexity reduces to the complexity of the Dijkstra algorithm $C_{Dijkstra} = O(N \log N + E)$. For $m > 1$, the complexity becomes NP-complete, since k can grow exponentially with N.

3.3 Proof that HAMCRA Is Exact

The proof that HAMCRA is exact in solving the MCP problem depends on the validity of the search-space reducing techniques. Obviously, if TAMCRA finds a feasible path at the beginning, then the MCP problem is exactly solved. For the case that this first step fails, we summarize the different steps in HAMCRA:

1. Paths with length $l(P) > 1$ need not be examined.
2. If in the k-shortest path algorithm the value of k is not restricted, HAMCRA returns all possible paths ordered in length between source and destination.
3. If, for all i, $w_i(P_1) \leq w_i(P_2)$, then $w_i(P_1) + u_i \leq w_i(P_2) + u_i$ for any u_i. For all definitions of length $l(.)$ satisfying the vector norm criteria (such as (1)) there holds that $l(\boldsymbol{w}(P_1) + \boldsymbol{u}) \leq l(\boldsymbol{w}(P_2) + \boldsymbol{u})$ for any vector $\boldsymbol{u}$. Hence, we certainly know that P_2 will never be shorter than P_1. Hence, dominated paths need not be stored in the queue.
4. If $\sum_{i=1}^{m} \alpha_i \left(w_i(P_{A\to n}) + w_i\left(P^*_{B\to n}\right)\right) > \sum_{i=1}^{m} \alpha_i L_i$, then sub-path $P_{A\to n}$ can never be complemented with a path $P^*_{B\to n}$ to satisfy the constraints $\boldsymbol{L}$. Hence, the sub-path $P_{A\to n}$ should not be considered further.

5. Finally, the insertion/extraction policy of the nodal queues uses a predicted end-to-end length instead of the real length of a (sub)-path. However, since the predicted end-to-end length and the real end-to-end length of a complete path between source and destination are the same, this path is returned if $l(P) \leq 1$ or removed if $l(P) > 1$. Thus, only feasible end-to-end paths will be examined and exactness is guaranteed.

4 Performance Evaluation of HAMCRA

In this section a performance evaluation of HAMCRA is presented based on simulation results. We have simulated on three classes of graphs, namely the class of random graphs $G_p(N)$ [1] with link-density $p = 0.2$, the class of Internet-like power law graphs with power $\tau = 2.4$ in the nodal degree distribution $\Pr[d = k] = k^{-\tau}$ and the extremely regular class of square two-dimensional lattices. We have performed two types of simulations.

The first type of simulations consisted of generating, for each simulation run, 10^4 graphs with all link weights independent uniformly distributed in the range (0,1]. In each graph, based on multiple constraints, we computed a path between two nodes (A and B) in the graph and stored the maximum queue-size k that was used. In all classes of graphs, the source A and destination B were randomly chosen. For the class of two-dimensional lattices, we also simulated with A chosen in the upper left corner and B in the lower right corner. We refer to this "worst-case" setting as Lattice 2 and to the case where the source and destination nodes are chosen randomly as Lattice 1. The constraints are chosen very strict, such that only one path can obey these constraints. In this case the MCP problem equals the more difficult MCOP problem.

The second type of simulations consisted of generating only one two-dimensional lattice with A chosen in the upper left corner and B in the lower right corner and then finding a path in this graph, subject to different constraints. To examine the influence of the constraints, we simulated with 10^4 different constraint vectors per topology.

4.1 Simulation Type 1

Fig. 2 presents the results as a function of the number of nodes N. The expected queue-size $E[k]$ and the variance in queue-size $var[k]$ are very close to one for the class of random graphs. Similar results were also obtained with SAMCRA [15], suggesting that exact QoS routing in the class of random graphs with uniformly distributed link weights is easy. The class of power law graphs also has a moderate $E[k]$, although the variability $var[k]$ is larger than in the class of random graphs. This was expected because the power law graphs are less random than the random graphs. The class of two-dimensional lattices gives the worst performance, especially when the source and destination nodes are furthest apart. The two-dimensional lattices have a large expected hopcount and many paths between source and destination, making this class of graphs hard to

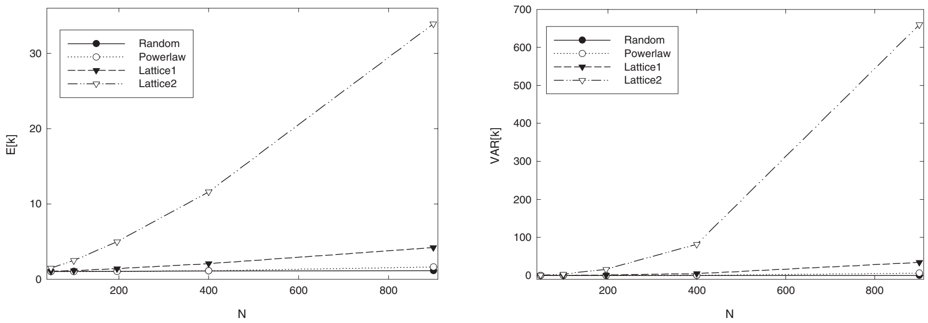


Fig. 2. The queue-size k for different topologies as a function of the number of nodes N.

solve. This was also observed and motivated in [12]. Finally, we have simulated the performance of HAMCRA as a function of the number of constraints m. The (undisplayed) results show that the queue-size k increases with m, but that this increase diminishes if m gets large enough. In fact, if $m \to \infty$, $E[k]$ will be exactly one as proved in [15].

4.2 Simulation Type 2

In this subsection we present the results for the class of two-dimensional lattices with A chosen in the upper left corner and B in the lower right corner. We believe that this class of two-dimensional lattices represents worst-case topologies. We have simulated on a single lattice topology with 100 values for constraint L_1 and 100 values for constraint L_2, leading to a total of 10^4 computations per simulation. With our choice of the constraints it can occur that no feasible path exists. We have performed simulations for different levels of link correlation ρ. We only observed a high complexity for an extremely negative correlation. The results indicate that the values of the constraints and the correlation between the link weights can have a serious impact on the complexity. If the link weights are negatively correlated and the constraints are close to the weights of the m-dimensional shortest paths, the complexity is highest. The impact of correlation and constraints on the complexity of QoS routing has been discussed in [12]. Fig. 3 presents our results for HAMCRA with different predictive length functions. Our results show that HAMCRA with TAMCRA predictions has a good complexity if a feasible path is present and if the constraints are not too strict. The complexity is better than with Dijkstra lower bound predictions. Unfortunately, the complexity may be large if no feasible path is present or if only one path can obey the constraints. In the worst-case scenario with $\rho = -1$, it is possible to reduce the complexity by including lower bounds based on $\frac{1}{m}\sum_{i=1}^{m}\frac{w_i(P)}{L_i} \leq 1$. In this case we could have verified in polynomial time if a feasible path existed. If exactness for MCOP is required, we recommend to use the linear length function (see Section 3.1) $l(P) = \frac{1}{m}\sum_{i=1}^{m}\frac{w_i(P)}{L_i}$ for search-space reduction as well as end-to-end path length prediction (instead of TAMCRA).

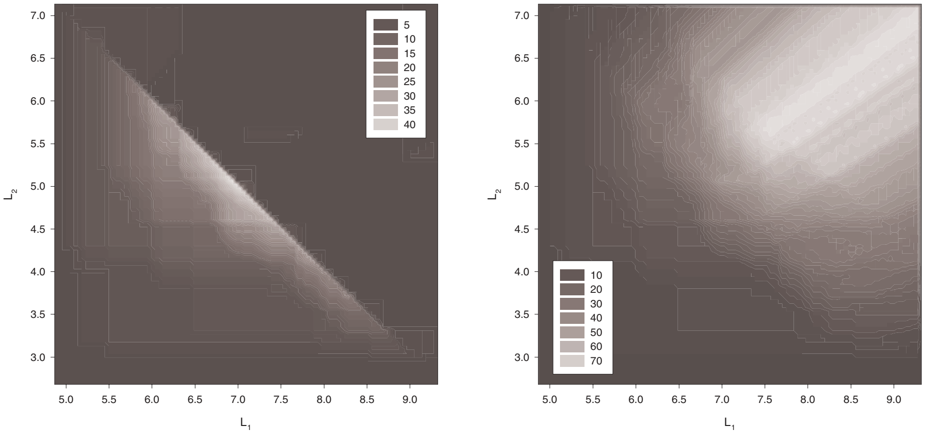


Fig. 3. The expected queue-size as a function of the constraints L_1 and L_2, whit $\rho = -1$, $N = 49$. (left) queue-size HAMCRA with TAMCRA predictions (right) queue-size HAMCRA with Dijkstra predictions.

5 Conclusions

In this paper we have revisited the use of bi-directional search in unicast routing. This method is powerful in (one-dimensional) unicast routing. To our knowledge an extension of one-dimensional bi-directional search towards multiple dimensions has never been examined. We have filled that gap in this paper and have shown that such an extension is not trivial. Some difficulties appear especially when the multi-dimensional shortest path is needed or when no feasible path is present. To avoid these difficulties, we have proposed and evaluated HAMCRA. This hybrid algorithm exactly solves the multi-constrained path (MCP) problem and is composed of the exact QoS algorithm SAMCRA and the heuristic QoS algorithm TAMCRA. HAMCRA uses TAMCRA to quickly follow a feasible path and uses SAMCRA to maintain its exactness for the MCP problem. Simulations with HAMCRA show that HAMCRA quickly finds a feasible path for nearly the entire range of feasible constraints. The complexity of HAMCRA can only be high when the constraints are closely situated around the weights of the multi-dimensional shortest paths, the link weights are negatively correlated and we have a specific topology (like the two-dimensional lattice). We believe that in practice this situation is unlikely to occur and that HAMCRA is expected to have a low complexity. If our assumption holds, then it is pointless to consider heuristics for QoS routing that cannot even guarantee QoS requirements to be met.

References

1. B. Bollobas, *Random Graphs*, Cambridge University Press, second edition, 2001.
2. B.V. Cherkassky, A.V. Goldberg and T. Radzik, "Shortest paths algorithms: theory and experimental evaluation", Mathematical Programming, Series A, no. 73, pp. 129-174, 1996.
3. B.V. Cherkassky, A.V. Goldberg and C. Silverstein, "Buckets, heaps, lists and monotone priority queues", SIAM Journal on Computing, no. 28, pp. 1326-1346, 1999.
4. G. Dantzig, "On the shortest route through a network", Mgmt. Sci., vol. 6, pp. 187-190, 1960.
5. H. De Neve, and P. Van Mieghem, "TAMCRA: A Tunable Accuracy Multiple Constraints Routing Algorithm", Computer Communications, Vol. 23, pp. 667-679, 2000.
6. E.W. Dijkstra, "A note on two problems in connexion with graphs", Numerische Mathematik, no. 1, pp. 269-271, 1959.
7. D. Eppstein, "Finding the k Shortest Paths", SIAM J. Computing, 28(2):652-673, 1998.
8. G. Gallo and S. Pallottino, "Shortest Paths Algorithms", Annals of Operations Research, no. 13, pp. 3-79, 1988.
9. M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, Freeman, San Francisco, 1979.
10. P. Hart, N. Nilsson and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths", IEEE Transactions on Systems Science and Cybernetics, 2:100-107, 1968.
11. F.A. Kuipers, T. Korkmaz, M. Krunz and P. Van Mieghem, "An Overview of Constraint-Based Path Selection Algorithms for QoS Routing", IEEE Communications Magazine, vol. 40, no. 12, December 2002.
12. F.A. Kuipers and P. Van Mieghem, "The Impact of Correlated Link Weights on QoS Routing", Proc. of IEEE INFOCOM 2003, April 2003.
13. G. Liu and K.G. Ramakrishnan, "A*Prune: An Algorithm for Finding K Shortest Paths Subject to Multiple Constraints", Proc. of IEEE INFOCOM 2001, April 2001.
14. T. Nicholson, "Finding the shortest route between two points in a network", the computer journal, vol. 9, pp. 275-280, 1966.
15. P. Van Mieghem, H. De Neve, and F.A. Kuipers, "Hop-by-hop Quality of Service Routing", Computer Networks, vol. 37. No 3-4, pp. 407-423, 2001.

The NAROS Approach for IPv6 Multihoming with Traffic Engineering

Cédric de Launois*, Olivier Bonaventure, and Marc Lobelle

Université catholique de Louvain
Department of Computing Science and Engineering
{delaunois,bonaventure,ml}@info.ucl.ac.be
http://www.info.ucl.ac.be

Abstract. Once multihomed, an IPv6 site usually wants to engineer its interdomain traffic. We propose that IPv6 multihomed hosts inquire a so called "Name, Address and ROute System" (NAROS) to determine the source and destination addresses to use to contact a destination node. By selecting these addresses, the NAROS server roughly determines the routing. It thereby provides features like traffic engineering and fault tolerance, without transmitting any BGP advertisement and without impacting on the worldwide routing table size. The performance of the NAROS server is evaluated by using trace-driven simulations. We show that the the load on the NAROS server is reasonable and that we can obtain good load-balancing performances.

Keywords: multihoming, traffic engineering, IPv6, BGP.

1 Introduction

The size of BGP routing tables in the Internet has been growing dramatically during the last years. The current size of those tables creates operational issues for some Internet Service Providers and several experts [1] are concerned about the increasing risk of instability of BGP.

Part of the growth of the BGP routing tables [2] is due to the fact that, for economical and technical reasons, many ISPs and corporate networks wish to be connected via at least two providers to the Internet. Nowadays, at least 60% of those domains are connected to two or more providers [3].

Once multihomed, a domain will usually want to engineer its interdomain traffic to reduce its costs. Unfortunately, the available interdomain traffic engineering techniques [4] are currently based on the manipulation of BGP attributes which contributes to the growth and the instability of the BGP routing tables.

It can be expected that IPv6 sites will continue to be multihomed and will also need to engineer their interdomain traffic. Although several solutions to the IPv6 multihoming problem have been discussed within the IETF [5,6,7,8,9,10,11], few have addressed the need for interdomain traffic engineering. We propose and

* Supported by a grant from FRIA (Fonds pour la Formation à la recherche dans l'Industrie et dans l'Agriculture, rue d'Egmont 5 - 1000 Bruxelles, Belgium).

G. Karlsson and M.I. Smirnov (Eds.): QofIS 2003, LNCS 2811, pp. 112–121, 2003.

evaluate in this paper an innovative host-centric solution to the IPv6 multihoming problem. This solution allows sites to engineer their incoming and outgoing interdomain traffic without any manipulation of BGP messages.

In the following section, we briefly present the technical and economical reasons for multihoming in the Internet, and situate our solution among other proposed multihoming solutions. Next, we describe the NAROS architecture and explain how it supports multihoming and traffic engineering. Finally, we use trace-driven simulations to evaluate the performance of our solution.

2 Multihoming Issues

IPv6 multihoming solutions are significantly different from IPv4 ones because they must allow the routing system to scale better. Morever, the IPv6 address space is much larger, which gives more freedom when designing multihoming. An IPv6 host may have several global addresses. Paradoxically this can help in reducing the BGP table sizes but it requires that hosts correctly handle multiple addresses. Requirements for IPv6 multihoming are stronger and multiple [12]. In this paper, we essentially focus on the following requirements.

Fault Tolerance. Sites connect to several providers mainly to get fault tolerance. A multihoming solution should be able to insulate the site from both link and ISP failure.

Route Aggregation. Every IPv6 multihoming solution is required to allow route aggregation at the level of their providers [1], [12]. This is essential for the scalability of the interdomain routing system.

Source Address Selection. A multihomed IPv6 host may have several addresses, assigned by different providers. When selecting the source address of a packet to be sent, a host could in theory pick any of these addresses. However, for security reasons, most providers refuse to convey packets with source addresses outside their address range. So, the source address selected by a host also determines the upstream provider used to convey the packet. This has a direct impact on the flow of traffic. Moreover, if a host selects a source address belonging to a failed provider, the packet will never reach its destination. Thus, a mechanism must be used to select the most appropriate source address.

Destination Address Selection. When a remote host contacts a multihomed host, it must determine which destination address to use. The destination address also determines the provider used. If a provider of the multihomed site is not available, the corresponding destination address cannot be used to reach the host. So we must make sure that an appropriate destination address is always selected.

Traffic Engineering. A multihomed site should be able to control the amount of inbound and outbound traffic exchanged with its providers.

ISP Independence. It is desirable that a multihoming solution can be set up independently without requiring cooperation of the providers.

2.1 Related Work

All current IPv6 multihoming approaches allow route aggregation and provide at least link fault tolerance. A summary of desired features provided by various multihoming solutions is provided in table 1. The solutions and their features are detailed in a survey on multihoming mechanisms [5].

Table 1. Features provided by current IPv6 multihoming solutions.

Feature	*[8]*	*[9]*	*[7]*	*[10]*	*[6]*	*[13]*	*[11]*	*NAROS*
Link fault tolerance	x	x	x	x	x	x	x	x
ISP fault tolerance			x	x	x	x	x	x
Stable configuration in case of long term failure			x	x		x	x	x
Explicit ISP selection					x	x		x
Allows load sharing	x	x			x	x	x	x
Explicit traffic engineering								x
Solve source address selection problem							x	x
Transport-layer survivability	x	x		x	x			
Site-ISP independency			x	x	x	x	x	x
Inter ISP independency		x	x	x	x	x	x	x
No changes for Internet routers	x	x	x	x	x	x	x	x
No changes for site exit routers	x	x	x	x	x			x
No changes for hosts	x	x	x					
No changes for correspondent nodes	x	x	x		x	x	x	x
No new security issues	x	x	x			x	x	x
No need of tunnels			x	x	x	x		x
No modification to current protocols	x	x	x		x			x
No new protocol	x	x	x	x	x	x	x	
Valid for both TCP and UDP	x	x	x		x	x	x	x

The first two approaches [8], [9] use tunnels and/or backup links with or between the providers. The third solution [7] uses the Router Renumbering [14] and Neighbor Discovery [15] protocols to deprecate addresses in case of ISP failure. The fourth approach [10], proposes to modify the TCP protocol to preserve active TCP connections. The fifth solution uses the IP mobility mechanisms to switch between delegated addresses in case of failure [7], [6]. The sixth approach [13] consists in enhancing the Neighbor Discovery protocol to help the hosts in selecting the appropriate site exit routers. The solution proposed in [11] defines new ICMP redirection messages to inform a host of the site exit router to use.

The last approach is the NAROS approach presented in this paper. It relies on the utilization of several IPv6 addresses per host, one from each provider. The basic principle of NAROS is that before transmitting packets, hosts contact the NAROS service to determine which IPv6 source address they should use to reach a given destination.

This approach has never been developped, although briefly suggested in [11]. To the best of our knowledge, this is the first approach that explicitly allows load-balancing and traffic engineering in IPv6 multihoming sites.

3 The NAROS Service

Figure 1 illustrates a standard multihomed site. Suppose three Internet Service Providers (ISPA, ISPB and ISPC) provide connectivity to the multihomed site. The site exit router connecting with ISPA (resp. ISPB and ISPC) is RA (resp. RB and RC). Each ISP assigns a site prefix. The prefixes (PA, PB and PC), together with a subnet ID (SA, SB or SC) are advertised by the site exit routers and used to derive one IPv6 address per provider for each host interface.

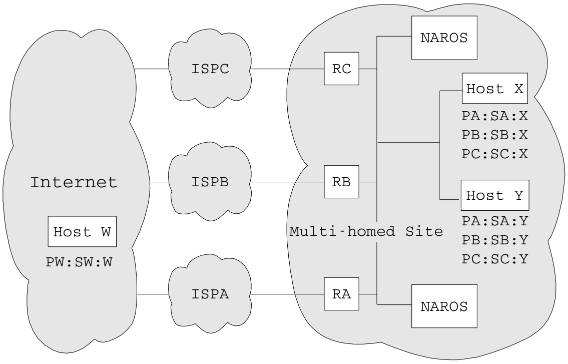


Fig. 1. A multihomed site connected with three providers.

In the NAROS architecture, the site advertises ISPA addresses only to ISPA, and ISPA only announces its own IPv6 aggregate to the global Internet.

Since each host has several IPv6 addresses, it must decide which address to use when transmitting packets. The basic principle of our solution is to let the NAROS service manage the selection of the source addresses. This address selection will influence how the traffic flows through the upstream providers and a good selection method will allow the site to engineer its interdomain traffic.

We now consider in details how NAROS addresses four main issues: source and destination address selection, fault-tolerance and traffic engineering.

Source Address Selection. When a host initiates a connection with a correspondent node, it must determine the best source address to use among its available addresses. The source address selection algorithm described in [16] already provides a way to select an appropriate address. However, this selection is arbitrary when a host has several global-scope IPv6 addresses as in our case.

The principle we propose is that the host asks the NAROS service which source address to use. It complements in this way the default IPv6 source address selection algorithm [16].

Many factors could possibly influence the selection process, such as the current loads and states of the links or administrative policies. A NAROS server could also rely on informations contained in BGP tables, e.g. the path length towards the destination.

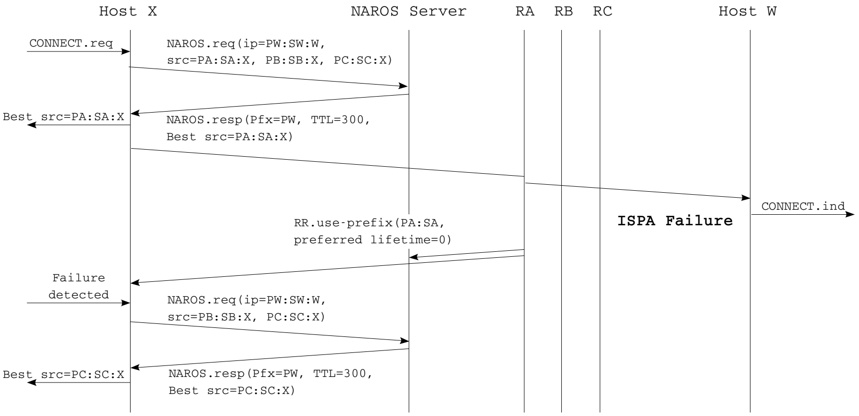


Fig. 2. Basic NAROS scenario example.

In its simplest form, the basic NAROS service is independent from any other service. A NAROS server does not maintain state about the internal hosts. It is thus possible to deploy several NAROS servers in anycast mode inside a site for redundancy or load-balancing reasons. A NAROS server can also be installed on routers such as the site exit routers. The NAROS protocol runs over UDP and contains only two messages: NAROS request and NAROS response [17].

The first message is used by a client to request its connection parameters. The parameters included in a NAROS request are at least the destination address of the correspondent node and the source addresses currently allocated to the client. The NAROS server should only be contacted when the default source address selection procedure [16] cannot select the source address.

The NAROS response message is sent by a NAROS server and contains the connection parameters to be used by the client. The parameters include at least the selected best source address, a prefix and a lifetime. It tells that the client can use the selected source address to contact any destination address matching the prefix. These parameters remain valid and can be cached by the client during the announced lifetime.

The upper part of figure 2 shows an example of how the NAROS messages and parameters are used. The exact format of the NAROS message is outside the scope of this paper. When Host X sends its first packet to remote Host W (PW:SW:W), it issues a NAROS request in order to obtain the source address to use to reach Host W. Upon receipt of the request, the NAROS server identifies the prefix PW associated with Host W and selects for example PA:SA:X as the best source address. The prefix can be determined arbitrarily, e.g. using the /8 prefix corresponding to the destination address. Another solution is to extract from a BGP table the prefix associated with the destination. The server then indicates the lifetime (e.g. 300 seconds) of these parameters in the NAROS response message.

After having processed the reply, Host X knows that it can use PA:SA:X to reach any destination inside prefix PW, including Host W. The selected source address should be used for the whole duration of the flow, in order to preserve the connection. If new TCP or UDP connections for the same destination are initiated before the announced lifetime expires, the client can use the cached parameter. Otherwise the host must issue a new NAROS request and it may get a different source address for the same destination. By using appropriate values for the lifetime and the prefix in the NAROS response, it is possible to reduce the number of NAROS requests sent by hosts as will be shown in section 4.

Destination Address Selection. A second case is when Host W on the Internet needs to contact Host X in the multihomed site. It first issues a DNS request. The DNS server of the multihomed site could reply with all the addresses associated to Host X. At worst, Host W will try the proposed addresses one by one. Eventually, a connection will work.

Fault Tolerance. A third problem to consider is when one of the upstream providers fails. As in the solution described in [7], [11], the site exit routers use router advertisement messages to communicate to hosts the available prefixes [15]. When a provider crashes, the site exit router connected to this provider detects the event and advertises a null preferred lifetime for that prefix. A client can take this event into account by immediately asking new parameters to the NAROS server. More generally, a host can ask updated parameters each time it detects a failure which affects one of its communications. Once the new source address is known, IP mobility or other mechanisms can be used in order to preserve the established TCP connections [6], [10].

In the lower part of fig. 2, consider for example that ISPA becomes unavailable. The site exit router connected to ISPA detects the failure and advertises a null preferred lifetime for prefix PA. The NAROS server immediately takes this advertisement into account and future NAROS replies will not contain this prefix. Host X will also receive this advertisement. The standard effect is that it should no longer use this source address for new TCP or UDP flows. If Host X is currently using a deprecated address, it can issue a new NAROS request to choose among its other available source addresses. The host can then use IP mobility mechanisms to switch to the new source address in order to maintain its connection alive.

Traffic Engineering. When a host selects a source address, it also selects the provider through which the packets will be sent. Since the source address to use is selected by NAROS, this can naturally be used to perform traffic engineering. For example, in order to balance the traffic among the three providers in figure 1, a NAROS server can use a round-robin approach. For each new NAROS request, the server selects another provider and replies with the corresponding source address. Except when a provider fails, this source address, and thus the upstream provider, remains the same for the whole duration of the flow.

NAROS Advantages. Beside the above functionalities, the NAROS approach has several advantages. First, the NAROS service can be set up independently from the providers. A provider only delegates a prefix to the site. This makes the solution applicable for small sites such as enterprise networks. Next, since routes to addresses delegated by one provider are not announced to other providers, full route aggregation is possible. Another advantage is that the solution allows traffic engineering without injecting any information in the internet routing system. Moreover, the NAROS service can easily support unequal load distribution, without any additionnal complexity. Next, NAROS allows the providers to perform ingress filtering, which benefits to security. Finally, changes are limited to hosts inside the multihomed site. Legacy hosts are still able to work, but they cannot benefit from all the NAROS advantages.

4 Performance Evaluations

The NAROS protocol depends on choosing two base parameters: the size of the prefix associated with the destination and its lifetime. We now evaluate the impact of these parameters on the cache size of the hosts, the number of NAROS requests and consequently the server load, and finally the load-balancing quality.

The evaluation of the NAROS service presented in this section is based on a real traffic trace [18]. This trace is a flow-trace collected during 24 hours on November 18, 2002 and contains all the interdomain traffic of a university site. 7687 hosts were active in the network and the volume of traffic exchanged is about 200 GB (18.8 Mb/s in average). The trace contains information about 322 million packets forming more than 17.5 million TCP and UDP flows. The average flow lifetime is 12 seconds. We evaluated the NAROS protocol with IPv4 because no significant IPv6 network is available today.

The first performance parameter to consider is the size of the NAROS cache maintained by the hosts. We evaluate the impact of the prefix length used in the NAROS replies on the cache size of the hosts. For example, if a host requests for destination 1.2.3.4, the NAROS may reply with a /24 prefix, meaning that the parameters are valid for all addresses in 1.2.3.0/24. It may also extract the corresponding prefix from a BGP table. In this case, the prefix length is variable because it depends on the prefix matched in the BGP table for this destination.

Figure 3 shows on a log-log scale $p_1(x)$: the percentage of hosts having a maximum cache size greater than x. It shows for example that if we use /24 prefixes as in the example, the cache size remained below 100 entries during the whole day for 95% of the hosts. We used a lifetime of 300s. The hosts which present the largest cache size were found to be either compromized machines sending lots of probes or very active peer-to-peer clients.

The use of lower lifetimes (not shown) yields to smaller cache sizes. A consequence of this figure is that small prefix lengths and low lifetime contribute to small cache sizes. A value of 300s seems appropriate for the studied site.

We also evaluate the impact of the lifetime on the cache performance. A good cache performance is necessary to limit the number of NAROS requests that a host issues. Figure 4 evaluates the percentage of cache hits versus the lifetime in seconds. It shows that the cache hit ratio is higher when longer lifetime or smaller prefix lengths are used. However, we get no significant improvement by

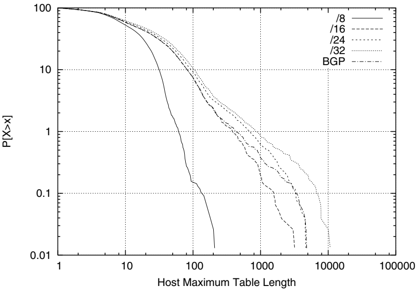


Fig. 3. NAROS Cache size for a lifetime of 300s and various prefix lengths.

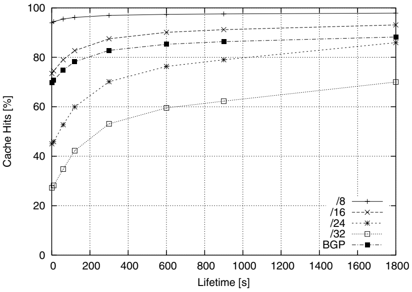


Fig. 4. Impact of the response lifetime on the cache performance.

using lifetimes longer than about 300 seconds. We also see that the lifetime has little impact on the cache hit ratio when /8, /16 or BGP prefixes are used.

A second element to consider is the server load. Figure 5 shows on a log-log scale $p_2(x)$: the percentage of hosts issuing more NAROS requests than x, during the whole day. We use here a lifetime of 300s and simulate various prefix lengths. Figure 5 shows that when BGP prefixes are used, 90% of the hosts issue less than about 300 requests during the whole day. The resulting server load is illustrated in figure 6. This load is proportional to the number of host and essentially follows the traffic load. The load average is about 35 requests per second, which is still reasonable. In comparison, this is no more than the number of DNS requests coming from the Internet and addressed to the site studied. The bandwidth overhead of the NAROS approach is evaluated to about 0.35%.

We now compare the performance of the NAROS load-balancing technique with the best widely used load-balancing technique which preserves packet ordering, i.e. CRC16 [19]. We focus on the common case of load-balancing between two outgoing links of the same capacity. For the NAROS load-balancing, we use a round-robin approach, i.e. a new flow is alternatively assigned to the first and the second links. CRC16 is a direct hashing-based scheme for load-balancing where the traffic splitter uses the 16-bit Cyclic Redundant Checksum algorithm as a hash function to determine the outgoing link for every packet. The index of the outgoing link is given by the 16-bit CRC checksum of the tuple (source IP, destination IP, source port, destination port, protocol number), modulo the number of links. CRC16 is often used on parallel links from the same router.

We measure the performance of the load-balancing by looking at the deviation from an even traffic load between the two links. Let $load_1$ and $load_2$ be respectively the traffic load of the first and the second link. We define the deviation as a number in $[-1, 1]$ computed by $(load_1 - load_2)/(load_1 + load_2)$. A null deviation means that the traffic is balanced, while a deviation of 1 or -1 means that all the traffic flows through the first or the second link. Fig. 7 compares the deviation in percent of the NAROS and CRC16 load-balancing. For NAROS, we used BGP prefixes and a lifetime of 300s. We see that the NAROS solution is able to provide load-balancing as good as the best current static load-balancing mechanism. Fig. 8 compares the NAROS load-balancing quality for a lifetime

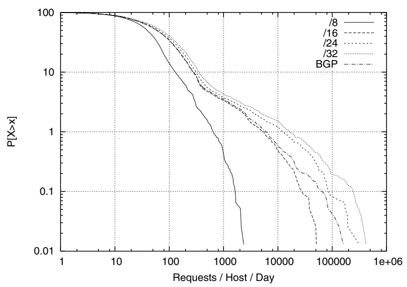


Fig. 5. Number of requests per host during the day, with a lifetime of 300s.

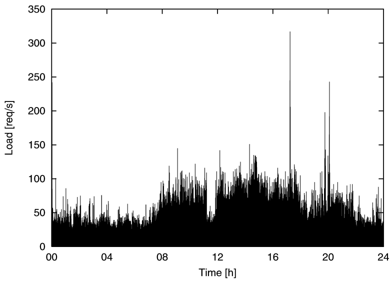


Fig. 6. Server load using BGP and a lifetime of 300s.

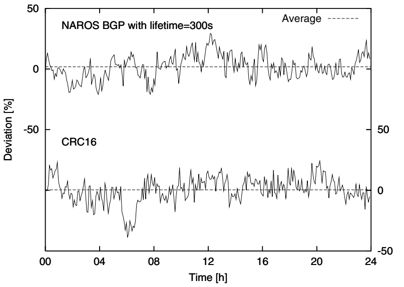


Fig. 7. NAROS and CRC16 load balancing comparison.

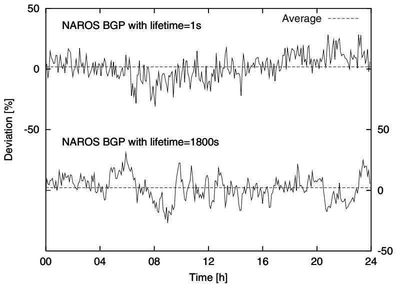


Fig. 8. NAROS load balancing with lifetime of 1s and 1800s.

of 1s and a lifetime of 1800s. It shows that the quality of the load-balancing is better when short lifetimes are used, at the expense of a larger server load.

5 Conclusion

In this paper, we have proposed a solution which provides fault-tolerance and traffic engineering capabilities without impacting on the Internet routing tables. When a host needs to communicate with a remote host, it contacts its NAROS server to determine the best source IPv6 address to use. The NAROS server does not maintain any per-host state, can easily be deployed as an anycast service inside each site, and can be set up independently from the providers. It allows to indirectly, but efficiently, engineer the interdomain traffic, without manipulating any BGP attribute. Changes are limited to hosts inside the multihomed site. Legacy hosts are still able to work, even if they cannot benefit from site-multihoming. We have also shown that the load on the NAROS server was reasonable and that, when used to load-balance the outbound traffic between two providers, the NAROS server obtained a similar performance as classical CRC-16 based load-balacing mechanisms. Further investigations include the address selection procedure used by the server and how NAROS can help in engineering the inbound traffic.

Acknowledgements

Steve Uhlig provided the traffic traces and contributed useful comments. We also thank Luca Deri for the `ntop` tool.

References

1. Atkinson, R., Floyd, S.: IAB concerns & recommendations regarding internet research & evolution. Internet Draft, IAB (2003) <draft-iab-research-funding-00.txt>, work in progress.
2. Bu, T., Gao, L., Towsley, D.: On routing table growth. In: Proc. IEEE Global Internet Symposium. (2002)
3. Agarwal, S., Chuah, C.N., Katz, R.H.: OPCA: Robust interdomain policy routing and traffic control. In: Proc. OPENARCH. (2003)
4. Quoitin, B., Uhlig, S., Pelsser, C., Swinnen, L., Bonaventure, O.: Interdomain traffic engineering with BGP. IEEE Communications Magazine (2003)
5. Bagnulo, M., et al.: Survey on proposed IPv6 multi-homing network level mechanisms. Internet Draft (2001) <draft-bagnulo-multi6-survey6-00.txt>.
6. Bagnulo, M., et al.: Application of the MIPv6 protocol to the multi-homing problem. Internet Draft (2003) <draft-bagnulo-multi6-mnm-00.txt>, work in progress.
7. Dupont, F.: Multihomed routing domain issues for IPv6 aggregatable scheme. Internet Draft, IETF (1999) <draft-ietf-ipngwg-multi-isp-00.txt>, work in progress.
8. Jieyun, J.: IPv6 multi-homing with route aggregation. Internet Draft, IETF (1999) <draft-ietf-ipng-ipv6multihome-with-aggr-00.txt>, work in progress.
9. Hagino, J., Snyder, H.: IPv6 multihoming support at site exit routers. RFC 3178, IETF (2001)
10. Tattam, P.: Preserving active TCP sessions on multi-homed networks (2001) http://jazz-1.trumpet.com.au/ipv6-draft/preserve-tcp.txt.
11. Huitema, C., Draves, R.: Host-centric IPv6 multihoming. Internet Draft (2003) <draft-huitema-multi6-hosts-02.txt>, work in progress.
12. Abley, J., Black, B., Gill, V.: Goals for IPv6 site-multihoming architectures. Internet Draft, IETF (2003) <draft-ietf-multi6-multihoming-requirements-04.txt>, work in progress.
13. Draves, R., Hinden, R.: Default router preferences, more-specific routes, and load sharing. Internet Draft, IETF (2002) <draft-ietf-ipv6-router-selection-02.txt>, work in progress.
14. Crawford, M.: Router renumbering for IPv6. RFC 2894, IETF (2000)
15. Narten, T., Nordmark, E., Simpson, W.: Neighbor discovery for IP version 6 (IPv6). RFC 2461, IETF (1998)
16. Draves, R.: Default address selection for internet protocol version 6 (IPv6). RFC 3484, IETF (2003)
17. de Launois, C., Bonaventure, O.: Naros: Host-centric ipv6 multihoming with traffic engineering. Internet Draft (2003) <draft-de-launois-multi6-naros-00.txt>, work in progress.
18. http://www.info.ucl.ac.be/people/delaunoi/naros/ (June 2003)
19. Cao, Z., Wang, Z., Zegura, E.W.: Performance of hashing-based schemes for internet load balancing. In: INFOCOM (1). (2000) 332–341

Adaptive Multipath Routing Based on Local Distribution of Link Load Information

Ivan Gojmerac, Thomas Ziegler, and Peter Reichl

Telecommunications Research Center Vienna (ftw.)
Donau-City-Str. 1, 1220 Vienna, Austria
{gojmerac,ziegler,reichl}@ftw.at

Abstract. Adaptive Multi-Path routing (AMP) is a new simple algorithm for dynamic traffic engineering within autonomous systems. In this paper, we describe an AMP variant which is related to the well-known Optimized Multi-Path (OMP) routing protocol. Whereas OMP requires global knowledge about the whole network in each node, the AMP algorithm is based on a backpressure concept which restricts the distribution of load information to a local scope, thus simplifying both signaling and load balancing mechanisms. The proposed algorithm is investigated using ns-2 simulations for a real medium-size network topologyand load scenarios by performing comparisons to several standard routing strategies.

1 Introduction

Over the last years, various measurements performed by different Internet Service Providers (ISPs) have shown that traffic demands in the Internet change dynamically and exhibit large time of day variations [1]. This provides a clear motivation for improving the performance of operational networks by optimizing the allocation of traffic with respect to the current demands. However, the current architecture of the Internet does not provide a straightforward framework for the optimal mapping of traffic demands to available network resources, establishing traffic engineering to become one of the most important Internet research areas in the last couple of years.

In contrast to physical capacity planning (which is usually performed on a time-scale of multiple months) or traffic control mechanisms (e.g. scheduling, which operates on the time-scale of single packet transmission), many competing approaches are currently being proposed for IP traffic engineering on the time-scale of minutes or hours. The approaches differ substantially in the amount of introduced network management overhead, complexity and requirements on the underlying networking technologies. In this paper we introduce and investigate a dynamic multipath routing algorithm that provides simple and automatized traffic engineering by performing continuous load balancing among multiple paths within an autonomous system, without adding any management overhead.

The rest of the paper is structured as follows: Section 2 surveys related approaches, before Section 3 introduces the AMP algorithm. Section 4 describes our ns-2 implementation as well as simulation scenarios and results, and Section 5 concludes the paper with summarizing remarks and an outlook on current and future research.

G. Karlsson and M.I. Smirnov (Eds.): QofIS 2003, LNCS 2811, pp. 122–131, 2003.

2 Related Work and Basic Idea

Over the last few years, multipath routing has already become a well-established research topic in the area of dynamic traffic engineering. This section briefly outlines the main directions which have been proposed so far and introduces the basic idea of our novel approach.

In today's operational Internet Service Provider (ISP) networks, links are usually assigned static *link costs* (sometimes also called *link weights*), as sketched in Figure 1 for the case of a simple network topology. If routing is performed based on minimizing these link costs, then traffic always takes the same path from source to sink. Whereas this works fine in uncongested networks, the appearance of congestion significantly reduces network efficiency, because in this case the traffic will be persistently routed over congested links, even if parallel uncongested paths exist.

Among the approaches proposed so far to enhance the outlined situation, a rather straightforward possibility employs standard IP routing [2] and tries to globally optimize the link costs for a given traffic matrix. The main advantage of this method is that no changes are required in the underlying network technologies. However, the estimation of the traffic matrix for a live IP network is a non-trivial task, as shown in [3]. Moreover, with respect to the typical time-of-day behavior of traffic matrices [1], it is necessary to regularly repeat the optimization procedure, which adds further network management complexity.

Unlike the traditional IP network architecture, in which paths between pairs of routers can be determined only implicitly by the appropriate setting of link costs, the Multi-Protocol Label Switching (MPLS) [4] technology enables the explicit choice and establishment of label switched paths (LSPs) between pairs of routers in an autonomous system. A number of proposals concentrate on performing traffic engineering by intelligent management of LSPs. However, MPLS-based traffic engineering has the obvious inherent drawback of extensive management overhead.

Another simple traffic engineering approach in the framework of the plain IP architecture is the adaptation of routing metrics to the current load conditions in the network as attempted in the early ARPAnet [5]. There, link costs were dynamically adjusted in proportion to the packet delays, which were used as an indicator of congestion. But due to the coarse granularity of traffic shifts (as multiple paths may be affected at the same time by a single link cost change), this scheme led to link load oscillations for the case of high network congestion (cf. [6]).

Finally, the Optimized Multi-Path (OMP) protocol [6] is a traffic engineering extension of currently deployed link-state routing protocols, which aims at distributing load optimally based on each router X having global knowledge about all link loads in the network. With this information, X can shift traffic from congested to less congested paths and thus perform load balancing decisions. In order to keep nodes up to date, every link propagates and regularly refreshes load information using the protocol's flooding mechanism which is triggered either by the time elapsed since the last update or the amount of load changes in the last measurement.

However, a closer look at OMP reveals some important disadvantages of this protocol, e.g. the sophisticated (and thus memory-consuming) data structures required for storing entire multipaths between all possible sources and sinks, and the inherently unpredictable signaling overhead necessary for disseminating link load updates by flooding.

AMP has been designed to avoid these drawbacks. Viewed from an OMP perspective, the central new feature of the AMP algorithm is to reduce the network view of any node from a global to a local one. Therefore, an arbitrary network node X does not know about the state of all possible link combinations between X and all other network nodes, but is only informed about its immediate neighborhood. The propagation of congestion information through the network resorts to a so-called “backpressure mechanism”. For an intuitive description of this concept, one could view the Internet as a meshed system of unidirectional porous rubber tubes transporting viscous fluid. As soon as the fluid hits upon resistance (congestion situation), this leads to a momentary situation of local pressure increase with two possible consequences: the pressure starts propagating *opposite to the flow direction* (backpressure), in fact tube after tube with decreasing intensity due to the viscosity of the fluid, eventually leading to the establishment of a new global pressure/loss equilibrium. Secondly, persistent congestion may also cause fluid drops to locally pour through the rubber, corresponding to packet loss in a network, which we aim to reduce by performing backpressure and load balancing.

To describe this interplay between local dissemination and global propagation of load information more formally, let A be a generic network node and Ω_A the generic set of all its immediate neighbor nodes. Additionally, let $\overline{AB}$ denote a generic directed link from node A to node B.

The main mechanism of our algorithm is explained in Figure 1. In contrast to OMP, where an increase in utilization on link $\overline{Y_0X}$ causes nodes all over the network to offload some of their paths containing link $\overline{Y_0X}$, under AMP the only node to react is Y_0 as end-node of $\overline{Y_0X}$, trying to shift traffic away from $\overline{Y_0X}$ to alternative paths. Additionally, Y_0 periodically sends out so-called “backpressure messages” (BMs) to each of its neighbor nodes $N_j \in \Omega_{Y_0}$, in order to inform N_j about its respective contribution to the congestion situation on link $\overline{Y_0X}$. All $N_j \in \Omega_{Y_0}$ in turn pass on this information

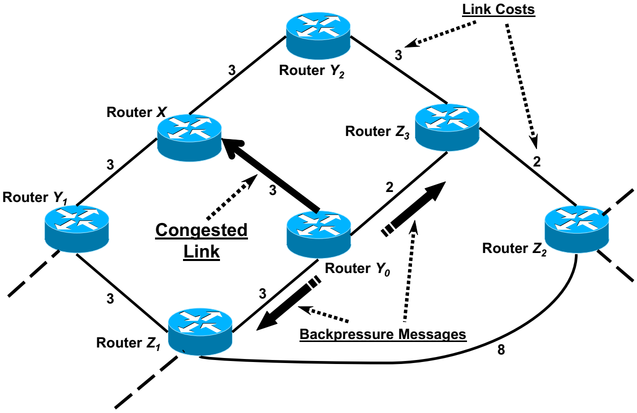


Fig. 1. Basic Idea of Backpressure Mechanism

to their own neighbor nodes, again in proportion to those nodes' respective contribution to the congestion situation, etc. This "quasi-recursive" mechanism provides for the propagation of congestion information through the network producing deterministic signaling traffic, because BMs are sent out periodically.

3 Algorithmic Description of AMP

This section provides an algorithmic description of our new AMP proposal. We start with a survey on multipath calculation, link load metrics and load balancing as key concepts we are able to take from the OMP proposal [6] and adapt to the AMP environment. The rest of the section describes the concept of using backpressure messages for signaling as the crucial new idea distinguishing AMP from OMP and other conventional multipath routing proposals. For further algorithmic details as well as considerations on stability and signaling overhead we refer to [7].

3.1 Multipath Calculation, Load Balancing and Link Load Metrics

In general, multipath routing approaches differ from other interior gateway routing protocols (e.g. OSPF [8]) by no longer employing only the best (minimal cost) path towards a destination, but allow for using more than one path. For instance, the so-called equal cost multipath approach (ECMP) provides a straightforward possibility, allowing to use multiple paths with the same minimal cost and split the traffic equally among them.

In order to further increase the number of candidate paths while avoiding complicated MPLS-like or source routing path establishment, [6] proposes to employ a "relaxed best path criterion". The basic idea is to consider any neighbor node which is closer in terms of cost to the destination than the current node as a viable next hop for multipath routing. Note how this condition immediately prevents the formation of routing loops as the cost to the destination node is forced to strictly decrease with every node along a path.

Going back to Figure 1, we can examine this criterion for an example topology comprising seven nodes. Consider e.g. node X, where the minimal cost to destination node Z_2 is obtained for path $\overline{XY_0Z_3Z_2}$ and equals 7. As the cost to destination Z_2 for X's neighbor Y_2 equals 5 < 7, the path $\overline{XY_2Z_3Z_2}$ also becomes viable under this criterion.

The use of the relaxed best path criterion allows the utilization of a number of different paths between source and destination, but at the same time requires load balancing between these alternatives. As shown in [7], AMP load balancing resorts to dynamic, but rather conservative load shifting, where, in contrast to OMP, load adjustment is not triggered by a complicated decision process, but performed in regular time intervals.

If an AMP node A has multiple paths towards a destination node B, it must have a mechanism for splitting traffic destined to node B among these paths without misordering of packets within microflows. To this end, AMP applies a hash function over the source and destination addresses and divides the hash space among the available paths by setting appropriate thresholds. In analogy to [6], this mechanism allows unequal splitting of traffic among the paths, where dynamic load balancing is achieved by suitable changes of these thresholds. As demonstrated by [9], the CRC-16 (16-bit

Cyclic Redundancy Check) hash function achieves very good load balancing performance due to evenly spreading the source/destination address pairs in the solution space for most examined realistic traffic traces. Note that conservative threshold shifts support an oscillation-free transition towards the new equilibrium.

As a suitable link load metric for elastic traffic, e.g. TCP, our load balancing algorithm employs the "equivalent load" (EL) as proposed in [6]:

$$EL = max\{\rho,\rho \times K \times \sqrt{P}\} \tag{1}$$

where P denotes the packet loss probability and K is a scaling factor determining the packet loss boundary at which the EL value exceeds the link utilization ρ, defined as

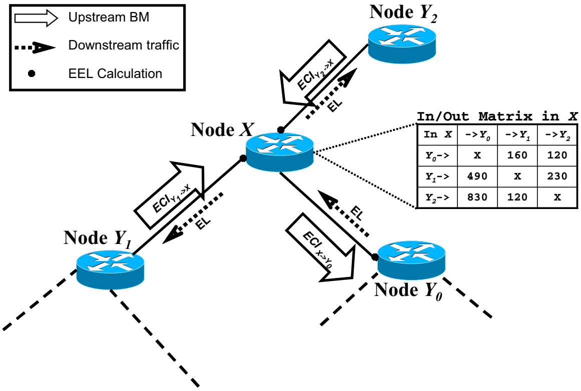
$$\rho = \frac{Carried\ Traffic\ Volume}{Link\ Capacity \times Time\ Period}\ . \tag{2}$$

Equation (1) reflects the dynamic behavior of elastic traffic passing through the link, as TCP flows slow down approximately in proportion to the inverse square root of the packet loss probability on the bottleneck link [10]. In our simulations, we set the value of K to 10, which corresponds to the EL value exceeding the link utilization value ρ for packet loss probabilities greater than 1%.

3.2 Recursive Backpressuring as AMP Signaling Mechanism

It has already been mentioned that the signaling mechanism is considered to be the key novelty of AMP. Figure 2 sketches a typical situation where node Y_0 has to take some load balancing decision for its output links. In the following, consider link $\overline{Y_0X}$ as an example for describing the information required by Y_0 for each of its output links. Besides the equivalent load on the link which can be measured directly, Y_0 requires also information about the extent to which traffic routed from node Y_0 via node X triggers congestion on links $\overline{XY_i}$ and also further downstream, i.e. on links beyond nodes Y_i, $i \geq 1$. Such information is the content of the backpressure messages (BMs).

To describe the mechanism in more detail, for the moment let us occupy the perspective of node X which receives traffic from node Y_0 over the link $\overline{Y_0X}$ and forwards the traffic to nodes Y_i over the links $\overline{XY_i}$, $i \geq 1$. We assume that node X (like any node) is able to measure only the equivalent load EL on its output links, i.e. $\overline{XY_i}$, $i \geq 1$, according to definition (1). While Y_0 is also to some extent responsible for the load situation on links $\overline{XY_i}$, $i \geq 1$, it has no possibility to directly find out about possible congestion on these links. Therefore, node X which has direct access to links $\overline{XY_i}$ can inform node Y_0 about the situation there as far as it is of interest for node Y_0's load balancing decision. This is done periodically by the backpressure messages (BMs). But it is not sufficient that X informs Y_0 only about the (directly measured) explicit loads on links $\overline{XY_i}$, $i \geq 1$, because the situation on links beyond nodes Y_i, $i \geq 1$, is of interest as well. On the other hand, for the latter links node X has no direct access itself, but luckily enough receives BMs from nodes Y_i which inform X about its own influence on the load situation beyond nodes Y_i, $i \geq 1$, i.e. exactly the desired information that should additionally be passed on to node Y_0.



In X	->Y_0	->Y_1	->Y_2
Y_0->	X	160	120
Y_1->	490	X	230
Y_2->	830	120	X

Fig. 2. Generation of ECI Messages

Summing up, any BM sent from node X to node Y_0 should contain both direct information on the explicit load situation on links $\overline{XY_i}$, as well as indirect information about the situation further behind as reported by nodes Y_i to node X, $i \geq 1$. The following equation describes the content of a BM more formally as a function f of these $2n$ parameters where n is the number of output links for node X. To this end, let $BM(A, B)$ denote the content (in terms of parameter values) of the BM sent from node A to node B. Then,

$$BM(X, Y_0) = f(EL_{\overline{XY_1}} \ldots, EL_{\overline{XY_n}}, BM(Y_1, X), \ldots, BM(Y_n, X)) . \tag{3}$$

In order to keep the BM small, f is supposed to map the $2n$ parameters eventually to a scalar describing the congestion situation to Y_0. To derive f, we first reduce the number of input parameters to one per output link by summarizing the situation on each link $\overline{XY_i}$ by a function g, i.e.

$$g_i = g(EL_{\overline{XY_i}}, BM(Y_i, X)) \qquad \forall i = 1, \ldots, n . \tag{4}$$

As neither the output link nor the network beyond should be overloaded, the maximum function is a good candidate for g: assuming that $BM(Y_i, X)$ is equal to the scalar $ECI_{Y_i \to X}$ (the so-called "explicit congestion indication", see below), this leads to the following definition of "effective equivalent load" (EEL):

$$EEL_{\overline{XY_i}} = max\left\{EL_{\overline{XY_i}}, ECI_{Y_i \to X}\right\}, \; i = 1, 2, \ldots, n \tag{5}$$

Note that $ECI_{Y_i \to X}$ summarizes the load information on the downstream links of Y_i as contained in the BM from Y_i to X. Similarly, $ECI_{X \to Y_0}$ is sent as BM from node X to Y_0 and can be viewed as the result of a further simplification of the function f in (3), where the n different parameters g_i resulting from (4) are summarized according to a function h:

$$ECI_{X \to Y_0} = h(g_1, \ldots, g_n) . \tag{6}$$

We use a weighted sum for the computation of h, where the weight for link $\overline{XY_i}$ corresponds to the ratio between traffic on link $\overline{XY_i}$ that has arrived from node Y_0 via X and the total traffic on link $\overline{XY_i}$. Thus, the weighted sum provides a compressed version of all information which is available to node X about Y_0's contribution to the congestion situation on the downstream part of the network and is therefore called "explicit congestion indication":

$$ECI_{X\rightarrow Y_0} = \sum_{Y_i \in \Omega_X \backslash Y_0} \frac{\beta_{\overline{XY_i}}(Y_0)}{\beta_{\overline{XY_i}}} \cdot EEL_{\overline{XY_i}}. \qquad (7)$$

Remember that Ω_X is the set of all neighbor nodes of node X, $\overline{XY_i}$ is the downstream link between node X and node Y_i, $\beta_{\overline{XY_i}}(Y_0)$ is the number of bytes sent from node Y_0 via X to Y_i, and $\beta_{\overline{XY_i}}$ the total number of bytes sent from any node $\in \Omega_X \backslash Y_i$ via X to Y_i.

Summarized shortly, $ECI_{X\rightarrow Y_0}$ is a one-dimensional parameter which describes the extent to which node Y_0 contributes the congestion situation of the network as seen from the perspective of node X. $ECI_{X\rightarrow Y_0}$ is described according to (3) in terms of the function f, which after all together with (4) and (6) results in the following structure:

$$\begin{aligned} BM(X, Y_0) &= ECI_{X\rightarrow Y_0} = f(EL_{\overline{XY_1}} \ldots, EL_{\overline{XY_n}}, BM(Y_1, X), \ldots, BM(Y_n, X)) \\ &= h(g_1, \ldots, g_n) = h(g(EL_{\overline{XY_1}}, ECI_{\overline{Y_1}\rightarrow X}), \ldots, g(EL_{\overline{XY_n}}, ECI_{Y_n\rightarrow X})) \end{aligned} \qquad (8)$$

The formulation given in (8) provides further insight into the recursive structure of the backpressure mechanism, as of course $ECI_{Y_i\rightarrow X}$ is the analogue information sent by node Y_i to X and as such enters (8) as $BM(Y_i, X)$.

In order to guarantee a quick propagation of congestion information throughout the complete network, which is an important prerequisite for suitable load balancing decisions, the interval between consecutive BMs on individual links has to be kept rather small, e.g. one second. If we consider the very small size necessary for BM-packets (protocol-dependent, typically around 50 bytes) and today's high backbone link capacities (typically higher than 155 Mbit/s), we can conclude that AMP signaling usually consumes less than $3*10^{-4}$ % of the link capacity.

Finally note that calculating $\beta_{\overline{XY_i}}(Y_0)$ in (7) requires node X to map precisely traffic on the input link $\overline{Y_0X}$ to the output links $\overline{XY_i}$, $i = 1, 2, \ldots, n$. This mapping is described in a so-called "In/Out Matrix" stored in node X (see Fig. 2). This matrix contains for every node pair (P, Q), $P, Q \in \Omega_X, P \neq Q$, the number of bytes carried between these nodes via X.

4 Simulation and Results

4.1 ns-2 Implementation of AMP

For our simulations, we have chosen the standard packet simulator *ns-2* as developed at the University of California, Berkeley [11]. Our ns-2 implementation comprises the full functionality of the AMP algorithm, i.e. relaxed-best path criterion,

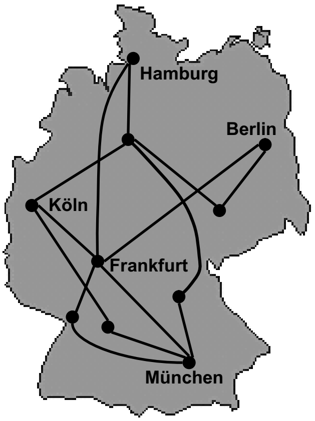


Fig. 3. German B-WiN Research Network

AMP-specific link metrics, backpressure signaling, load balancing and packet forwarding mechanisms.

In [12], we have described simulations with the AT&T-US topology, which is comprised of 27 nodes in major US cities and 47 links interconnecting them. Our evaluations show that AMP achieves significant performance improvements terms of Web page response times and total TCP goodput in the network.

In this paper we wish to investigate the performance of AMP for a smaller network. We have chosen the German B-WiN research network (sketched in Figure 3) as a representative example of medium-size European ISP networks, comprised of 10 nodes and 14 links with link capacities ranging from 53.0 to 133.6 Mbit/s [13]. As far as the link costs are concerned, they are set arbitrarily either to 1 or 2.

The following simulation results demonstrate the performance of AMP compared to standard shortest path routing (SPR) and equal-cost multipath routing (ECMP). We generated traffic according to a Web traffic model similar to the SURGE model [14]. Each node hosts a virtual number of Web users that is assumed to be proportional to the approximate population size of the corresponding German city. The spatial distribution (fan-out) of HTTP-requests in each node is assumed to be proportional to the relative population sizes of the other German cities, i.e. nodes of the topology.

4.2 Simulation Scenarios and Results

Using the B-WiN network topology, we have simulated the SPR, ECMP and AMP routing strategies under various load conditions, ranging from low load to very high load, close to the level of network saturation. The different load levels were produced in direct proportion to the number of users in the individual nodes. Our simulations differ from the related work performed for the OMP protocol [15, 16, 17] as follows:

- In contrast to [16], we have implemented and used the relaxed best path criterion as described in Section 3.1.
- Our Web-user model is closer to reality than the client-server model used in [17].

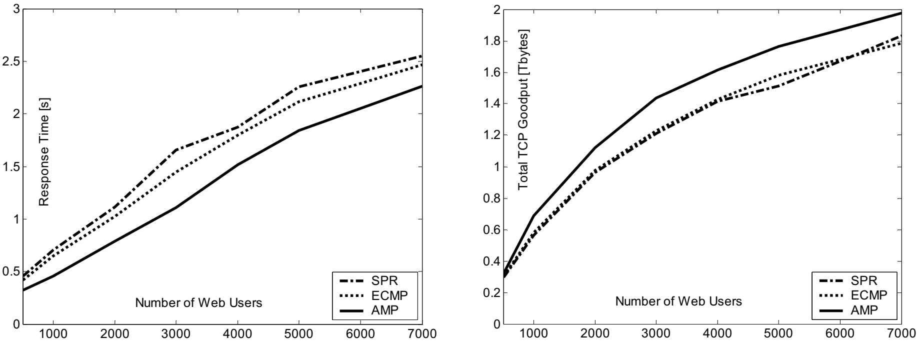


Fig. 4. Comparison of Average Web Page Response Times (left) and Total TCP Goodput (right) for AMP vs. SPR and ECMP

- Whereas [16] uses stochastic traffic sources sending out packets with a constant mean rate, and [17] restricts itself only to FTP traffic, we claim that our Web-traffic model as described above is much closer to reality.

Figure 4 displays the most interesting of our simulation results. Note that in each case we have dropped the initial simulation phase and started measuring as soon as the network had reached equilibrium state. Each simulation has been run for 20000 seconds.

In terms of average Web page response times (Figure 4 left), AMP clearly outperforms the other two routing strategies. Compared to SPR and ECMP, AMP achieves response time reductions of up to 35%. Significant performance gains are also achieved with respect to the total TCP goodput in the network, where AMP provides increases of up to 18% (Figure 4 right). For shorter flows, these gains increase further, e.g. in the case of response times up to nearly 50%, as additional simulations have demonstrated [7]. It is important to notice that AMP performs consistently better than its competitors for practically all investigated traffic loads in this topology. Note however, that for very low loads all three routing strategies perform similarly well, as the traffic flows do not experience significant congestion in the network.

5 Conclusions and Outlook

This paper presents the design of Adaptive Multi-Path (AMP) as a low complexity algorithm for dynamic traffic engineering. In contrast to existing traffic engineering approaches, which usually employ global signaling of link load information, our algorithm resorts to local signaling, thus making the signaling overhead deterministic and minimal, and reducing memory consumption in routers, as AMP does not have to store entire multipaths towards all possible destinations. After implementing AMP in the *ns-2* simulator, we have demonstrated significant performance improvements of AMP compared to standard routing strategies for a real ISP topology and a spectrum of realistic traffic loads. Current and future work deals with simulative performance evaluation of AMP for traffic loads displaying characteristic time-of-day variations and for different parameter settings of the algorithm as well as analytical approaches.

Acknowledgments

This work has been performed in the framework of the Austrian Kplus Competence Center Program. Special thanks to our colleague Eduard Hasenleithner for his restless efforts in maintaining the simulation environment.

References

[1] S. Bhattacharyya, C. Diot, J. Jetcheva, N. Taft: *POP-Level and Access-Link-Level Traffic Dynamics in a Tier-1 POP.* ACM SIGCOMM Internet Measurement Workshop, San Francisco, CA, 2001.

[2] B. Fortz, M. Thorup: *Internet Traffic Engineering by Optimizing OSPF Weights*. Proc. IEEE Infocom, Tel Aviv, Israel, 2000, pp. 519-528.

[3] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, C. Diot: *Traffic Matrix Estimation: Existing Techniques and New Directions*. ACM SIGCOMM, Pittsburgh, PA, 2002.

[4] E. Rosen, A. Viswanathan, R. Callon: *Multiprotocol Label Switching Architecture*. IETF RFC 3031, 2001.

[5] A. Khanna, J. Zinky: *The Revised ARPANET Routing Metric.* ACM SIGCOMM Symposion on Communications Architectures and Protocols, Austin, TX, 1989.

[6] C. Villamizar: *OSPF Optimized Multipath (OSPF-OMP)*. IETF Internet Draft, 1999.

[7] I. Gojmerac, T. Ziegler, P. Reichl: *Adaptive Multi-Path (AMP) - a Novel Routing Algorithm for Dynamic Traffic Engineering*. Technical Report FTW-TR-2003-007, Vienna, 2003.

[8] J. Moy: *OSPF version 2*. IETF RFC 2328, 1998.

[9] Z. Cao, Z. Wang, E. Zegura: *Performance of Hashing-Based Schemes for Internet Load Balancing.* IEEE Infocom, Tel Aviv, Israel, 2000.

[10] M. Mathis, J. Semke, J. Mahdavi, T. Ott: *The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm*. ACM Computer Communications Review, 27(3), 1997.

[11] *ns-2 Homepage* - http://www.isi.edu/nsnam/ns/

[12] I. Gojmerac, T. Ziegler, F. Ricciato, P. Reichl: *Adaptive Multipath Routing for Dynamic Traffic Engineering*. To appear in Proc. of IEEE Globecom, San Francisco, CA, December 2003.

[13] K. Below, C. Schwill, U. Killat: *Erhöhung des Nutzungsgrades eines ATM Netzes für den Wissenschaftsbereich (ERNANI).* Technical Report, Dept. Communication Networks, Technical University Hamburg-Harburg, September 2001, pp. 45-53 (in German).

[14] P. Barford, M. E. Crovella: *Generating Representative Web Workloads for Network and Server Performance Evaluation*. ACM Sigmetrics, Madison, WI, 1998.

[15] C. Villamizar: *OMP Simulations*. http://www.fictitious.org/omp/simulations.html

[16] G. M. Schneider, T. Nemeth: *A Simulation Study of the OSPF-OMP Routing Algorithm. J. Computer Networks*, Vol. 39 (4), 2002, pp. 457-468.

[17] K. Farkas: *OMP Simulation Results in IP Networks.* PCH Conference, Budapest, Hungary, 2001.

Statistical Point-to-Set Edge-Based Quality of Service Provisioning[*]

Satish Raghunath and Shivkumar Kalyanaraman

Department of ECSE,
Rensselaer Polytechnic Institute
Troy, NY 12180
{raghus,kalyas}@rpi.edu

Abstract. In this paper we propose an edge-based quality of service architecture aimed at site-to-site private networks over the Internet. We extend the traditional point-to-point service model to a point-to-set service model, assuming a finite, bounded set of destination sites. Instead of provisioning point-to-point links between a source and its set of destinations, a point-to-set service allows the user to have an allocated bandwidth, which could be flexibly assigned to traffic going toward any destination within the set. The proposed point-to-set service provides low loss rates and *flexibility* to users while allowing providers to obtain multiplexing gains by employing a probabilistic admission control test. Simulation results are provided to demonstrate the utility of deploying such a model.

1 Introduction

The best-effort traffic in Internet is inherently of the point-to-anywhere nature, i.e., sources direct packets to any possible destination. In contrast, traditional quality-of-service (QoS) models set up premium services on a *point-to-point* basis (eg: virtual leased lines, frame-relay etc). Recently, with the advent of IP differentiated services [1,2,12] there has been interest in expanding the *spatial granularity* of QoS models. Clark and Fang [2] proposed that a pool of "assured" service tokens could be allocated to a user or site with the flexibility to employ the tokens toward any arbitrary destination. The large spatial granularity of such a service makes efficient admission control and provisioning virtually impossible [12].

We consider a subset of this problem by examining assurances to a fixed set of destinations and provide a novel solution wherein a user has considerable *flexibility* in apportioning the allocated bandwidth among the destinations and the provider also sees multiplexing gains.

1.1 The Point-to-Set Concept

Consider a private network of sites I_1, I_2, E_1, E_2, E_3 and E_4 as shown in Fig. 1(a). The aggregate traffic from I_1 (called the *point*) toward E_1, E_2 or E_3

[*] Supported in part by NSF contract ANI9806660, NSF contract ANI9819112 and grants from AT&T Labs, Intel Corp. and Nortel Networks Limited.

G. Karlsson and M.I. Smirnov (Eds.): QofIS 2003, LNCS 2811, pp. 132–141, 2003.

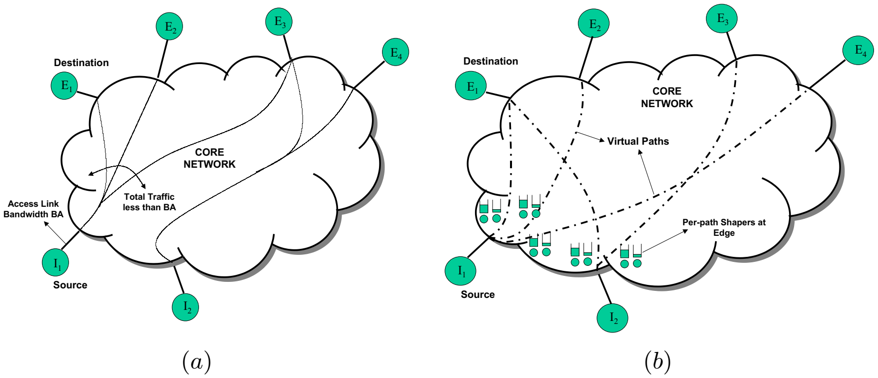


Fig. 1. (a) The Point-to-Set Concept (b) The Point-to-Set architecture (described in §4).

(called the *set*) is bounded by the capacity of the access link (say, "peak"). Given the point-to-point allocation model, site I_1 would require a link with capacity equal to "peak", to each destination in the *set*. As such, the total purchased capacity from the provider exceeds the access link capacity leading to wastage of resources. We propose a *point-to-set service* wherein a customer buys a bandwidth *less than or equal* to his peak requirement (or a given total bandwidth), but is assured that his traffic needs to any destination in the set are met with a *probability* close to 1. In other words, the user buys bandwidth to a set of destinations, instead of purchasing point-to-point links to the destinations and retains the freedom of deciding the fraction of bandwidth allocated to a specific destination. Thus there is a cost saving to the customer (the point-to-point links need not be leased) and multiplexing gains to the provider (the paths from I_1 to $\{E_1, E_2, E_3\}$ can be multiplexed with other contracts).

1.2 Building a Deployable Model

In an ideal point-to-set model, the provider would build techniques to accurately estimate customer demand and appropriately provision bandwidth toward various destinations. However, our previous work [9] indicates that time-varying traffic statistics make such estimation and dynamic provisioning components difficult to implement. In order to simplify the model from the perspective of making the network simple, the user could be required to conform to a certain traffic profile.

The resource wastage in a point-to-point allocation model is due to over-provisioning caused by lack of knowledge regarding the fraction of total load offered toward a destination. The solution to this could be in assuming something about the per-destination load. In order to allow for the dynamic nature of traffic we could strike a *middle ground* between the two extremes of assuming all (as in point-to-point models) or nothing (e.g., hose model) about the per-destination traffic. We could assume that the fraction of traffic toward a given destination is random, but has a given mean and variance (m, v).

This approach would allow the traffic fraction toward a destination to vary within the limits specified by (m, v). Further, knowing the leaky-bucket parameters shaping the total traffic, one could compute bounds on the probability of observing a particular load toward a given destination. We show that these (m, v) parameters can be enforced using simple deterministic shaper elements and lead to a probabilistic admission control scheme that can be easily deployed.

The contributions of this paper are thus as follows: a) A novel architecture for edge-based provisioning toward a set of destinations; b) A simple means to capture and enforce the per-destination traffic statistics; and c) A probabilistic admission control test that allows a flexible service for the user and multiplexing gains to the provider.

2 Related Work

Clark et al [2] introduced the idea of going beyond point-to-point services and providing flexibility to users, while LIRA [12] presents one of the first models where QoS assurances are for a large set of destinations (possibly unlimited). LIRA faces scalability and deployability issues due to per-packet admission control and changes required in routing protocols. Duffield et al [3] propose a framework for Virtual Private Network (VPN) resource management and introduce the idea of a "hose" as a dynamically provisioned access link for a VPN node. However, estimation errors due to uncertainties in traffic statistics and the complexity of signaling-based provisioning mechanisms are the drawbacks of the model. In this paper, we do not require bandwidth estimation or signaling-based reservation while achieving higher spatial granularity of QoS.

3 Notations and Assumptions

Table 1 provides a brief description of the symbols that are used in the succeeding sections. In the following sections, a "user" refers to a customer network offering traffic. A "flow" is a traffic aggregate emanating from a network. The user traffic is assumed to be shaped by a dual-leaky-bucket regulator of the form (π, ρ, σ). Thus, the cumulative offered traffic $A(t)$ in time t always satisfies $\{A(t) \leq \pi t, A(t) \leq \rho t + \sigma\}$. A QoS commitment to the user is termed as a contract (defined in §4.1). The admission control module is assumed to know the paths connecting ingresses and egresses. Each "path" is assumed to be uniquely associated with an ingress-egress pair. Routes are assumed to remain stable.

4 The Point-to-Set Architecture

The point-to-set architecture is depicted in Fig. 1(b). Each user network that enters into a contract with the provider is assumed to specify the set of destinations and the mean and variance of per-destination traffic fraction. This fraction

Table 1. Table of Notations.

Symbol	Meaning
π_j,ρ_j, σ_j	User j Peak rate, Avg rate, Bucket
π_{ij},ρ_{ij}, σ_{ij}	User j Peak, Avg, Bucket toward dest i
p_{ij}, m_{ij}, v_{ij}	Random var for traffic fraction toward i for user j, its mean and variance
C_i	Capacity of path i
D_{max}	Max permissible delay at ingress
ϵ	Max allowed capacity violation probability
Y_j	Random variable for total traffic (bps) due to user j
X_{ij}	Random variable for traffic (bps) toward node i for user j
F, f	Flexibility
Γ_j	Capacity of link j

is enforced via dual-leaky buckets as shown in Fig. 1(b). The admission control module is a central entity to the provider network that knows the paths connecting the provider edge nodes. A new contract can be admitted only if the bandwidth requirement can be accommodated along each *path* connecting the ingress node to a destination. Thus while deciding to admit a contract, the module checks whether adding this contract would cause input rate to exceed the “path capacity”. We define path capacity in the next section. Here we provide an intuitive description of the concept.

A given path from an ingress to an egress may share one or more physical links with other paths in the network. Hence its capacity is not the same as that of the physical links constituting the path. It is convenient to introduce a notion of a *virtual path* of fixed capacity connecting the ingress to the egress. As shown in Fig. 1(b) a virtual path between an ingress-egress pair appears as if it is dedicated to this pair. Then the task of the admission control test is to verify that the probability that the input traffic exceeds the path capacity is less than a given threshold for every path affected by the new contract.

4.1 Definitions

We first define the path capacity and the contract specification.

Definition 1. Consider a path defined by the sequence of links $\{M_j\}$. Let Γ_j be the capacity of M_j. Let n_j be the number of paths passing through M_j. Then, the capacity of the path, is defined as: $C = \min_j \frac{\Gamma_j}{n_j}$.

Although this algorithm for path capacity is simplistic in not allowing differentiation among paths, in succeeding sections we shall use this definition and treat an improved algorithm in future work.

Definition 2. A Contract for user network j consists of the dual-leaky-bucket characterization of the total traffic given by $(\pi_j, \rho_j, \sigma_j)$, the finite set of destination nodes, S_j, the set of pairs $\{(m_{ij}, v_{ij}) \,|\, i \in S_j\}$ where (m_{ij}, v_{ij}) are the mean and variance of the random variable p_{ij} indicating the fraction of total traffic toward i. So if total traffic (bits/second) is given by Y_j and X_{ij} indicates the traffic toward destination i, $X_{ij} = p_{ij}Y_j$, $p_{ij} \in (0, 1]$ and $\sum_i X_{ij} = Y_j$.

4.2 Admission Control Test

The key idea that we exploit here is that of getting an *a priori* estimate of the fraction of total traffic that is offered along a given path. Denote the total traffic (bits per second) offered by customer j as Y_j. Let X_{ij} denote the traffic due to customer j on the path leading to the destination i. If p_{ij} are fixed constants, the provider can provision the right amount of bandwidth toward each destination. A more interesting and realistic situation is when p_{ij} are not fixed. Let p_{ij} be a random variable with mean and variance (m_{ij}, v_{ij}). For simplicity, we assume p_{ij} are independent of Y_j, i.e., the fraction of traffic toward a destination is independent of the total volume of traffic offered by the network. We now impose the constraint that Y_j is policed to a peak rate π_j and shaped by a leaky-bucket shaper (ρ_j, σ_j). The admission control condition can then be as follows - admit a new contract if:

$$\forall i,\ Pr\{\sum_j X_{ij}(t) > C_i\} < \epsilon \ \ and \ \forall i,\ \sum_j m_{ij}\rho_j < C_i \tag{1}$$

Here $\epsilon < 1$ is a given constant. Observe that Equation (1) serves our objectives well. It reserves per-path bandwidth depending on the amount of traffic that the contract might offer and (via ϵ) provides a control on how conservative the admission control gets.

4.3 Quantifying Flexibility

An ideal Point-to-Set service provides an abstraction of a point-to-point link toward each destination for a contract. The source network has the *flexibility* to offer an arbitrary fraction of its total traffic toward any destination. To measure how close to ideal an implementation is, we could examine the flexibility it offers. We expect a measure of flexibility to satisfy these intuitive requirements: (a) higher the flexibility, greater is the freedom to the user in terms of load distribution with respect to the destinations, (b) if loss rates are kept low, higher the flexibility, closer is the service to a point-to-point regime.

Define flexibility, F so that: $\frac{\sqrt{v_{ij}}}{m_{ij}} \leq F \ \forall i, j$. The definition implies that a higher value of F allows for higher variance in per-path offered load. In order to attain lower loss rates and still allow for higher F one would have to admit lesser number of contracts, i.e., employ a lower value of ϵ.

5 Evaluating the Admission Control Decision

In order to evaluate Equation (1), we would need the distribution of X_{ij}. An alternative approach would be to *bound* the distribution somehow, exploiting the fact that Y_j was constrained by $(\pi_j, \rho_j, \sigma_j)$. We thus obtain an upper bound on the mean and variance of the process. To do this we employ a technique similar to [5] and observe that the extremal "on-off" source has the maximum variance among all rate patterns that can be obtained given the $(\pi_j, \rho_j, \sigma_j)$ characterization, if mean is set at ρ_j (Proposition 1). Our approach differs from that of [5] in having a bound independent of a specific interval or duration, and in considering the dual leaky-bucket shaped inputs specified by $(\pi_j, \rho_j, \sigma_j)$.

We note that although the extremal on-off source has maximum variance it does not necessarily maximize buffer overflow probability [4]. The extremal source has been used in the past [7,11] with reference to bandwidth and buffer allocation. Here we employ the source owing to the fact that it leads to more conservative provisioning while easing analysis.

Proposition 1. *Consider a source shaped as $(\pi_j, \rho_j, \sigma_j)$. A transmission pattern with mean ρ_j, that maximizes the variance of rate is given by the periodic extremal on-off source, wherein the source transmits at the peak rate π_j for a duration $T_{on} = \frac{\sigma_j}{\pi_j - \rho_j}$ and switches off for $T_{off} = \frac{\sigma_j}{\rho_j}$.*

The interested reader is referred to [10] for proofs of the propositions. With this proposition, we can now consider the first and second moments of the per-path traffic due to a contract, namely, X_{ij} at the edge of the network.

Proposition 2. *If Y_j, the total traffic due to customer j, shaped by a dual leaky-bucket shaper (π_j,ρ_j,σ_j) has a mean ρ_j and X_{ij} is the fraction of Y_j along path i, the mean and variance of X_{ij} are given as follows.*

$$E\{X_{ij}\} = m_{ij}\rho_j \tag{2}$$

$$Var\{X_{ij}\} \leq m_{ij}\rho_j(\pi_j(\frac{v_{ij}}{m_{ij}} + m_{ij}) - m_{ij}\rho_j) \tag{3}$$

The proof [10] exploits Proposition 1 and uses the corresponding extremal source to bound the variance. Observing that for each path, the statistical characteristics of the traffic offered by a given customer is independent of those of others at the *edge of the network* we now propose an approximation.

Proposition 3. *Define the Gaussian random variable Z_i with mean $m_{Z_i} = \sum_j m_{ij}\rho_j$ and variance $v_{Z_i} = \sum_j m_{ij}\rho_j(\pi_j(\frac{v_{ij}}{m_{ij}} + m_{ij}) - m_{ij}\rho_j)$. Then for sufficiently large number of admitted customers, we have the following approximation.*

$$Pr\{\sum_j X_{ij} > C_i\} \leq Pr\{Z_i > C_i\} \approx \frac{1}{\sqrt{2\pi}} exp\left(-\frac{(C_i - m_{Z_i})^2}{2v_{Z_i}}\right) \tag{4}$$

5.1 Enforcing the Per-Path Limits

Once a contract is admitted, the provider needs to ensure that the offered traffic adheres to the per-path mean and variance restrictions. As demonstrated by the following proposition, it is straightforward to derive the dual leaky bucket shaper $(\pi_{ij}, \sigma_{ij}, \rho_{ij})$ for the path i to enforce (m_{ij}, v_{ij}).

Proposition 4. *Define the dual leaky bucket shaper $(\pi_{ij}, \sigma_{ij}, \rho_{ij})$ such that:*

$$(\pi_{ij}, \sigma_{ij}, \rho_{ij}) = (\pi_j(\frac{v_{ij}}{m_{ij}} + m_{ij}), \sigma_j, m_{ij}\rho_j) \tag{5}$$

This dual leaky bucket shaper ensures that the per-path traffic fraction with mean $m_{ij}\rho_j$ has variance less than $m_{ij}\rho_j(\pi_j(\frac{v_{ij}}{m_{ij}} + m_{ij}) - m_{ij}\rho_j)$

With the above proposition, we now have the ability to implement the model with simple shaping elements.

5.2 Buffer Dimensioning

In order to decide the size of buffers at each hop, we can either set a limit on the maximum tolerable per-hop delay or constrain the maximum burstiness of the input traffic at each node. While the first strategy is simpler, it can result in higher loss rates owing to increased burstiness inside the network. To limit the burstiness of a flow incident at a given node, we must limit the increase in burstiness due to every previous hop through which this flow passed. We do this by limiting the maximum increase in burstiness at the ingress.

Consider a multiplexer M. Let P denote the set of multiplexers feeding traffic to M and L denote the set of incident flows at M. Let D_i^{max} denote the maximum tolerable delay at multiplexer i. We can set the buffer size at a multiplexer i to $\sum_{l \in L} \sigma_l$ or a quantity that upper bounds it, as given below.

$$\sum_{l \in L} \sigma_l = \sum_{p \in P} \sum_{l \in p} \sigma_l \leq \sum_{p \in P} D_p^{max} C_p = D_M^{max} \tag{6}$$

If we set D_{max} to be the maximum tolerable delay at every ingress, we can recursively compute the bound given in Equation (6) for a specific topology. Thus higher buffers are allocated at a multiplexer further along a path.

6 Performance Evaluation

The performance evaluation is performed with the following objectives: (a) to verify the superiority of the probabilistic admission control condition in terms number of admitted contracts in comparison to point-to-point allocation model, (b) to explore the properties of flexibility and ϵ as "control knobs" for the model. We begin by detailing the method used to setup the simulations.

6.1 Methodology

In order to evaluate the scheme, we employ Auckland IV traffic traces [8] with the MCI backbone topology (see [10]) in NS-2 simulation environment. The Auckland data trace is a good fit for this evaluation since it corresponds to traffic at an access link.

Each simulation consists of two phases - an admission control phase where a randomly generated set of contracts is subject to the admission test, and a traffic generation phase where the admitted contracts are simulated. To generate a contract randomly (e.g., with 4 destinations), three uniform random numbers, $r_i, i = 2 \ldots 4$ are generated in the range $[min, max]$. Then setting $r_1 = 1$ and $\sum_i r_i m_{1j} = 1$ we obtain $m_{ij} = r_i m_{1j}$. The total traffic is then apportioned according to a Normal random variable with mean and variance (m_{ij}, v_{ij}) with negatives mapped to a small positive fraction. In the simulations, $(\pi_i, \rho_i, \sigma_i)$ were set to $(0.75\ Mbps, 0.5\ Mbps, 100\ kb)$. The link capacities were set to 10 *Mbps* and their delay was chosen to be 10 *ms*. In the succeeding sections, each point in a graph indicating a simulation result, is the average of 10 simulation runs.

6.2 Comparing with the Point-to-Point Model

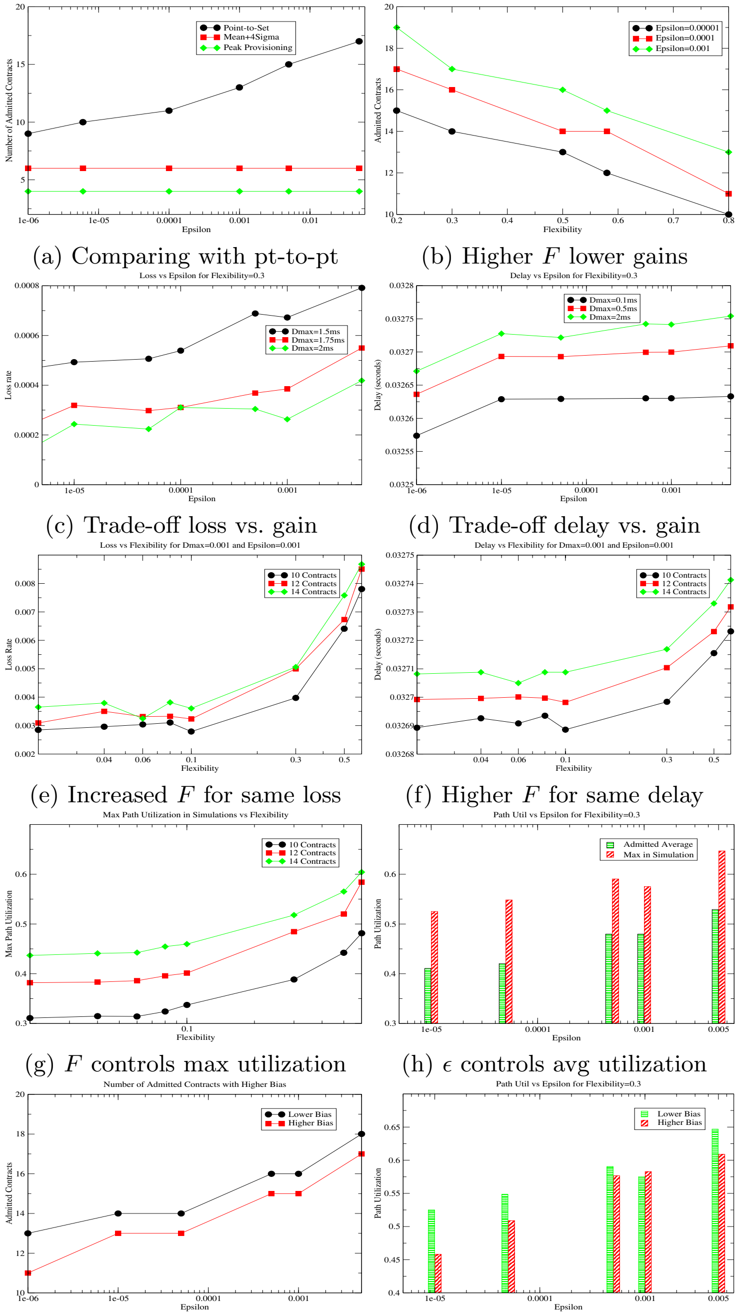
The motivation for deploying point-to-set services is in the fact that there are multiplexing gains for the provider. A point-to-point service provisions links at peak rate toward each of the destinations in the set. In addition to this, we introduce a model where the provider reserves $m_{ij} + 4\sqrt{v_{ij}}$ instead of doing a probabilistic admission control. Although deterministic, it exploits the additional information regarding per-destination traffic fraction. Fig. 2(a) shows the number of admitted contracts under these three schemes and clearly a probabilistic scheme performs much better.

6.3 ϵ and Flexibility as Control Knobs

- **Flexibility and gains:** A higher flexibility means higher variance of per-destination load. Thus for a given ϵ, higher F leads to lower number of admitted contracts as seen in Fig. 2(b).
- **Loss and delay trade-off:** While lower losses can be achieved with more buffers (Fig. 2(c)), it comes at the cost of higher average delays (Fig. 2(d)). Thus setting D_{max} higher allows higher multiplexing gain (higher ϵ) but with higher delays.
- **Cost of Flexibility:** If the provider has to maintain roughly the same loss rates and delays, providing higher F means lower multiplexing gains (Fig. 2(e) and (f)).
- **Controlling utilization:** ϵ and F provide two dimensions of control on utilization. While higher ϵ can achieve higher *average* (Fig. 2(h)), a higher F can lead to higher *maximum* utilization (Fig. 2(g)).

6.4 Effect of Bias in Traffic

If there is more demand toward certain destinations, i.e. the load is biased, there would be some resource wastage. Here we just present this effect and do not provide a solution. In the simulations, if we increase *max* (see §6.1) we increase the bias of traffic toward certain destinations. In Fig. 2(i) and Fig. 2(j) we see that the number of admitted contracts and maximum measured utilization are lower for the same ϵ if the bias is higher.



(a) Comparing with pt-to-pt

(b) Higher F lower gains

(c) Trade-off loss vs. gain

(d) Trade-off delay vs. gain

(e) Increased F for same loss

(f) Higher F for same delay

(g) F controls max utilization

(h) ϵ controls avg utilization

(i) Higher bias lower gain

(j) Bias lowers utilization

Fig. 2. Results of Performance Evaluation.

7 Summary and Conclusions

This paper proposed a novel QoS architecture called the point-to-set architecture. The traditional point-to-point model was extended to be able to provide considerable freedom to the user network in dynamically apportioning the allocated bandwidth among a *finite set* of destinations. The NS-2 implementation results demonstrated the superiority of the model over point-to-point models. The significance of flexibility and the permissible capacity violation probability (ϵ) in providing control over the trade-off between multiplexing gains and loss rates was demonstrated. Future work will involve studying means to further improve multiplexing gains by possible improvements to the admission control test.

References

1. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, *An Architecture for Differentiated Services*, Dec. 1998, IETF RFC 2745.
2. D. Clark, W. Fang, "Explicit Allocation of Best-Effort Packet Delivery Service", *IEEE/ACM Trans. on Networking*, Vol. 6 No. 4, pp. 362-373, August 1998.
3. N.G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K.K. Ramakrishnan, J.E. van der Merive, "A Flexible Model for Resource Management in Virtual Private Networks", *ACM SIGCOMM 99*
4. G. Kesidis, T. Konstantopoulos, "Extremal Shape-Controlled Traffic Patterns in High-Speed Networks," *Trans. on Communications*, Vol. 48, No. 5, pp. 813-819, May 2000.
5. E.W. Knightly, "Second Moment Resource Allocation in Multi-Service Networks," *ACM SIGMETRICS 97*, Vol 25 No. 1, pp. 181-191, Jun. 1997.
6. E.W. Knightly, N.B. Shroff, "Admission control for statistical QoS: theory and practice," *IEEE Network*, Vol. 13 No. 2, pp. 20-29, March 1999.
7. F. Lo Presti, Zhi-Li Zhang, J. Kurose, D. Towsley, "Source Time Scale and Optimal Buffer/Bandwidth Tradeoff for Heterogeneous Regulated Traffic in a Network Node," *IEEE/ACM Trans. on Networking*, Vol. 7, No. 4, pp. 490-501, August 1999.
8. J. Micheel, I. Graham, N. Brownlee, "The Auckland data set: an access link observed," *Proceedings of the 14th ITC Specialists Seminar on Access Networks and Systems*, Catalonia, Spain, April 2001.
9. S. Raghunath, K. Chandrayana, S. Kalyanaraman, "Edge-based QoS Provisioning for Point-to-Set Assured Services," *IEEE ICC 2002*, Vol. 2, pp. 1128-1134, 2002.
10. S. Raghunath, S. Kalyanaraman, "Statistical Point-to-Set Edge-based Quality of Service Provisioning (Extended Version)," *Networks Lab Technical Report ECSE-NET-2003-1*, 2003. `http://networks.ecse.rpi.edu/tech_rep/qofis03-tr.pdf`
11. V. Sivaraman and F. Chiussi, "Providing end-to-end statistical delay guarantees with earliest deadline first scheduling and per-hop traffic shaping," *IEEE INFOCOM 2000*, Vol. 2, pp. 631-640, 2000.
12. I. Stoica, H. Zhang, "LIRA: An Approach for Service Differentiation in the Internet," *NOSSDAV 98*, Cambridge, England, pp. 115-128, July 1998.

A Dynamic Bandwidth Allocation Algorithm for IEEE 802.11e WLANs with HCF Access Method[⋆]

Gennaro Boggia, Pietro Camarda, Claudio Di Zanni,
Luigi A. Grieco, and Saverio Mascolo

Dipartimento di Elettrotecnica ed Elettronica
Politecnico di Bari, Via Orabona
4 – 70125 Bari, Italy
{g.boggia,camarda,dizanni,a.grieco,mascolo}@poliba.it

Abstract. This paper proposes a dynamic bandwidth allocation algorithm for supporting QoS in IEEE 802.11e WLANs with Hybrid Coordination Function (HCF) access method. It distributes the limited WLAN capacity by taking into account the desired queueing delay that multimedia data flows would expect. The algorithm has been designed by following a control theoretic approach and its properties have been analytically investigated. The effectiveness of our approach has been also proved by computer simulations, involving both audio and video flows. Both mathematical analysis and simulation results show that the algorithm guarantees queueing delays that are bounded by the QoS specifications.

1 Introduction

IEEE 802.11 WLAN standard [1] has been widely accepted and employed as a fundamental tool for ubiquitous wireless networking due to its simplicity and robustness against failures. Moreover, the increasing popularity of Personal Device Assistants (PDAs) with embedded speakers, a microphone, and sometimes even a miniature camera, is boosting the interest for WLAN-based media transmission as an alternative to classic cellular wireless communication. However, for a WLAN-based media delivering equipment to be successful, it is important that the limited first-hop bandwidth be managed efficiently to take into account the time constraints of audio/video applications [2,3]. The new IEEE 802.11e proposal has been introduced to support Quality of Service (QoS) in WLANs and represents the natural framework to address the first hop bandwidth allocation issue. Within the 802.11e, the Enhanced Distributed Coordination Function (EDCF) and Hybrid Coordination Function (HCF) have been proposed as basic mechanisms to support the QoS in WLANs. In this paper, we propose a novel control theoretic approach for supporting bandwidth allocation in IEEE

[⋆] This work was funded in part by the Italian Government in the framework of the project TANGO (Traffic models and Algorithms for Next Generation IP networks Optimization).

G. Karlsson and M.I. Smirnov (Eds.): QofIS 2003, LNCS 2811, pp. 142–151, 2003.

802.11e WLANs with HCF. It distributes the limited WLAN capacity by taking into account the desired queueing delay that audio/video applications would expect. For that purpose, a proportional controller and feedforward disturbance compensation have been exploited. It has been designed by following a control theoretic approach and its properties have been analytically investigated. The effectiveness of our approach has been also proved by computer simulations, involving both audio and video flows. Both mathematical analysis and simulation results show that the algorithm ensures bounded queueing delays as required by the QoS specifications.

The paper is organized as follows: Section 2 gives an overview of 802.11 access methods; Section 3 describes the proposed bandwidth allocation algorithm; Section 4 shows simulation results and, finally, the last section draws the conclusions.

2 IEEE 802.11 Access Methods

This section describes the IEEE 802.11 [1] basic functionalities and the IEEE 802.11e standard EDCF and HCF mechanisms. In the sequel we will consider an infrastructure WLAN [4,5] composed by an Access Point (AP) and a group of stations (associated with the AP) that communicate through the wireless medium. The APs behave like base stations in a cellular communication system by interconnecting the mobile stations with the backbone.

The primary 802.11 MAC protocol is the Distributed Coordination Function (DCF), based on CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance): for each MAC Protocol Data Unit to transmit, all stations contend for accessing the radio channel by listening if the channel is idle after a minimum duration called DCF Interframe Space (*DIFS*). Instead, if the channel is sensed as busy, station waits until the channel becomes idle for a *DIFS* plus an additional random backoff time, uniformly distributed in a range called *Contention Window* (*CW*).

Each successfully receipted frame is acknowledged by the receiving station with an ACK frame, which is sent after a SIFS (Short Interframe Space) period. If there is an unsuccessful transmission, the CW value is doubled and the station restarts sensing the channel.

In order to support time-bounded services, the 802.11 standard defines the Point Coordination Function (PCF) access method as optional capability, which provides a contention-free medium access [2]. A Point Coordinator (PC), typically the AP, polls each stations enabling them to transmit without channel contention. With PCF, time is always divided into repeated periods, called *SuperFrames* (SFs), formed by a Contention Period (CP) and a Contention Free Period (CFP). During the CP the DCF is used for medium access, instead, during the CFP the PCF is used.

At the nominal beginning of each CFP interval, the PC senses if the medium is idle for a PIFS (PCF Interframe Space) interval and transmits a beacon frame with synchronization and timing functions, specifying the time when the next beacon frame will arrive. Then, after a SIFS interval, the PC asks to each station for pending frames, polling them with a CF-Poll frame (with no data) or a Data+CF-Poll frame (with pending data for the polled station). A polled station responds, after a SIFS period, with a CF-ACK (no data) or a Data+CF-ACK frame, and the PC can pool another station. At the end of the CFP, the PC transmits a specific control frame, called CF-End [1].

There are some issues with PCF, which are currently under investigation: the PC cannot adaptively distribute the wireless channel capacity by taking into account the status of mobile stations, because the starting time and the transmissions duration of polled stations is unknown and is not under the control of the PC [2].

2.1 IEEE 802.11e QoS Enhancements

Stations operating under 802.11e specifications are usually known as enhanced stations or QoS Stations (QSTAs). The superframe utilization is preserved by using the EDCF in the CP only and the HCF in the CFP. The use of HCF requires a centralized controller (generally the AP) which is called the Hybrid Coordinator (HC). In order to obtain service differentiation, 802.11e introduces up to 8 Traffic Categories (TCs), each one characterized by specific QoS needs that have to be taken into account to properly distribute the WLAN bandwidth. For that purpose, the concept of TXOP (Transmission Opportunity) is introduced which is defined as an interval of time when a station has the right to initiate transmission [2,6].

The EDCF access method is the same of DCF, except for contention parameters that are used per each TC. In this way, a statistical service differentiation among TCs is obtained. Each QSTA has a queue for each traffic class $TC(i)$ that acts as a virtual station with its contention window CW(i) [2,6].

The HCF access method is an optional access method only usable in infrastructure configurations. It combines some EDCF functions with PCF basic aspects. There is a CP period during which QSTAs use EDCF access method and a CFP during the which only polled and granted QSTAs are allowed to transmit. The CFP ends after the time announced in the beacon frame or by a CF-End frame sent by the HC. The HC (typically performed by the AP) allocates TXOPs to QSTAs to match predefined service rate, delay and/or jitter requirements. During the CFP, the HC uses special QoS CF-Poll frame to specify the starting time and the maximum duration of TXOPs assigned to each QSTA. Moreover, IEEE 802.11e specifications consider two basic mechanisms allowing QSTAs to feed back queue levels of each TC to the HC. The first one exploits an appropriate header field used during data transmission (during both the CFP and CP); the second one allows the AP to query QSTAs in order to get feedbacks during the Controlled Contention Interval, which is generally located at the end of the superframe.

In the following section, we will propose a control scheme based on feeding back queue levels to control queueing delays and provide the typical QoS needs of audio/video applications.

3 The Control Theoretic Framework

In this section a QoS control scheme for WLANs supporting HCF access method is proposed. It is based on the assumption that the HC assigns TXOPs only during the CFP and the superframe duration is constant and equal to T_{SF}. Moreover, with the proposed scheme, the HC assigns TXOPs to each TC queue within each station thus solving the internal contention among TCs too. In the sequel we will refer to a WLAN system composed by an Access Point (AP) and a set of mobile stations (QSTAs) supporting QoS WLAN capabilities. For each QSTA there are N traffic sources feeding N queues, respectively (in the 802.11e proposal $N \leq 8$). At the beginning of each superframe, the AP has to allocate the channel bandwidth that will drain each queue during the following CFP. We assume that at the beginning of each superframe the AP is aware of all the queue levels $q_i,\ i = 1, \ldots, M$, where M is the total number of queues in the WLAN system (i.e., the total number of traffic sources within the BSS).

The dynamics of the i^{th} queue can modelled as follows:

$$q_i(k+1) = q_i(k) + d_i(k) \cdot T_{SF} - u_i(k) \cdot T_{SF}, \qquad i = 1, \ldots, M \tag{1}$$

where $q_i(k+1)$ is the i^{th} queue level at the beginning of the $(k+1)^{th}$ superframe, and $d_i(k)$ and $u_i(k)$ are, respectively, the i^{th} source rate and the bandwidth assigned for draining the i^{th} queue during the k^{th} superframe.

In each superframe, the TXOP value assigned to drain the i^{th} queue during the k^{th} superframe is given by:

$$TXOP_i(k) = u_i(k) \cdot (T_{SF}/C) + N_i^p \cdot (2SIFS + T_{ACK} + T_{MH} + T_{PHY}). \tag{2}$$

where N_i^p is the number of packets that is possible to transmit in the time interval $u_i(k)T_{SF}/C$, C is the WLAN capacity, and T_{ACK}, T_{MH}, and T_{PHY} are, respectively, the duration to transmit the ACK frame, the MAC_Header, and the frame preamble.

The behavior of $d_i(k)$ is unpredictable and does not depend on the AP, (i.e. using a control theoretic approach, it can be modelled as a disturbance). In particular, the following model for $d_i(k)$ can be assumed: $d_i(k) = \sum_{j=0}^{+\infty} d_{0j} \cdot 1(k - T_j)$, where $1(k)$ is the unitary step function and T_j is a time lag [7]. Due to this assumption and to the linearity of system (1), we will study the system response to the signal $d_i(k) = d_0 \cdot 1(k)$ without loss of generality. We will first propose the QoS control scheme under the assumption that the WLAN capacity C is never exceeded by the allocation algorithm. Then we will discuss the effects of the limited capacity C.

3.1 The Control Scheme

The proposed QoS control algorithm aims at driving the queueing delay τ_i experienced by the packets going through the i^{th} queue to the target value τ_i^T. The target queuing delay τ_i^T is related to the QoS requirements for the i^{th} TC.

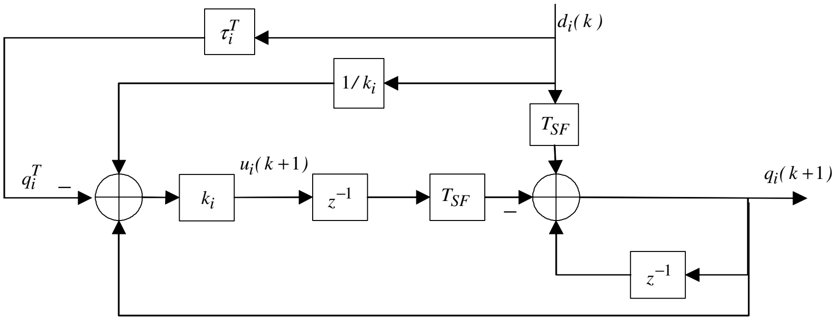


Fig. 1. Control scheme using a propotional control with feedforward disturbance compensation.

To mach our control objective, we will exploit both proportional controller and feedforward compensation as shown in Fig. 1.

For the system reported in Fig. 1 the following transfer functions can be obtained:

$$\frac{Q_i(z)}{D_i(z)} = \frac{T_{SF}(z + k_i\tau_i^T - 1)}{z(z-1+k_iT_{SF})}; \qquad \frac{U_i(z)}{D_i(z)} = \frac{k_iT_{SF}z + (z-1)(1-k_i\tau_i^T)}{z(z-1+k_iT_{SF})} \tag{3}$$

where $Q_i(z)$,$U_i(z)$, and $D_i(z)$ are, respectively, the $\mathscr{Z}$-transforms of $q_i(k)$,$u_i(k)$, and $d_i(k)$. This system is asymptomatically stable iff all its poles z_p satisfy the condition $|z_p| < 1$, that is iff $0 < k_i < 2/T_{SF}$ [9]. In the sequel, we will assume that k_i guarantees system stability.

By applying the final value theorem to Eq. (3), with $d_i(k) = d_0 \cdot 1(k)$, it turns out $q_i(+\infty) = d_0 \cdot \tau_i^T$ and $u_i(+\infty) = d_0$. Thus, the steady state queueing delay is $\tau_i(+\infty) = q_i(+\infty)/u_i(+\infty) = \tau_i^T$

Moreover, by considering the system reported in Fig. 1 where $d_i(k) = d_0 \cdot 1(k)$ and by transforming back to the time domain Eq. (3), it results:

$$\begin{cases} q_i(k) = d_0 \cdot (T_{SF} - \tau_i^T)(1 - k_i \cdot T_{SF})^{(k-1)} \cdot 1(k-1) + d_0 \cdot \tau_i^T \cdot 1(k-1) \\ u_i(k) = d_0 \cdot k_i \cdot (T_{SF} - \tau_i^T)(1 - k_i \cdot T_{SF})^{(k-1)} \cdot 1(k-1) + d_0 \cdot 1(k-1) \end{cases} \tag{4}$$

From these equations it is straightforward to show that the following relations hold when $0 < k_i < 1/T_{SF}$:

$$\begin{aligned} &\begin{cases} q_i(k) \le d_0 \cdot T_{SF} = q_i(1) \\ u_i(k) \le d_0 \cdot [1 + k_i \cdot (T_{SF} - \tau_i^T)] = d_i(1) \end{cases} && \text{if} \quad T_{SF} \ge \tau_i^T \\ &\begin{cases} q_i(k) \le d_0 \cdot \tau_i^T \\ u_i(k) \le d_0 \end{cases} && \text{if} \quad T_{SF} \le \tau_i^T \end{aligned} \tag{5}$$

Eq. (5) implies that if we attempt to drive the queueing delay to $\tau_i^T < T_{SF}$ there is an overshoot of the queue which is equal to $d_0 \cdot (T_{SF} - \tau_i^T)$. Whereas if $\tau_i^T \ge T_{SF}$ and $0 < k_i < 1/T_{SF}$ there is no overshoot.

The control system in Fig. 1 drives queue lengths to the desired value using the following control law:

$$u_i(k+1) = k_i \cdot [q_i(k+1) - d_i(k) \cdot \tau_i^T] + d_i(k). \tag{6}$$

The control law (6) requires a measure of $d_i(k)$. Such measure can be obtained from Eq. (1) as follows:

$$d_i(k) = [q_i(k+1) - q_i(k)]/T_{SF} + u_i(k), \qquad i = 1, \dots, M \tag{7}$$

In fact at step $k+1$ the controller knows all variables on the right side of Eq. (7).

Capacity Constraint, Data Rate Reductions and Robustness to Delayed Feedback. The overall control system we have proposed is based on the assumption that the WLAN capacity is larger than the sum of the bandwidth assigned to each queue. If $\exists k_0 \ni' \sum_{i=1}^{M} u_i(k_0) > C$ then it is necessary to reallocate the bandwidth C so that the sum of the bandwidth assigned to each queue is equal to C. This can be done by assigning to each queue a bandwidth equal to $u_i(k_0) - \Delta u_i(k_0)$, which satisfies the relation $\sum_{i=1}^{M} [u_i(k_0) - \Delta u_i(k_0)] = C$, where $\Delta u_i(k_0), i = 1, \dots, M$ are chosen by taking into account TC priorities. From Eq. (1), it turns out that the signal $\Delta u_i(k)$ can be modelled as an equivalent disturbance $d_i^{eq}(k) = d_i(k) + \Delta u_i(k)$. It should be pointed out that when the WLAN capacity is exceeded, queues build up and queuing delays grow over the target values. This is an obvious results because when capacity is exceeded congestion happens and no QoS requirements can be satisfied. However what is important is that, as soon as the aggregate traffic diminishes under the WLAN capacity, the control loop drives the queuing delays towards the target values.

The proposed control algorithm implicitly assumes that all mobile stations transmit at the same constant data rate, which is equal to the WLAN capacity C. However, mobile stations can select a data rate $C_i < C$ to counteract the effect of signal attenuation and be able to transmit data. This effect can be easily taken into account by using C_i instead than C in Eq. (2).

The control law (6) is based on the assumption that QSTAs feed queue levels $q_i(k+1)$ back to the AP at the beginning of each superframe. If a measure of the i^{th} queue level has not be sent during the last superframe, then the AP can query the QSTAs containing the i^{th} queue to obtain the feedback. Thus the feedback signal might be delayed up to T_{SF} seconds. Fig. 2 depicts the considered control system where a delay of 1 step affects the feedback branch.

By considering Fig. 2, the following transfer function can be obtained:

$$\frac{Q_i(z)}{D_i(z)} = \frac{z \cdot T_{SF} + k_i T_{SF}(1/k_i - \tau_i^T)}{z^2 - z + k_i \cdot T_{SF}} \cdot \tag{8}$$

From Eq. (8) it results that the system pole is $z_p = \frac{1 \pm \sqrt{1 - 4k_i \cdot T_{SF}}}{2}$, which is asymptotically stable iff $|z_p| < 1$, that is iff $0 < k_i < 1/T_{SF}$.

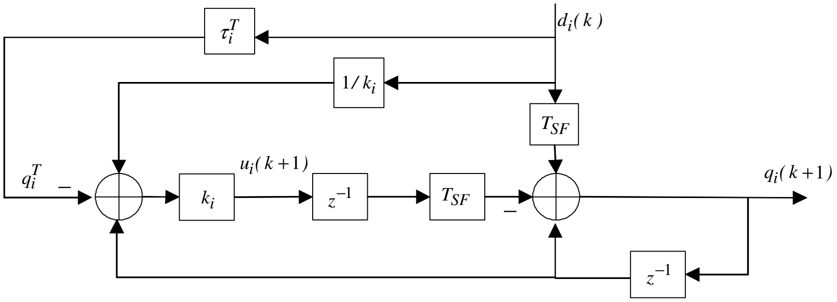


Fig. 2. Control scheme with delayed feedback.

4 Performance Evaluation

Starting from the main results of the previous section, herein we first show how to choose the system parameters and then we present computer simulation results to confirm the effectiveness of the proposed approach. Typical QoS specifications for audio/video applications impose a one way delay less than 150ms (up to 400ms with echo control) for conversational voice or videophone and up to 10s for video streaming [10]. Being the WLAN the first hop traversed by the media flow, we can assume a minimum target delay equal to 50ms. In order to avoid overshoots (see Sec. 3), which might be critical for the time constraints of audio/video applications and saturate the WLAN limited capacity, we set $T_{SF} \leq \min_{i=1..M}(\tau_i^T) = 50ms$. Moreover, to have transfer functions without zeros while guaranteeing stability also in the presence of delayed feedback, we choose $k_i = 1/\tau_i^T$.

To highlight the influence of the T_{SF} parameter on system performance, we should consider that, due to the granularity of the system, the transmission of a packet can be delayed up to $\tau_i^T + T_{SF}$. Thus, a large T_{SF} would critically affect the queuing delay of media flows carrying conversational voice or video. Therefore, in the following we will use the system settings $T_{SF} = 20ms < 50ms$, which leads to a tolerable degradation of the expected queuing delay.

Moreover, assuming flows with constant packet size, the algorithm needs to round the assigned TXOPs in order to transmit an exact number of packets. We round TXOPs in excess, which can lead to queuing delay smaller than $\tau_i^T + T_{SF}$.

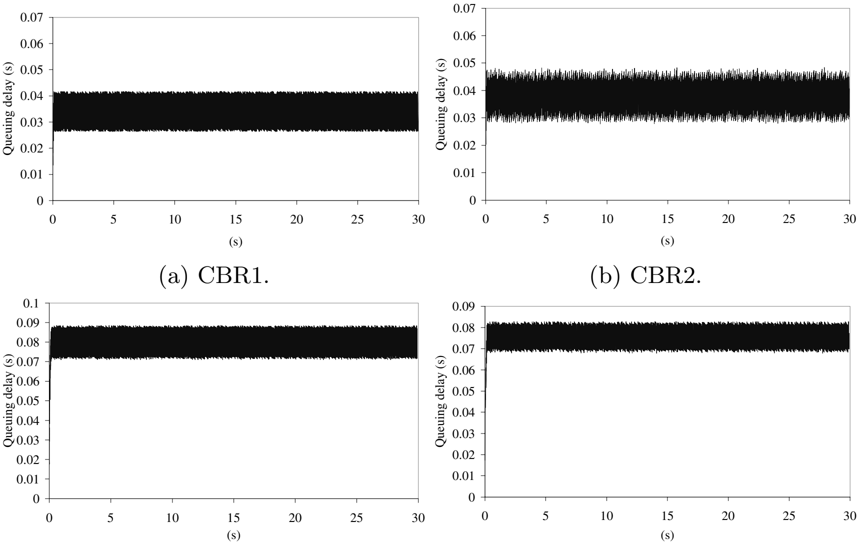
To confirm the effectiveness of the proposed control scheme we have run computer simulations involving audio/video data transfers. For that purpose, we have enhanced the simulator developed by Singla and Chesson [11]. In all simulations a WLAN capacity $C = 11$ Mbps is assumed.

The first considered scenario involves 4 CBR flows CBR1-CBR4 sharing the WLAN capacity. CBR flows characteristics are reported in Table 1. Fig. 3 shows that queuing delays of the CBR flows are bounded as required by the specifications.

The second scenario (see Fig. 4) involves two CBR flows, two voice flows encoded with the G.729 standard [12] and two video flows that represents the encoded version of the *Asterix* film obtained from an online video trace library

Table 1. Characteristics of the considered CBR flows.

Flow	**Transmission rate**	**Packet size**	**Target One Way Delay** $\tau_i^T + T_{SF}$
CBR1	1Mbps	500Bytes	70ms
CBR2	512 Kbps	500Bytes	70ms
CBR3	1Mbps	256Bytes	95ms
CBR4	2Mbps	2048Bytes	110ms



(a) CBR1.

(b) CBR2.

(c) CBR3.

(d) CBR4.

Fig. 3. Queuing delay for CBR flows in Tab. 1.

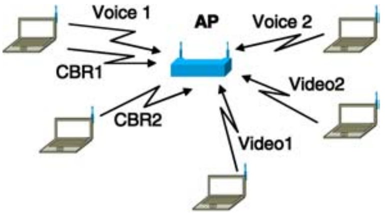
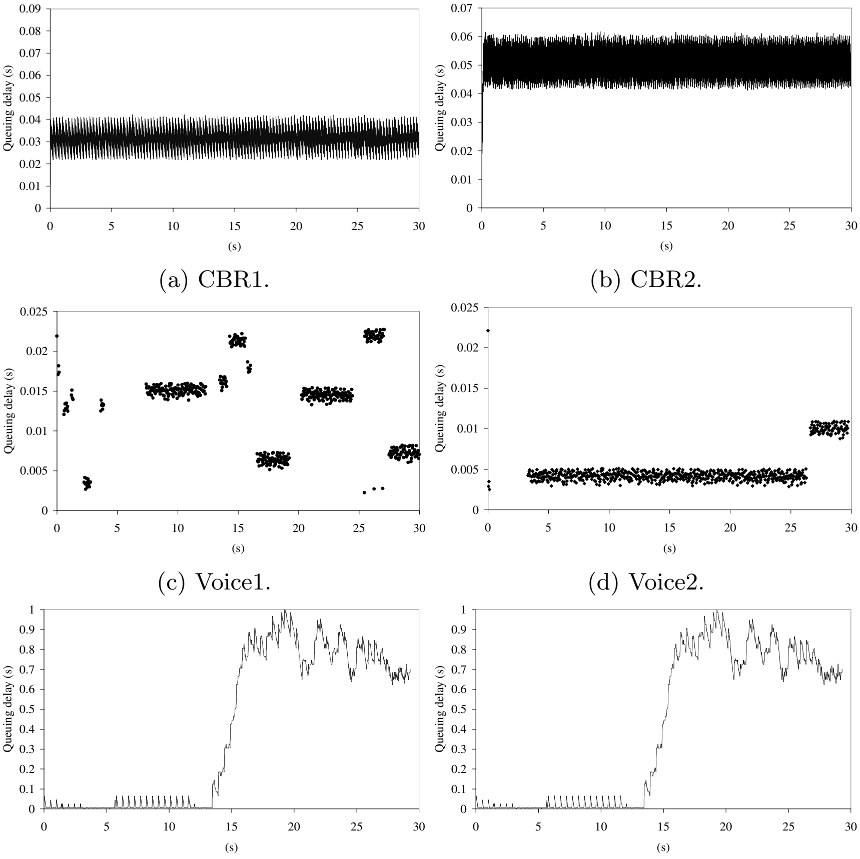


Fig. 4. Simulated Scenario with multimedia flows.

[13]. Flows characteristics are reported in Table 2. Fig. 5 shows that queuing delays of the flows are bounded as required by the specifications in Table 2. Regarding the Voice1 and Voice2 flows, the black spots refer to the talk spurts of the data source.

Table 2. Characteristics of the considered multimedia flows.

Flow	Transmission rate	Packet size	Target One Way Delay $\tau_i^T + T_{SF}$
CBR1	384kbps	500Bytes	90ms
CBR2	512 Kbps	512Bytes	90ms
Voice1,2	G.729-8Kbps	40Bytes	50ms
Video1,2 Asterix	MPEG1 encoding	1536Bytes	1.020s



(a) CBR1.

(b) CBR2.

(c) Voice1.

(d) Voice2.

(e) Asterix1.

(f) Asterix2.

Fig. 5. Queuing delay for multimedia flows in Tab. 2.

5 Conclusion

The paper proposed a novel bandwidth allocation algorithm for supporting QoS in WLANs. The algorithm is based on classic control theory and has been theoretically analyzed. A procedure to properly designing a WLAN access network supporting HCF capabilities has been developed and computer simulation results, involving several audio/video flows, have been shown to confirm the effectiveness of the proposed algorithm.

References

1. N. Prasad and A. Prasad, editors. *WLAN Systems and Wireless IP*. Artech House universal personal communications series. Artech House, 2002.
2. S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz, and L. Stibor. IEEE 802.11e Wireless LAN for Quality of Service. In *European Wireless Conference 2002*, Florence, Italy, Feb. 2002.
3. L. Qiu, P. Bahl, and A. Adya. The effect of first-hop wireless bandwidth allocation on end-to-end network performance. In *NOSSDAV'02*, pages 85–93, Miami, Florida, May 2002.
4. B. P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai. IEEE 802.11 wireless local area networks. *IEEE Commun. Mag.*, pages 116–126, Sep. 1997.
5. R.O. LaMaire, A. Krishna, and P. Bhagwat. Wireless LANs amd mobile networking: Standards and future directions. *IEEE Commun. Mag.*, pages 86–94, Aug. 1996.
6. IEEE 802.11 WG. *Draft Supplement to Standard for Telecommunications and Information Exchange between Systems - LAN/MAN Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS).* IEEE 802.11e/D2.0, Nov. 2001.
7. S. Mascolo. Congestion control in high-speed communication networks using the Smith principle. *Automatica, Special Issue on Control methods for communication networks*, 35:1921–1935, December 1999.
8. V. Jacobson. Congestion avoidance and control. In *ACM Sigcomm '88*, pages 314–329, Stanford, CA, USA, August 1988.
9. K. J. Astrom. and B. Wittenmark. *Computer controlled systems: theory and design.* Prentice Hall Information and System Sciences serie. Prentice Hall, Englewood Cliffs, 3 edition, 1995.
10. International Telecommunication Union (ITU). *Transmission Systems and Media, Digital Systems and Networks.* ITU-T Recommendation G.1010, Nov. 2001.
11. Singla A and G. Chesson. *HCF and EDCF Simulations.* Atheros Communications, Inc., October 2001. doc: IEEE 802.11-01/525r0.
12. International Telecommunication Union (ITU). *Coding of Speech at 8 kbit/s using Conjugate-Structure Algebraic-Code-Excited Linear Prediction (CS-ACELP).* ITU-T Recommendation G.729, Mar. 1996.
13. Video trace library. `http://www-tkn.ee.tu-berlin.de/research/trace/trace.html`.

Performance Analysis of an Enhanced IEEE 802.11 Distributed Coordination Function Supporting Service Differentiation[*]

Bo Li and Roberto Battiti

Department of Computer Science and Telecommunications, University of Trento,
38050 POVO Trento, Italy
{li,battiti}@dit.unitn.it

Abstract. As one of the fastest growing wireless access technologies, Wireless LANs (WLANs) must evolve to support adequate degrees of service differentiation. Unfortunately, current WLAN standards like IEEE 802.11 Distributed Coordination Function (DCF) lack this ability. Work is in progress to define an enhanced version capable of supporting QoS for multimedia traffic at the MAC layer. In this paper, we aim at gaining insight into two mechanisms to differentiate among traffic categories, i.e., scaling the minimum contention window size and the length of the packet payload according to the priority of each traffic flow. We propose an analysis model to compute the throughput and packet transmission delays. In additions, we derive approximations to get simpler but more meaningful relationships among different parameters. Comparisons with simulation results show that a very good accuracy of performance evaluation can be achieved by using the proposed analysis model.

Keyword: Wireless LAN, IEEE 802.11, Quality of Service Guarantee, Service Differentiation

1 Introduction

One of the major challenges of the wireless mobile Internet is to provide Quality of Service (QoS) guarantees over IP-based wireless access networks [2]. Wireless access may be considered just another hop in the communication path for the whole Internet. Therefore, it is desirable that the architecture supporting quality assurances follows the same principles in the wireless networks as in the wireline Internet, assuring compatibility between the wireless and wireline parts. A good example for such a wireless technology is the IEEE 802.11 Distributed Coordination Function (DCF) standard [3], compatible with the current best-effort service model of the Internet.

In order to support different QoS requirements for various types of service, a possibility is to support service differentiation in the IEEE 802.11 MAC layer. Some differentiated services-capable schemes for the enhancement of IEEE 802.11 MAC have been proposed [4][5]. In [4], service differentiation is supported by setting different Minimum Contention Window $CW_{\min}$ for different types of services. The work in [5] proposes three service differentiation schemes for IEEE 802.11 DCF.

[*] This work is supported by the project of WILMA funded by Provincia Autonoma di Trento (www.wilmaproject.org)

G. Karlsson and M.I. Simirnov (Eds.): QofIS 2003, LNCS 2811, pp. 152–161, 2003.

Moreover, in [6], both the Enhanced Distributed Coordination Function (EDCF) and the Hybrid Coordination Function (HCF), defined in the IEEE 802.11e draft, are evaluated. In the literature, performance evaluation of the basic 802.11 MAC protocol has been done by using simulation [7] or by means of analytical models [8]-[12].

By building on previous papers dealing with the analysis of the IEEE 802.11 MAC, we extend the analysis to the Enhanced IEEE 802.11 MAC with service differentiation support. The new results of the presented analysis provide a compact explanation about the effect of the different parameters on the service differentiation.

2 IEEE 802.11 DCF and Enhanced Versions

In the 802.11 MAC sub-layer, two services have been defined: the Distributed Coordination Function (DCF), which supports delay-insensitive data transmissions, and the optional Point Coordination Function (PCF) to support delay-sensitive transmissions. The DCF works as a listen-before-talk scheme, based on CSMA. Moreover, a Collision Avoidance (CA) mechanism is defined to reduce the probability of collisions. Briefly, when the MAC receives a request to transmit a frame, a check is made of the physical and virtual carrier sense mechanisms. If the medium is not in use for an interval of DIFS, the MAC may begin transmission of the frame. If the medium is in use during the DIFS interval, the MAC will select a backoff time and increment the retry counter. The backoff time is uniformly chosen in the range $(0, W-1)$, W being the contention window. The MAC decrements the backoff value each time the medium is detected to be idle for an interval of one slot time. The terminal starts transmitting a packet when the backoff value reaches zero. After the transmission of a packet, the sender waits for the ACK frame from the receiver after SIFS (plus the propagation delay). If the sender does not receive the ACK within ACK_Timeout, or if a different packet is on the channel, it reschedules the packet transmission according to the given backoff rules. If there is a collision, the contention window is doubled, a new backoff interval is selected. At the first transmission attempt, W is set equal to a value $CW_{\min}$ called minimum contention window. After each unsuccessful transmission, W is doubled, up to a maximum value $CW_{\max} = 2^m \cdot CW_{\min}$.

The basic DCF method is not appropriate for handling multimedia traffic requiring guarantees about throughput and delay. Because of this weakness, work is in progress to define an enhanced version capable of supporting QoS [13]. In this paper, we are not interested in exploring all details of the new proposed standard but to gain insight into two of the mechanisms, i.e. scaling minimum contention window sizes and lengths of packet payload according to the priority of each traffic category.

3 Performance Analysis

We assume that the channel conditions are ideal (i.e., no hidden terminals and capture) and that the system operates in saturation: a fixed number of stations always have a packet available for transmission.

L $(L \geq 1)$ different types of traffic flows are considered with n_i traffic flows for traffic type i $(i = 1,2,...,L)$. Let $b_i(t)$ be the stochastic process representing the backoff time counter for a given traffic flow with type i. Moreover, let us define for convenience $W_i = CW_{\min,i}$ as the minimum contention window for traffic type i. Let m_i, "maximum backoff stage" be the value such that $CW_{\max,i} = 2^{m_i} \cdot W_i$, and let $s_i(t)$ be the stochastic process representing the backoff stage $(0,1,\ldots,m_i)$ for a given traffic flow with type i .

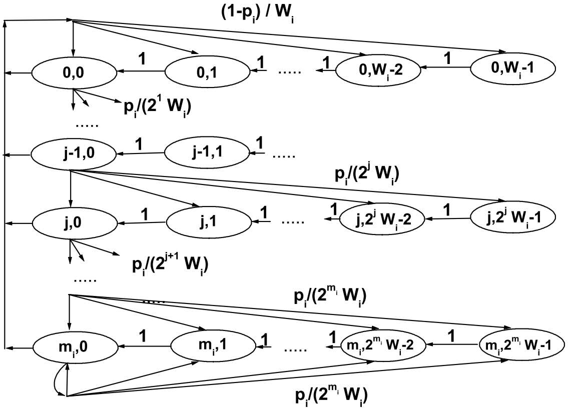


Fig. 1. Markov model of backoff process for traffic type *i*

The key approximation in the model is that, at each transmission attempt for a traffic flow of type i, regardless of the number of retransmissions suffered, each packet collides with constant and independent probability p_i. This assumption has been shown by simulation in [12] to be very accurate as long as W_i and n_i get larger. In this paper, p_i is referred to as conditional collision probability: the probability of a collision seen by a packet belonging to a traffic flow with type i at the time of its being transmitted on the channel.

We use a two-dimensional discrete-time Markov chain to model the behavior of a traffic flow with type i. The states are defined as couples of two integers $\{s_i(t), b_i(t)\}$. The Markov chain can be presented as follows (see Fig.1)

Case 1: Before packet transmissions,

$$P\{j,k \mid j,k+1\} = 1 \quad k \in [0, 2^j \cdot W_i - 2], j \in [0, m_i] \tag{1.1}$$

Case 2: After packet transmissions,

$$\begin{cases} P\{j+1,k \mid j,0\} = \dfrac{p_i}{2^{j+1} W_i} & (j < m_i), (k \in [0, 2^{j+1} \cdot W_i - 1]) \qquad (2.1) \\ P\{m_i,k \mid m_i,0\} = \dfrac{p_i}{2^{m_i} W_i} & (k \in [0, 2^{m_i} \cdot W_i - 1]) \qquad (2.2) \\ P\{0,k \mid j,0\} = \dfrac{(1-p_i)}{W_i} & (j \in [0, m_i]), (k \in [0, W_i - 1]) \qquad (2.3) \end{cases}$$

Based on the above descriptions of state transitions for traffic flows, we can solve the Markov chain for type i traffic. Let $q_i(j,k)$, $j \in [0,m_i]$ and $k \in [0,2^j W_i - 1]$, be the stationary distribution of the chain. It can be found that all the state probabilities can be expressed by $q_i(0,0)$. Because $\sum_{j=0}^{m_i} \sum_{k=0}^{2^j W_i - 1} q_i(j,k) = 1$, we have:

$$q_i(0,0) = \frac{2(1-2p_i)(1-p_i)}{(1-2p_i)(W_i+1) + p_i W_i [1-(2p_i)^{m_i}]} \tag{3}$$

Let τ_i be the probability that a station carrying type i traffic transmits in a randomly chosen slot time. We have:

$$\tau_i = \sum_{j=1}^{m_i} q_i(j,0) = \frac{2(1-2p_i)}{(1-2p_i)(W_i+1) + p_i W_i [1-(2p_i)^{m_i}]} \tag{4}$$

With the above probabilities defined, we can express packet collision probabilities p_i as:

$$p_i = 1 - (1-\tau_i)^{n_i - 1} \prod_{j=1, j\neq i}^{L} (1-\tau_j)^{n_j} \tag{5}$$

After combining equations (4) and (5) and by using a numerical method, we can get all the values for p_i and τ_i. It can be easily found that this system has a unique set of solutions. For example, assume that there are two different solutions for p_1 and τ_1, i.e., (p_1', τ_1') and (p_1'', τ_1''). Assume that $\tau_1'' < \tau_1'$, from equation (4), we have that $p_1'' > p_1'$, which is because that in equation (4) τ_i increases with the decreasing of p_i [11]. From equation (5), on the contrary, it can be found that we have $p_1'' < p_1'$. Therefore, this system must have a unique set of solutions for $L \geq 1$.

Let P_{tr} be the probability that there is at least one transmission in the considered slot time. We have

$$P_{tr} = 1 - \prod_{j=1}^{L} (1-\tau_j)^{n_j} \tag{6}$$

The probability P_s that there is one and only one transmission occurring on the channel can be given as

$$P_s = \frac{\sum_{j=1}^{L} \left\{ n_j \tau_j (1-\tau_j)^{n_j - 1} \cdot \prod_{k=1, k\neq j}^{L} (1-\tau_k)^{n_k} \right\}}{P_{tr}} \tag{7}$$

Moreover, we define $P_{str,i}$ as the probability that there is one and only one transmission of a traffic flow with type i occurring on the channel, and we have

$$P_{str,i} = n_i \tau_i (1-\tau_i)^{n_i - 1} \cdot \prod_{k=1, k\neq i}^{L} (1-\tau_k)^{n_k} \tag{8}$$

The normalized system throughput S can be expressed as

$$S = \frac{\sum_{i=1}^{L} P_{str,i} \cdot E[P_{Len,i}]}{(1-P_{tr})\sigma + \sum_{i=1}^{L} P_{str,i} \cdot T_{s,i} + (P_{tr} - \sum_{i=1}^{L} P_{str,i}) \cdot T_c} = \sum_{i=1}^{L} S_i \tag{9}$$

where S_i denotes the throughputs contributed by type i traffic flows. $E[P_{Len,i}]$ is the average duration to transmit the payload for type i traffic (the payload size is measured with the time required to transmit it). If all packets of type i traffic have the same fixed size, we have $E[P_{Len,i}] = P_{Len,i}$. σ is the duration of an empty slot time. $T_{s,i}$ is the average time of a slot because of a successful transmission of a packet of a traffic flow with type i. $T_{s,i}$ can be expressed as

$$T_{s,i} = PHY_{header} + MAC_{header} + E[P_{Len,i}] + SIF + \delta + ACK + DIFS + \delta \tag{10}$$

where δ is the propagation delay. T_c is the average time the channel is sensed busy by each station during a collision, and it can be expressed as

$$T_c = PHY_{header} + MAC_{header} + E[P_{c\_Len}] + DIFS + \delta \tag{11}$$

where $E[P_{c\_Len}]$ is the average length of the longest packet payload involved in a collision. With the definition of the set $\Omega_1(k)$ as

$$\Omega_1(k) \equiv \{c_1,...,c_L \mid \sum_{i=1}^{L} c_i = k, 0 \le c_i \le n_i, i = 1,...,L\}$$

$E[P_{c\_Len}]$ can be given as:

$$E[P_{c\_Len}] = \frac{\sum_{k=2}^{n_1+..+n_L} \sum_{\Omega_1(k)} \left\{ \prod_{i=1}^{L} \binom{n_i}{c_i} \cdot \tau_i^{c_i} (1-\tau_i)^{n_i - c_i} \cdot \max\left[\theta(c_1) P_{Len,1},...,\theta(c_L) P_{Len,L}\right] \right\}}{P_{tr} - \sum_{i=1}^{L} P_{str,i}} \tag{12}$$

where

$$\theta(x) \equiv \begin{cases} 1 & x > 0 \\ 0 & x = 0 \end{cases}$$

Next, we analyze the average packet delay. We define delay $T_{D,i}$ as the average time period between the instant of a traffic flow with type i beginning its backoff procedure to transmit a packet to the instant that the packet can be transmitted without collision. Therefore, $T_{D,i}$ dose not include the transmission time for the packet.

It can be easily found that there is a simple relationship between $T_{D,i}$ and the throughput S_i. We have the relation as:

$$s_i \equiv \frac{S_i}{n_i} = \frac{E[P_{Len,i}]}{T_{D,i} + T_{s,i}} \tag{13}$$

Therefore, $T_{D,i}$ can be given as

$$T_{D,i} = \frac{E[P_{Len,i}]}{s_i} - T_{s,i} \tag{14}$$

4 Approximation Analysis

In order to gain a deeper insight into the whole system, we make some approximations to get simpler but more meaningful relationships among different parameters. We start from equation (5) to derive:

$$(1-p_i)(1-\tau_i) = (1-p_j)(1-\tau_j) = \prod_{j=1}^{L}(1-\tau_j)^{n_j} \quad (1 \le i, j \le L) \tag{15}$$

From the above equation, it can be seen that if $\tau_i \ne \tau_j$, then we must have $p_i \ne p_j$. When the minimum contention window size $W_i >> 1$ and $W_j >> 1$, the transmission probabilities τ_i and τ_j are small, that is, $\tau_i << 1$ and $\tau_j << 1$. Therefore, from equation (15), we have the following approximation

$$p_i \approx p_j \tag{16}$$

Furthermore, when $W_i >> 1$, $W_j >> 1$ and $m_i \approx m_j$, we have the following approximation based on equation (4)

$$\frac{\tau_i}{\tau_j} \approx \frac{W_j}{W_i} \tag{17}$$

From equations (5) and (9), we have

$$\frac{S_i}{S_j} = \frac{P_{str,i} \cdot E[P_{Len,i}]}{P_{str,j} \cdot E[P_{Len,j}]} = \frac{n_i\tau_i(1-\tau_j)E[P_{Len,i}]}{n_j\tau_j(1-\tau_i)E[P_{Len,j}]} \approx \frac{n_i\tau_i E[P_{Len,i}]}{n_j\tau_j E[P_{Len,j}]} \approx \frac{n_i W_j E[P_{Len,i}]}{n_j W_i E[P_{Len,j}]} \tag{18}$$

Then, we have

$$\frac{s_i}{s_j} \approx \left(\frac{E[P_{Len,i}]}{W_i}\right) \Bigg/ \left(\frac{E[P_{Len,j}]}{W_j}\right) \tag{19}$$

From the above equation, we can see that the throughput differentiation is mainly determined by the scaling of minimum contention window sizes and the length of packet payloads.

Moreover, from equation (14) and (19), we can see that under the conditions $T_{s,i} << \frac{E[P_{Len,i}]}{s_i}$ and $T_{s,j} << \frac{E[P_{Len,j}]}{s_j}$, which holds when the number of traffic flows $n_i >> 1$ and $n_j >> 1$, we have

$$\frac{T_{D,i}}{T_{D,j}} \approx \frac{\dfrac{E[P_{Len,i}]}{s_i}}{\dfrac{E[P_{Len,j}]}{s_j}} \approx \frac{W_i}{W_j} \tag{20}$$

Equation (20) is another important approximation relationship obtained. From above equation, we can see that packet delay differentiation among different types of traffic flows is mainly determined by the ratio of the corresponding minimum contention window sizes.

Table 1. System parameters

MAC Header	272 bits
PHY Header	192 μs
ACK	112 bits +PHY header
Channel Bit Rate	11Mbps
Propagation Delay	1 μs
Slot Time	20 μs
SIFS	10 μs
DIFS	50 μs

5 Results and Discussions

In this section, we present some simulation and numerical results according to our analysis model. In our examples, we assume that two types of traffic coexist in the system. The parameters for the system are summarized in Table 1 based on IEEE 802.11b.

In Fig. 2 and Fig. 3, we validate our proposed analysis model by comparing simulation results and numerical results. For our simulator, which is implemented by using C++, we consider that there are 20 stations, 10 of them carrying type 1 traffic and the others carrying type 2 traffic. In the simulation, ideal channel conditions (i.e., no hidden terminals and capture) are assumed. The other parameters are set as follows: $W_2 = 1024$, $E[P_{Len,1}] = 2000\,bytes$, $E[P_{Len,2}] = 1000\,bytes$, $m_1 = m_2 = 8$. Different simulation values are obtained by varying the minimum contention window size W_1. Each simulation value is obtained by running our simulator to simulate the actual behavior of the system within the period of 30 minutes. In Fig. 2, the total system throughput S and throughput S_1 are shown versus W_1. In Fig. 3, average packet delays $T_{D,1}$ and $T_{D,2}$ are shown versus W_1. As expected, when W_1 decreases, traffic flows with type 1 occupy larger portion of channel resources. From these two figures, we can see that the simulation results agree so well with the theoretical ones that they overlap, especially in the case of larger W_1, which justifies the assumption made in section III.

In Fig. 4 to Fig. 5, we keep the total number of traffic flows constant, and we change the number n_1 of traffic flows with type 1. In Fig. 4, throughputs s_1 and s_2 versus the number of traffic flows n_1 are shown with the variation of W_1. In Fig. 5, packet delays $T_{D,1}$ and $T_{D,2}$ are shown with the variation of W_1. We can see that when W_1 decreases, traffic flows with type 1 gain priority over type 2 traffic flows: throughput s_1 becomes larger than s_2, and packet delay $T_{D,1}$ becomes smaller than

$T_{D,2}$. However, in the case of large n_1 (such as $n_1 > 40$), both the performance on throughput and packet delays are worse than the case of $W_1 = W_2 = 256$, which indicates that providing service differentiation with very large number of traffic flows belonging to the higher priority group makes the system performance worse than in the case of no service differentiation support. The reason is that with the increase of n_1, collision rates p_1 and p_2 increase drastically, reducing the bandwidth utilization. If the number of traffic flows with higher priority is sufficiently small, both throughput and packet delays for higher priority traffic are improved significantly with only small influence on traffic flows with lower priority. Therefore, the number of traffic flows with higher priority must be strictly controlled to only small proportions of the total number of traffic flows by suitable access control schemes.

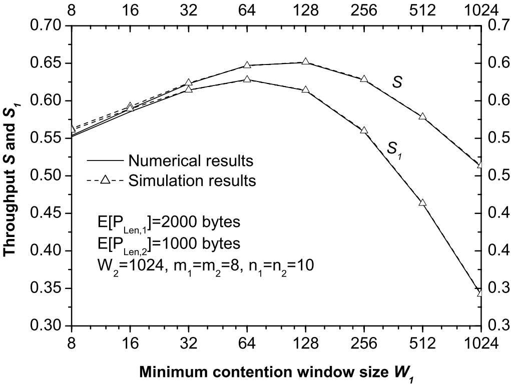


Fig. 2 Throughput S and S_1 versus W_1

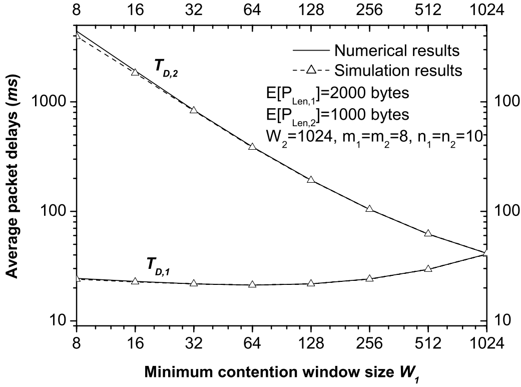


Fig. 3 Average packet delays $T_{D,1}$ and $T_{D,2}$ versus W_1

6 Conclusions

We propose an analysis model to compute the throughput and packet transmission delays in a WLAN with Enhanced IEEE 802.11 Distributed Coordination Function, which supports service differentiation. In our analytical model, service differentiation is supported by scaling the contention window and the packet length according to the priority of each traffic flow. Comparisons with simulations results show that good accuracy of performance evaluations can be achieved by using the proposed analysis model.

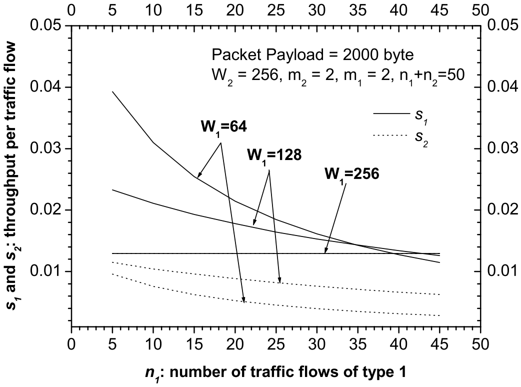


Fig. 4 Throughput s_1 and s_2 versus the number of traffic flows n_1

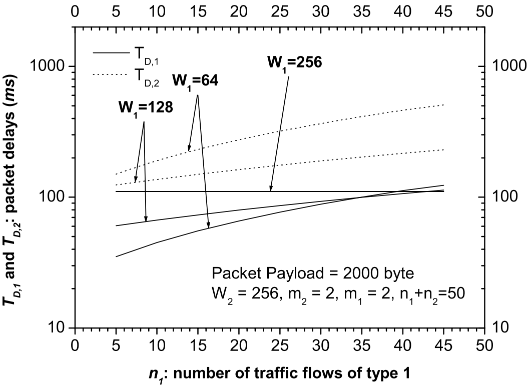


Fig. 5 Packet delays $T_{D,1}$ and $T_{D,2}$ versus the number of traffic flows n_1

References

1. L. Bos and S. Leroy, "Toward an All-IP-Based UMTS System Architecture," IEEE Network, vol. 15, No. 1, Jan, 2001, pp. 36-45.
2. Y. Cheng and W. H. Zhuang, "DiffServ Resource Allocation for Fast Handoff in Wireless Mobile Internet," IEEE Communications Magazine, vol. 40, No. 5, 2002, pp. 130-136.
3. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, IEEE Standard 802.11, Aug. 1999.

4. A. Veres, A. T. Campbell, M. Barry and L. H. Sun, "Supporting Service Differentiation in Wirelss Packet Networks Using Distributed Control," IEEE Journal on Selected Areas In Communications, Vol. 19, No. 10, Oct 2001, pp. 2081-2093.
5. Aad and C. Castelluccia, "Differentiation Mechanisms for IEEE 802.11," IEEE Inforcom 2001, pp. 209-218.
6. S. Mangold, S. Choi, P. May, O. Klein, G. Hietz and L. Stibor, “IEEE 802.11e wireless lan for quality of service,” *Proceedings of the European Wireless*, 2002 Feb.
7. Weinmiller, M. Schlager, A. Festag, and A. Wolisz, “Performance study of access control in wireless LANs IEEE 802.11 DFWMAC and ETSI RES 10 HIPERLAN,” Mobile Networks and Applications, vol. 2, pp. 55-67, 1997.
8. H. S. Chhaya and S. Gupta, "Performance modeling of asynchronous data transfer methods of IEEE 802.11 MAC protocol," Wireless Networks, vol. 3, pp. 217-234, 1997.
9. T. S. Ho and K. C. Chen, "Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LAN's," in Proc. IEEE PIMRC, Taipei, Taiwan, Oct. 1996, pp.392-396.
10. F. Cali, M. Conti, and E. Gregori, "IEEE 802.11 wireless LAN: Capacity analysis and protocol enhancement," Proc. INFOCOM'98, San Francisco, CA, March 1998, vol. 1, pp. 142 -149.
11. G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," IEEE Journal on Selected Areas In Communications, Vol. 18, No. 3, March 2000.
12. Y. C. TAY and K. C. CHUA, "A Capacity Analysis for the IEEE 802.11 MAC Protocol," Wireless Networks, 7, 2001, pp. 159-171.
13. M. Benveniste, G. Chesson, M. Hoehen, A. Singla, H. Teunissen, and M.Wentink, “EDCF proposed draft text,” IEEE working document 802.11-01/131r1, March 2001.

Scheduling Time-Sensitive Traffic on 802.11 Wireless LANs

Martin Heusse, Paul Starzetz, Franck Rousseau,
Gilles Berger-Sabbatel, and Andrzej Duda

LSR-IMAG Laboratory
BP. 72, 38402 Saint Martin d'Hères, France
{heusse,starzetz,rousseau,gberger,duda}@imag.fr
http://www-lsr.imag.fr

Abstract. In contrast to the common wisdom stating that 802.11 wireless LANs are not suitable for time-sensitive traffic, we have observed that in some conditions packet traffic transmitted over 802.11b may benefit from low delays even in saturation. Our analysis and measurements show that low delays can be obtained irrespectively of the greedy behavior of other hosts and without any traffic control mechanisms: when some hosts try to gain as much as possible of the transmission capacity of the radio channel, it is still possible for other hosts to experience low delay provided their packet rates are below some threshold value. The only situation in which a time-sensitive traffic source fails to obtain low delay is when its packet rate is too high with respect to its share of the channel capacity. We provide an analytical formula for determining the limiting packet rate that can be used to guide rate adaptive applications such as audio or video codecs to keep their output rates under the limiting rate and benefit in this way from low delays without any coordinated traffic control mechanisms.

1 Introduction

The common wisdom concerning time-sensitive traffic over wireless LANs such as 802.11 states that this kind of communication links cannot provide low latency [6, 12, 11]. Usually, it is assumed that the delay may be fairly long because of the Distributed Coordination Function (DCF) based on the CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) medium access method. Such multiple access randomized protocols are considered as not suitable for time-sensitive traffic. Another access method defined in the 802.11 standard, the Point Coordination Function (PCF) oriented towards time-bounded services, is not implemented in most of current products. Many proposals try to alleviate this problem by modifying the MAC layer [2] [3] [7].

In this paper, we show that hosts generating time-sensitive traffic in a 802.11 cell may benefit from low delays even in saturation conditions. We observe that low delays can be obtained irrespectively of the greedy behavior of other hosts and without any traffic control mechanisms: even if some hosts try to gain as

G. Karlsson and M.I. Smirnov (Eds.): QofIS 2003, LNCS 2811, pp. 162–171, 2003.

much as possible of the transmission capacity of the radio channel, other hosts may experience low delays provided their packet rates are below some threshold value.

We consider the case of several hosts that generate traffic over the wireless channel of 802.11. Some hosts send high priority time-sensitive flows that require low delay while generating packets with relatively small rate (for example H.261 typical rates start at 64 kbit/s and increase in multiples of 64 kbit/s). They compete for the radio channel with other hosts that do not care about the delay, but present a greedy behavior by trying to gain as much of the available bandwidth as possible.

In another work [10], we have proposed to use the DiffServ model to provide QoS differentiation at the IP level over the standard DCF method of 802.11. By scheduling packets according to their DiffServ classes (BE, AF, EF) and by constraining the output rate of each host via DiffServ traffic shaping mechanisms, we can keep the 802.11 network in the state of non-saturation so that the time critical high priority EF class benefits from stable short delays. Achieving such service differentiation requires traffic control mechanisms and collaboration of all hosts in a cell, for example a coordinator at the access point may configure the DiffServ mechanisms implemented in all hosts to reflect current allocations of the available bandwidth to aggregated traffic classes [8].

We begin with the analysis of the channel utilization in the 802.11 cell. The analytical results provide us with a limiting packet rate: if a host keeps its traffic below this limit, even if other hosts try to gain as much as possible, the host will experience short delays. We then verify experimentally this behavior. In our setup, we measure the throughput and the delay of two hosts in a wireless cell that generate traffic of different classes. We designate different traffic classes according the DiffServ model [5]: one host generates high priority EF traffic and the other one lower priority AF traffic. We use token buckets to control the source rates for which we want to measure the performance indices (they are only used for measurements, they are not needed for obtaining low delays). The experience confirms the analysis showing that if the channel is saturated by the lower priority AF class, it is still possible for the EF class to benefit from low delay provided that the EF packet rate remains lower than the limiting rate. The only situation in which the EF class fails to obtain low delay is when its packet rate is too high with respect to its share of the channel capacity. Note that the results of this paper can be used to guide rate adaptive applications such as audio or video codecs to keep their output rates under the limiting rates and benefit in this way from low delays without any coordinated traffic control mechanisms.

Our results show that the important parameter for scheduling traffic over the 802.11 WLAN is the packet rate and not the overall throughput. The importance of the packet rate results from the fairness properties of the CSMA/CA access method—in fact hosts in 802.11 share the channel capacity according to equal packet rates and not equal throughput shares [9, 4].

The paper is structured as follows. First, we analyze the utilization in 802.11b to derive the limiting packet rate (Section 2). Then, we describe the setup of the measurement experiments (Section 3). We show the performance results in a saturated cell (Section 4). Finally, we present some conclusions (Section 5).

2 Limiting Packet Rate in 802.11b

In this section, we model the behavior of a 802.11b cell [1] with hosts sending packets of different sizes to derive the limiting packet rate. The results of this paper follow up the analysis of the 802.11 performance anomaly [9], in which we have derived simple expressions for the useful throughput, validated them by means of simulation, compared with several performance measurements, and analyzed the performance of the 802.11b cell when one slow host (transmitting at a degraded rate e.g. 1 Mbit/s) competes with other fast hosts. Here, we modify the model to take into account different packet sizes and rates.

The DCF access method of 802.11b is based on the CSMA/CA principle in which a host wishing to transmit senses the channel, waits for a period of time (DIFS – Distributed Inter Frame Space) and then transmits if the medium is still free. If the packet is correctly received, the receiving host sends an ACK frame after another fixed period of time (SIFS – Short Inter Frame Space). If this ACK frame is not received by the sending host, a collision is assumed to have occurred. The sending host attempts to send the packet again when the channel is free for a DIFS period augmented of a random amount of time.

If there are multiple hosts attempting to transmit, the channel may be sensed busy and hosts enter a collision avoidance phase: a host waits for a random interval distributed uniformly over $\{0, 1, 2, ...CW - 1\} \times SLOT$. The congestion window CW varies between $CW_{\text{min}} = 32$ and $CW_{\text{max}} = 1024$, the value of $SLOT$ is 20 μs (these parameters are for 802.11b). The host that chooses the smallest interval starts transmitting and the others freeze their intervals until the transmission is over. When hosts choose the same value of the random interval, they will try to transmit at the same slot, which results in a collision detected by the missing ACK frame (only the transmitting hosts may detect a collision). Each time a host happens to collide, it executes the exponential backoff algorithm – it doubles CW up to CW_{max}.

We assume that each host i sends packets of size s_{i} at rate x_{i} packets per second. The frame transmission time depends on the size: $t_{\text{tr}} = s_{\text{i}}/R$, where R is the nominal transmission rate (11 Mbit/s for 802.11b). The overall frame transmission time experienced by a single host when competing with $N - 1$ other hosts can be expressed as:

$$T_{\text{i}} = t_{\text{ov}} + \frac{s_{\text{i}}}{R} + t_{\text{cont}}.$$

where the constant overhead

$$t_{\text{ov}} = DIFS + t_{\text{pr}} + SIFS + t_{\text{pr}} + t_{\text{ack}}$$

is composed of the PLCP (Physical Layer Convergence Protocol) preamble and header transmission time $t_{\mathrm{pr}} = 96\ \mu s$ (short PLCP header), $SIFS = 10\ \mu s$, t_{ack} is the MAC acknowledgment transmission time (10 μs if the rate is 11 Mbit/s as the ACK length is 112 bits), and $DIFS = 50\ \mu s$.

Under high load, to evaluate the impact of contention, we consider that the hosts always sense a busy channel when they attempt to transmit and that the number of transmissions that are subject to multiple successive collisions is negligible. In this case, we find:

$$t_{\mathrm{cont}}(N) \simeq SLOT \times \frac{1 + P_{\mathrm{c}}(N)}{N} \times \frac{CW_{\mathrm{min}}}{2},$$

where $P_{\mathrm{c}}(N)$ is the proportion of experienced collisions for each packet successfully acknowledged at the MAC level ($0 \leqslant P_{\mathrm{c}}(N) < 1$).

A simple expression for $P_{\mathrm{c}}(N)$ can be derived by considering that a host attempting to transmit a frame will eventually experience a collision if the value of the chosen backoff interval corresponds to the residual backoff interval of at least one other host. Such an approximation holds if multiple successive collisions are negligible. So we have

$$P_{\mathrm{c}}(N) = 1 - (1 - 1/CW_{\mathrm{min}})^{N-1}. \tag{1}$$

At this point we have all the elements of T_{i}, the global transmission time of host i. Now we want to find the overall performance—the channel utilization when hosts transmit packets at rate x_{i} while alternating transmissions. The utilization will determine the limiting packet rate beyond which the network enters the saturation state. We can evaluate the channel utilization by considering that host i uses the channel with rate x_{i} during time T_{i} as:

$$U = \sum_{i=1}^{N} x_{\mathrm{i}} T_{\mathrm{i}} + x_{\mathrm{coll}}\ T_{\mathrm{coll}}, \tag{2}$$

where $x_{\mathrm{coll}}, T_{\mathrm{coll}}$ are the collision rate and the time spent in collisions, respectively. If all hosts are greedy, their rates in the saturation state will be equal, so the limiting rate can be found from:

$$x^{\mathrm{sat}} \sum_{i=1}^{N} T_{\mathrm{i}} + x_{\mathrm{coll}}\ T_{\mathrm{coll}} = 1, \tag{3}$$

which finally yields:

$$x^{\mathrm{sat}} = \frac{1 - x_{\mathrm{coll}}\ T_{\mathrm{coll}}}{\sum_{i=1}^{N} T_{\mathrm{i}}}. \tag{4}$$

$x_{\mathrm{coll}}, T_{\mathrm{coll}}$ can be easily found for the case of two stations. To make the comparison with experimental results easier, we identify hosts by their type of traffic: host 1 generates time-sensitive EF traffic while host 2 generates AF packets:

$$x_{\mathrm{coll}} = x^{\mathrm{sat}} P_{\mathrm{c}}(2), \tag{5}$$

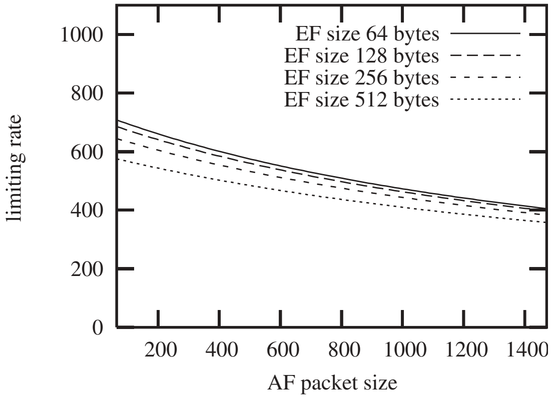


Fig. 1. Limiting packet rate for two hosts and different packet sizes.

$$T_{\text{coll}} = \max(T_{\text{EF}}, T_{\text{AF}}) \tag{6}$$

so for $N = 2$, and assuming that AF packets are longer or equal to EF packets, we obtain the following formula for the limiting rate:

$$x^{\text{sat}} = \frac{1}{T_{\text{EF}} + [1 + P_{\text{c}}(2)]T_{\text{AF}}}. \tag{7}$$

Figure 1 presents the limiting rate for two hosts in function of different packet sizes.

For $N > 2$, a simple approximation consists of not taking into account collisions. In this case we obtain the following upper bound for the limiting packet rate:

$$x^{\text{sat}} = \frac{1}{\sum_{i=1}^{N} T_{\text{i}}}. \tag{8}$$

3 Experimental Setup

We have set up a platform to measure the delay and the throughput that hosts can obtain when sharing a 11 Mbit/s 802.11b wireless channel. We have used two notebooks running Linux RedHat 8.0 (kernel 2.4.20) with 802.11b cards based on the same chipset (Lucent Orinoco and Compaq WL 110). The wired part of the network is connected by an access point based on a PC box (SuSE 7.3) running software access point `hostap`. The notebooks use the Wvlan driver for the wireless cards. The cards do not use the RTS/CTS option that may optimize performance in case of the hidden terminal problem.

To avoid interferences in the use of the wireless channel, we measure the round trip time (RTT) in a configuration in which a host sends a packet over 802.11b and the reply returns via another interface (100 Mbit/s Ethernet). Figure 2 presents the experimental setup.

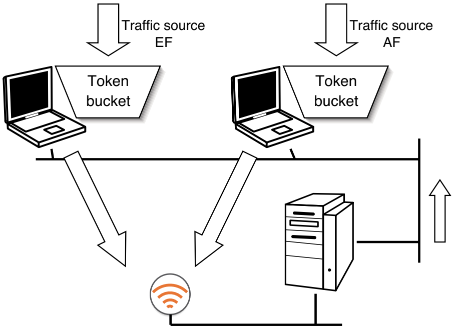


Fig. 2. Experimental setup.

As said previously, we limit the experimental study to two hosts designated according to their type of traffic: the EF host sends time-sensitive traffic of a given packet rate whereas the AF host will try to increase its traffic as much as possible starting from 256 kbit/s to 10 Mbit/s in steps of 256 kbit/s.

The measurement results in the rest of the paper are presented in function of the *offered load*, which is the sum of the EF and AF traffic in kbit/s. The packet size given in figures corresponds to the UDP payload size.

4 Performance in Saturation Conditions

In this section we provide experimental results in saturation conditions. Figures 3, 4, 5 present the RTT of the EF class transmitting at different rates (128, 256, 512 kbit/s) when competing with the AF class. We can see that when the EF packet rate is small (128 kbit/s, 64 byte packets means 250 p/s packet rate), the RTT of the EF class remains small (under 6 ms) even if the cell is already saturated (offered load increased to 10 Mbit/s). As the limiting rate is 383 p/s for 1472 byte AF packets (cf. Eq. 7) and more for shorter packets, the EF class of 250 p/s packet rate will always benefit from low delays.

We can also see that for 512 byte AF packets and 256 kbit/s EF traffic (500 p/s packet rate), the delay is still short, because the limiting rate for this case is 644 p/s. Figure 5 shows the case in which the delay becomes very high due to queueing delays—for 1472 byte AF packets the limiting rate is 383 p/s, so the EF packet rate of 1000 p/s (512 kbit/s with 64 byte packets) is too high.

These result show that even if the channel is saturated by AF traffic, it is still possible for the EF class to benefit from low delay provided that the EF packet rate remains lower than the limiting packet rate. The reason for this behavior is the basic CSMA/CA channel access method which provides good fairness properties (contrary to the common wisdom concerning the fairness of 802.11 [4]) —the channel access probability is equal for all competing hosts. Hence at saturation, each competing class obtains an equal packet rate. And when the classes use different packet sizes, the throughput of each class may be

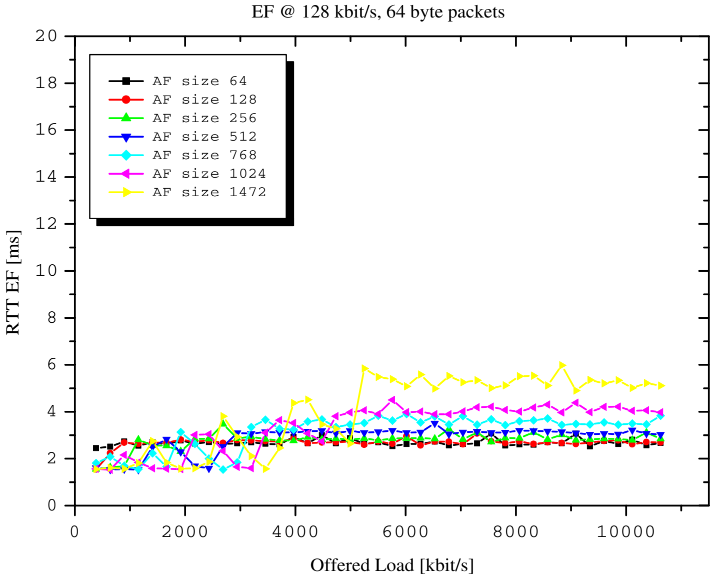


Fig. 3. RTT of the EF class for increasing offered load, constant 128 Kb/s EF traffic.

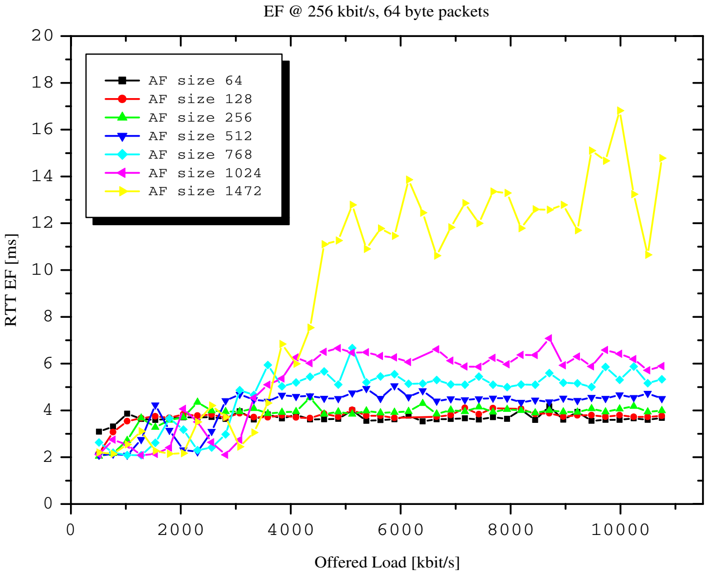


Fig. 4. RTT of the EF class for increasing offered load, constant 256 Kb/s EF traffic.

different. So, even if the AF class sends packets with a rate exceeding its packet rate share, the EF class still benefits from its share of the packet rate. If the EF rate is lower than the packet rate share, its delay remains small.

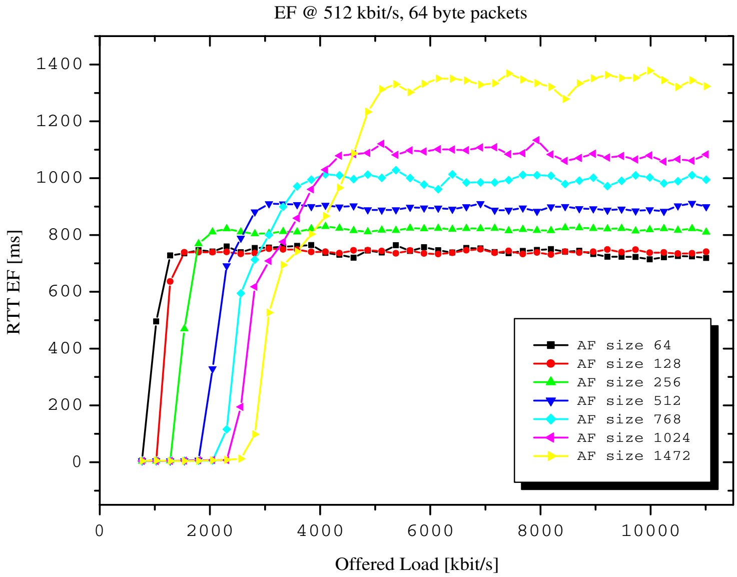


Fig. 5. RTT of the EF class for increasing offered load, constant 512 Kb/s EF traffic.

The only situation in which the EF class fails to obtain a low delay is when the host packet rate is greater than its packet rate share. Figure 6 illustrates this case in a similar setup: for the constant EF rate of 1024 kbit/s, 64 byte packets, we increase the rate of the AF class for different packet sizes. This rate of the EF class corresponds to the packet rate of 2000 p/s, which is greater than the limiting rate for any AF packet size. We can observe from the figure that the EF class does not obtain this rate, so that the RTT increases because of queueing delays: the corresponding RTT measurements appear in figure 7. We can also observe how the packet rates of both classes tend towards equal values when the cell becomes saturated.

5 Conclusions

The analysis and measurements in this paper show that the time-sensitive EF class may benefit from low delays irrespectively of the greedy behavior of other hosts and without any traffic control mechanisms. The only condition for obtaining such desired behavior is to keep the packet rate under the limiting value that we have analytically derived in Section 2. The only situation in which the EF class fails to obtain low delay is when its packet rate is too high with respect to its packet rate share.

Our results show that the packet rate is the most important parameter for QoS guarantees on the 802.11 WLAN. Its importance results from the fact that in the CSMA/CA access method, every time the EF host has a packet to transmit, it will contend with other hosts and gain the channel with probability $1/N$, where N is the number of hosts wanted to send a packet. If it does not succeed,

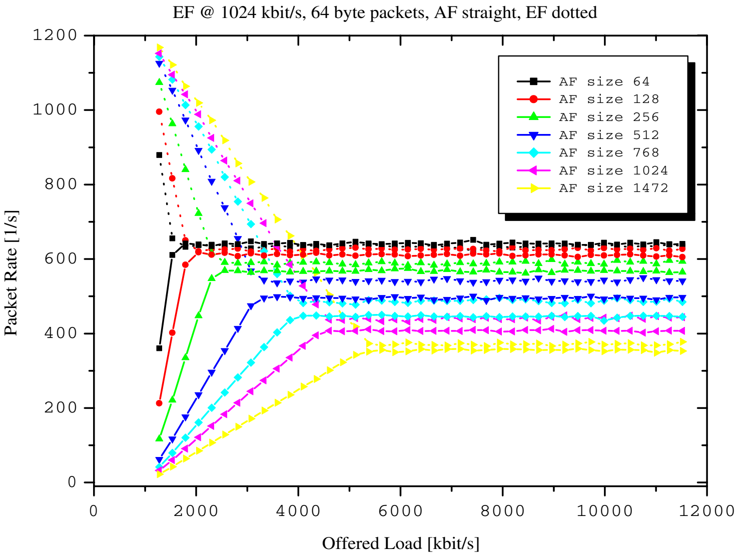


Fig. 6. Packet rates obtained by the AF and EF traffic for increasing offered load.

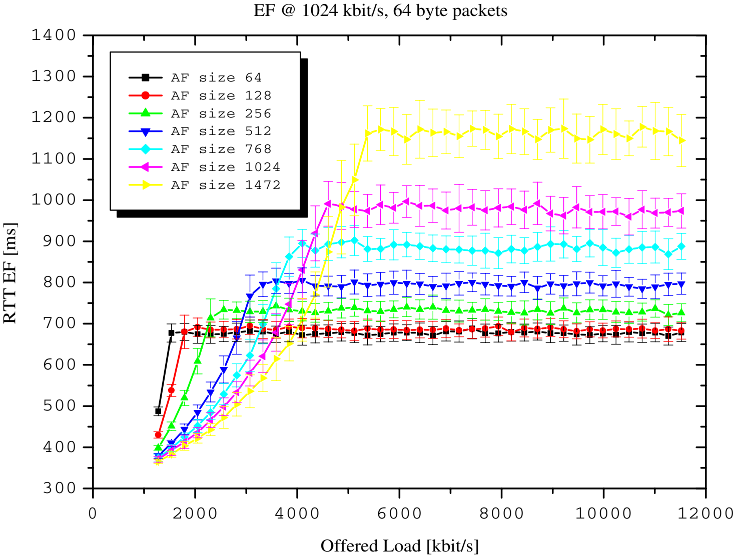


Fig. 7. RTT of the EF class, for an increasing offered load. High relative EF packet rate.

it will attempt another time with a higher probability than the host that has gained the channel: its residual contention interval is smaller on the average than the contention interval of the successful host. Note also that our results apply to other variants of WLANs such as 802.11a and 802.11g, because they use the same MAC access method as 802.11b.

The results of this paper show that it is possible to provide some QoS guarantees over the standard DCF method of 802.11. In particular, adaptive applications such as audio or video codecs can keep their output rates under the limiting packet rate and benefit in this way from low delays without any coordinated traffic control mechanisms.

References

1. ANSI/IEEE. 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2000.
2. A. Banchs, M. Radimirsch, and X. Perez. Assured and expedited forwarding extensions for IEEE 802.11 wireless LAN. In *Tenth IEEE International Workshop on Quality of Service*, pages 237–246, 2002.
3. M. Barry, A.T. Campbell, and A. Veres. Distributed control algorithms for service differentiation in wireless packet networks. In *INFOCOM 2001*, volume 1, pages 582 –590, 2001.
4. Gilles Berger-Sabbatel, Andrzej Duda, Olivier Gaudoin, Martin Heusse, and Franck Rousseau. On the Fairness of 802.11. In *submitted for publication*, 2003.
5. S. Blake et al. An Architecture for Differentiated Services, 1998. `Internet RFC 2475`.
6. K.C. Chen. Medium access control of wireless LANs for mobile computing. *IEEE Network*, 8(5):50–63, 1994.
7. Garg, P. et al. Using IEEE 802.11e MAC for QoS over Wireless. In *IPCCC 2003*, Phoenix USA, 2003.
8. J. Antonio García-Macías, Franck Rousseau, Gilles Berger-Sabbatel, Toumi Leyla, and Andrzej Duda. Quality of Service and Mobility for the Wireless Internet. *Wireless Networks*, 9(4):341–352, 2003.
9. Martin Heusse, Franck Rousseau, Gilles Berger-Sabbatel, and Andrzej Duda. Performance Anomaly of 802.11b. In *Proceedings of IEEE INFOCOM 2003*, San Francisco, USA, 30– 3 2003.
10. Martin Heusse, Paul Starzetz, Franck Rousseau, Gilles Berger-Sabbatel, and Andrzej Duda. Bandwidth Allocation for DiffServ based Quality of Service over 802.11b. In *Proceedings of IEEE Globecom 2003*, San Francisco, USA, 2003.
11. L. Romdhani, Q. Ni, and T. Turletti. Adaptive EDCF: Enhanced Service Differentiation for IEEE 802.11 Wireless Ad Hoc Networks. In *WCNC'03*, New Orleans, USA, 2003.
12. J.L. Sobrinho and A.S. Krishnakumar. Real-time traffic over the IEEE 802.11 MAC. *Bell Labs Technical Journal*, 1(2):172–187, 1996.

Throughput Analysis of a Probabilistic Topology-Unaware TDMA MAC Policy for Ad-hoc Networks

Konstantinos Oikonomou[1] and Ioannis Stavrakakis[2]

[1] INTRACOM S.A., Development Programmes Department
19.5 Markopoulou Avenue, Paiania 190 02, Athens, Greece
okon@intracom.gr
Tel: +30 210 6677023, Fax: +30 210 6671312

[2] University of Athens, Department of Informatics & Telecommunications
Panepistimiopolis, Ilissia 15 784, Athens, Greece
istavrak@di.uoa.gr
Tel: +30 210 7275343, Fax: +30 210 7275333

Abstract. The existing topology-unaware TDMA-based schemes are suitable for ad-hoc networks and capable of providing a minimum guaranteed throughput by considering a deterministic policy for the utilization of the assigned scheduling time slots. In an earlier work, a probabilistic policy that utilizes the non-assigned slots according to an access probability, common for all nodes in the network, was proposed. The achievable *throughput for a specific transmission* under this policy was analyzed. In this work, the *system throughput* is studied and the conditions under which the system throughput under the probabilistic policy is higher than that under the deterministic policy and close to the maximum are established.

1 Introduction

Ad-hoc networks require no infrastructure and nodes are free to enter, leave or move inside the network without prior configuration. This flexibility introduces new challenges and makes the design of an efficient Medium Access Control (MAC) a challenging problem. CSMA/CA-based MAC protocols have been proposed, [1], [2], [3], whereas others have employed handshake mechanisms like the Ready-To-Send/Clear-To-Send (RTS/CTS) mechanism [4], [5], to avoid the *hidden/exposed terminal* problem.

Chlamtac and Farago, [6], as well as Ju and Li, [7], have proposed an original TDMA-based scheme for topology transparent scheduling. However, both schemes employ a deterministic policy for the utilization of the assigned time slots that fails to utilize non-assigned time slots that could result in successful transmissions, as it is shown here.

In a previous work, [8], the general approach proposed in [6] and [7] (to be referred to as the *Deterministic Policy*) was considered and the idea of allowing the nodes to utilize (according to a common access probability) scheduling slots

G. Karlsson and M.I. Smirnov (Eds.): QofIS 2003, LNCS 2811, pp. 172–181, 2003.

not originally assigned (according to the rules in [6], [7]) to them, was introduced. In this work, the *system throughput* under the Probabilistic Policy is analyzed extensively and the conditions for the existence of an efficient range of values for the access probability are established, based on a properly defined topology density metric.

In Section 2, a general ad-hoc network is described as well as both the Probabilistic Policy and the Deterministic Policy. In Section 3, the preliminary system throughput analysis shows that it is difficult or impossible, in the general case, to fully analyze it. An approximate analysis is presented in Section 4 that establishes the conditions for the existence of an efficient range of values for the access probability. Furthermore, this analysis determines the maximum value for the system throughput and the corresponding value for the access probability; bounds on the latter probability are determined analytically as a function of the topology density. In Section 5, the accuracy of the approximate analysis is studied. Simulation results, presented in Section 6, show that a value for the access probability that falls within the bounds, as they are determined based on the topology density, results in a system throughput that is close to the maximum. Section 7 presents the conclusions.

2 Scheduling Policies

An ad-hoc network may be viewed as a time varying multihop network and may be described in terms of a graph $G(V, E)$, where V denotes the set of nodes and E the set of links between the nodes at a given time instance. Let $|X|$ denote the number of elements in set X and let $N = |V|$ denote the number of nodes in the network. Let S_u denote the set of neighbors of node u, $u \in V$. These are the nodes v to which a direct transmission from node u (transmission $u \to v$) is possible. Let D denote the maximum number of neighbors for a node; clearly $|S_u| \leq D$, $\forall u \in V$.

The transmission(s) that corrupts transmission $u \to v$ may or may not be successful itself. Specifically, in the presence of transmission $u \to v$, transmission $\chi \to \psi$, $\chi \in S_v \cup \{v\} - \{u\}$ and $\psi \in S_\chi \cap (S_u \cup \{u\})$, is corrupted. If $\psi \in S_\chi - (S_\chi \cap (S_u \cup \{u\}))$, then transmission $\chi \to \psi$ is not affected by transmission $u \to v$.

Under the Deterministic Policy, [6], [7], each node $u \in V$ is randomly assigned a unique polynomial f_u of degree k with coefficients from a finite Galois field of order q ($GF(q)$). Polynomial f_u is represented as $f_u(x) = \sum_{i=0}^{k} a_i x^i (\text{mod } q)$ [7], where $a_i \in \{0, 1, 2, ..., q-1\}$; parameters q and k are calculated based on N and D, according to the algorithm presented either in [6] or [7]. For both algorithms it is satisfied that $k \geq 1$ and $q \geq kD + 1$ (k and D are integers).

The access scheme considered is a TDMA scheme with a frame consisted of q^2 time slots. If the frame is divided into q subframes s of size q, then the time slot assigned to node u in subframe s, ($s = 0, 1, ..., q-1$) is given by $f_u(s) \text{mod } q$ [7]. Let the set of time slots assigned to node u be denoted as Ω_u. Clearly, $|\Omega_u| = q$. The deterministic transmission policy, [6], [7], is the following.

The Deterministic Policy: Each node u transmits in a slot i only if $i \in \Omega_u$, provided that it has data to transmit.

Depending on the particular random assignment of the polynomials, it is possible that two nodes be assigned overlapping time slots (i.e., $\Omega_u \cap \Omega_v \neq \emptyset$). Let $C_{u\to v}$ be the set of overlapping time slots between those assigned to node u and those assigned to any node $\chi \in S_v \cup \{v\} - \{u\}$. $C_{u\to v}$ is given by:

$$C_{u\to v} = \Omega_u \cap \left(\bigcup_{\chi \in S_v \cup \{v\} - \{u\}} \Omega_\chi \right) . \tag{1}$$

Let $R_{u\to v}$ denote the set of time slots i, $i \notin \Omega_u$, over which transmission $u \to v$ would be successful. Equivalently, $R_{u\to v}$ contains those slots not included in set $\bigcup_{\chi \in S_v \cup \{v\}} \Omega_\chi$. Consequently,

$$|R_{u\to v}| = q^2 - \left| \bigcup_{\chi \in S_v \cup \{v\}} \Omega_\chi \right| . \tag{2}$$

$R_{u\to v}$ is the set of *non-assigned eligible* time slots for transmission $u \to v$; if such slots are used by transmission $u \to v$, the probability of success for the particular transmission could be increased.

Theorem 1. *$|R_{u\to v}|$ is greater than or equal to $q(k-1)D$.* □

The proof of Theorem 1 is included in [8].

From Theorem 1 it is obvious that for $k > 1$, $|R_{u\to v}| > qD$. Consequently, the number of non-assigned eligible slots may be quite significant for the cases where $k > 1$ (this case corresponds to large networks, [7]). Even for the case where $k = 1$, $|R_{u\to v}| \geq 0$, that is, $|R_{u\to v}|$ can still be greater than zero. For those nodes for which the set of overlapping slots is not the largest possible $\left(\text{i.e., } \left| \bigcup_{\chi \in S_v \cup \{v\}} \Omega_\chi \right| < (|S_v| + 1)q\right)$, $|R_{u\to v}|$ is greater than zero, even for $k = 1$. Furthermore, if the neighborhood of node v is not dense, or $|S_v|$ is small compared to D, then $|R_{u\to v}|$ is even higher.

In general, the use of slots i, $i \in R_{u\to v}$, may increase the average number of successful transmissions, as long as $R_{u\to v}$ is determined and time slots $i \in R_{u\to v}$ are used efficiently.

The Probabilistic Policy: Each node u always transmits in slot i if $i \in \Omega_u$ and transmits with probability p in slot i if $i \notin \Omega_u$, provided it has data to transmit.

The Probabilistic Policy does not require specific topology information (e.g., knowledge of $R_{u\to v}$, etc.) and, thus, induces no additional control overhead. The access probability p is a simple parameter common for all nodes.

3 System Throughput

The *throughput for a specific transmission* under the Probabilistic and Deterministic policies was investigated in [8]. In this section the expressions for the *system throughput* under both policies are provided and the conditions under which the Probabilistic Policy outperforms the Deterministic Policy are derived. When these conditions are satisfied it is shown that there exists an *efficient* range of values for p (such that the system throughput under the Probabilistic Policy is higher than that under the Deterministic Policy). The analysis assumes heavy traffic conditions; that is, there is always data available for transmission at each node, for every time slot.

Let $P_{D,succ}$ ($P_{P,succ}$) denote the probability of success of a transmission (averaged over all transmissions) under the Deterministic (Probabilistic) Policy (be referred to as the *system throughput* for both policies) assuming that each node u may transmit to only one node $v \in S_u$ in one frame. According to the work presented in [8], it can be concluded that $P_{D,succ}$ and $P_{P,succ}$ are given by the following equations.

$$P_{D,succ} = \frac{1}{N} \sum_{\forall u \in V} \frac{q - |C_{u \to v}|}{q^2}, \tag{3}$$

$$P_{P,succ} = \frac{1}{N} \sum_{\forall u \in V} \frac{q - |C_{u \to v}| + p|R_{u \to v}|}{q^2} (1-p)^{|S_v|}, \tag{4}$$

where $v \in S_u$. From Equation (4) it can be seen that for $p = 0$, $P_{P,succ} = P_{D,succ}$, while for $p = 1$, $P_{P,succ} = 0$. In general, $P_{P,succ}$ may or may not be greater than $P_{D,succ}$. Consequently both equations have to be analyzed to establish the conditions under which $P_{P,succ} \geq P_{D,succ}$.

Theorem 2. *Provided that* $\sum_{\forall u \in V} \left(|R_{u \to v}| - (q - |C_{u \to v}|)|S_v|\right) \geq 0$ *is satisfied, there exist an efficient range of values for p of the form* $[0, p_{max}]$*, for some* $0 \leq p_{max} < 1$*.* □

The proof of this theorem can be extracted by calculating and analyzing the first derivative of $P_{P,succ}$ with respect to p.

The following theorem is based on Theorem 2 for "large networks" ($k > 1$, see [7]).

Theorem 3. *For* $k > 1$*, there always exists an efficient range of values for p of the form* $[0, p_{max}]$*, for some* $0 \leq p_{max} < 1$*, for which* $P_{P,succ} \geq P_{D,succ}$*.* □

The proof of Theorem 3 can be easily extracted from a relevant theorem presented in [8].

Theorem 2 and Theorem 3 establish the conditions for the existence of an efficient range of values for the access probability p of the form $[0, p_{max}]$, for some $0 \leq p_{max} < 1$. Given that Equation (4) is difficult or impossible to be analyzed for $D > 1$ (correspond to a polynomial of degree greater than 2), respectively, an approximate analysis is considered and presented in the following section.

4 Approximate Analysis

The approximate analysis presented in this section is based on a polynomial that is more tractable than that in Equation (4). Let the system throughput $P_{P,succ}$, be approximated by $\tilde{P}_{P,succ}$:

$$\tilde{P}_{P,succ} = \frac{1}{N} \sum_{\forall u \in V} \frac{q - |C_{u \to v}| + p|R_{u \to v}|}{q^2} (1-p)^{\overline{|S|}}, \tag{5}$$

where $\overline{|S|} = \frac{1}{N} \sum_{\forall u \in V} |S_u|$, denotes the *average number of neighbor nodes.* Let $\overline{|S|}/D$ be referred to as the *topology density.* For a given pair of N and D, numerous topologies exist that can be categorized according to the average number of neighbor nodes. In the sequel, the conditions under which $\tilde{P}_{P,succ} \geq P_{D,succ}$, are established and the value for p (denoted by $\tilde{p}_0$) that maximizes $\tilde{P}_{P,succ}$ is determined as well.

Let $\phi_{u \to v} = \frac{\sum_{\chi \in S_v \cup \{v\} - \{u\}} |\Omega_\chi \cap \Omega_u|}{|S_v|+1}$ denote the *average number of overlapping slots* of node u with each node $\chi \in (S_v \cup \{v\} - \{u\})$. As it can be seen from Appendix 1, the following inequality holds.

$$|R_{u \to v}| \geq q^2 - (|S_v| + 1)(q - \phi_{u \to v}). \tag{6}$$

Let $\overline{\phi} = \frac{1}{N} \sum_{\forall u \in V} \phi_{u \to v}$.

Theorem 4. *Provided that $\sum_{\forall u \in V} \left(|R_{u \to v}| - (q - |C_{u \to v}|)\overline{|S|}\right) \geq 0$ is satisfied, there exists a range of efficient values of p of the form $[0, \tilde{p}_{max}]$, for some $0 \leq \tilde{p}_{max} < 1$. $\tilde{P}_{P,succ}$ assumes a maximum for $p = \frac{\sum_{\forall u \in V} \left(|R_{u \to v}| - (q - |C_{u \to v}|)\overline{|S|}\right)}{\sum_{\forall u \in V} \left(|R_{u \to v}|(\overline{|S|}+1)\right)}$ $(\equiv \tilde{p}_0)$.* □

This theorem can be proved by calculating and analyzing the first derivative of $\tilde{P}_{P,succ}$ with respect to p.

Theorem 6 determines the lower and upper bounds of $\tilde{p}_0$ as a function of $\overline{|S|}$ only.

Theorem 5. *There exists an efficient range of values for p, provided that $\overline{\phi} \geq \frac{2\overline{|S|}+1}{4}$.*

Proof. According to Theorem 4, there exists an efficient range of values of p if $\sum_{\forall u \in V} \left(|R_{u \to v}| - (q - |C_{u \to v}|)\overline{|S|}\right) \geq 0$ holds. From Equation (6), $|R_{u \to v}| \geq q^2 - (\overline{|S|} + 1)(q - \phi_{u \to v})$. Therefore, $\sum_{\forall u \in V} \left(|R_{u \to v}| - (q - |C_{u \to v}|)\overline{|S|}\right) \geq 0$ always holds if $\sum_{\forall u \in V} \left(q^2 - (\overline{|S|} + 1)(q - \phi_{u \to v}) - (q - |C_{u \to v}|)\overline{|S|}\right) \geq 0$. Given that $|C_{u \to v}| \geq \phi_{u \to v}$ (see Appendix 2), it is enough to show that $q^2 - (2\overline{|S|} + 1)q + (2\overline{|S|} + 1)\overline{\phi} \geq 0$.

Let Δ be equal to $(2\overline{|S|} + 1)^2 - 4(2\overline{|S|} + 1)\overline{\phi} = (2\overline{|S|} + 1)(2\overline{|S|} + 1 - 4\overline{\phi})$. For $\Delta \leq 0$, $q^2 - (2\overline{|S|} + 1)q + (2\overline{|S|} + 1)\overline{\phi} \geq 0$. Since $2\overline{|S|} + 1 > 0$, in order for $\Delta \leq 0$, $2\overline{|S|} + 1 - 4\overline{\phi} \leq 0$ should hold, or $\overline{\phi} \geq \frac{2\overline{|S|}+1}{4}$. □

The condition of Theorem 5 (or Theorem 4) is sufficient but not necessary in order for $\tilde{P}_{P,succ} \geq P_{D,succ}$. Notice also that these theorems do not provide for a way to derive $\tilde{p}_{max}$. In addition, $\tilde{p}_0$ depends on parameters that are difficult to know for the entire network. In the sequel, Theorem 6 not only provides for a range of efficient values for p but also determines simple bounds ($\tilde{p}_{0_{max}}$, $\tilde{p}_{0_{min}} : \tilde{p}_{0_{max}} \leq \tilde{p}_0 \leq \tilde{p}_{0_{min}}$) on the values of $\tilde{p}_0$ (that maximizes $\tilde{P}_{P,succ}$) as a function of the simple topology density $\overline{|S|}$.

Theorem 6. *$\tilde{p}_{0_{max}} = \frac{1}{\overline{|S|}+1}$, and $\tilde{p}_{0_{min}} = \frac{q^2-(2\overline{|S|}+1)\left(q-\frac{2\overline{|S|}+1}{4}\right)}{\left(q^2-(\overline{|S|}+1)\left(q-\frac{2\overline{|S|}+1}{4}\right)\right)(\overline{|S|}+1)}$, provided that there exists an efficient range of values for p.*

Proof. According to Theorem 4, there exists an efficient range of values of p, if $\sum_{\forall u \in V}\left(|R_{u\to v}| - (q - |C_{u\to v}|)\overline{|S|}\right) \geq 0$ holds. Then, it can be calculated that $\tilde{p}_0 = \frac{\sum_{\forall u \in V}\left(|R_{u\to v}|-(q-|C_{u\to v}|)\overline{|S|}\right)}{\sum_{\forall u \in V}\left(|R_{u\to v}|(\overline{|S|}+1)\right)} = \frac{1}{(\overline{|S|}+1)}\left(1 - \frac{\sum_{\forall u \in V}(q-|C_{u\to v}|)\overline{|S|}}{\sum_{\forall u \in V}|R_{u\to v}|}\right) \leq \frac{1}{\overline{|S|}+1}$. Therefore, $\tilde{p}_{0_{max}} = \frac{1}{\overline{|S|}+1}$. From Equation (6), $|R_{u\to v}| \geq q^2 - (\overline{|S|}+1)(q - \phi_{u\to v})$ and given that $\tilde{p}_0 = \frac{1}{(\overline{|S|}+1)}\left(1 - \frac{\sum_{\forall u \in V}(q-|C_{u\to v}|)\overline{|S|}}{\sum_{\forall u \in V}|R_{u\to v}|}\right)$, it is concluded that $\tilde{p}_0 \geq \frac{\sum_{\forall u \in V}\left(q^2-(\overline{|S|}+1)(q-\phi_{u\to v})-(q-|C_{u\to v}|)\overline{|S|}\right)}{\sum_{\forall u \in V}\left(q^2-(\overline{|S|}+1)(q-\phi_{u\to v})\right)(\overline{|S|}+1)}$. Since $|C_{u\to v}| \geq \phi_{u\to v}$ (Appendix 2), it can be calculated that $\tilde{p}_0 \geq \frac{q^2-(2\overline{|S|}+1)(q-\overline{\phi})}{\left(q^2-(\overline{|S|}+1)(q-\overline{\phi})\right)(\overline{|S|}+1)} = \tilde{p}'_{min}(\overline{\phi})$. It can be calculated that $\frac{d\tilde{p}'_{min}(\overline{\phi})}{d\overline{\phi}} > 0$, and therefore, $\tilde{p}'_{min}(\overline{\phi})$ increases as $\overline{\phi}$ increases. Consequently, the minimum value for $\tilde{p}_0$, $\tilde{p}_{0_{min}}$, corresponds to the minimum value of $\overline{\phi}$ for which there exists an efficient range of values for p. This value according to Theorem 5, corresponds to $\overline{\phi} = \frac{2\overline{|S|}+1}{4}$ and as a result, $\tilde{p}_{0_{min}} = \frac{q^2-(2\overline{|S|}+1)\left(q-\frac{2\overline{|S|}+1}{4}\right)}{\left(q^2-(\overline{|S|}+1)\left(q-\frac{2\overline{|S|}+1}{4}\right)\right)(\overline{|S|}+1)}$. □

Both Theorem 5 and 6 are important for the realization of a system that efficiently implements the Probabilistic Policy. Given a polynomial assignment that satisfies Theorem 5, for a value of p between $\tilde{p}_{0_{min}}$ and $\tilde{p}_{0_{max}}$, the achievable system throughput is close to the maximum. For the determination of $\tilde{p}_{0_{min}}$ and $\tilde{p}_{0_{max}}$ it is enough to have knowledge of the topology density $\overline{|S|}$.

5 On the Accuracy of the Approximation

The analysis presented in Section 4 has established the conditions under which $\tilde{P}_{P,succ} \geq P_{D,succ}$, as well as the range of values of p for which $\tilde{P}_{P,succ}$ is maximized. This section presents the cases for which (a) $\tilde{P}_{P,succ}$ is *close* to $P_{P,succ}$ in the sense that there exists a small number ε_1 such that $|P_{P,succ} - \tilde{P}_{P,succ}| \leq \varepsilon_1$ and (b) if the condition, for which $\tilde{P}_{P,succ} \geq P_{D,succ}$ holds, is satisfied, then

$P_{P,succ} \geq P_{D,succ}$ holds as well. This section presents the case for which the conditions of Theorem 2 and Theorem 4 are *close* in the sense that $\exists \varepsilon_2$: $\left| \sum_{\forall u \in V} \left(|R_{u \to v}| - (q - |C_{u \to v}|)\overline{|S|} \right) - \sum_{\forall u \in V} \left(|R_{u \to v}| - (q - |C_{u \to v}|)|S_v| \right) \right| \leq \varepsilon_2$. It is also shown that as ε_2 increases linearly, ε_1 increases exponentially. For the case where ε_1 is not small, it is possible that ε_2 is small and $P_{P,succ} \geq P_{D,succ}$ holds if the condition corresponding to $\tilde{P}_{P,succ} \geq P_{D,succ}$ is satisfied. In general, ε_1 reflects the *accuracy* of $\tilde{P}_{P,succ}$ while ε_2 reflects the *accuracy* of the conditions that if satisfied, $\tilde{P}_{P,succ} \geq P_{D,succ}$ holds.

First, it is enough to determine the cases when ε_1 is close to zero. $|P_{P,succ} - \tilde{P}_{P,succ}| \leq \frac{1}{N} \sum_{\forall u \in V} \left(\frac{q - |C_{u \to v}| + p|R_{u \to v}|}{q^2} (1-p)^{\overline{|S|}} \left| (1-p)^{|S_v| - \overline{|S|}} - 1 \right| \right) \leq \varepsilon_1$ holds. Let $Var\{|S|\} = \frac{1}{ND} \sum_{v \in V} \left| \overline{|S|} - |S_v| \right|$ be defined as the *topology density variation.* It is evident that as $Var\{|S|\}$ increases, ε_1 increases exponentially. Consequently, $\tilde{P}_{P,succ}$ is a good approximation of $P_{P,succ}$, for rather small values of $Var\{|S|\}$ ($\varepsilon_1 \to 0$).

Second, the absolute difference between the two conditions is calculated to be equal to $\left| \sum_{\forall u \in V} (q - |C_{u \to v}|)(\overline{|S|} - |S_v|) \right| \leq \sum_{\forall u \in V} (q - |C_{u \to v}|) \left| \overline{|S|} - |S_v| \right|$ $\leq \varepsilon_2$. Consequently, as $Var\{|S|\}$ approaches zero, ε_2 approaches zero linearly and consequently, the condition under which $\tilde{P}_{P,succ} \geq P_{D,succ}$ holds, approaches linearly the condition under which $P_{P,succ} \geq P_{D,succ}$ holds. On the other hand, as $Var\{|S|\}$ increases, ε_2 increases linearly but ε_1 increases exponentially.

Let p_0 denote that value for p that maximizes (global maximum) $P_{P,succ}$. Obviously, $\left. \frac{dP_{P,succ}}{dp} \right|_{p=p_0} = 0$. Equation (4) is a polynomial of degree D and it is difficult or impossible to be solved to obtain an analytical form for p_0. It is obvious that for $Var\{|S|\} = 0$, $\tilde{p}_0 \equiv p_0$ and therefore, $p_0 \in (\tilde{p}_{0,min}, \tilde{p}_{0,max})$. In general, p_0 may or may not belong in $(\tilde{p}_{0,min}, \tilde{p}_{0,max})$ but any value $p \in (\tilde{p}_{0,min}, \tilde{p}_{0,max})$ for which $\tilde{P}_{P,succ} \geq P_{D,succ}$ holds and $\tilde{P}_{P,succ}$ is close to its maximum value, possibly (depending on the value of ε_2) leads to $P_{P,succ} \geq P_{D,succ}$ and it is possible (depending on the value of ε_1) that $P_{P,succ}$ is close to its maximum value as well.

6 Simulation Results

For the simulation purposes four different topology categories are considered. The number of nodes in each topology category is $N = 100$, while D is set to 5, 10, 15 and 20. These four topology categories are denoted as D5N100, D10N100, D15N100 and D20N100 respectively. The algorithm presented in [7] that maximizes the minimum guaranteed throughput, is used to derive the sets of scheduling slots. Time slot sets are assigned randomly to each node, for each particular topology. The particular assignment is kept the same for each topology category throughout the simulations.

The simulation results presented demonstrate the performance for $k = 1$ (the resulting value for k is equal to 1 for the four topology categories, [7]), that is the case that the number of non-assigned eligible time slots is expected to be rather small and, thus, the effectiveness of the Probabilistic Policy rather low.

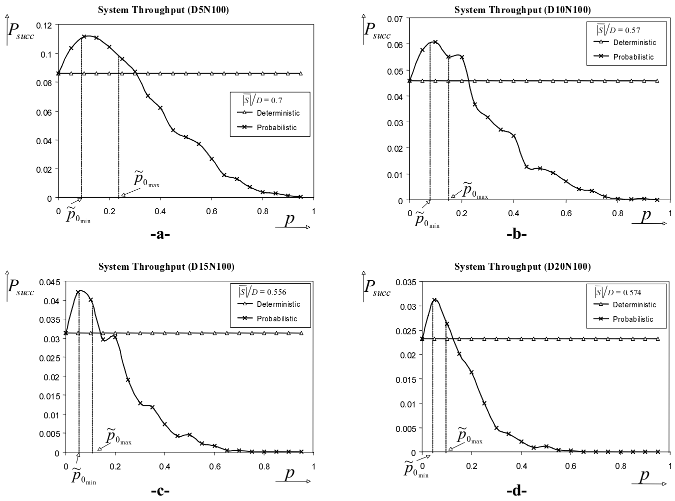


Fig. 1. System throughput simulation results for different values of p for both the Deterministic and the Probabilistic Policy.

The value of $\overline{\phi}$ calculated for each topology satisfies the condition of Theorem 5 for all cases. If $p \in (\tilde{p}_{0,min}, \tilde{p}_{0,max})$ then the achieved system throughput is possible to be close to the maximum, as it appears from Theorem 6.

Figure 1 depicts simulation results for the system throughput (P_{succ}), under both the Deterministic and the Probabilistic Policies, as a function of the access probability p. It can be observed, as expected, that the system throughput achieved under the Deterministic Policy is constant with respect to p. Obviously, $(\tilde{p}0_{min}, \tilde{p}0_{max})$ determines a range of the values of p for which $P_{P,succ} > P_{D,succ}$ and it appears that $P_{P,succ}$ is close to its maximum value.

For the comparison between the two schemes, it is set $p = \tilde{p}0_{min}$. From Figure 2, it can be seen that the achieved system throughput under the Probabilistic Policy is higher than that under the Deterministic Policy. It is a fact that for high topology density values and small networks ($k = 1$) the gain of the Probabilistic Policy is negligible but for any other case (small topology density values and $k = 1$ or any topology density values and $k > 1$) the gain is significantly high.

7 Summary and Conclusions

The Probabilistic Policy was introduced in [8] and the *throughput for a specific transmission* was studied. In this work, this policy is considered and a *system throughput* analysis is presented in order to (a) identify the suitable range of values for the access probability p for which the Probabilistic Policy outperforms the Deterministic Policy; (b) identify the maximum value for the system throughput and the corresponding value of the access probability; (c) determine simple bounds on the access probability that maximize the system throughput as a function of the topology density.

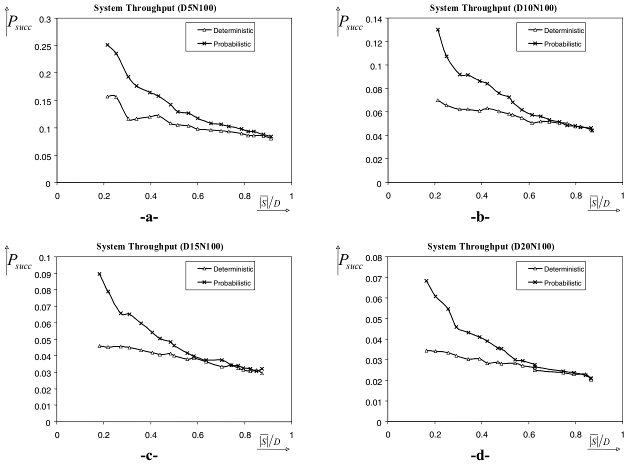


Fig. 2. System throughput simulation results (P_{succ}), for both policies, for different values of the topology density $\overline{|S|}/D$ ($p = \tilde{p}_{0_{min}}$).

Simulation results have been derived for four network topology categories $\big($for four pairs $(N, D)\big)$. The derived results have supported the claims and expectations regarding the comparative advantage of the Probabilistic Policy over the Deterministic Policy and that the approximate system analysis determines the range of values that, under certain conditions, maximize the system throughput under the Probabilistic Policy, or induce a system throughput close to the maximum.

References

1. IEEE 802.11, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications," Nov. 1997. Draft Supplement to Standard IEEE 802.11, IEEE, New York, January 1999.
2. V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LAN's," Proceedings of ACM SIGCOMM'94, pp. 212-225, 1994.
3. C.L. Fullmer, J.J. Garcia-Luna-Aceves, "Floor Acquisition Multiple Access (FAMA) for Packet-Radio Networks," Proceedings of ACM SIGCOMM'95, pp. 262-273, 1995.
4. P. Karn, "MACA- A new channel access method for packet radio," in ARRL/CRRL Amateur Radio 9th Computer Networking Conference, pp. 134-140, 1990.
5. J. Deng and Z. J. Haas, "Busy Tone Multiple Access (DBTMA): A New Medium Access Control for Packet Radio Networks," in IEEE ICUPC'98, Florence, Italy, October 5-9, 1998.
6. I. Chlamtac and A. Farago, "Making Transmission Schedules Immune to Topology Changes in Multi-Hop Packet Radio Networks," IEEE/ACM Trans. on Networking, 2:23-29, 1994.

7. J.-H. Ju and V. O. K. Li, “An Optimal Topology-Transparent Scheduling Method in Multihop Packet Radio Networks,” IEEE/ACM Trans. on Networking, 6:298-306, 1998.
8. Konstantinos Oikonomou and Ioannis Stavrakakis, “A Probabilistic Topology Unaware TDMA Medium Access Control Policy for Ad-Hoc Environments”, Personal Wireless Communications (PWC 2003), September 23-25, 2003, Venice, Italy.

Appendices

Appendix 1 $|R_{u\to v}| \geq q^2 - (|S_v|+1)(q-\phi_{u\to v})$

$\left|\bigcup_{\chi\in S_v+\{v\}}\Omega_\chi\right|$ can be written as $\left|\bigcup_{j=1}^{|S_v|+1}\Omega_j\right|$ by assigning numbers, $j = 1, ..., |S_v|+1$, to each node $\chi \in S_v \cup \{v\}$. Without loss of generality, it is assumed that node u corresponds to number $|S_v|+1$ or $\Omega_u \equiv \Omega_{|S_v|+1}$.

$$\left|\bigcup_{j=1}^{|S_v|+1}\Omega_j\right| = |\Omega_1| + \left|\bigcup_{j=2}^{|S_v|+1}\Omega_j\right| - \left|\Omega_1\cap\left(\bigcup_{j=2}^{|S_v|+1}\Omega_j\right)\right|$$

$$\vdots \qquad\qquad \vdots$$

$$\left|\bigcup_{j=|S_v|}^{|S_v|+1}\Omega_j\right| = |\Omega_{|S_v|}| + \left|\bigcup_{j=|S_v|+1}^{|S_v|+1}\Omega_j\right| - \left|\Omega_{|S_v|}\cap\left(\bigcup_{j=|S_v|+1}^{|S_v|+1}\Omega_j\right)\right|$$

$\left|\bigcup_{j=|S_v|+1}^{|S_v|+1}\Omega_j\right| = |\Omega_{|S_v|+1}|$. $|\Omega_j| = q$ and by adding all lines: $\left|\bigcup\limits_{j=1}^{|S_v|+1}\Omega_j\right| = (|S_v|+1)q - \sum\limits_{j=1}^{|S_v|}\left|\Omega_j\cap\left(\bigcup\limits_{l=j+1}^{|S_v|+1}\Omega_l\right)\right|$. Let $\theta_{u\to v} = \frac{\sum_{j=1}^{|S_v|}\left|\Omega_j\cap\left(\bigcup_{l=j+1}^{|S_v|+1}\Omega_l\right)\right|}{|S_v|+1}$. The latter expression can be written as $\left|\bigcup_{j=1}^{|S_v|+1}\Omega_j\right| = (|S_v|+1)q - (|S_v|+1)\theta_{u\to v} = (|S_v|+1)(q-\theta_{u\to v})$, and therefore, given Equation (2), $|R_{u\to v}| = q^2 - (|S_v|+1)(q-\theta_{u\to v})$. $\phi_{u\to v}$ can be written as follows: $\phi_{u\to v} = \frac{\sum_{j=1}^{|S_v|}|\Omega_j\cap\Omega_{|S_v|+1}|}{|S_v|+1}$ or $\phi_{u\to v} = \frac{\sum_{j=1}^{|S_v|}|\Omega_j\cap\Omega_u|}{|S_v|+1}$. Given that $\Omega_u \equiv \Omega_{|S_v|+1}$, it is concluded that $\theta_{u\to v} \geq \phi_{u\to v}$ and consequently, Equation (6) is proved.

Appendix 2 $|C_{u\to v}| \geq \phi_{u\to v}$

From Equation (1), $|C_{u\to v}| = \left|\Omega_u\cap\left(\bigcup_{\chi\in S_v\cup\{v\}-\{u\}}\Omega_\chi\right)\right| \geq |\Omega_u\cap\Omega_j|$, for all nodes $\chi\in(S_v\cup\{v\}-\{u\})$, denoted by numbers $j = 1, ..., |S_v|$, while node u is denoted by $|S_v|+1$. Consequently, $|S_v||C_{u\to v}| \geq \sum_{j=1}^{|S_v|}|\Omega_u\cap\Omega_j|$, or $|C_{u\to v}| \geq \frac{\sum_{j=1}^{|S_v|}|\Omega_u\cap\Omega_j|}{|S_v|} > \frac{\sum_{j=1}^{|S_v|}|\Omega_u\cap\Omega_j|}{|S_v|+1}$, or $|C_{u\to v}| > \phi_{u\to v}$. Consequently, $|C_{u\to v}| \geq \phi_{u\to v}$ (the equality holds when $\Omega_u\cap\left(\bigcup_{\chi\in S_v\cup\{v\}-\{u\}}\Omega_\chi\right) = \emptyset$ in which case $|C_{u\to v}| = \phi_{u\to v} = 0$).

Study of the Capacity of Multihop Cellular Networks*

Antonis Panagakis, Elias Balafoutis, and Ioannis Stavrakakis

Dept. of Informatics and Telecommunications
University of Athens, Athens, Greece
{grad0260,balaf,istavrak}@di.uoa.gr

Abstract. Recently, the application of the peer to peer networking paradigm (typical for an ad hoc network) has been proposed for wireless local area networks (WLANs), instead of the traditional cellular networking paradigm. In this paper the performance of a WLAN employing the peer to peer networking paradigm is studied via simulations; the results indicate that the direct application of the peer to peer networking paradigm in a WLAN leads to a substantially decreased throughput for the traffic directed to the Access Point (AP). The study also reveals that the cumulative receiving throughput of nodes located at the periphery of relatively small circular areas around the AP is substantially higher. Thus, the capacity of the multihop cellular network may be enhanced by employing the peer to peer paradigm only outside a circular area around the AP and the cellular paradigm inside this circular area. Examples are provided of environments where the aforementioned idea of distributing the traditional AP functionality to a set of nodes at the periphery of a circular area around the AP can be effectively applied.

1 Introduction

The cellular networking paradigm is the traditional networking paradigm for WLANs. Recently, the application of the peer to peer networking paradigm (typical for an ad hoc network) has been proposed for WLANs ([7,9,6,2,3]). However, the direct application of the peer to peer networking paradigm in a WLAN demands that the AP be "downgraded" to an ordinary node, in the sense that the AP no longer has the control of the shared medium (as in typical cellular networks), and its transmission range is reduced. These can have important side-effects on the network's performance, especially for the case of inter-cell traffic (traffic that traverses the AP). In this paper, the performance of a Multihop Cellular Network (MCN) ([6]) is studied via simulations for the case of inter-cell traffic and a more efficient approach for the employment of the peer to peer paradigm in WLANs is proposed. The main difference of a Multihop Cellular Network (MCN) compared to the cellular network is that peer

* This work has been supported in part by the IST Programme under contract IST-2001-32686 (BROADWAY).

G. Karlsson and M.I. Smirnov (Eds.): QofIS 2003, LNCS 2811, pp. 182–192, 2003.

to peer (between nodes) communications are allowed and the operation of the network is distributed. The transmission range of both the mobile nodes and the AP is reduced so that multi-pair peer to peer communication be possible unihibited by interference. On the other hand the transmission range should not be reduced to the extent that node connectivity be compromised[1]. The benefits of the adaptation of the peer to peer paradigm within a WLAN include mainly reduced energy consumption and the possibility for multiple simultaneous transmissions over the shared medium (spatial reuse of the shared medium). In an effort to show the advantages of the MCN architecture due to the possibility for spatial reuse of the shared medium, past studies [6,2] have considered network traffic conditions in which all traffic is intra-cell. Under inter-cell traffic conditions though, the benefits of spatial reuse – if any – are questionable and the effectiveness of the MCN architecture is quite poor. In this paper the capacity of the MCN architecture is evaluated via simulations under inter-cell traffic conditions. The derived results indicate that the MCN architecture supports a substantially decreased throughput at the AP. The study also reveals that the cumulative receiving throughput of nodes located at the periphery of relatively small circular areas around the AP is substantially higher. Thus, the capacity of the multihop cellular network may be enhanced by employing the peer to peer paradigm only outside a circular area around the AP and the cellular paradigm inside this circular area in order to allow for a coordinated and high throughput (last) one-hop access to the AP of the nodes within this circular area.

The structure of this paper is as follows. In section 2 the capacity of the MCN is evaluated for the case of inter-cell traffic. In section 3 an enhancement is proposed which is shown to incorporate several of the advantages of the peer to peer communication into the cellular architecture. Related work is summarized in section 4.

2 Evaluation of the Capacity of the MCN Architecture

When all traffic is intra-cell, the MCN architecture is expected to be efficient due to the advantage of the spatial reuse of the shared medium in the peer to peer communications. No node is expected to be different than any other node in terms of the amount of traffic generated or received under the intra-cell traffic conditions. The uniform distribution of the peer to peer communications across the cell (a) maximizes the benefit of the spatial reuse by spreading evenly the one-hop transmissions (and minimizing the interference) and (b) results in a geographically evenly distributed traffic load avoiding the formation of problem-

[1] The reduced coverage area of the AP is referred to as the sub-cell; its radius is denoted as R_{sc} and is assumed to be equal to the transmission range of all the nodes. R_c denotes the radius of the geographical area that the AP is responsible for covering; in the cellular networking paradigm it is assumed that the transmission range of the AP is equal to R_c. In addition, neighboring APs are assumed of the same radius and not overlapping; that is, if L denotes the minimum distance between two APs then $L \geq 2R_c$.

atic bottlenecks; the latter occurs since a relatively low traffic load is directed to each receiving node and, thus, can be supported, even if (due to interference) its throughput is only a small portion of the medium's capacity.

However, in a WLAN architecture it is expected that the traffic would mostly be directed toward the node that supports accessing the web and nodes outside the cell. Thus, the long-term ratio of the inter-cell over the total (inter-cell and intra-cell) traffic is expected to be high and possibly very close to 1. The previous implies that the uniform distribution of the peer to peer transmission and traffic load are far from valid under realistic conditions, since one node, the AP, would be offered a high traffic load and have a high concentration of peer to peer transmissions in its vicinity. Due to the latter, the (receiving) throughput of the AP's shared medium would be drastically reduced and would be far from adequate in supporting the high traffic load offered to the AP (that would require a very high throughput). In this section the aforementioned issues are investigated by studying the performance of the MCN architecture under inter-cell traffic conditions.

2.1 Simulation Environment

The simulator used is the ns simulator [1] (version 2.1b7) with the multihop wireless extensions from the CMU Monarch group. At the medium access level the Distributed Coordination Function (DCF) mode of the IEEE 802.11 MAC standard is used with the default parameters of the simulator. The path loss model used is the two ray ground path loss model of the ns simulator according to which the signal attenuates in proportion to $1/d^2$ up to a certain distance (default value $\approx 86m$) and in proportion to $1/d^4$ beyond that. The only modifications applied concern the transmission power (in order to vary the transmission range) and the carrier sense threshold (CST) (in order to vary the range of interference). In the simulation results the ratio of the interference range over the transmission range is referred to as the interference factor and is denoted as a. For example, for the default values of the simulator $a \approx 2.2$ [2]. The UDP transport protocol is employed with a packet size of 512 bytes; the Distributed Source Routing (DSR) protocol is employed for packet routing. The nodes are assumed to be static so that any impact of node mobility on the derived results be avoided.

The simulated topology consists of an 11x11 (121-node) grid in which some (or all) nodes transmit to the node at the center of the grid, which plays the role of the AP. The side of the grid is 40m and the transmission range is set to 60m in order for the diagonal neighbors to be reachable in one hop. The investigation is limited to the case of the uplink (where the benefits in terms of power consumption due to the multihop operation are more important) and only inter-cell traffic is considered, that is, traffic directed to the AP.

[2] In this work the term interference range is used as in [5] to refer to the maximum distance for which a receiver can sense a transmitter; the term carrier sensing range (used for example in [4]) is rather more appropriate for this quantity (see [4] for a relevant discussion and [8] for details on the physical layer modeling in ns simulator).

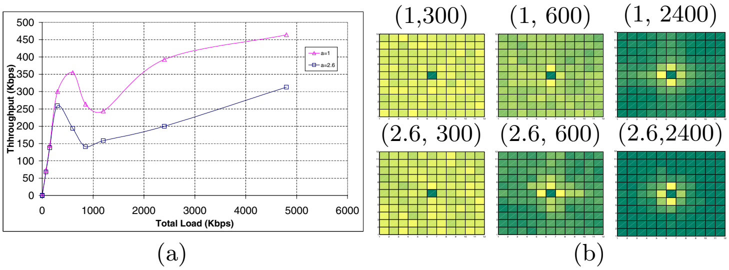
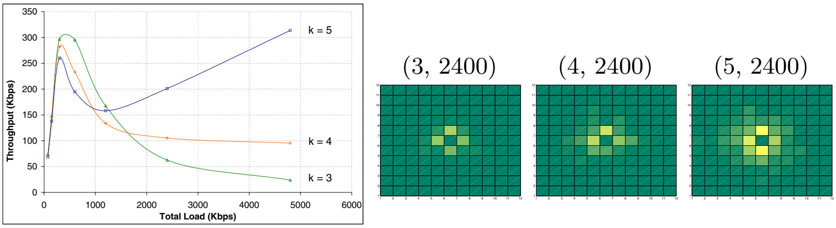


Fig. 1. (a) Total throughput vs total load for two different ranges of interference and (b) Throughput distribution for (a, total load (kbps))=(1, 300), (1, 600), (1, 2400), (2.6, 300), (2.6, 600), (2.6, 2400) .

2.2 Simulation Results

The Effect of Interference. All nodes generate the same amount of traffic to the AP, which is assumed to be the node located in the center of the grid. Figure 1(a), illustrates the throughput achieved for different loads for the systems with $a = 2.6$ and $a = 1$, respectively. Two basic observations can be made from these results. The first is that the throughput of the MCN system is much lower than the channel capacity (1.3 Mbits) [3]. The second is that the throughput achieved for $a = 2.6$ is lower than that for $a = 1$. There is also a paradox observed in this figure regarding the shape of both throughput curves: initially, the throughput increases as the load increases, then decreases for a small range of loads and then increases again. Figure 1(b) attempts to provide for some insight into the aforementioned paradox. Each cell in each of the six grids shown in Figure 1(b) corresponds to a node in the simulated topology. The different colors in each cell represent the different ratios of packets sent by the source of the grid corresponding to that cell that have been successfully received at the destination. Only the traffic originated by each node is considered (packets that are relayed are not accounted for). The lighter the color, the larger the ratio of packets sent by the corresponding source that reach the AP. The very dark color at the center corresponds to the AP, which does not send any packets. In all figures the total load is evenly distributed among all sources. The first set of figures is obtained for a total load of 300Kbps. The received packet ratio at the destination for this load is 92% and 86% for $a = 1$ and $a = 2.6$, respectively; that is, the 92% and 86% of all packets reach the AP. As seen in the figure this ratio is almost evenly distributed among all nodes in the grid. The difference in throughput for the two systems is attributed to the level of interference, which is larger for $a = 2.6$. The second set of figures is obtained for total load of 600Kbps. The received packet ratio at the destination for this load is 60% and 32% for

[3] The maximum throughput that can be achieved between two nodes using the IEEE 802.11 DCF, (with UDP traffic and packet size of 512 bytes) was found (via simulations) to be 1.3Mbits. This value may be considered as the channel capacity and is used for comparison with the throughput of the simulated system.



(a) (b)

Fig. 2. (a) Total throughput vs total load for three different values of number of hops (3, 4 and 5) (b) Throughput distribution for (number of hops, total load (kbps))=(3, 2400), (4, 2400), (5, 2400).

$a = 1$ and $a = 2.6$, respectively. It is observed that mostly the nodes around the AP contribute to the total throughput, while the other nodes block each other's transmissions. However, this small fraction of nodes that do contribute to the system's throughput is rather low (5 Kbps per node) and the total throughput is low. In the last set of figures, which is obtained for total load of 2400Kbps, the situation is similar to the previous one. The received packet ratio for this load is 16% and 8% for $a = 1$ and $a = 2.6$, respectively. Again only the nodes around the AP contribute to the system's throughput but now their rate (20Kbps) is such that the total throughput becomes larger.

The Effect of the Number of Hops. In order to investigate the effect of the length of the path on the throughput only the nodes that lie k or less hops away from the AP transmit to the AP. Figure 2(a) illustrates the results for the system with interference factor $a = 2.6$ and for $k = 3$, $k = 4$ and $k = 5$. A general comment from Figure 2(a) is that the shape of the throughput curves remains the same for all values of k depicted in the figure; for light load situations (up to 300Kbps, which is far below the channel capacity), the impact of interference is negligible and the achieved throughput is about the same for the three cases ($k = 3, 4$ and 5).That is, the larger number of hops does not lead to a throughput reduction under light traffic load condition. When the interference becomes significant and a throughput reduction is observed, (at a load about 300Kbps in Figure 2(a)), the impact of the interference is larger on the larger hops paths and thus, the throughput decrease is larger for larger k. A "paradox" appears under heavy load conditions, where the aforementioned relative performance is reversed. Figure 2(b) attempts to shed some light into this behavior. Again, each cell in the figure corresponds to a node in the simulated topology. The different colors in each cell represent the different number of packets sent by each node that manage to reach the destination. The lighter the color, the larger the number of sent packets that reach the AP. This set of figures are derived for a system load of 2400Kbps (50, 30 and 20 Kbps per node for $k = 3, 4$ and 5, respectively). By observing the lighter color for the nodes around the AP it can be concluded that despite the fact that the nodes send less packets (20 Kbps),

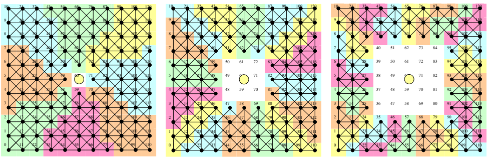

(a) (b) (c)

Fig. 3. Simulated topologies. The receivers are located at ring 1, ring 2 and ring 3 in (a), (b) and (c), respectively (,that is, 1, 2 and 3 hops away from the AP). Different colors are used to distinguish the different groups of nodes; in each group of nodes the node located nearer to the AP is the destination of the traffic originated by the other nodes belonging to the same group.

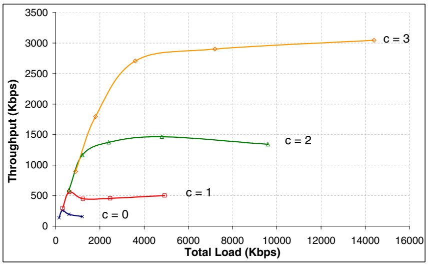


Fig. 4. Avaialble bandwidth at the rings around the AP for $a = 2.6$.

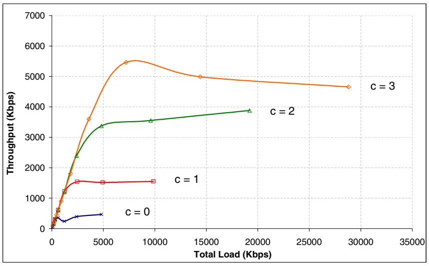


Fig. 5. Avaialble bandwidth at the rings around the AP for $a = 1$.

the system manages to deliver a larger number of packets when $k = 5$. It seems that for $k = 3$ and 4 the nodes block each other, while for $k = 5$, the nodes at the 5th hop block the nodes at the 4th and 3rd hop and leave space for the nodes at the first two hops to transmit to the AP. Similar observations can be made when comparing the cases for $k = 3$ and $k = 2$. Similar comments regarding the effect of the path length also apply for the case of $a = 1$, which is not presented here.

While the length of the paths does affect the performance of the system, the throughput remains significantly lower than the channel capacity even for a small number of hops around the AP and even for light loads. This fact highlights the inefficiency of the MCN architecture for the inter-cell traffic and confirms the related discussion in the beginning of this section.

Available Throughput at "Rings" around the AP. In an attempt to shed light into the observed performance degradation the throughput of the system is measured at rings of several (hops) distance from the access point. The ring located c hops away from the access point has m nodes where $(c, m) \in \{(1, 8), (2, 16), (3, 24), (4, 32), (5, 40)\}$ and is referred to as $ring_c$. In order to exclude the effect of the interference caused by the internal (with respect to each ring) nodes, these nodes do not transmit (or relay) any traffic. Thus, when the receiving nodes belong to $ring_c$, only nodes located m, $m > c$, hops away from

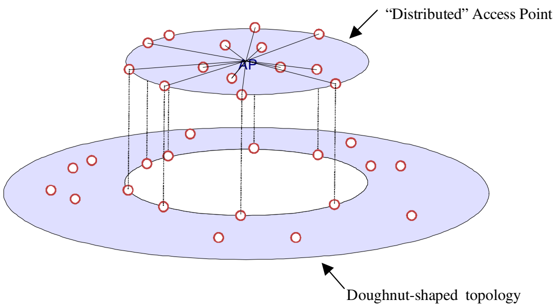


Fig. 6. The topology of a traditional cellular network may be decomposed into a doughnut-shaped topology and a small circular area around the AP.

the AP are considered as potential sources; the exact association of each source to one receiver is shown in Figure 3. Figures 4 and 5 illustrate the throughput for $a = 2.6$ and $a = 1$, respectively, for the cases where the receivers belong to $ring_c$, $c = 0, 1, 2, 3$. It can be observed that the throughput increases as c increases, which can be much larger than the capacity of the channel (1.3Mbps) due to spatial reuse.

The presented results indicate that although the achievable throughput relatively far from the AP can be much larger than the capacity of the channel, this throughput cannot be "transferred" to the AP due to the bottleneck formed around it. In the sequel an enhancement is proposed, which attempts to alleviate the AP bottleneck problem by exploiting the higher throughput achieved in the rings around it.

3 Enhancing the Efficiency of an MCN

Simulation results indicate that the achievable bandwidth at the AP for the inter-cell traffic in an MCN is low due to contention and interference. On the other hand, it is observed that the receiving throughput at nodes located at the periphery of a circular area around the AP is significantly higher. This is because when the destination nodes are nodes located on this periphery, the multihop peer to peer transmissions from nodes beyond this periphery benefit from spatial reuse and are more efficiently supported.

An enhancement could be based on the idea that the topology of a traditional cellular network be decomposed on a conceptual level into a small circular area around the AP and a doughnut-shaped topology. In a doughnut-shaped topology, where the traffic that would be directed to the AP in the traditional cellular environment is destined to nodes located on the inner boundary of the doughnut-shaped topology, it is expected that the multihop peer to peer transmissions would be effective, since spatial reuse of the shared medium would be possible along the entire path including the destination. In the real topology all traffic is destined to the AP and, thus, there would be the need to distribute (or pseudo-

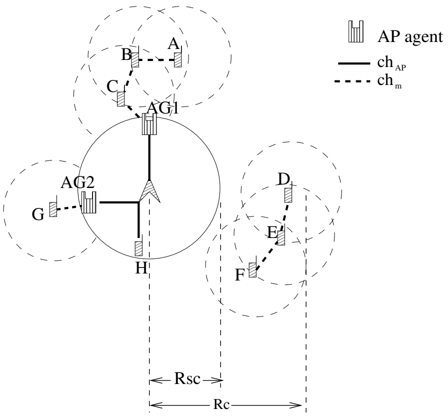


Fig. 7. Proposed enhancement.

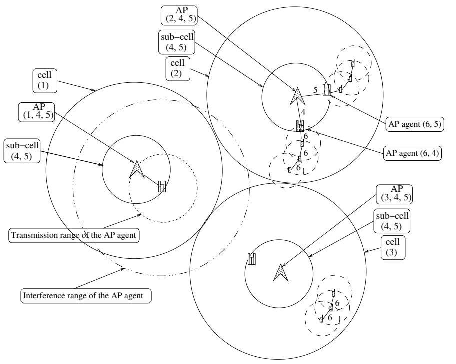


Fig. 8. An example of the generalization of the proposed enhancement for three neighboring cells.

distribute) the AP functionality at the inner boundary of the multihop doughnut-shaped area (see figure 6). To address the latter a dedicated channel may be used in order to pseudo-distribute the AP, that is, the AP would operate in a different channel than that used by the ordinary nodes for their peer to peer communications. In order for the communication between the nodes and the AP to be possible, some nodes, referred to as the AP agents, should operate on a time division basis in these two channels. Thus, at least two channels would be required in each cell. However, if one of them were reusable in adjacent cells (its use were not prohibited due to interference) the additional cost (in terms of required channels) would be rather low[4].

The proposed enhancement is depicted in figure 7. The nodes located outside the sub-cell operate as in the simple MCN architecture in a channel (say ch_m). The difference is that the AP and the nodes located within its coverage area (the sub-cell) operate in a different channel (say ch_{AP}) by employing a centralized protocol, and thus, the AP keeps its distinguished role within the sub-cell. Some nodes within the sub-cell (the AP agents) operate on a time division basis between the two channels (ch_{AP} and ch_m) providing for connectivity between the AP and the nodes located outside the sub-cell. By employing the concept of the AP agents, and as far as the nodes located outside the sub-cell are concerned, the functionality of the AP is distributed to a set of nodes (the AP agents) that undertake the effort to concentrate the traffic directed to the AP in the multihop channel. Due to the fact that there is no interference between communications in and outside the sub-cell the maximum achievable throughput of the system (for inter-cell traffic) is expected to be greater than the throughput of the sim-

[4] A potential enhancement such as the above requires that the AP agents be capable of operating on a time division basis between two channel. The exact description of such a mechanism, as well as the exact description of the mechanism used for determining which of the nodes should act as AP agents, is out of the scope of this paper.

ple MCN and approximately equal to the minimum between the capacity of the channel utilized within the sub-cell and the throughput of the system measured at the AP agents. The throughput of the system measured at the AP agents depends on the number and the location of the AP agents which in their turn depend on the transmission range of the AP.

The transmission range of the AP does not have to be the same as that of the other nodes, as in the simple MCN network. However, it should be noted that we restrict our presentation to the case where the dimensions of the sub-cell are such that the use of the channel utilized within the sub-cell does not prohibit (due to interference) the use of the same channel within the sub-cell of adjacent cells. It is straightforward to determine under which conditions this requirement is fulfilled. More specifically, and assuming that all cells are identical the use of a channel within the subcells of adjacent cells is permitted if (it is assumed that nodes located inside the sub-cell have a transmission range no greater that the transmission range of the AP)$L - 2R_{sc} \geq aR_{sc}$. Assuming that $L = 2R_c$ it follows that $\frac{R_c}{R_{sc}} \geq \frac{a}{2} + 1$ should hold. For typical cases (more precisely for the values of the interference ratio considered in this work ($a \leq 4$)) it is concluded that for $R_c \geq 3R_{sc}$ the use of an additional channel within the sub-cell does not affect adjacent cells. Thus, only the case of MCNs with at least three hops (and interference conditions no worse then those described in section 2.1) is considered in this work. It should be noted that the expansion of the sub-cell involves a tradeoff between the bandwidth that is available at the margins of the sub-cell (at the AP agents) and the total power consumption along of the paths between the nodes and the AP (since the last hop is longer).

The proposed enhancement may be generalized in order to combine the cellular and the peer to peer networking paradigms under the assumption that the AP is capable of operating simultaneously in multiple channels (at least two). This can be realized by adding additional antennas to the AP; the cost of such an approach is much less than the cost of adding additional APs since there are no extra infrastructure/administrative costs. In addition, a single channel is assumed outside the sub-cell for peer to peer communications. According to the proposed generalization one of the AP's operating channels is utilized exactly as in the cellular networking paradigm; it covers the entire cell and centralized control is employed by the AP. The rest of the channels are used within the sub-cell (whose dimensions are assumed to be subject to the same constrains as those discussed above, in order for the corresponding channels to be reusable in neighboring cells) and AP agents that operate on a time division basis between one of these channels and the channel used for multihop communications outside the sub-cell are employed.

The main difference compared to the MCN is that the peer to peer paradigm is not replacing the cellular one, but is used to form alternative multihop paths in order to offload the cellular system (by being utilized for intra-cell traffic) and provide power consuming alternative paths to the AP (for example, for traffic that can tolerate delay and losses) by exploiting the concept of the AP agents.

For the peer to peer communications the same channel can be utilized in all neighboring cells; interference occurring on cells' boundaries should not be of great concern since there always is available the cellular mode for nodes that face strong interference. Note that this does not hold in the case of MCN where the multihop path is the only one available. For the same reason (existence of the cellular mode), there is no need for the exact determination of the transmission range of the nodes in order for the peer to peer network to remain fully connected.

An example of the proposed generalization is illustrated in figure 8. Assume that there are three neighboring AP's, whose cells are of the same dimensions and do not overlap, that operate according to the cellular networking paradigm. That is, each AP uses a channel with such a transmission range that its entire cell is covered; let these channels be denoted as ch_1, ch_2, ch_3. In order to double the capacity of the system three additional channels (say ch_4, ch_5, ch_6) would be required, since in each cell a different channel must be utilized. According to the proposed enhancement two of these channels (say ch_4 and ch_5) are employed by all three APs with a transmission range such that interference between nodes operating at these channels in different subcells is avoided, that is, the dimensions of the subcells corresponding to ch_4 and ch_5 are subject to the aforementioned restrictions. The other channel (ch_6) is employed in all three cells for the peer to peer communications between the nodes outside the sub-cell. In each sub-cell there are some AP agents for each channel utilized within the sub-cell, that is, there are some nodes that on a time division basis operate in (ch_4 and ch_6) or (ch_5 and ch_6). In this way the available capacity at the AP is tripled at the cost that the 2/3 of this capacity is not available with the same characteristics to all nodes (multihop paths are typically characterized by greater delay, lower availability but also lower power consumption). Given the diversity of the QoS requirements of the applications and the existence of (high demand) hot spots the proposed enhancement, which combines the cellular and the peer to peer networking paradigms, might be efficient in many environments.

4 Related Work

In [7] a cellular architecture is proposed in which some ad hoc relay stations are placed inside the cells and relay traffic dynamically from overloaded cells to lightly loaded cells in order to balance the system's traffic. In [9] a similar architecture is presented, in which an ad hoc overlay layer is added to the fixed cellular infrastructure for the forwarding of the traffic from a heavy loaded cell to neighboring cells. In [2] the multihop paradigm is adopted but with the coordination of a base station as in the cellular paradigm. The performance in this case is evaluated for intra-cell traffic and highlights the good spatial reuse characteristics of the peer to peer communication. In [6] the multihop cellular network architecture (MCN) is proposed and an analytical evaluation of the capacity of the network is presented. In [3] a simulation study of the MCN is provided for TCP traffic and in the presence of mobility, and three approaches to alleviate the bottleneck around the AP are presented. The effect of interference in peer to peer communications is studied in [5] where the capacity of the ad-hoc networking paradigm is evaluated and scalability issues are addressed.

References

1. The network simulator ns-2, http://www.isi.edu/nsnam/ns/.
2. Hung-Yun Hsieh and Raghupathy Sivakumar. Performance comparison of cellular and multi-hop wireless networks: A quantitative study. In *Proc. of the ACM SIGMETRICS*, 2001.
3. Hung-Yun Hsieh and Raghupathy Sivakumar. On using the ad-hoc network model in wireless packet data networks. In *Proc. of the ACM MOBIHOC*, 2002.
4. Sang Bae Kaixin Xu, Mario Gerla. How effective is the ieee 802.11 rts/cts handshake in ad hoc networks? In *Proc. of the IEEE Globecom 2002*.
5. Jinyang Li, Charles Blake, Douglas S. J. De Couto, Hu Imm Lee, and Robert Morris. Capacity of ad hoc wireless networks. In *Proc. of ACM/IEEE MOBICOM*, pages 61–69, 2001.
6. Ying-Dar Jason Lin and Yu-Ching Hsu. Multihop cellular: A new architecture for wireless communications. In *Proc. of IEEE INFOCOM*, 2000.
7. Chunming Qioa and Hongyi Wu. icar: An integrated cellular and ad hoc relay system. In *Proc. of IC3N*, Las Vegas, NV, USA, October 2000.
8. Mineo Takai and Jay Martin. Effects of wireless physical layer modeling in mobile ad hoc networks. In *Proc. of the ACM MOBIHOC*, 2001.
9. Xiaxin Wu, Gary S. H. Chan, and Biswanath Mukherjee. Madf: A novel approach to add an ad-hoc overlay on a fixed cellular infrastructure. In *Proc. of the IEEE WCNC*, 2000.

Influence of Power Control and Link-Level Retransmissions on Wireless TCP

Niels Möller and Karl Henrik Johansson

KTH, Stockholm*

Abstract. A fundamental assumption of the TCP protocol is that packet losses indicate congestion on the network. This is a problem when using TCP over wireless links, because a noisy radio transmission may erroneously indicate congestion and thereby reduce the TCP sending rate. Two partial solutions, that improve the quality of the radio link, are power control and link-level retransmissions. By modeling these two lower layers of control loops, we derive an analytical model of the delay distribution for IP packets traversing a link. We investigate the effect on TCP, in particular the performance degradation due to spurious timeouts and spurious fast retransmits caused by delays and reorder on the link. It is shown that the models allow us to quantify the throughput degradation. The results indicate that link-level control and TCP interact, and that tuning one or the other is needed in order to improve performance.

1 Introduction

The TCP algorithms assume that packet losses indicate congestion on the network [1,2]. When using TCP over wireless links, packet drops due to radio noise makes TCP reduce its sending rate, hurting performance [3,4,5,6]. Two partial solutions, that improve the quality of the radio link, are power control and link-level retransmissions.

Power control tries to adapt the transmission power to varying channel characteristics, "fading", which can be thought of as a disturbance. Link-level retransmission, such as Automatic Repeat Request (ARQ), tries to hide losses from upper layers, e.g. TCP/IP, by resending damaged radio blocks. The IP packets still, however, experience random delays or even reorderings when they are transmitted across such a link. How to deal with these problems on the TCP level is the topic of intensive research, e.g., [6,7,8,9]. Despite recent progress, the influence of the radio link properties on TCP is far from fully understood.

In this article, we use models for these two link-layer processes, and derive the link properties, in particular the delay distribution, experienced by IP-packets. From these properties, we derive the effects on TCP throughput.

* This work was supported by the Swedish Research Council and the Swedish Center for Internet Technologies through the SRI project.

G. Karlsson and M.I. Smirnov (Eds.): QofIS 2003, LNCS 2811, pp. 193–202, 2003.

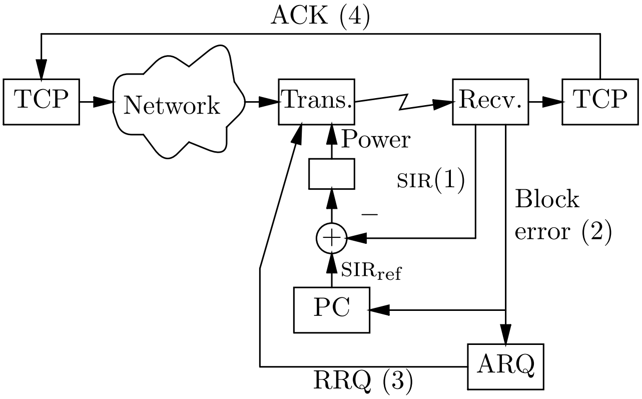


Fig. 1. System overview, including four feedback loops marked (1) – (4).

1.1 System Overview

When using TCP over a wireless link, there are several interacting control systems stacked on top of each other, illustrated in Figure 1. At the lowest level, the transmission power is controlled in order to keep the signal to interference ratio (SIR) at a desired level. This is a fast inner loop intended to reject disturbances in the form of "fading", or varying radio conditions. On top of this, we have an outer power control loop that tries to keep the block error rate (BLER) constant, by adjusting the target SIR of the inner loop. The radio channel and power control are modeled in Section 2. Next, we have local, link-level, retransmissions of damaged blocks, described in Section 3. Finally, we have the end-to-end congestion control of TCP. The effects the link layer control has on TCP performance is investigated in Section 4.

2 Radio Model

Data is transmitted over the radio link as a sequence of *radio blocks* (RB). One radio block corresponds to a *transmission time interval* (TTI) of 10 or 20 ms. Depending on the bandwidth (typically from 64 kbit/s to 384 kbit/s) the size of a RB can vary from 160 octets (the small 10 ms TTI is not used for the lowest data rates) to 960 octets.

The transmission of the radio blocks is lossy. Let p denote the overall probability that a radio block is damaged. The power of the radio transmitter is controlled, so that the loss probability stays fairly constant. The target block error rate is a deployment trade-off between channel quality and the number of required base stations. For UMTS the reference block error rate is often chosen to be about 10%, see [10]. In the following, we thus assume $p = 0.1$.

2.1 Power Control

The typical power control uses an inner loop that tries to keep the signal to interference ratio (SIR) close to a reference value SIR_{ref}. This loop typically has a sample frequency of 1500Hz, and a one bit feedback that is subject to a delay of two samples, i.e. 1.3 ms. In simulations [11], the inner loop is able to track the reference SIR within 2-3 dB, with a residual oscillation due to the severe quantization. The period of this oscillation is typically less than 5 samples, i.e. 3.3 ms.

As there is no simple and universal relationship between the SIR and the quality of the radio connection, there is also an outer loop that adjusts SIR_{ref}. This loop uses feedback from the decoding process; in this article we assume that the power control outer loop is based on block errors.

As it is hard to estimate the BLER accurately, in particular if the desired error rate is small, one approach is to increase SIR_{ref} significantly when an error is detected, and decrease the SIR_{ref} slightly for each block that is received successfully. It is interesting to note that this strategy resembles the TCP "additive increase, multiplicative decrease" congestion control strategy. For a survey of modern power control techniques for systems such as WCDMA, see [11].

2.2 Markov Model

The outer loop of the power control sets the reference value for the SIR. Given a particular reference value $\text{SIR}_{\text{ref}} = r$, the obtained SIR is a stochastic process. Together with the coding scheme for the channel, we get an expected probability for block errors. If the coding scheme is fixed, the probability of block errors is given by a function $f(r)$.

One can compute $f(r)$ from models of the channel and coding. For the BPSK example given in [12] we get

$$f(r) \approx 1 - \left(1 - Q(\sqrt{2\alpha e^{\beta r + \frac{1}{2}\beta^2\sigma^2}}\,)\right)^N \tag{1}$$

where α is the coding gain, $\beta = \frac{\ln 10}{10}$, σ is the standard deviation of the received SIR, N is the number of bits in a radio block, and $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-x^2} \mathrm{d}x$. We take $\alpha = 4$, from [12], $\sigma = 1$ dB, corresponding to a small SIR oscillation as in [11], and $N = 2560$, corresponding to a modest link capacity of 128 kbit/s.

However, for control purposes, even a crude approximation of the true $f(r)$ should be sufficient. In general, $f(r)$ is a decreasing, threshold-shaped function. For small enough r, $f(r) \approx 1$, i.e. almost all blocks are lost. For large enough r, $f(r) \approx 0$, i.e. almost no blocks are lost. The operating point of the power control is the point close to the end of the threshold where $f(r) = p = 0.1$, so it is the shape of $f(r)$ close to this point that is important.

The outer loop of the power control uses discrete values for SIR_{ref}. One way to keep the block error probability close to the desired probability p, is to use a fixed step size $\varDelta$ as follows. Whenever a radio block is received successfully, SIR_{ref}

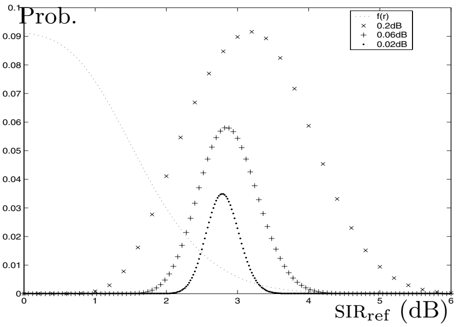


Fig. 2. Stationary distribution for the power control. Each mark represents one state of the power control, the corresponding value of SIR_{ref}, and its stationary probability. The dotted curve is the threshold-shaped function $f(r)$, scaled to fit in the figure, which represents the block error probability as a function of SIR_{ref}.

for the next block is decreased by $\varDelta$. And whenever a radio block is damaged, SIR_{ref} for the next block is increased by $K\varDelta$, where $1/(1+K) = p$. The value of $\varDelta$ is an important control parameter, which determines the performance of the power control.

For an integer K the varying SIR_{ref} can be viewed as a discrete Markov chain [12]. To do this, we have to make the assumption that block errors depend only on the value of SIR_{ref} when the block is transmitted; there is no relevant history except the SIR_{ref}. In our case $p = 0.1$ implies $K = 9$.

For a given function $f(r)$ and a step size $\varDelta$, we also modify f by setting $f(r) = 1, r < r_{\min}$ and $f(r) = 0, r > r_{\max}$. This results in a finite Markov chain, and it is straight forward to compute the stationary distribution for SIR_{ref}.

Figure 2 shows the threshold function $f(r)$, together with the stationary distribution for three values of $\varDelta$. Note that a smaller step size gives a more narrow distribution, and a smaller average SIR, which means that there is a trade-off between energy efficiency and short response times.

For a large $\varDelta$, the power control state, i.e. SIR_{ref}, will move quite far along the tail of the $f(r)$ threshold. For a smaller $\varDelta$, the control state will stay close to the operating point, so that a linearization of $f(r)$ would be a good approximation.

3 Link-Layer Retransmissions

The simplest way to transmit IP packets over the wireless link is to split each IP packet into the appropriate number of radio blocks, and drop any IP packet where any of the corresponding radio blocks were damaged. But as is well known, TCP interprets all packet drops as network congestion, and its performance is therefore very sensitive to non-congestion packet drops. An IP packet loss probability on the order of 10% would be detrimental.

There are several approaches to recover reasonable TCP performance over wireless links. In this paper we concentrate on a local and practical mechanism:

The link can detect block damage (this is needed for power control anyway), and it can use that information to request that damaged blocks be retransmitted over the link. This capability is an option in standard wireless network protocols, see [13] for an evaluation of these options in the IS-2000 and IS-707 RLP standards. Alternative approaches include changes to the TCP algorithms (e.g. Eifel, [7], and TCP Westwood [6]), and the use of forward error correction coding to add redundancy to the packets, either end-to-end as in [14] or at the link as in [15].

The effect of link level retransmissions is to transform a link with constant delay and random losses into a link with random delay and almost no losses.

There are several schemes for link level retransmission. We will consider one of the simpler, the (1,1,1,1,1)-Negative Acknowledgement scheme [16], which means that we have five "rounds", and in each round we send a single retransmission request. When the receiver detects that the radio block in time slot k is damaged, it sends a retransmission request to the sender. The block will be scheduled for retransmission in slot $k + 3$ (where the delay 3 is called the RLP NAK guard time). If also the retransmission results in a damaged block, a new retransmission request is sent and the block is scheduled for retransmission in slot $k + 6$. This goes on for a maximum of five retransmissions.

Consider the system at a randomly chosen start time, with the state of the power control distributed according to the stationary distribution. For any finite loss/success sequence (for example, the first block damaged, the next six received successfully, the eighth damaged), we can calculate the probability by conditioning on the initial power control state and following the corresponding transitions of the Markov chain. In the following sections, we use these probabilities to investigate the experience of IP packets traversing the link.

4 TCP/IP Properties

When transmitting variable size IP packets over the link, each packet is first divided into fix size radio blocks. We let n denote the number of radio blocks needed for the packet size of interest. For the links we consider, we have $n \leq 10$.

The outermost feedback loop we consider is the end-to-end congestion control of TCP. The role of TCP is to adapt the sending rate to the available bandwidth. The inputs to the TCP algorithms are measured roundtrip time, and experienced loss events, both of which depend on the links traversed by the TCP packets. To understand and predict the behavior of TCP, we must consider how it interacts with the power control and the link level retransmissions.

4.1 IP Packet Delay

One important link property for TCP is the delay distribution of packets traversing the link. Using the loss/success sequence probabilities from Section 3, we can explicitly compute the IP packet delay distribution. The expected value and standard deviation for three values of the step size are shown in Table 1. The times are measured in TTIs, and only the delay due to retransmissions is included (an n-block packet is of course also subject to a fix delay of n TTI).

Table 1. Mean and standard deviation of the IP packet delay distribution.

n	$\Delta = 0.2$	$\Delta = 0.06$	$\Delta = 0.02$
1	0.31±0.94	0.32±0.99	0.33±1.03
2	0.51±1.08	0.53±1.15	0.54±1.20
3	0.62±1.08	0.64±1.18	0.65±1.24
4	0.72±1.09	0.74±1.21	0.76±1.28
5	0.83±1.09	0.85±1.23	0.87±1.31
6	0.94±1.09	0.96±1.25	0.98±1.35
7	1.06±1.09	1.07±1.27	1.09±1.38
8	1.17±1.09	1.18±1.29	1.19±1.41
9	1.28±1.09	1.29±1.30	1.30±1.44
10	1.39±1.09	1.40±1.31	1.41±1.46

4.2 Timeout

A timeout event occurs when a packet, or its acknowledgement, is delayed too long. Let RTT_k denote the roundtrip time experienced by packet k and its corresponding acknowledgement. The TCP algorithms estimates the mean and deviation of the roundtrip time. Let $\widehat{\text{RTT}}_k$ and $\hat{\sigma}_k$ denote the estimated roundtrip time and deviation, based on measurements up to RTT_k. TCP then computes the timeout value for the next packet as $\widehat{\text{RTT}}_k + 4\hat{\sigma}_k$, which means that the probability that packet k causes a timeout is given by

$$P_{\text{TO}} = P(\text{RTT}_k > \widehat{\text{RTT}}_{k-1} + 4\widehat{\sigma}_{k-1})$$

We assume that the values RTT_k are identically and independently distributed according to the delay distribution given in Section 4.1. For simplicity, we also assume that the estimates $\widehat{\text{RTT}}_k$ and $\hat{\sigma}_k$ are perfect and equal to the true mean and standard deviation of RTT_k.

The value of P_{TO} will of course depend on the delay distribution. In TCP, timeout is the last resort recovery mechanism, and in order to get reasonable performance, a timeout must be a very rare event. Let us look at some examples:

- If RTT_k is uniformly distributed on an interval, then $P_{\text{TO}} = 0$.
- If RTT_k happen to be normally distributed, then timeouts will be rare: We get $P_{\text{TO}} \approx 1/(1.5 \cdot 10^4)$.
- For an arbitrary distribution with finite mean and variance, Chebyshev's inequality, $P(|X - \mu| \geq a\sigma) \leq 1/a^2$, yields the bound $P_{\text{TO}} \leq 1/16$.

We can expect the P_{TO} to lie somewhere between the extreme cases of a normally distributed delay, and the pessimistic value given by Chebyshev's inequality. When calculating P_{TO} from the delay distribution of Section 4.1, we get the probabilities in Figure 3.

We see that the probability for spurious timeout is significant for all packet sizes. The dependence on step size and packet size is complex, it seems to be peculiarities of the ARQ scheme, e.g. the "RLP NAK guard time", shining through.

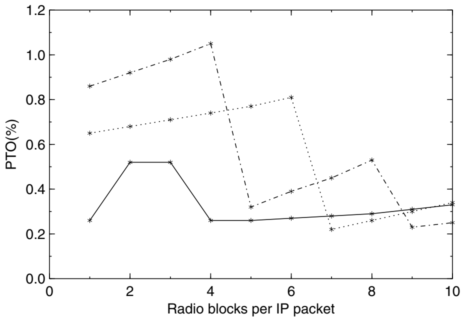


Fig. 3. Probability of spurious timeout. The solid line is for $\varDelta = 0.2$, dotted for $\varDelta = 0.06$ and dash-dotted for $\varDelta = 0.02$.

5 Throughput Degradation

In this section, we derive an estimate for the performance degradation due to either of spurious fast retransmits and spurious timeouts. The expression for the relative degradation depends only on the probability of the event in question, the type of event, and the number of packets in the maximum congestion window. We illustrate the procedure by calculating the degradation for an example link.

When computing TCP throughput, there are two distinct cases: Depending on the bandwidth-delay product, throughput can be limited either by the bandwidth of the path across the network, or by the maximum TCP window size.

For a small bandwidth-delay product, a modest buffer before the bottleneck link (which we will assume is our radio link) will be enough to keep the radio link fully utilized. Timeouts and fast retransmit events, if they occur with a low frequency, will not affect throughput at all. This can be seen for example in the performance evaluation [16]: In the scenarios that have a large maximum window size compared to the bandwidth-delay product, we get a throughput that is the nominal radio link bandwidth times $1 - p$, and there is no significant difference between different link retransmission schemes. Only when bandwidth or delay is increased, or the maximum window size is decreased, do we see a drastic changes in throughput when the BLER or retransmission-scheme varies.

Therefore, we will concentrate on the case of a large bandwidth-delay product. For a concrete example, we will consider the following scenario: Radio link bandwidth 384 kbit/s, maximum TCP window size $w = 8192$ bytes, packet size $m = 1500$ bytes, and a constant roundtrip delay time, excluding the radio link itself, of 0.2 s. We will also consider a smaller packet size of $m = 960$ bytes, making each IP packet fit in a single radio block.

T will denote the mean end-to-end roundtrip time, 0.25 s for the larger packets and and 0.23 s for the smaller packets. This implies that we have the large bandwidth-delay product case, with an ideal throughput w/T of 32 Kbyte/s for the large packets and slightly larger for the smaller ones. Note that this is smaller than the available radio bandwidth of the link, $384000(1 - p)/8 \approx 42$ Kbyte/s

Table 2. Performance degradation.

Degradation	Packet size	$\Delta = 0.2$	$\Delta = 0.06$	$\Delta = 0.02$
due to P_{TO}	Small	5.6%	13.0%	16.5%
	Large	4.9%	6.4%	8.4%
due to P_{FR}	Small	2.5%	5.9%	7.3%
	Large	0.0%	0.1%	0.2%

Assume that a spurious timeout occurs independently with a small probability P_{TO} for each packet. Consider a typical cycle starting with a timeout event, followed by $1/P_{\mathrm{TO}}$ packets that are sent without timeouts or retransmissions. We compare the throughput during this cycle with the ideal throughput w/T. For simplicity, we use only congestion windows that are an integral number of packets, rounding down when needed, and we also assume that the cycle length $N = 1/P_{\mathrm{TO}}$ is an integer. The cycle can be divided into two phases: An initial recovery phase of r roundtrip times, where we send ℓ packets, followed by the second phase of $(N - \ell)m/w$ roundtrip times where we send at full speed, w/m packets each roundtrip time, $N - \ell$ packets in all.

The throughput during one cycle can be expressed as

$$\text{throughput} = \frac{N}{r + (N - \ell)m/w} \frac{m}{T} \tag{2}$$

Compared to the ideal throughput w/T, the relative degradation boils down to

$$\left(\frac{w}{T} - \text{throughput}\right) / \frac{w}{T} = \frac{d}{d + N} = \frac{1}{1 + 1/(dP_{\mathrm{TO}})} \tag{3}$$

where we have substituted $d = rw/m - \ell$, the number of additional packets we would have sent during the cycle if we had never decreased our congestion window.

5.1 Degradation due to Timeouts

Consider the effect of timeout. When recovering from timeout, the slowstart threshold is set to $w/2$. The congestion window is reset to one packet, and doubled each roundtrip time, until it reaches the slowstart threshold. Above the slow start threshold, the usual increment of the congestion window of one packet per roundtrip time is used until we get pack to the maximum value w.

For $m = 1500$, we take $w = 5m = 7500$ (so that $w/m = 5$ is an integer). The length of the recovery phase is 4 roundtrip times, during which we send 1, 2, 3 and 4 packets. We get $d = 10$, and from the P_{TO} calculated in section 4.2 we get a performance degradation of 4.9%, 6.4%, and 8.4% for our three step sizes.

For $m = 960$ we take $w = 8m = 7680$. The length of the recovery phase is 6 roundtrip times, during which we send 1, 2, 4, 5, 6, 7 packets. Thus, $d = 23$, and we get a performance degradation of 5.6%, 13.0% and 16.5% for the three step sizes.

5.2 Degradation due to Fast Retransmit

With large delay variations, packets can get so severely reordered that they trigger the TCP fast retransmit. The probability of spurious fast retransmit (P_{FR}) can be estimated from the loss/success-sequence probabilities. It turns out that unless the link is configured to do "in-order delivery", P_{FR} is significant for $n = 1$ ($0.25\% < P_{\mathrm{FR}} < 0.8\%$). P_{FR} decreases very rapidly with increasing n.

The TCP performance degradation for both spurious timeout and spurious fast retransmit is summarized in Table 2. We lose up to 7% due to fast retransmits, when using small packets, and up to 16% due to timeouts.

6 Conclusions

The properties of the lossy radio link considered in this paper reduces TCP throughput. By modeling the underlying processes controlling transmission power and retransmission scheduling on the link (Section 2 and 3), we can calculate the link properties, in particular the delay distribution, which are relevant for TCP (Section 4). This helps us getting a better understanding of the TCP behavior when communicating over modern wireless links.

In Section 5 we found that when using IP packets that fit in one or two radio blocks, as is common for high bandwidth wireless links, the link will generate spurious fast retransmits. Spurious timeouts will also occur, and this effect is significant for both large and small packets. The timeout value in TCP makes timeout a very rare event if the roundtrip time has a uniform or normal distribution, but that is not the case for the delay due to link layer retransmissions.

Some possible approaches to improved TCP performance over radio links are:

- Change the TCP algorithms to make them more robust to link irregularities.
- Engineer the link layer, to give it properties that plain TCP handles well.
- Aim for small delays, both over the wireless link and in the rest of the network, and for saturation of the wireless link.

Improvements of the TCP algorithms is an area of intense research. A drawback is that deployment of new algorithms affect all Internet end systems, which makes it a slow and costly process.

Tuning the link properties is more practical from a deployment point of view, at least if the tuning can be done before widespread adoption of a new link type. For example, spurious fast retransmits can be eliminated by having the wireless receiver buffer packets, making sure that packets are never forwarded out of order (this "in-order" option is already available on real systems [10]). To reduce the number of spurious timeouts, the link or either TCP end point could artificially increase the delay variation by adding an extra uniformly distributed delay to packets or acknowledgements. Also the link level retransmissions themselves can be seen as an example of changing the link properties to make the link more TCP friendly.

The point of reducing delay is that if the bandwidth-delay product is small enough that the radio link can be saturated by an ordinary TCP sender, then a modest size buffer just before the radio link is enough to keep the link saturated, regardless of fluctuations in the TCP congestion window. It is hard to say if reducing bandwidth-delay product is a realistic objective, as it depends on the speed and delay of future link technologies. Extensions to TCP that increases the maximum window size would also help, for similar reasons [17].

References

1. Jacobson, V.: Congestion avoidance and control. In: Proc. of SIGCOMM. Volume 18.4. (1988) 314–329
2. Floyd, S.: TCP and explicit congestion notification. ACM Computer Communication Review **24** (1994) 10–23
3. DeSimone, A., Chuah, M., Yue, O.: Throughput performance of transport-layer protocols over wireless LANs. In: IEEE Globecom'93. (1993) 542–549
4. Xylomenos, G., Polyzos, G.C.: Internet protocol performance over networks with wireless links. IEEE Network **13** (1999) 55–63
5. Zorzi, R., Chockalingam, A., Rao, R.R.: Throughput analysis of TCP on channels with memory. IEEE J-SAC **18** (2000) 1289–1300
6. Mascolo, S., Casetti, C., Gerla, M., Sanadidi, M.Y., Wang, R.: TCP Westwood: bandwidth estimation for enhanced transport over wireless links. In: MobiCom, Rome, Italy (2001)
7. Ludwig, R., Katz, R.H.: The Eifel algorithm: Making TCP robust against spurious retransmissions. ACM Computer Communication Review **30** (2000)
8. Canton, A., Chahed, T.: End-to-end reliability in UMTS: TCP over ARQ. In: Globecom 2001. (2001)
9. Garrosa, P.M.: Interactions between TCP and channel type switching in WCDMA. Master's thesis, Chalmers University of Technology (2002)
10. Dahlén, A., Ernström, P.: TCP over UMTS. In: Radiovetenskap och Kommunikation 02. RVK (2002)
11. Gunnarsson, F., Gustafsson, F.: Power control in wireless communications networks — from a control theory perspective. Survey paper in IFAC World Congress, Barcelona (2002)
12. Sampath, A., Kumar, P.S., M.Holtzman, J.: On setting reverse link target SIR in a CDMA system. In: Proc. IEEE Vehicular technology conference. (1997)
13. Bai, Y., Rudrapatna, A., Ogielski, A.T.: Performance of TCP/IP over IS-2000 based CDMA radio links. In: Proc. of IEEE 52th VTC'2000-Fall, IEEE (2000)
14. Lundquist, H., Karlsson, G.: TCP with forward error correction. `http://www.it.kth.se/~hen/lundqvist_tcpfec.pdf` (2002)
15. Liu, B., Goeckel, D.L., Townsley, D.: TCP-cognizant adaptive forward error correction in wireless networks. `http://www-net.cs.umass.edu/~benyuan/pub/wirelessTCP.pdf` (2002)
16. Khan, F., Kumar, S., Medepalli, K., Nanda, S.: TCP performance over CDMA2000 RLP. In: Proc. IEEE 51st VTC'2000-Spring. (2000) 41–45
17. Jacobson, V., Braden, R., Borman, D.: TCP extensions for high performance. RFC 1323 (1992)

A Users' Satisfaction Driven Scheduling Strategy for Wireless Multimedia QoS

Leonardo Badia, Michele Boaretto, and Michele Zorzi*

DipInge, University of Ferrara, Italy
{lbadia,mboaretto,zorzi}@ing.unife.it

Abstract. In this work, we exploit game-theoretical concepts to depict the behaviour of multimedia users for the Radio Resource Management. Moreover, we also include pragmatic economic considerations, which allow studies of provider's revenue and possible charging mechanisms. These concepts are in particular applied to HSDPA scheduling procedures, whose main aim is to improve the performance of 3G networks and to allow extensions to a plethora of services. We briefly discuss a model for the users' satisfaction that includes both perceived QoS and pricing, already proposed to determine the QoS provisioning and network dimensioning. We apply the users' satisfaction function of this model in the scheduler. In this way we achieve improvements of the QoS as it is seen from the users' point-of-view, i.e., by involving the satisfaction of service constraints but also paid price. This analysis will be extended also to the provider's side, with considerations on the achievable revenue, that is an important aspect to take into account in service supplying.

1 Introduction

The recent standardisation of High Speed Downlink Packet Access (HSDPA) interface opens up the availability of a faster adaptation to the channel state for packet transmission in WCDMA networks. In HSDPA, the downlink channel is shared among all users, basically in a TDMA-like fashion. Thus, the following factors have an impact on the performance and determine different outcomes of the offered Quality of Service (QoS): the radio propagation conditions, the scheduling technique and the coexistence of many users in the same sector.

In [1] the concept of Channel-State Dependent Scheduler is introduced, i.e., it is shown that it is fundamental for the scheduling technique to be aware of the user's instantaneous link state condition. A general conclusion, that is still valid in the case under exam, is that the overall throughput for the sector can be maximised if the scheduler uses its knowledge of the channel state, so that only users with good channel are served. On the other hand, to meet QoS constraints, it can be necessary to supply a certain degree of fairness, that implies to serve also users with bad channel condition.

The issue of the satisfaction of users' QoS requirements is not trivial, as it can be related to the economic problem for the provider to achieve an adequate revenue. For a real operator, this aspect can not be neglected, since the network maintainance is possible only if the costs of service provisioning are overcome.

* This work has been partially supported by Ericsson Research.

G. Karlsson and M.I. Smirnov (Eds.): QofIS 2003, LNCS 2811, pp. 203–213, 2003.

In order to emphasise the analysis of this matter in more depth, we adopt a utility-based approach. Utility functions have been widely used in the recent technical literature, with particular focus on the modeling of users' satisfaction in Radio Resource Management (RRM) problems [2] [3]. Moreover, for the scheduling, several solutions can be seen as application of a utility-based framework: the scheduling issue is often seen and solved as a problem of priorising users in a queue, and this can be done by defining appropriate weights, like in the Weighted Fair Queuing (WFQ) [4] or WF^{2}Q [5] schedulers. In [6] a mathematical formulation of the RRM issue, directly applicable to our case, has been developed, by involving utility functions with assigned properties, that depict the soft tunability of the QoS requirements. Hence, the RRM goal is seen as a strategy of maximizing the system utility, that in this approach includes only the users' welfare.

In [7] this concept was extended, by including also pricing and demand effects. This means that the welfare of the network is also related to the money exchange between user and service provider. In fact, users are likely to be satisfied if they obtain an acceptable QoS, but also if they pay a fair price for the offered service. Thus, the users' satisfaction should be an increasing function of the offered QoS; however, the higher the price, the lower the satisfaction. Hence, we should refer to economics in two directions, i.e., for utility functions and for what concerns the pricing strategies [8].

In this work, we take into account this trade-off between users' and provider welfare by considering a model of the users' satisfaction as previously discussed. Moreover, we apply different strategies of scheduling on the HSDPA channel within the satisfaction framework. It has also to be taken into account in the welfare maximisation, that the operator can not offer the service without adequate revenue. In this way, the provider welfare can be expressed in economic terms; we will discuss how a satisfactory revenue can be obtained from the given algorithms, and we compare different scheduling techniques also in terms of achievable revenue. Thus, the main contribution of the present paper is to investigate if well-known strategies which obtain an optimal scheduling under the technical point-of-view, i.e., by maximising the throughput, are efficient if revenue is taken into account, or there is margin of improvement, and how large.

The work is organised as follows: in Section 2 we present the analytical model for the users' satisfaction, including both pricing and utilities. In Section 3 we discuss different scheduling strategies under the theoretical point-of-view and we outline how they can be modified to take into account the revenue improvement aspect. Section 4 presents simulation results and Section 5 concludes the work.

2 Model for the Users' Satisfaction

We present a model that employs the concept of utility function, used in micro-economics to classify and sort the customer preferences. Thus, we are considering the distribution among the users of a scarce resource, represented with a generic quality-related parameter g ($g \geq 0$), and the results of this operation are seen with a mapping through a utility function $u(g)$. In our study, the utility and the preferences are related to multimedia wireless services, in particular under the point-of-view of the Radio Resource assignment. Hence, g represent the assigned network resource. Note that the analysis can be easily extended by replacing g with a multi-dimensional vector. For the scheduling issue, g can be identified with the assigned rate. In this analysis we will study the rate assignment *a priori*, i.e., as it is before the transmission. Thus, g is the assigned transmission rate on the HSDPA channel. Indeed, to have a better adherent model we should replace this parameter with the throughput in terms of correctly delivered packets, as this is a more appropriate metric to depict the customer satisfaction. However, it is still possible to do this in the given framework, with only small modifications. In fact, note that these two quantities (assigned rate and achieved throughput) are strictly connected.

If there are N users in the network, in general each user will have a different utility function $u_i(g)$, with $i = 1, 2, \ldots, N$. The utility level of the ith user will depend on the assignment of g_i. Even though we do not investigate in detail how the utility functions can be derived in different networks, we exploit some of their general properties. For example, since a larger amount of bandwidth can be useless but can not hurt, the utility functions $u(g)$ are assumed to be non decreasing functions of g. On the other hand, the economic law of diminishing marginal utilities states that the improvement of the quality due to a larger amount of g becomes smaller as g increases. Formally:

$$\forall i = 1, 2, \ldots, N \ \ \frac{\mathrm{d}u_i(g)}{\mathrm{d}g} \geq 0 \quad \text{and} \quad \lim_{g \to \infty} \frac{\mathrm{d}u_i(g)}{\mathrm{d}g} = 0\,. \tag{1}$$

The right part of previous equation can be also almost equivalently expressed as the observation that there is a value g_{max} such that the utility $u(g_{max})$ is the upper limit for each $u_i(g)$, in other words:

$$\forall i = 1, 2, \ldots, N \ \ \lim_{g \to \infty} u_i(g) \approx u_i(g_{max})\,. \tag{2}$$

This formulation is not fully equivalent to (1) but it is almost always verified when technological constraints are involved. In this case, g_{max} can be seen as the maximum amount of resource that can be received from the technological support (user terminal). Note that the conditions on the utilities expressed above imply that every $u_i(g)$ is necessarily concave for g greater than a given value. In this work we will consider also 0 as the minimum achievable utility, that corresponds to the utility of not receiving the service at all, i.e. $u(0) = 0$. This condition can also be changed if run-time service degradation are considered, as in this case the utility could go even below 0; in fact, it is commonly assumed preferable to be not admitted at all than to be disconnected from the network while receiving the service. Thus, a lowest utility equal to 0 corresponds to a condition of *ideal Admission Control.*

Different types of utility functions can be introduced to model different kinds of service, as discussed in [9]. For example, discrete-value utilities (i.e., combination of step functions) are an appropriate characterisation for the simplest kinds of traffic, like GSM-like voice call. Here, the aim of introducing the utilities is only to determine whether the service is acceptable or not, but it does not matter how much it is appreciated.

According to the chosen mathematical representation, there are several ways to combine the users' utilities. Commonly, the utilities are assumed to be additive [3], thus, their aggregate is simply a sum; in other models, different combinations are proposed. In any case, however, the aggregate of the users' utilities contributes to the network welfare, and the first goal of the RRM can be seen with a *naïve* point-of-view as the welfare maximisation.

This leads to a straightforward application of the general model as presented in [6], by considering the RRM as an optimisation problem:

$$\max W(\mathbf{g}) \quad W(\mathbf{g}) = \sum_{i=1}^{N} u_i(g_i) \tag{3}$$

$$\text{s.t. } \textstyle\sum_{i=1}^{N} g_i < K_{max} \; , \tag{4}$$

where W is the network welfare, defined as an aggregate of the utilities and thus function of the vector $\mathbf{g}$ of the assigned g's. Equation (3) can be replaced with other similar aggregation of the utilities. Also the kind of capacity characterising the network can be represented with different conditions than Eq. (4). In this case, Equation (4) represents a *hard capacity* system, that is a TDMA- or FDMA-like systems with fixed assigned maximum quantity K_{max} of allocable resource. In general it can be written: $\mathcal{K}(\mathbf{g}) < K_{max}$ with an appropriately defined capacity constraint $\mathcal{K}$. The main point, yet, is that in the expression of the welfare $W(\mathbf{g})$ only the single user utilities are worth.

It seems to be more realistic to consider at the same time instead also the effect of the pricing [10] This is not only suggested for the sake of a realistic model. In fact, the strategies of charging users for the offered service generate revenue, by improving at the same time the network management. For example, a cheap service is also likely to be abused of. Moreover, between the pricing strategies that are almost equally acceptable by a willing-to-pay user, the operator should choose the one that provides the highest revenue, since in this way its own satisfaction is increased. As long as this can be done without decreasing too much the users' welfare, this implies indeed a more efficient resource usage; in other words, wastes are avoided.

However, it should also be considered that in general a price variation can greatly affect the users' demand, resulting in a translation of (3)-(4) into different optimisation problems. From the point-of-view of the network provider, it can be assumed that users that do not face both adequate QoS and affordable price are unsatisfied customers. Thus, it can be assumed that these users pay only a fraction of the tariff they are supposed to. This can be seen also with a probabilistic approach: if we map the satisfaction A_i of the ith user into the range $[0, 1]$, we can express the expected value of the revenue as

$$R = \sum_{i=1}^{N} A_i p_i \tag{5}$$

where p_i is the price paid from the ith user. This model was proposed in [7] to depict the users' satisfaction coming from a static heuristic rate assignment. Actually it is possible to extend the framework to the case under exam, i.e., a more detailed scan of the solution to the allocation problem, which can constitute an adequate scheduling strategy.

In more detail, A_i represents the satisfaction, or the *acceptance probability* of the ith user. The latter denomination better identifies the meaning of this parameter, that is indeed probabilistic. In sufficiently large networks, or after a sufficiently long run, it is likely that A_i is also the fraction of satisfied customers, i.e., the ones who keep paying for the service without abandoning it or being driven to other operators.

There are several possibilities to define A_i, with different kinds of parameters. However, the basic dependences are the offered QoS, that is better described through its perception represented by the utility u_i, and the paid price p_i. Hence, we will consider $A_i = A(u_i, p_i)$ for each i, by assuming that every user in the network follows the same decision criterion to decide whether they are satisfied or not. On the other hand, note that the users are differentiated in the perception of the offered service ($u_i(g)$ changes according to the index i). Also the price can vary among the users.

For each definition of the acceptance probability, the following properties have to hold:

$$\begin{aligned} \forall(u,p) \neq (0,0) \ : \quad & \frac{\partial A}{\partial u} \geq 0\,,\ \lim_{u\to 0} A(u,p) = 0\,,\ \lim_{u\to\infty} A(u,p) = 1\,, \\ & \frac{\partial A}{\partial p} \leq 0\,,\ \lim_{p\to 0} A(u,p) = 1\,,\ \lim_{p\to\infty} A(u,p) = 0\,. \end{aligned} \tag{6}$$

where the limits $u \to \infty, p \to \infty$ should be intended only in a mathematical sense, as it is likely that they do not represent a feasible situation. For example, as discussed above, u is upper-bounded to $u(g_{max})$. This justifies the fact that the condition $u_i = p_i = \infty$ could not have a completely coherent definition with the above properties. Note that even the value of A on the edge point $u_i = p_i = 0$ is also difficult to be properly defined; however, this point does not affect the definition as $A_i p_i$ of the revenue contribution. In fact, this case represent a user not admitted, thus the paid tariff is 0, and this is verified for any value we decide to assign to $A(0,0)$ in the interval $[0,1]$.

A satisfactory definition of A is henceforth:

$$A(u(g_i), p(g_i)) \triangleq 1 - \exp(-C\frac{u^{\mu}}{p^{\epsilon}})\,. \tag{7}$$

Note, however, that almost identical results can be obtained from any expression of $A(u,p)$ that verifies the above properties. Furthermore, note that in the more general case of p seen as a function of g it is realistic to require that also $p(g)$ is an increasing function of g.

Thus, the total revenue expressed by Equation (5) can be rewritten as $R(\mathbf{g})$, where $\mathbf{g} = (g_1, g_2, \ldots, g_N)$ is the vector of the assigned g's. With these considerations, the optimisation problem (3)-(4) can be rewritten as

$$\max R(\mathbf{g}) = \sum_{i=1}^{N} A_i(g_i) p_i(g_i) \tag{8}$$

$$\text{s.t. } \mathcal{K}(\mathbf{g}) < K_{max}\,. \tag{9}$$

In this work we do not consider analytical solutions to the problem. Instead, we analyse the behaviour of classical scheduling algorithms for what concerns the revenue and present possibilities of improvement. Moreover, we will quantify how much the revenue can be increased with appropriate techniques.

3 Scheduling Algorithms Framework

Let us present how classical scheduling algorithms can be extended to the HSDPA release of UMTS. The choice of the scheduling strategy has a major impact on the system performance: however, in HSDPA the channel conditions might be fastly tracked to improve the system throughput. In this kind of system the MAC features are located in the node-B, in order to evaluate the rapid variations of the wireless channel, i.e., fast fading.

In a wireless network, considering the instantaneous radio conditions is a fundamental task, because of the location-dependent and bursty errors typical of this kind of systems. For example, a user in a fading dip may experience a bad channel and may be unable to transmit for a certain period of time. The scheduling framework has to consider the channel conditions and to give priority to users that perceive a clean channel; users with a poor SIR will be delayed until they have a better propagation scenario.

Such a policy permits the maximisation of total throughput because it minimises packet retransmissions. Nevertheless some degree of fairness is required, in order to prevent users' starvation. In a Round Robin (RR) scheduling resources are allocated to the communication link without taking into account the channel conditions but only on a sequential basis, with a high degree of fairness but with the potential risk of not considering the propagation scenario, causing a possible high number of retransmissions. Due to the poor performance exhibited in this sense by the pure RR scheduler, in the following we will not analyse this strategy. In fact, to guarantee the QoS requirements it is necessary to find a trade-off between a pure SIR-based heuristic and a round robin scheduling, i.e., between the throughput maximisation and the number of users that can achieve a given QoS.

We consider various kinds of heuristics involved in the scheduling process. For example, we might introduce a traditional SIR-based heuristic, called C/I, with a greedy assignment of the available resources [11]; such a policy permits to obtain the maximum sector throughput, but users that perceive a bad channel may have a poor assignment of resources. In other words, the sharing of the codes is basically characterised by a high degree of unfairness.

As opposed to such a policy we introduce, as an original contribution, a utility-function-based assignment, with the aim of increasing the achievable revenue. Considering an assigned rate dependent on a utility function permit not only to obtain a better degree of fairness, but also a generally better allocation, in particular in terms of fairness. This happens because the rate is assigned by following the perceived user-utility and not only the channel state.

The policies will be compared in terms of revenue and users' admission in order to highlight the consequences on the provider side. In a utility-based approach it is possible to assign the resources according to more complex issues compared to a simple C/I policy; user parameters like SIR, buffer state, deadline of the packets can be considered and mixed in a more efficient manner.

In our proposed strategy the scheduling process starts from a solution obtained with a greedy heuristic and modifies this assignment giving more resources to the user with the highest marginal utility, in order to improve the total sector utility. Resources are subtracted to user with the minimum marginal utility, to obtain a variation of the total utility as little as possible. This algorithm is based on a local search of the optimal solution, ending when the goal function reaches a local maximum.

The heuristic for the starting solution affects the result of the local-search in two ways: firstly, it should be a good solution in itself, as this improves the convergence of the algorithm. Secondly, it has to be general enough to scan the solution range without being specialised to a peculiar subset of cases. Even though the solution coming from the C/I algorithm is good, we prefer for these reasons to choose another option.

In the following, the starting solution g_{i0} for the rate assignment for the ith user will be determined by means of the marginal utility, i.e., $u'_i(g)$. In particular, g_{i0} is the highest feasible value that implies a marginal utility equal to a given threshold ϑ. According to the value of ϑ, different heuristics are determined in this way. The meaning of this heuristic, and the meaning of ϑ, are the following: the marginal utility $u'(g)$ represents the increase of the QoS perception for increasing g. Thus, g_{i0} is the highest value that can guarantee a relative increase larger than ϑ; beyond this value, the increase is always lower. Obviously the choice of ϑ depends on the kind of strategy that the provider wants to follow in the assignment, as a value of ϑ close to 0 implies to assign almost the maximum meaningful value, that is, $u_i(g_{i0}) \approx u_i(g_{max})$. On the other hand, the larger ϑ, the lower the starting assigned rate g_{i0}.

After this initial condition, the assignment can be modified, by means of a local-search algorithm obtaining a local optimum solution.

Moreover, the revenue will also depend on the pricing strategy. Thus, the choice of the function $p(g)$ should be indicated to clarify the above definition of revenue, as given in Equation (5). In the literature [12], different pricing strategies have been proposed, and obviously the pricing strategy choice heavily affects the value of the total revenue. In this work we will consider two kinds of pricing policies, mainly for their simplicity of concept. The first one is a *flat price* strategy, i.e., the price is fixed for any value of the assigned rate. The second policy represents instead a simple usage-based pricing with linear price. This means that $p(g) = kg$ is linearly related to g through a given constant k. It is interesting to observe that in Equation (5) there is expressed a double dependence of the revenue on the pricing, as also A_i is a function of the price.

For what concerns the utilities $u_i(g)$, they are assumed to be sigmoid functions of g, with general definition:

$$\forall i = 1, 2, \ldots, N \quad u_i(g) \triangleq \frac{(g/K_i)^{\zeta_i}}{1 + (g/K_i)^{\zeta_i}}, \tag{10}$$

where the parameters $K_i > 0$ and $\zeta_i \geq 2$ depend on the index i, so that different users may follow different utility functions. In the simulations, K_i and ζ_i are randomly generated with uniform distribution within a given interval.

With these definitions, in the next we will study the behaviour of the classical C/I scheduling policy against our proposal introduced to improve the revenue by simulations. However, this aspect of the scheduling could be studied even from the point-of-view of finding a theoretically optimal resource allocation.

4 Results

In this Section we will present the results relative to the HSDPA interface obtained with a UMTS simulator developed at the University of Ferrara, in which detailed user dynamics have been implemented. The simulation environment consists of a 3×3 hexagonal cells' structure. The cells' cluster is wrapped onto itself in order to avoid the

Table 1. List of Parameters of Simulation Scenario.

Parameter (symbol)	value
cell radius (d)	250 m
gain at 1 m (A)	-30dB
Hata path loss exponent (α)	3.5
shadowing parameter (σ)	4dB
Doppler frequency (f_d)	2Hz
mean SNR at cell border	40dB
max assignable rate (g_{max})	96 codewords
utility parameter (ζ)	$5.0 \div 8.0$
utility parameter (K)	$0.2 \div 6.0$
acceptance prob. parameter (C)	0.5
acceptance prob. parameter (μ)	2.0
acceptance prob. parameter (ϵ)	4.0

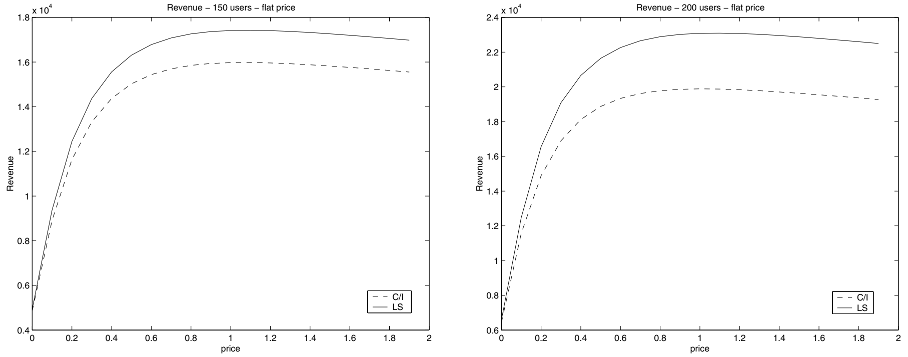


Fig. 1. Revenue for flat price, 150 (left) and 200 (right) users, as a function of the price.

"border effect". In radio channel propagation, path loss, fast fading and shadowing have been included. By considering the environment mobility, a non-zero Doppler frequency is assigned, even though stationary users are considered. Table 1 reports the data for the simulation scenario and the Acceptance-probability model.

The compared scheduling strategies are the previously introduced C/I and the original proposal based an iterative search of the local maximum for the revenue, from a heuristic starting point in which the marginal utilities are allocated to the value $\vartheta = 0.5$. With the Acceptance-probability model it is possible to evaluate the earned revenue. This is done in Figure 1 for the flat pricing policy and in Figure 2 for the usage-based policy. In both cases, two different situations have been represented, with 150 and 200 users respectively.

As it was to be expected, the revenue obtained with the local-search strategy (indicated in the Figures as "LS") is better than for the C/I strategy. In fact, the former searches for solutions that improve the revenue. However, note that the qualitative behaviour is similar. Moreover, simulations also show that the number of iterations of the local-search procedure is low (usually 4–5 iterations); thus, it might be said that a sim-

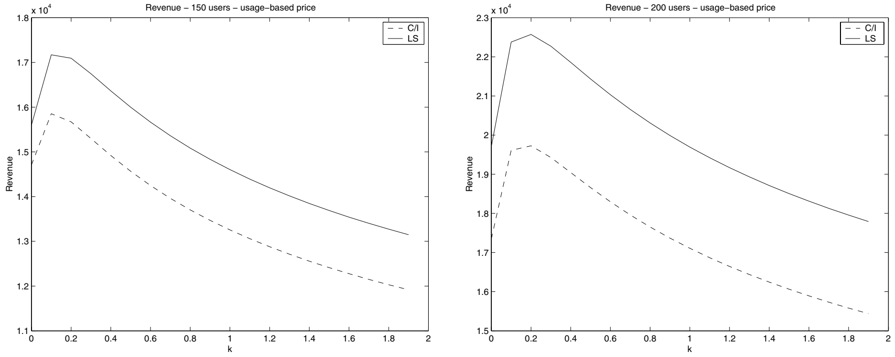


Fig. 2. Revenue for $p(g) = kg$, 150 (left) and 200 (right) users, as a function of k.

ple variation from the initial solution allows to greatly improve the earned revenue[1]. With the LS strategy we are able to obtain a revenue improvement of 10% approximately in the case with 150 users for both pricing policies. For the 200 users' network, the improvement is even larger, being greater than the 20%. In this way the provider welfare is increased, by leading to a more efficient resource usage, at least from the operator's point-of-view.

As a further observation, note that the curves present an optimal price in both strategies. The existence of a price value that maximises the revenue comes directly from the conditions given by Equation (6). However, the fact that the maximising price is approximately the same for both curves implies that the gap is not due to a different behaviour with respect to the price, but it is structural. In other words, for the same pricing conditions, the LS strategy is able to achieve a higher revenue since it allocates the resources in a more satisfactory way for the users.

In Figure 3, the admission rate is represented, i.e., the percentage of users which are satisfied and achieve a rate g larger than 0. The acceptance probability model is used here to determine whether a user is satisfied or not. This satisfaction rate can be equivalently seen as admission rate, since users with very low assignments are likely to be unsatisfied, so they can be considered as not admitted. In the most extreme case, a user with assigned rate equal to 0 can be seen as a user surely blocked. Here, 150 users and flat price policy are depicted; however, the curves are similar also for different number of users or pricing strategy. It can be observed that the larger the price, the lower the admission rate. Yet, it should also be noted that the LS strategy allows a larger admission rate than the C/I strategy. This is due to the revenue improvement that the LS strategy tries to accomplish. In other words, not only the revenue, but also the total users' satisfaction is larger with the LS strategy. This confirms that the LS strategy succeeds in achieving a higher revenue for the operator without hurting the users.

Finally, in Figure 4 the sensitivity of the results to the value of ϑ is highlighted. Here, three cases ($\vartheta = 0.3, 0.5, 0.7$) are shown for both revenue and admission rate in a flat price situation with 200 users. The situation with different pricing policies or

[1] Note that the most interesting price region is the one which leads to the highest values of the revenue. Here, the improvement is obviously more consistent.

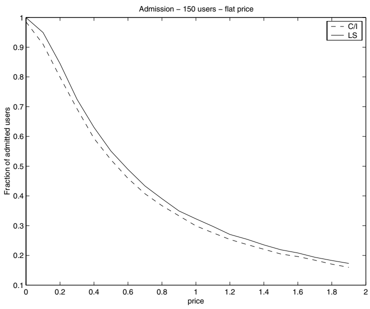


Fig. 3. Admission rate for flat price, 150 users, as a function of the price.

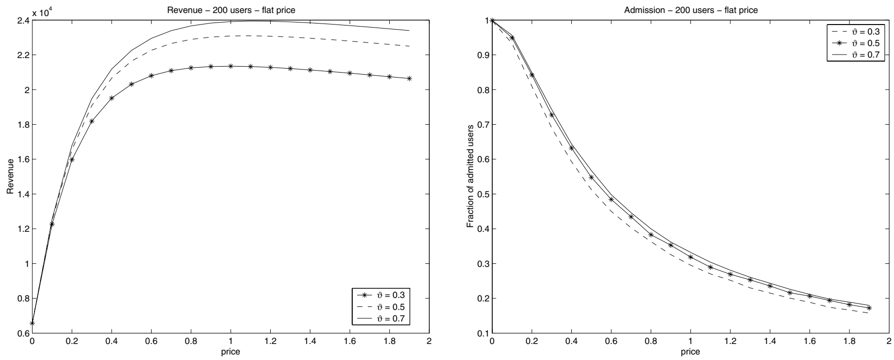


Fig. 4. Revenue (left) and Admission rate (right) for 200 users and variable ϑ.

users' number is yet quite similar. It can be observed that different LS solutions are obtained for different values of the marginal utility starting value ϑ. Even though the qualitative behaviour is similar for the three curves, it can be concluded that a parameter optimisation can improve even further the performance. Thus, a deeper investigation on the effect of the choice of ϑ, in which also considerations about the market strategy of the provider play a role (see [7]) is necessary and can be the subject of a future analysis.

5 Conclusions and Future Work

The analysis from the provider's point-of-view of the HSDPA scheduler shows that there are several possibilities of improving the network management under this aspect. This can be easily seen with the introduction of the *Acceptance-probability* model, which considers the joint effect of user utility and price, by allowing to account for economic considerations.

The results show that the application of a classical efficient strategy, like the C/I scheduler, by neglecting the economic counterpart of the allocation, can lead to unsatisfactory results for the operator, even though the C/I strategy provides a maximised throughput. On the other hand, a simple strategy that locally searches for higher values of the revenue is able to greatly improve the profit and the economic efficiency of the resource management, by keeping the users' satisfaction level almost constant, if not increased. Thus, the usefulness of the economic considerations is emphasised. Besides, several further observations open up on how the network welfare can be improved. For example, the simple heuristic strategies discussed here offer the advantages of simplicity and fast evaluation; however, the optimisation of the internal parameters can improve even further the performance and/or the convergence rate.

Finally, from a theoretical point-of-view it could be possible to study within the given framework the behaviour of a more general scheduling strategy, in which the revenue maximisation is considered as the goal of the optimisation problem (8)–(9). This study, that can allow to gain a better understanding of the RRM issues, is left for future research.

References

1. C. Fragouli , V. Sivaraman, M. Srivastava, "Controlling Multimedia wireless link via enhanced class-based queueing with channel state dependent packet scheduling," *Proceedings of INFOCOM 1998*, Joint conference of IEEE Computer and Communication Society, Volume 2, 1998.
2. M. Xiao, N.B. Shroff, E.K.-P. Chong, "Utility-Based Power Control in Cellular Wireless Systems," *Proceedings of INFOCOM 2001*, Joint Conference of the IEEE Computer and Communication Societies, pp 412–421, 2001.
3. L. Song, N. Mandayam, "Hierarchical SIR and Rate Control on the Forward Link for CDMA Data Users under Delay and Error Constraints," *IEEE JSAC*, IEEE Journal on Selected Areas in Communications, Volume 19, Issue 10, pp 1871–1882, 2001.
4. A. Demers, S. Keshav, S. Shenker, "Analysis and simulation of a fair queuing algorithm," *Proceedings of ACM SIGCOMM 1998*, Conference on Applications, Technologies, Architectures and Protocols for Computer Communication, pp 1–12, 1998.
5. J.C.R. Benner, H. Zhang, "WF^{2}Q: worst case fair weighted fair queuing," *Proceedings of INFOCOM 1996*, Joint Conference of the IEEE Computer and Communication Societies, pp 120–128, 1996.
6. F.P. Kelly, A. Maulloo, D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability", *Journal of the Operational Research Society*, Volume 49, pp 237–252, 1998.
7. L. Badia, M. Lindström, J. Zander, M. Zorzi, "Demand and Pricing Effects on the Radio Resource Allocation of Multimedia Communication Systems," *Proceedings of GLOBECOM 2003* (to appear), 2003.
8. H.R. Varian, "Intermediate Microeconomics: A Modern Approach," *Norton*, New York, 1999.
9. D. Famolari, N. Mandayam, D. Goodman, V. Shah, "A New Framework for Power Control in Wireless Data Networks: Games, Utility and Pricing", *Wireless Multimedia Network Technologies*, Kluwer Academic Publishers, pp 289–310, 1999.
10. E.W. Fulp, M. Ott, D. Reininger, D.S. Reeves, "Paying for QoS: an optimal distributed algorithm for pricing network resources," *Proceedings of IWQoS98*, International Workshop on Quality of Service, pp 75–84, 1998.
11. T.J. Moulsley, "Performance of UMTS HSDPA for Data Streaming," *Proceeding of Third International Conference on 3G Mobile Communication Technologies*, 2002.
12. C. Courcoubetis, F.P. Kelly, V.A. Siris, R. Weber, "A study of simple usage-based charging schemes for broadband networks," *Telecommunications Systems*, Volume 15, pp 323–343, 2000.

Effects on TCP from Radio-Block Scheduling in WCDMA High Speed Downlink Shared Channels

Ulf Bodin[1] and Arne Simonsson[2]

[1] Luleå University of Technology, SE - 971 87 Luleå, Sweden
uffe@sm.luth.se.
Tel.: +46 920 492 897,
[2] Ericsson Research, SE - 971 87 Luleå, Sweden.

Abstract. Avoiding delay jitter is essential to achieve high throughput for TCP. In particular, delay spikes can cause spurious timeouts. Such timeouts force TCP into slow-start, which may reduce congestion window sizes drastically. Consequently, there may not always be data available for transmission on bottleneck links. For HS-DSCH, jitter can occur due to varying interference. Also, properties of the radio-block scheduling influence the jitter. We evaluate, through simulations, effects on TCP from scheduling. Our evaluation shows that round-robin (RR) schedulers can give more jitter than SIR schedulers. SIR schedulers discriminates low SIR users to improve spectrum utilization while RR schedulers distribute transmission capacity fairly. The high jitter with RR scheduling cause however both lower utilization and decreased fairness in throughput among users than with SIR scheduling. The Eifel algorithm makes TCP more robust against delay spikes and reduces thereby these problems.

1 Introduction

The High-Speed Downlink Shared Channel (HS-DSCH) in Wideband CDMA (WCDMA) release 5 introduces support for peak bit-rates for data services exceeding 8 Mbps [6][7]. Moreover, delays considerably lower than for data channels in previously releases of WCDMA are supported. This makes HS-DSCH suitable for bursty Internet traffic (e.g., web page data objects transferred using TCP).

HS-DSCH implements higher order modulation (i.e., 16QAM) and fast link adaptation for good spectrum efficiency and to offer high bit-rates. More robust QPSK modulation is also implemented to transmit to users experiencing high interference. This is needed since WCDMA systems are expected to be interference limited (i.e., caused by concurrent transmissions).

In HS-DSCH, fast radio-block scheduling can be used to further increase the efficiency in spectrum usage when the system is interference limited. E.g., a scheduler giving precedence to connections with high signal-to-interference ratios (SIR) is likely to result in higher spectrum utilization [7]. This gain may however come at the price of unfairness among competing users.

Avoiding delay jitter is important for TCP. In particular, delay spikes may cause spurious timeouts, which results in unnecessary retransmissions and multiplicative decreases in congestion window sizes [5]. Fortunately, these problems are reduced

G. Karlsson and M.I. Simirnov (Eds.): QofIS 2003, LNCS 2811, pp. 214–223, 2003.

through the Eifel algorithm [1][2][3]. Also, a more conservative management of the TCP retransmit timer and a more careful response of the TCP sender on duplicate ACKs reduce the problems of spurious [4].

Interference can cause jitter in HS-DSCH. The choice of channel coding and modulation for a user is based on its SIR, which is affected by path-loss, fading, and interference from transmissions. The interference originating from transmissions within a users´ own cell is limited through orthogonal coding, while interference from transmissions in other cells is limited by distance.

Radio-block scheduling influences the jitter. With round-robin (RR) schedulers the delay of individual IP packets is determined by the number of users being active and by the SIR for each receiving user. With schedulers giving precedence to high SIR users the delay of IP packets to these users depends less on the number of users being active. Consequently, with SIR scheduling, the delay and jitter can be low for high SIR users while low SIR users may experience considerable delay. Long delay does not however necessary mean high jitter. I.e., with SIR scheduling, IP packets are likely to be either sent rapidly or delayed until high SIR users no longer have IP packets available for transmission.

We evaluate, through simulations, effects on TCP Sack [9] from RR and SIR scheduling respectively. Also, we study effects on TCP Reno with the Eifel algorithm. The simulation platform used, the network simulator version 2 (ns-2) [8], include a fairly detailed model of TCP.

Our evaluation shows that a RR scheduler may give more jitter than a SIR scheduler. The jitter is severe enough for TCP to experience delay spikes causing spurious timeouts. TCP Sack sources suffering frequently from such timeouts may not always have data available for transmission in HS-DSCH (i.e., since they are forced into slow-start by spurious timeouts). This reduces the average number of users being active. Consequently, delay spikes decrease the spectrum utilization and fairness in throughput among users. This means that transmission fairness cannot always be directly translated to fairness in throughput for TCP.

With TCP Reno and the Eifel algorithm, delay spikes causing spurious timeouts do not force TCP sources into slow-start. Thereby, this algorithm can reduce the fairness and utilization problem of TCP with RR scheduling.

2 Primer on HS-DSCH

The higher order modulation is considered useful for HS-DSCH although CDMA systems are typically interference limited. This is because allocating a large part of the available downlink power for a transmission time interval (TTI)[1] to a single user can give fairly high SIR. Periods of high SIR can however rapidly pass onto periods of considerable lower SIR. Consequently, fast link adaptation to change coding and modulation between TTIs to adjust for instantaneous channel conditions is essential.

Reasons for SIR to vary include changes in available power due to other channels transmitting, path-loss and shadow fading due to user movements, and fluctuating multi-path fading. Changes in available power can be considered to affect users

[1] HS-DSCH uses a TTI of two ms. This short TTI enables low delays, high granularity in the radio-block scheduling process, and accurate tracking of time-varying channel conditions.

within a cell randomly. However, path-loss and fading affects users individually. While path-loss and shadow fading changes at rather long time-scales, the time-scales of multi-path fading are typically around the time-scale of an HS-DSCH TTI.

The approach for HS-DSCH is to use the remaining cell power after serving other dedicated and common channels. The power for HS-DSCH is thus likely to change over time. However, this power is expected to change on considerable longer time-scales than a HS-DSCH TTI and it can therefore be assumed constant for evaluations of TCP performance over HS-DSCH.

HS-DSCH is shared primarily in the time-domain. All codes and power available for HS-DSCH can be allocated to one user for a TTI. It may however not be possible to always utilize the available payload space with one user. To avoid wasting payload space, a limited degree of code multiplexing is supported. E.g., up to four simultaneous transmissions can be allowed to increase the amount of payload data for each TTI. Then, appropriate coding and modulation are chosen separately for the estimated SIR of each respective user.

The short TTI used for HS-DSCH enables radio-block scheduling at high granularity to improve the radio spectrum utilization. Using predications on channel quality made for fast link adaptation a scheduler can give users with high SIR precedence to transmission slots. Then, fairness in the distribution of transmission capacity may however need to be sacrificed for improved spectrum utilization.

Other schedulers not accounting for SIR estimates such as RR schedulers can be used for HS-DSCH to distribute transmission capacity fairly. Also, schedulers exploiting the trade-off between high spectrum utilization and transmission fairness can be used for HS-DSCH.

Unfortunately, for data transfers using TCP, transmission fairness cannot always be directly translated to fairness in application data throughput. This is because TCP throughput may be affected by additional forwarding quality metrics such as delay and jitter. In particular, delay spikes may reduce spectrum utilization and fairness in throughput among users.

3 Delay Spikes in HS-DSCH

Delay spikes can be considered as a general problem in radio networks. Mechanisms such as ARQ and changing radio conditions cause varying transmission delays, which may be interpreted as delay spikes by TCP. In this section we focus on the specific properties of HS-DSCH that can be expected to cause transmission delays to vary.

In HS-DSCH, low SIR users experience lower transmission rates than high SIR users (i.e., due to lower rate coding and lower order modulation). Sudden decreases in SIR can therefore appear as delay spikes for individual IP packets. As mentioned in Section 2 there are several reasons for why SIR is likely to vary.

In addition to lower rate coding and lower order modulation, a SIR based scheduler may cause delay spikes to users with low SIR (since data transfers for users with low SIR will be interrupted when data arrives for users with higher SIR). These delay spikes are however likely to be followed by periods of longer delay with low variation because high SIR users occupying the channel.

A potential advantage with SIR scheduling regarding jitter is that high SIR users finish their transfers quickly. A reasonable assumption is that users become inactive for a period immediately after finishing a transfer. Then, users with worse radio conditions can transmit during periods of lower interference at which users with superior radio conditions are inactive.

With RR schedulers the delay of individual IP packets is determined by the number of users being active and by the SIR for each receiving user. Since transmission capacity is distributed fairly, the scheduler cannot explore variations in SIR to optimize the spectrum utilization as a SIR scheduler does. While users with temporally high SIR may finish transfers before their SIR degrades with SIR scheduling, such users may need longer time to finish transfers with RR scheduling (i.e., they stay active for longer periods). Changes in SIR are then likely to translate into IP packet transmission delay variations, which may appear as delay spikes for TCP. This suggests higher variations in transmission delay for IP packets with RR scheduling than with SIR scheduling.

For HS-DSCH, IP packets are stored in separate buffers for each user. This means that traffic flows only experiences limited degrees of statistical multiplexing in these buffers. Consequently, variations in queue length would cause considerable jitter even with fixed IP packet transmission delays. Varying transmission delay is likely to make the jitter experienced by TCP worse.

4 TCP and Delay Spikes

For optimally throughput, users waiting for data should have all their data buffered and available for HS-DSCH immediately as it is requested (i.e., assuming that HS-DSCH is a bottleneck). The congestion control mechanisms of TCP precludes however such buffering. In particular, being in slow-start TCP does not always have data buffered at bottlenecks. Typically, a TCP source in slow-start initially transmits two segments of data and then waits for the receiver to acknowledge them before releasing more data[2]. This results in TCP sources alternating between releasing bursts of data and being idle until they have opened their congestion window enough to always have data buffered for HS-DSCH. For short transfers, TCP may never reach such a window size.

TCP detects congestion through two complementary indicators. Firstly, when receiving three duplicate acknowledgements (ACKs) a TCP source assumes that the next expected bytes are lost due to buffer overflow somewhere in the data path. The missing bytes are retransmitted and the congestion-window is reduced by half to avoid further buffer overflows. This action is referred to as fast retransmit.

For duplicate ACKs to be sent, new data must be delivered to the receiver. This is not always possible since the data lost may be the final bytes of a transfer or the congestion can be severe enough to cause multiple packet losses for the TCP session in question. Therefore, TCP also maintains a timer to detect lost data. A retransmit timeout occurs when transmitted data is not acknowledged before a certain time limit (i.e., the retransmit timeout (RTO)). This limit is adjusted using measured round-trip-times (RTTs).

[2] An initial sending window between two and four segments is recommended when wireless links are in the path [5].

When the RTT suddenly increases it may exceed the RTO. This is because the RTO is determined using previously measured RTTs that were considerably shorter. Such increases can occur due to decreases in available forwarding capacity and due to high jitter appearing as delay spikes to TCP.

A spurious timeout occurs when a RTT suddenly increases and exceeds the RTO. When facing a timeout, TCP retransmit the segment of data it assumes to be lost (i.e., the segment for which the timeout occurred). However, for spurious timeouts, an ACK for the originally transmitted segment will eventually arrive. Since TCP cannot distinguish different ACKs for the same data, it must interpret the ACK as acknowledging the retransmitted segment. This means that it also must assume all other outstanding segments to be lost. Thus, the TCP sender goes into slow-start and retransmits all these segments.

Unfortunately, the TCP receiver generates a duplicate ACK for each segment received more than one. This is because it must assume the ACKs for these segments are lost. These duplicate ACKs may cause spurious fast retransmits and thus multiplicative decreases in the congestion window size of the TCP source. The unnecessary retransmission of data is undesirable since they may increase the traffic load considerably. This is particularly important to avoid for radio channels (i.e., because forwarding capacity often is scarce at such links due to limited radio spectrum).

The Eifel algorithm uses timestamps to distinguish ACKs for originally transmitted segments from ACKs for retransmitted segments [1]. Thereby, only one segment needs to be retransmitted at a spurious timeout and the TCP source is not entering the slow-start phase.

5 Evaluation

In this section we evaluate through simulations effects on TCP from scheduling. For the simulations, we have implemented a model of HS-DSCH into NS-2 [8]. We evaluate TCP Sack and TCP Reno with the Eifel algorithm, and with a RR and a SIR scheduler respectively.

5.1 Models and Assumptions

The radio model includes lognormal shadow fading with a standard deviation of 8dB and exponential path loss with a propagation constant of 3.5. Multi-path fading is not included. The block error rate (BER) is modeled as uniformly distributed errors. Self-interference is assumed limited to 10 percent and the interference from other transmissions within a user's own cell is assumed limited to 40 percent (i.e., through code division). The interference from transmissions in other cells than a user's cell is limited by distance only.

The ARQ mechanism modeled immediately retransmit damaged radio-blocks (i.e., no fast hybrid ARQ is implemented). We assume that 12 codes out of 16 are allocated for HS-DSCH. The coding and modulation combinations used in the simulations are listed in Table 1. Users experiencing SIR less than –3.5 dB faces a BER of 50 percent, while users with SIR equal to or higher than –3.5 dB are exposed to a BER of 10 percent. Up to three users may transmit in the same slot.

Table 1. Coding and modulation combinations

Coding (rate)	Modulation (type)	SIR (dB)	Bit-rate (Mbps)	Radio-block size (bytes)
0.25	QPSK	-3.5	1.44	360
0.50	QPSK	0	2.88	720
0.38	16QAM	3.5	4.32	1080
0.63	16QAM	7.5	7.20	1800

For TCP, the three-way handshake is excluded (i.e., sources send data with the initial SYN segment). A typical TCP implementation sends data after the three-way handshake. The expected consequence of this is that we are likely to overestimate throughputs for short transfers. Besides the segment size, which is set to 1450 bytes, the default TCP parameters in ns-2 are left unchanged for the simulations [8]. This means that the minimum retransmit timeout (RTO) is set to one second and that the TCP timer granularity is set to 0.1 second.

5.2 Simulation Setup

For each simulation, 56 mobile terminals are randomly distributed on a cell plan consisting of seven cells. Antennas are omni-directional with 500 m cell radius. The transmission power is 10W for all TTIs. Mobile terminals are stationary and thus do not change cells during simulations. Simulations with 12 different seeds are however made to test different locations of the mobile terminals.

Each user (mobile terminal) downloads a number of files. The file size of each transfer is randomly chosen among seven different sizes: 4350, 10150, 21750, 44950, 91350, 184150, and 369750 bytes. Transfer file sizes are selected aiming at liner increase in number of file transfers with decreasing file sizes. Seven times more 4350 bytes files are transmitted than 369750 bytes files. File sizes are selected to result in MTU sized packets only.

The MTU is set to 1500 bytes[3]. With this MTU, the segment (payload) size is 1460 bytes for TCP Sack. However, with Eifel, which requires the time-stamp option to be used, the segment size becomes 1450 bytes. To compare results for TCP Sack and for TCP Reno with the Eifel algorithm we set the segment size to 1450 for all simulations.

The one-way propagation delay of wired links between sources and base stations is set to 75 ms for all users. Wired links are over-dimensioned and the RTTs are thus 150 ms plus delays introduced in HS-DSCH. Buffer capacity to store 32 IP packets is allocated for ach user.

The waiting time between downloads is exponentially distributed with mean two seconds. Waiting periods are initiated when transfers are finished. This means that the system load increases with increasing transfer rates (i.e., the sooner a transfer is finished, the sooner the preceding transfer is initiated). Each simulation runs for five simulated minutes.

[3] The MTU of 1500 bytes is used since this is the payload frame size of Ethernet.

With the loads resulting from the above given parameters, the system is interference limited without causing numerous users with particularly bad radio-conditions to be completely locked-out. Such users may however be locked-out for periods and thus get only a few transfers through. The length of these lockout periods can be considerable longer without the Eifel algorithm than with this algorithm due to exponential back-offs.

5.3 Results

The mean IP packet transmission delay is longer with TCP Reno and the Eifel algorithm (from now referred to as TCP Eifel) than with TCP Sack (i.e., 44.4389 ms and 30.7101 ms respectively with SIR scheduling, and 47.9291and 20.5483 ms respectively with RR scheduling). As mentioned in Section 5.2, the system load increases with increasing transfer rates (i.e., transfers finishing faster results in higher load since that increases the number of transfers made during the simulation period). Hence, the longer packet transmission delays for TCP Eifel in the simulations indicate that it manages to achieve higher transfer rates than TCP Sack. The total aggregated throughputs verify this observation. For TCP Sack the aggregated throughput is 3.27325 Mbps with SIR scheduling and 2.57319 Mbps with RR scheduling. The difference in aggregated throughput is less evident (i.e., 4.77384 Mbps with SIR scheduling and 4.71201 Mbps with RR scheduling).

The aggregated throughput can be higher for TCP Eifel than for TCP Sack although the mean delay is longer. The reason for this is delay spikes causing spurious timeouts, which forces TCP Sack into slow-start. TCP Eifel recovers from such timeouts and continues transmitting in the congestion-avoidance mode. In the simulations, because of higher jitter TCP Sack faces considerable more severe degradations in throughput with RR scheduling than with SIR scheduling.

The difference in throughput between RR and SIR scheduling is likely to be larger with higher load causing more interference [7]. Thus, the minor difference in throughput for TCP Eifel with SIR and RR scheduling respectively may be larger at higher interference. However, the jitter can also be expected to vary with the load for both the RR scheduler and the SIR scheduler. This may affect the degradation in aggregate throughput for TCP Sack. We consider effects from varying loads as for further studies.

The IP packet transmission delay distributions differ between SIR and RR scheduling (Fig. 1). With SIR scheduling, IP packets are transmitted with low delay, or they face considerable delays while the delay is more evenly distributed with RR scheduling. This indicates that the jitter is higher with RR scheduling than with SIR scheduling[4]. Therefore, we expect that with RR scheduling TCP connections are more likely to experience delay spikes causing spurious timeouts than with SIR scheduling.

Fig. 2 shows that for long transfers the numbers of TCP timeouts are higher for TCP Sack with RR scheduling than with SIR scheduling. The corresponding numbers for TCP Eifel are however similar. The reason for this is for further studies. A possible explanation can however be that since TCP Eifel allow for less varying load and less varying interference than with TCP Sack. This may reduce the number of delay spikes experienced by TCP.

[4] In addition to transmission delay variations, the jitter experienced by TCP includes delay variations caused by queuing.

We look closer into aggregated throughputs by studying them individually for the different files sizes used in the simulations. In Fig. 3 it can be seen that for short transfers (i.e., small file sizes) aggregated throughputs are similar for the TCP versions and schedulers evaluated. Differences in aggregated throughputs are however evident for longer transfers (expect for TCP Eifel with SIR and RR scheduling respectively).

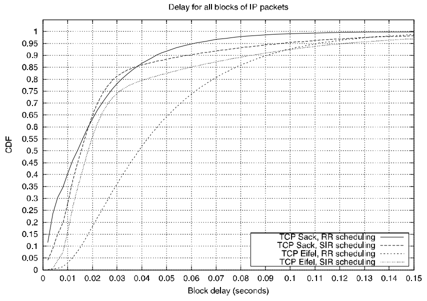


Fig. 1. IP packet delay distributions

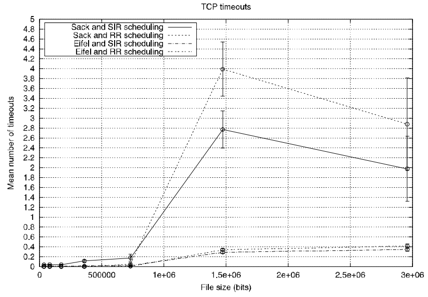


Fig. 2. Mean number of TCP timeouts

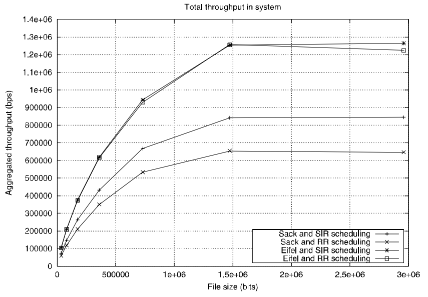


Fig. 3. Aggregated system throughputs

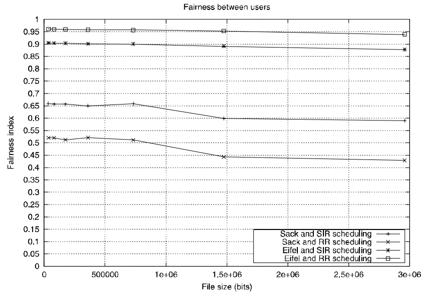


Fig. 4. Fairness in user throughputs

The larger degradation for long transfers appears because they may be in the congestion-avoidance mode before timeouts force them into slow-start. Short transfers never leaves slow-start and a timeout does therefore not degrade the throughput as much as for longer transfers. The considerable lower aggregate throughput for TCP Sack compared to for TCP Eifel indicates that the jitter is high with both schedulers. We consider the issue of whether throughputs for TCP Eifel can be improved or not by reducing the jitter as for further studies.

The degradations in aggregate throughputs caused by delay spikes also affect the fairness in throughputs among users. Fig. 4 shows worse fairness for TCP Sack with RR scheduling than with SIR scheduling[5]. The RR scheduler distributes transmission capacity fairly. This fairness do not translates into fairness in throughput among users

[5] The fairness index is defined by R. Jain in [11]. An index equal to one implies perfect fairness, while lower values of this index means unfairness in throughput among users.

for TCP Sack because of delay spikes causing spurious timeouts (Fig. 4). For TCP Eifel, which can handle such timeouts without going into slow-start, the fair distribution of transmission capacity provided by RR scheduling does however translate into fairness in throughput.

In Fig. 5, it can be seen that for short transfers the mean throughputs are higher for TCP Eifel with the RR scheduler than with the SIR scheduler. For long transfers, the mean throughputs are however higher with the SIR scheduler. This can be explained by that long transfers facing low interference can capture the channel with SIR scheduling, but not with RR scheduling. New transfers arriving are thus more likely to achieve transmission slots without considerable delay with RR scheduling than with SIR scheduling. This gives higher throughput to short transfers.

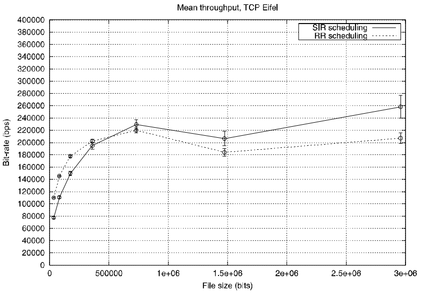


Fig. 5. Mean throughputs (90% confidence intervals), Eifel

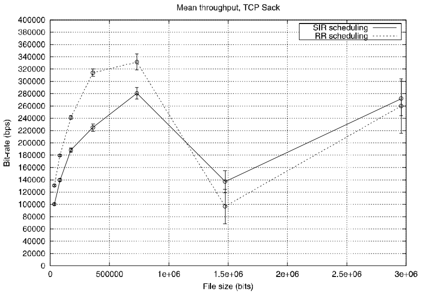


Fig. 6: Mean throughputs (90% confidence intervals), Sack

As for TCP Eifel, the mean throughputs are higher for short transfers with RR scheduling than with SIR scheduling for TCP Sack[6]. The larger difference with TCP Sack compared with TCP Eifel can be explained by the low fairness for TCP Sack with RR scheduling (Fig. 6). I.e., user throughputs are distributed over a larger span since users experiencing multiple spurious timeouts go frequently into slow-start, which reduces the traffic load. This leaves other users to experience exceptional good radio conditions.

6 Conclusions

In this paper we evaluate effects on TCP from radio-block scheduling in WCDMA HS-DSCH. TCP is sensitive to sudden increases in delay since this can cause spurious timeouts. Such timeouts results in unnecessary retransmissions and multiplicative decreases in congestion window sizes. The Eifel algorithm makes TCP more robust against sudden increases in delay.

High jitter can appear as sudden increases in delay for TCP. We refer to such increases as delay spikes. We show that round-robin (RR) radio-block scheduling can

[6] Since confidence intervals overlap for long transfers we cannot say anything about these transfers.

give higher jitter than SIR scheduling. Because of this high jitter, which causes spurious timeouts, the fair distributions of transmission capacity may not be translated to fairness in throughput among users for TCP. Moreover, the spectrum utilization can be decreased due to this jitter.

With the Eifel algorithm, the fair distribution of transmission capacity is better translated to fairness in throughput among users. Also, the utilization of the radio spectrum is improved. This indicates that the Eifel algorithm should be used when RR scheduling is used in HS-DSCH.

RR scheduling can be preferable with the Eifel algorithm since it can give similar spectrum utilization as SIR scheduling does, but better fairness in throughput among users and higher throughput for short transfers (i.e., small file sizes). We believe users to be more sensitive to delay for short transfers, which often belong to interactive applications such as web browsing. At scenarios in which interference is high, SIR scheduling may give considerable higher spectrum utilization than RR scheduling for TCP Eifel. We consider this as for further studies.

When the Eifel algorithm is not used, our study indicates however that SIR scheduling is clearly preferable in HS-DSCH. Without the Eifel algorithm, RR scheduling gives in our simulations worse fairness in throughput and lower spectrum utilization than SIR scheduling.

Multi-path fading is not used in the simulations. Such fading is likely to occur at similar time-scales as the transmission time interval for HS-DSCH. Hence, the results presented herein may be affected by multi-path fading. We consider this as for further studies.

References

1. Ludwig R. and Katz R. H., The Eifel Algorithm: Making TCP Robust Against Spurious Retransmissions, ACM CCR, Vol. 30, No. 1, Jan. 2000.
2. Ludwig R. and Mayer M., The Eifel Detection Algorithm for TCP, Internet DRAFT (work in progress), Dec. 2002.
3. Ludwig R. and Gurtov A., The Eifel Response Algorithm for TCP, Internet DRAFT (work in progress), Mar. 2003.
4. Gurtov A., On Treating DUPACKs in TCP, Internet DRAFT (work in progress), Oct. 2002.
5. Inamura H., Montenegro G., Ludwig R., Gurtov A., Khafizov F., TCP over Second (2.5G) and Third (3G) Generation Wireless Networks, Internet RFC 3481, Feb. 2003.
6. Parkvall S. et al., The Evolution of WCDMA Towards Higher Speed Downlink Packet Data Access, Proceedings of VTC 2001 (Spring).
7. Parkvall S., Peisa J., Furuskär M., Samuelsson M., and Persson M., Evolving WCDMA for Improved High Speed Mobile Internet, Proceedings of FTC 2001, Nov. 2001.
8. The network simulator – ns-2, URL: http://www.isi.edu/nsnam/ns/.
9. Mathis M., Mahdavi J, Floyd S., and Romanow A., TCP Selective Acknowledgment Options, Internet RFC 2018, Oct. 1996.
10. Jacobson V., Congestion Avoidance and Control, ACM Sigcomm, Aug. 1988.
11. Jain R., The Art of Computer Systems Performance Analysis, techniques for experimental design, measurement, simulation, and modeling, John Wiley & Sons, Inc. ISBN 0-471-50336-3, 1991.

Author Index