Lecture Notes in Computer Science 2698
Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

Springer
Berlin
Heidelberg
New York
Hong Kong
London
Milan
Paris
Tokyo

Wojciech Burakowski Berthold Koch
Andrzej Bęben (Eds.)

Architectures for Quality of Service in the Internet

International Workshop, Art-QoS 2003
Warsaw, Poland, March 24-25, 2003
Revised Papers

Springer

Volume Editors

Wojciech Burakowski
Andrzej Bęben
Warsaw University of Technology
Institute of Telecommunications
ul. Nowowiejska 15/19
00-665 Warsaw, Poland
E-mail: {wojtek/abeben}@tele.pw.edu.pl

Berthold Koch
PMC
Johann-Keller-Weg 8 a
86919 Utting, Germany
E-mail: Bert.Koch@t-online.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, H.3, H.4, D.2, K.4

ISSN 0302-9743
ISBN 3-540-40444-9 Springer-Verlag Berlin Heidelberg New York

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

http://www.springer.de

Printed in Germany

Typesetting: Camera-ready by author, data conversion by PTP-Berlin GmbH
Printed on acid-free paper SPIN: 10928004 06/3142 5 4 3 2 1 0

Preface

Providing Quality of Service (QoS) in the Internet is currently the most challenging topic for researchers, industry and network providers. Now, the only available service in the Internet is the best effort service, which assumes traffic is processed as quickly as possible, but there is no guarantee as to timely or actual delivery. On the other hand, there is pressure to offer new applications in the Internet (like VoIP, videoconferencing, on-line games, e-commerce, etc.) but this requires some packet transfer guarantees from the network (e.g., low packet transfer delay, low packet losses). To meet these requirements, new architectures for providing IP- based QoS in the Internet are proposed: Integrated Services (*IntServ*) and Differentiated Services (*DiffServ*). However, these architectures need some enhancements to provide adequate solutions for resource management, signaling, traffic engineering, traffic handling mechanisms, etc.

In the European research community, a number of projects inside the Fifth Framework Programme were addressed solving the above issues; among these AQUILA (*Adaptive Resource Control for QoS Using an IP-Based Layered Architecture*), CADENUS (*Creation and Deployment of End-User Services in Premium IP Networks*), and TEQUILA (*Traffic Engineering for Quality of Service in the Internet at Large Scale*) are excellent examples. The main achievements from these projects are the prototypes for fixed QoS IP networks. The extension of these proposed solutions into the wireless environment is the next step.

The *Workshop on Architectures for Quality of Service in the Internet*, jointly held with the *Final AQUILA IST Seminar – Art-QoS 2003*, was organized to bring together researchers working on providing Quality of Service for IP-based networks. The intention was to discuss architectural aspects and traffic control mechanisms supporting end-to-end QoS.

The AQUILA project started in January 2000 with 12 partners from 6 European countries (Austria, Finland, Germany, Greece, Italy and Poland). The project has defined a comprehensive framework for the support of QoS in IP-based networks. The proposed solutions were implemented in the form of prototypes and tested at the AQUILA trial sites in Helsinki, Vienna and Warsaw. During the 2003 Workshop two special sessions devoted to AQUILA were held.

March 2003

Wojciech Burakowski
Berthold Koch
Andrzej Bęben

Organization

The Art-QoS 2003 Workshop, jointly held with the Final AQUILA IST Seminar, was organized by the Institute of Telecommunications, Warsaw University of Technology, Poland.

Program Committee

Co-chairs
Wojciech Burakowski, Warsaw University of Technology, Poland
Berthold Koch, Siemens AG, Germany

Members
Jose Brazio, Telecommunications Institute, Lisbon, Portugal
Andrzej Dąbrowski, Warsaw University of Technology, Poland
Franco Davoli, University of Genoa, Italy
Gerald Eichler, T-Systems Nova, Germany
Hermann Granzer, Siemens AG, Germany
Ulrich Hofmann, Salzburg Research, Austria
Heinrich Hussmann, Dresden University of Technology, Germany
Laszlo Jereb, Budapest University of Technology and Economics, Hungary
Yannis Karadimas, Q-Systems, Greece
Ilkka Norros, VTT Information Technology, Finland
James Roberts, France Telecom R&D, France
Stefano Salsano, University of Roma "Tor Vergata," Italy
Paulo de Sousa, European Commission
Phuoc Tran-Gia, University of Wuerzburg, Germany
Iakovos S. Venieris, National Technical University of Athens, Greece
Manuel Villen Altamirano, Telefonica I+D, Spain
Józef Woźniak, Technical University of Gdansk, Poland

Referees

A. Bąk, WUT, Poland
A. Bęben, WUT, Poland
C. Brandauer, SPU, Austria
J. Brazio, TIL, Portugal
W. Burakowski, WUT, Poland
D. Bursztynowski, WUT, Poland
T. Czachórski, SUT, Poland
M. Dąbrowski, WUT, Poland
F. Davoli, Univ. of Genoa, Italy
G. Eichler, T-Systems Nova, Germany
A. Elizondo, Telefonica I+D, Spain
T. Engel, Siemens AG, Germany
H. Hussmann, TUD, Germany
C. Jędrzejek, ITTI, Poland
S. Kaczmarek, GUT, Poland
Y. Karadimas, Q-Systems, Greece
D. Katzengruber, TAA, Austria
B. Koch, Siemens AG, Germany
S. Koehler, Univ. Wuerzburg, Germany
U. Krieger, Univ. Frankfurt, Germany
J. Lubacz, WUT, Poland
K. Malinowski, WUT, Poland
M. Menth, Univ. Wuerzburg, Germany
J. Milbrandt, Univ. Wuerzburg, Germany
M. Pióro, WUT, Poland
F. Ramón, Telefonica I+D, Spain
F. Ricciato, Univ. of Rome, Italy
J. Roberts, FT R&D, France
S. Salsano, Univ. of Rome, Italy
F. Stohmeier, SPU, Austria
A. Tomaszewski, WUT, Poland
M. Villen, Telefonica I+D, Spain
I. Venieris, NTUA, Greece
J. van der Wal, TNO, The Netherlands
J. Woźniak, GUT, Poland

Local Organizing Committee

A. Bąk, WUT, Poland
A. Bęben, WUT, Poland
M. Dąbrowski, WUT, Poland
M. Fudała, WUT, Poland
H. Tarasiuk, WUT, Poland
E. Tarwacka, WUT, Poland

Sponsoring Institutions

NASK – Research and Academic Computer Network, Poland
ATM S.A., Poland
DGT Sp. z o.o., Poland
IEEE Chapter 19, Warsaw, Poland

Table of Contents

Architectures for Next Generation Networks

The Evolving Telecommunications Network 1
Martin Potts

An IP QoS Architecture for 4G Networks 18
Janusz Gozdecki, Piotr Pacyna, Victor Marques, Rui L. Aguiar, Carlos Garcia, Jose Ignacio Moreno, Christophe Beaujean, Eric Melin, Marco Liebsch

Integration of Mobility-, QoS-, and CAC-Management for Adaptive Mobile Applications 29
Daniel Prokopp, Michael Matthes, Oswald Drobnik, Udo Krieger

Architectures and Services

A Control Architecture for Quality of Service and Resource Allocation in Multiservice IP Networks 49
Raffaele Bolla, Franco Davoli, Matteo Repetto

Control Plane Architecture for QoS in OBS Networks Using Dynamic Wavelength Assignment 64
Sungchang Kim, JinSeek Choi, Minho Kang

IP Services Market: Modelling, Research, and Reality 76
Piotr Arabas, Mariusz Kamola, Krzysztof Malinowski

Signalling

Prototype Implementation for the Analysis of SIP, RSVP and COPS Interoperability 88
Tien Van Do, Barnabás Kálmán, Csaba Király, Zsolt Pándi

Reinforcement Learning as a Means of Dynamic Aggregate QoS Provisioning 100
Nail Akar, Cem Sahin

Admission Control

Calculating End-to-End Queuing Delay for Real-Time Services on an IP Network 115
Robert E. Kooij, Olaf Østerbø, J.C. van der Wal

Admission Control Method Based on Effective Delay for Flows Using EF PHB 127
Marcin Narloch, Sylwester Kaczmarek

QoS Provisioning for VoIP Traffic by Deploying Admission Control 139
Hung Tuan Tran, Thomas Ziegler, Fabio Ricciato

AQUILA: Resource Control

Overview of the Project AQUILA (IST-1999-10077) 154
Bert F. Koch, Heinrich Hussmann

Application Support by QoS Middleware 165
Falk Kemmel, Sotiris Maniatis, Anne Thomas, Charilaos Tsetsekas

BGRP Plus: Quiet Grafting Mechanisms for Providing a Scalable End-to-End QoS Solution 177
Eugenia Nikolouzou, Petros Sampatakos, Lila Dimopoulou, Stefano Salsano, Iakovos S. Venieris

AQUILA: QoS at Work

Measurement-Based Admission Control in the AQUILA Network and Improvements by Passive Measurements 189
Marek Dąbrowski, Felix Strohmeier

An Implementation of a Service Class Providing Assured TCP Rates within the AQUILA Framework 203
Christof Brandauer, Peter Dorfinger

Evaluation of the AQUILA Architecture: Trial Results for Signalling Performance, Network Services and User Acceptance 218
Marek Dąbrowski, Gerald Eichler, Monika Fudała, Dietmar Katzengruber, Tero Kilkanen, Natalia Miettinen, Halina Tarasiuk, Michael Titze

MPLS Traffic Engineering

CSPF Routed and Traffic-Driven Construction of LSP Hierarchies 234
Michael Menth, Andreas Reifert, Jens Milbrandt

Load Balancing by MPLS in Differentiated Services Networks 252
Riikka Susitaival, Jorma Virtamo, Samuli Aalto

Traffic Control Mechanisms

An Integrated Scheduling for Multiple Loss Priority Traffic in E-PON OLT Switches 265
Myoung Hun Kim, Hong Shik Park

Differentiation and Interaction of Traffic: A Flow Level Study 276
Eeva Nyberg, Samuli Aalto

Application-Oriented Evaluation of Measurement Estimation 291
Adam Wierzbicki, Lars Burgstahler

Author Index 305

The Evolving Telecommunications Network

Martin Potts

Martel GmbH, Bern, Switzerland
Tel: +41 31 994 2525, martin.potts@martel-consulting.ch

Abstract. This paper is based on a services and technologies Roadmap produced by the NGN Initiative (NGN-I) project[1] within the European Union's 5th Framework Programme IST (Information Society Technologies). It includes not only issues concerning technological evolution, but also highlights the more interactive way that the actors will communicate with each other in the future, and the types of end-user interaction.

1 Introduction

The term "Next Generation Networks" is wide-ranging and is interpreted variously by the broad variety of players involved in the communications business.

However, there is general agreement that the goal of Next Generation Networks is to bring services to customers in a manner that is:

- in accordance with the trend to separate the roles of the various stakeholders involved, eg. Service Providers, Network Providers and Content Providers
- interoperable (seamless transition between different networks (at the physical layer and above) and different services - internationally)
- future-proof (the easy incorporation of new services and network technologies)

Further requirements are given by the specific actors. For example:
End-users require:

- ubiquitous mobile access
- reliability from the network
- simplicity from the services
- security from both the services and the network
- negotiable QoS
- a single point of contact for billing

Service Providers require:

- a fast, open, service creation platform

[1] www.ngni.org

W. Burakowski et al. (Eds.): Art-QoS 2003, LNCS 2698, pp. 1–17, 2003.

- the capability to inform end-users of the services that are available
- QoS guarantees from the network regarding availability, throughput, delay, delay variation, loss, and security
- the ability to adapt services according to the available network QoS or device type/capabilities

Network Providers require convergence (where realistic) in order to maximise efficiency and minimise costs.
Even from the short list of top-level requirements above, it can be immediately appreciated that the issue is complex:

- To allow new service developers to operate independently and efficiently, requires a Service Provider - Network Provider interface to be well-defined and agreed internationally.
- It is difficult to be future-proof, when the future services are unknown.
- As mobile terminals become more sophisticated, the (mobile) transitioning between networks owned by different providers mid-session (and the corresponding charging issue) has to be solved.
- Applications may demand certain levels of QoS from the network or, alternatively, may be capable of adapting to the prevailing conditions.
- QoS parameters have to be agreed between Network Providers (the 'inter-domain' problem).
- The same service used on a different terminal, or transmitted over a different access network, will require different QoS values.

QoS on IP

Convergence in the network is essentially limited at the moment to the network layer protocol IP, and the acceptance that if users are going to demand increasingly multimedia applications, then the optical infrastructure will continue to extend outwards from the core network, via metro networks towards the access …. up to the point where the demand for mobility takes preference.

Legacy infrastructure, the expense of upgrading the local loop, niche markets for particular technologies, and even the impact of regulation (the threat of unbundling is deterring incumbent operators from upgrading the local loop) are also factors that have to be taken into account.

In most cases, these topics have been addressed by working groups within the NGN-I project.

This paper has 4 main purposes:

a) to identify what capabilities are expected of next generation of networks by the various players
b) to examine the key factors that determine in practice how networks evolve
c) to identify some evolutionary trends
d) to take a look into the future

The top-down assessment "a)" considers the general service trends and aims to capture the big picture, as opposed to a bottom-up approach, where an inventory of the technological details, ingredients and necessary components is made and systematically analysed (Figure 1).

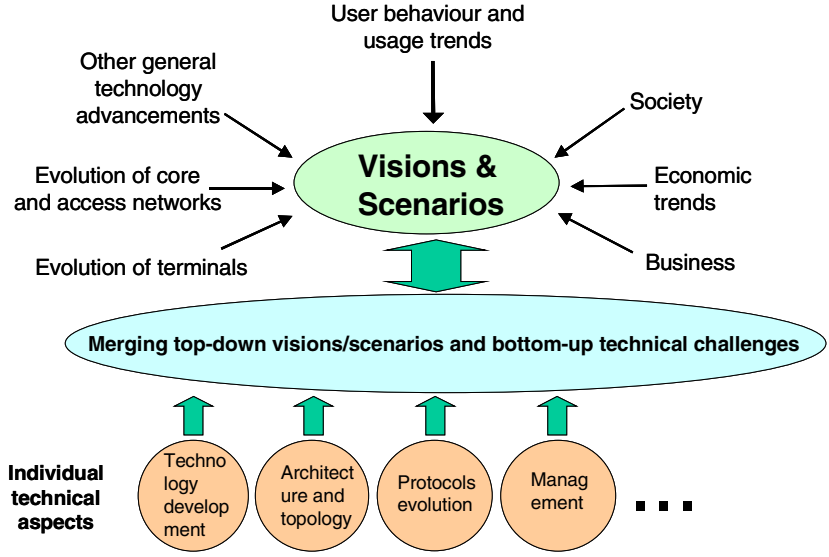


Fig. 1. Mapping of the top-down and bottom-up approaches

2 The User Requirements for NGNs

The definition of "users" differs depending upon whether the issue is being viewed from the perspective of the vendor, operator, network provider, or service provider. In many instances, the user of one service can also be a provider of another service. For example, to equipment vendors, the users are the operators and network providers, but from a service provider perspective, the users are ISPs, corporate users and residential end users. For ISPs, the users are corporate users and residential end users. Depending upon the user segment addressed, user expectations and requirements of NGNs vary significantly.

The "User Behaviour and Usage Trends" part of the NGN-I Roadmap considers the types of functionality that future users will demand. These generally have implications throughout all parts of the network. This part of the paper therefore represents the "top-down" view.

Significant increases in Internet traffic are expected, as connectivity in countries such as China and India becomes more widespread. This increase in volume is being accompanied by a fundamental change in the nature of telecommunications traffic generally. The almost exclusively circuit switched voice traffic became first mixed with - and is now heavily dominated by - packet based data traffic.

End users will also tend to generate more content themselves at the edge of the network. Consumer devices capable of producing large amount of digital content, such as digital photo and video cameras appear on the market at increasingly affordable prices. End users are expected to share the wealth of information and content they create with their peers. This will further change both the nature of the traffic, which is currently dominated by web browsing, and also the currently predominantly passive consumption of centralised content to a more balanced passive consumption and content creation.

Another trend influencing traffic evolution is online entertainment and online gaming. Interactive virtual reality games can boost network traffic significantly. They also present more stringent requirements towards the networks not only in terms of bandwidth, but also in terms of transmission delay.

Obviously, these trends, which rely on the buying power of the consumer and the quick adoption of new technologies are heavily influenced by the prosperity of the world economy. These technologies will almost certainly also have long reaching social impacts that are often under-estimated and not sufficiently understood.

Finally, an example for unexpected, so-called disruptive events that can have long reaching impact is the tragic terrorist attack on 11 September 2001. As a result of that, there has been a considerable increase in public security and surveillance increasing the volume of the traffic on the networks. In the course of 15-20 years, almost certainly there will be other disruptive events. Including their impact into the final picture is very difficult, because of their unexpected nature, but they will also shape the future in one way or another.

Whilst the precise successful services of the future are impossible to predict, surveys on user expectations from services have been conducted by many EU IST projects [2], consultancy companies and other organisations.

With a fairly high degree of confidence, we can therefore assume that most of the findings that are common to the surveys to be highly relevant. These indicate that users will require services to:

- be more sophisticated (service interoperability), yet simple to use
- be secure
- run on less obtrusive hardware
- be reliable, yet cost-effective
- work effectively when mobile
- offer more than just Internet access
- enable group communication
- be transportable between networks
- enable the same data to be available on different devices
- filter information
- exploit the existence of more embedded sensors
- exploit communication between embedded devices

[2] For example, TRUST, NOMAD, MOBIVAS, MOBIX, WISE, NGN-I (SMONET), SB3G cluster (ADAMAS, BRAHMS, BRAIN, DRIVE, EMBRACE, PRODEMIS, SUITED, VIRTUOUS, WIND-FLEX, WINE GLASS)

At a more specific level, (residential) end-users have indicated the desire for:

- quick communication via short messages to friends (point-point SMS -> multicasted, multimedia messages)
- faster access to more - and more sophisticated - information ….. and whilst they are on the move
- "personal agents" that are able to find the best services, and filter the deluge of unsolicited e-mail
- when travelling: language translation, maps, yellow pages, location based information of the city
- combined PDA and phone services
- sending current position information
- knowing the quality of a service before paying
- multimedia communications services for entertainment (gaming). For example, the construction, organisation and governance of a virtual parallel world

The business community will require more flexibility (but with security) to access company resources from outside the office, and to communicate with colleagues, via a variety of different devices (fixed, wireless and mobile). They need to instantly send and receive e-mail, retrieve information stored on corporate networks, or participate in Net meetings - regardless of location. And if the mobile worker moves during a session forcing a change in network, the application session should be maintained, so that work can continue uninterrupted without any reconfiguration, rebooting, or restarting of the programs in use. The amounts of data to be transferred will be more than today, but generally not to the level that will be required by residential users for gaming and other forms of entertainment.

However, specialised businesses (eg. space, meteorology, oil exploration, environmental, medical, TV) will continue to push the network capacity to the limit.

With a particular focus on making sophisticated services simple to use, the IST Advisory Group (ISTAG) identified 4 scenarios for daily life and work, projected for the year 2010, that highlighted a trend towards so-called "Ambient Intelligence".

The concept of Ambient Intelligence is one of greater user-friendliness, more efficient services support, user-empowerment, and interfaces that are embedded in all kinds of objects (clothing, buildings, vehicles, packaging, products, bodies, …). Ambient Intelligence works in a seamless, unobtrusive and often invisible way. This vision is one of people benefiting from services and applications whilst supported by new technologies in the background and intelligent user interfaces.

The main trends for users requirements for future services can therefore be summarised as:

- more entertaining, visually attractive, life-like services
- more mobility (and flexibility to move between networks)
- more security
- more peer-to-peer - and group - interaction
- the exploitation of Ambient Intelligence (to both add value to services and hide the complexity from the user)

3 The Mobile Services Value Network

A strong growth area is expected to be in new mobile services. By 2010, world-wide revenues for 3G services are expected to exceed $300B annually[3]. There are different business models for selling mobile services, depending upon complex agreements between network providers, content developers/owners and aggregators.

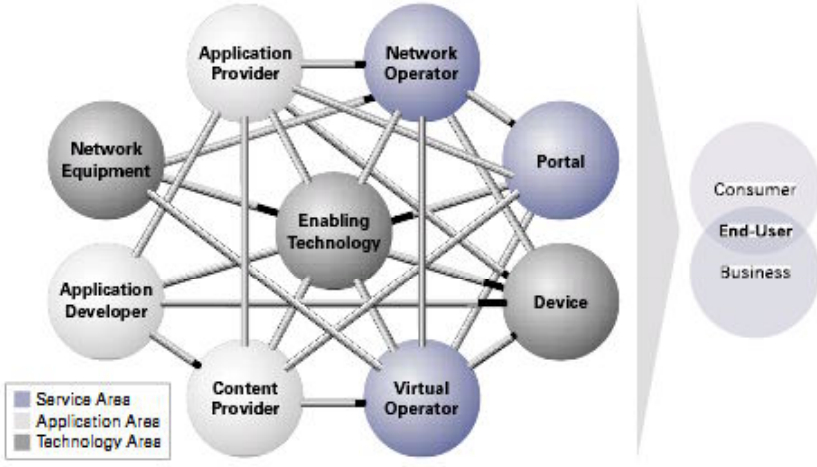


Fig. 2. A Value Network for Mobile Services

While mobile portals are often viewed as an extension of fixed Internet portals, mobile portals are faced with additional challenges of content optimisation due to small form factor devices, and the necessity of delivering that content to the mobile user–independent of the user's location and time.

Three main areas that the industry is working to resolve have been identified:

- Portal Application Development
- Mobile Terminals
- Portal Service Delivery Capabilities

The traditional Internet space is more concerned with the consistent packaging, delivery and presentation of content across mobile and fixed devices; while those from the mobile industry are more focused on resolving the technical issues to enable data transmission on historically voice-centric networks and devices. Both parties are now beginning to appreciate the need to work together to resolve the market and technical issues involved.

[3] UMTS Forum, Report 13, Table 10

4 The Evolution of the Network

The "Evolution of the Network" part of the Roadmap discusses the general trends taking place in the Core and Access networks, with special attention to the optical and mobile environments. Optical technologies play a major role in the Core and Metro areas, whereas the exploitation of the existing copper access, and new wireless technologies, are dominant in the access environment. Apart from the overall characteristics of these networks, aspects such as scalability and interoperability are addressed.

The following implications for the networks in the way that services will be delivered have been identified:

- increased use of Internet-based services
- increased emphasis on mobility and roaming
- increasing use of peer-to-peer applications
- needs for QoS, security, flexibility
- IPv6[4] and Post IP Protocols
- the support of services in ways that are cost effective and easy to use

Metro and Core Network Evolution

Optical Transport Networking (OTN) represents a natural next step in the evolution of transport networking. For evolutionary reasons, OTNs will follow many of the same high-level architectures as followed by SONET/SDH, ie. optical networks will remain connection-oriented, multiplexed networks. The major differences will derive from the form of multiplexing technology used: TDM for SONET/SDH vs. wavelength division for OTN. To satisfy the short-term need for capacity gain, the large-scale deployment of WDM point-to-point line systems will continue. As the number of wavelengths grows, and as the distance between nodes grows, there will be an increasing need to add or drop wavelengths at intermediate sites. Hence, flexible, reconfigurable Optical Add-Drop Multiplexers (OADMs), will become an integral part of WDM networks. As more wavelengths become deployed in carrier networks, there will be an increasing demand to manage capacity. In much the same way that digital cross-connects emerged to manage capacity into the electrical layer, Optical cross-connects (OXCs) will emerge to manage capacity at the optical layer.

Figure 3 depicts an OTN architecture covering the Core, Metro, and high-capacity Access domains. Initially the need for optical-layer bandwidth management was most acute in the Core environment, but increasingly the Access network at the client or server is becoming the bottleneck for data transfer. The logical mesh-based connectivity found in the core will be supported by means of physical topologies, including OADM-based shared protection-rings, and OXC-based mesh restoration architectures. As bandwidth requirements grow for the Metro and Access environments, OADMs will be used there too.

[4] IPv6 as a new networking protocol is addressed in most of the working groups, as it increasingly appears to offer benefits for the future networking convergence at the IP layer, across wired and wireless networks.

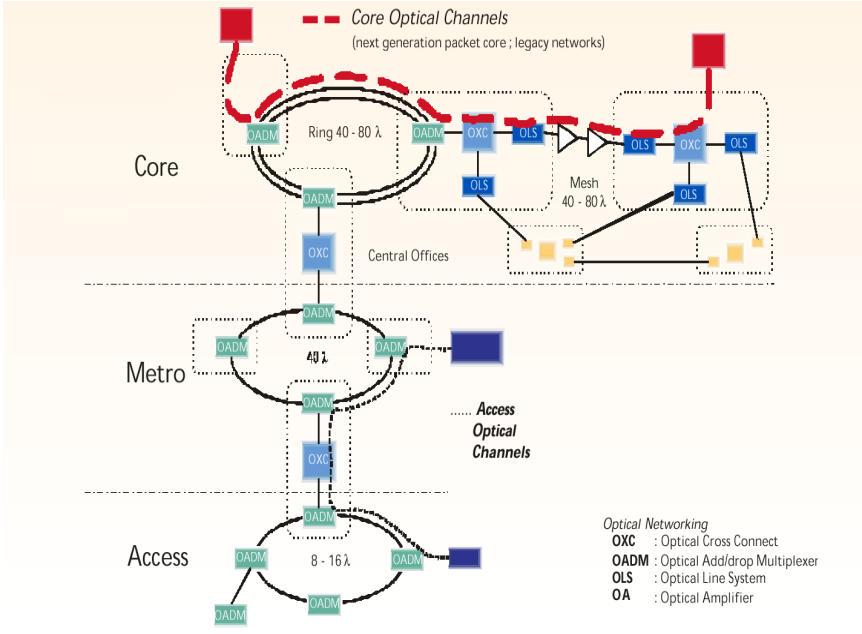


Fig. 3. Optical Transport Network Architecture

We expect the Core and Metro network to consist only of IP- and WDM-technologies. The architecture of the next generation network will take advantage of the provision of an integrated IP network layer directly on top of a WDM transport layer. The encapsulation of IP over WDM can be accomplished in different ways with simplified network stacks deploying protocols such as Packet over SONET/SDH, Gigabit Ethernet or Simple Data Link.

The basic guideline for the integrated IP/WDM architecture is that WDM is considered as a backbone technology and IP is interconnected to the WDM equipment at the edges of the Core network. Such a network is mainly considered by Internet Service Providers and in particular Competitive Operators, deploying optical infrastructure, leased or owned, willing to provide IP services on top of it using IP Points of Presence (PoPs).

The optical infrastructure will gradually evolve from ATM/SDH. Different topologies of WDM equipment may be deployed in the metropolitan and backbone areas. Incumbent operators could also deploy such a network, where in that case they integrate their existing ATM and SDH infrastructure with the DWDM equipment by using the WDM backbone or core to carry the ATM and SDH traffic.

Figure 4 depicts a future ISP's metropolitan network consisting of a WDM optical Metro core and IP Metro access. The IP section is composed of a number of IP PoPs, where customers can access the IP network services and traffic is groomed and forwarded to other PoPs or networks through the backbone. Access is facilitated to customers through the interconnection of the ISP's Provider Edge (PE) IP routers with the Customer Edge (CE) IP routers. Existing ATM and SDH equipment is shown for completeness. Provider equipment can be collocated or not with the customer

equipment, depending upon the distance between customer and provider premises and on the amount of traffic generated by the customer, and the tele-housing policies.

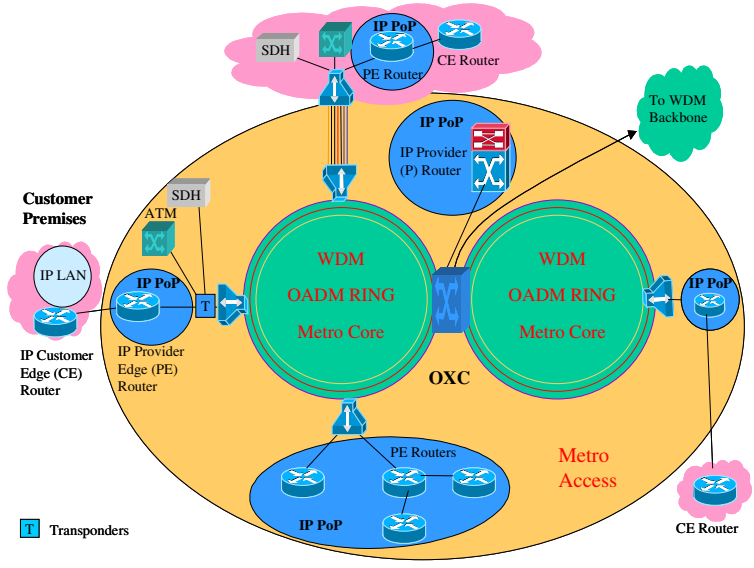


Fig. 4. Metropolitan Area IP over WDM Example

The optical WDM Metro core is usually composed of a ring of re-configurable OADMs, while additional point-to-point WDM links with Terminal Multiplexers can be considered for large customers. OADMs offer management interfaces so that they can be remotely re-configured to add and drop wavelengths (optical channels) to the ring through the tributary cards and multiplex them in the form of optical line signal in the corresponding line cards of the ring in each direction.

In the case where there are two WDM Metro core rings, then an optical cross-connect is needed, to route wavelengths from one ring to the other supporting all-optical networking. Such cross-connects are the most expensive pieces of optical networking equipment, capable of performing additional tasks, such as wavelength switching and conversion for hundreds of ports in an all-optical form without O-E conversion.

The metropolitan network should extend the transparency and the scalability of the LAN through to the optical core network. The IP Metro access is composed of a set of PE routers interconnected via optical interfaces with OADMs. At the access side of the metropolitan network, Fast Ethernet is becoming commonplace.

However, a more-compatible methodology would be the use of optical Ethernet (40-Gigabit speeds (SONET OC–768) have already been demonstrated). Network operators may limit their customers to a few Mbit/s, but the links are gigabit-capable; and someday the fees for gigabit-scale Ethernet services will be affordable. In the meantime, the protocols and techniques for bandwidth segregation over shared links exist, work well, and are used in thousands of sites. It is a simple step to run parallel optical Ethernet trunks, each on a separate wavelength, all multiplexed over a single fibre pair using DWDM technology. In this way, a point-to-point Ethernet link could

have scores of 10 Gbit/s channels, with an aggregate Ethernet bandwidth of perhaps 400 Gbit/s. Of course, this kind of network requires very large Ethernet switches at the ends of the fibres.

The limits on optical Ethernet bandwidth may be only the limit of fibre optic bandwidth (perhaps 25 Tbit/s per second for the available spectrum on today's fibre) which is still well beyond the capabilities of today's lasers and electronics. However, extrapolating from recent trends brings us to that level in only 5 or 10 years.

In the case that the router provides interfaces working in the 15xx nm range for transmission and reception, there is no need for a transponder in the OADM. The usual case, however, is that the routers' optical interfaces work at 1310 nm and there is a need to adapt this to the 15xx nm wavelength, which is done by the corresponding two-way transponder. The transponder converts the optical signal of 1310 nm to electrical and back to optical.

The Wide Area Network is usually composed of a partial mesh-type optical WDM network. Transmission rates of more than 10 Gbit/s per wavelength are providing access to terabits of bandwidth between metropolitan areas. The power budget is generally sufficient for distances up to 1000km without regeneration, reshaping and retiming. Optical Amplification is deployed either to boost the aggregate multiplexed optical line signal (eg. with an Erbium Doped Fibre Amplifier) or to separately regenerate each optical channel at the corresponding tributary.

Fixed Access Network Evolution

Broadband access needs are changing very rapidly. Content-intensive applications are driving up the need for speed. New peer-to-peer applications such as instant messaging with text, voice and - in the future - video will push the envelope even further since they require bi-directional data streaming.

Today, the copper-pair-based star structure, originally installed for telephony, is still dominant in most residential areas, with ADSL widely used to boost the throughput to some hundreds of kbit/s downstream, depending upon the distance from the exchange, and the quality of the copper pairs.

The CATV network is being upgraded to support 2-way digital transmission, and is capable of similar data flow rates upstream and downstream to ADSL. Being a shared medium, however, the instantaneous throughput experienced is dependent upon the number of simultaneous users and their usage pattern.

Fibre is penetrating into access areas, but the dream of fibre to the home (FTTH) or desktop has yet to materialise, mainly because of the cost-sensitive nature of this part of the network.

In the near future, residential access is expected to remain copper-based, using technologies such as xDSL to boost the capacity of traditional copper lines. However, for business offices, optical technology is already being used to bring high bandwidth to the end-user, with ATM and SDH access equipment at the customer premises. The next step is to use WDM technology for these environments. WDM will first be used in industrial and campus LAN environments. The DWDM network at the Microsoft headquarters in Redmond is a good example of a trial of these latest technologies, which use DWDM in the enterprise environment. This will become technically and economically feasible due to the very large number of wavelengths that a single fibre can carry, thus spreading the cost to more subscribers. Introducing more wavelengths

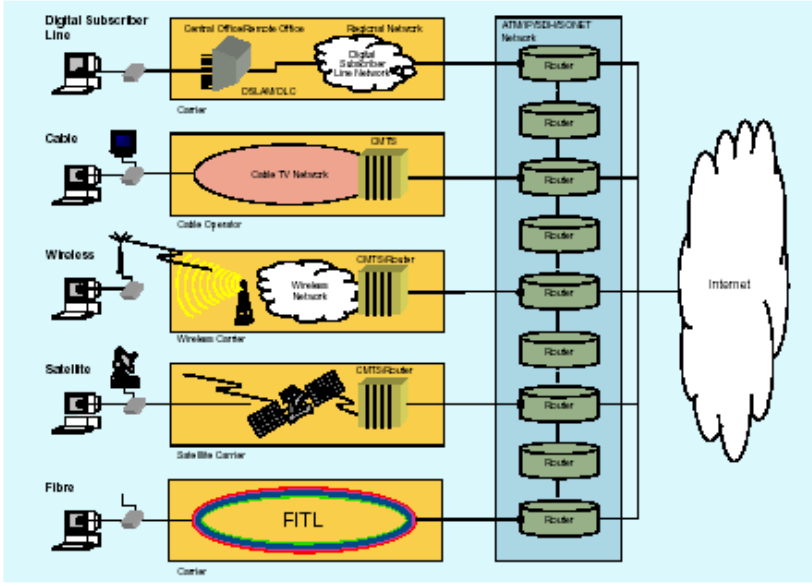


Fig. 5. Access Network Technologies

per fibre can also lead to new topologies for home access by using ring or bus like structures with an add/drop port per home so that each home has its own wavelength.

Nevertheless, the point-to-point optical fibre star structure is preferred for business customers with critical security requirements. In major cities, fibre already connects most big business offices (FTTO) and some residential buildings with ring or star structures. Fibre is getting closer to small business customers and residential customers with double star or tree-branch structures. Fibre to the cabinet (FTTCab) and fibre to the curb (FTTC) are becoming more common, also fibre to the town (FTTT) and fibre to the village (FTTV) are increasingly popular.

Wireless Access Network Evolution

The increase in mobile communications and user expectations for diversified wireless services has led to the development of a variety of wireless access systems. Considerable effort is underway to reconcile the different standards, typically by using multimode terminals and interworking devices. However, this approach does not seem to have all the ingredients to make the multiple existing and emerging mobile access technologies appear to the user as a single, seamless, and homogeneous network.

DVB-based[5] access networks are deployed through satellite transmission, for Digital TV broadcasting services. Interactive services are provided through the use of eg. telephone-lines for the (narrow-band) return channels. The technology has the main characteristic to be a broadcast and reliable (with very low error rate) link supporting around 1Gbit/s in total, and thereby able to transport hundreds of

[5] DVB, http://www.dvb.org/

compressed TV programs. In parallel, some data-based services can be carried, adding extra features around the TV programmes, such as electronic programme guides (EPGs) and encryption keys. For terrestrial transmission of digital TV, the DVB-T standard has been standardised and will be deployed in the near future progressively. Its purpose is the same, but the number of carried TV programmes will be limited to about 40.

Some wireless fixed broadband-access solutions have also been standardised, with relatively poor success. The local multipoint distribution service (LMDS) is being used for point-to-multipoint applications, like Internet access and telephony. It only has a 3-mile coverage radius, however. The multichannel multipoint distribution service (MMDS) was initially used to distribute cable television service. Currently it is being developed for residential Internet service. However, installations have not been profitable and service delays have been widespread. Currently, new standards are being defined: The IEEE 802.16 (WirelessMAN)[6] standard addresses metropolitan-area networks. The initial standard was approved in December 2001. The Working Group is currently developing amendment 802.16a to expand the scope to licensed and license-exempt bands from 2 to 11 GHz. ETSI is following a similar track for Europe.

Mobile-service networks offer access from mobile terminals when away, but it is expected that also at home these will play an important role. Various standards exist, and they vary over geographical locations. The first services to be offered were voice communication using a digital circuit-switched 10kbit/s connection (eg. GSM). Improvements come with the packet-switched connections (eg. GPRS) that will offer always-on connections. Third generation (3G) technologies (eg. UMTS, i-Mode) will provide multi-media applications (voice, data, audio and video) and nominal connection capabilities will be increased to around 64 kbit/s.

A possible way forward is the development of an open radio-access concept; that is, an access network which on one hand is based on a versatile air interface, and on the other hand is capable of satisfying different applications in different radio environments, when combined with IP-based backbone networks.

Besides flexibility in the air interface, such an open network paradigm requires a corresponding redefinition of layers above the physical one. In order to integrate heterogeneous mobile access networks, it is necessary to break the tie between mobile users and networks, and to move towards ways of operating that are:

- compatible with IP-based networks
- scalable; and
- distributed

The resource management should provide an independent performance calibration ("tuning knobs") allowing network operators to set target levels, tailored to user needs, on a unified IP-based access interface.

There will be a lot of different technologies and systems that will be used for the cellular communications. Therefore in the future, software radio solutions will be developed to enable dynamic reconfiguration (for all layers) and to offer a multifrequency and multimode system.

---

[6] IEEE 802.16, http://grouper.ieee.org/groups/802/16/

The IP protocol will be used by all types of terminals and by all networks. The 4G terminals will be a mobile and a wireless terminal with integrated Mobile IP and Cellular IP protocols.

Compared to wired networks, wireless networks have specific features such as loss of packets and bit rate modifications that have a significant impact on some applications requiring a constant QoS such as video. Adaptation of data transport to the constraints of wireless networks with techniques such as error resilience, scalability or joint source-channel coding is therefore critical.

Home Network Evolution

Homes contain many kinds of network technologies, for example:

- analogue/ISDN/ADSL/CATV/Ethernet/WLAN for communicative, interactive services
- CATV, satellite links, etc.. for entertainment services
- various low speed smart devices, interconnected and controlled by radio, fixed, infrared, … types of network

Interworking and interoperability, as well as the seamless provision of services, independent of the underlying networks is the most challenging topic yet to be addressed in the Access and Home Network environments. The standards arena of Home Networks is another area, which is currently too diversified and hence there is a number of proprietary technologies and interfaces. This is not a cost-effective solution.

In general, the home networking standards can be divided into two large groups: in-home networking standards, that provide interconnectivity of devices inside the home, and home-access network standards, that provide external access and services to the home via networks like cable TV, broadcast TV, phone net and satellite. Additionally, there are the mobile-service networks that provide access from mobile terminals when the user is away from home.

Many in-home networking standards require cabling between the devices. One option is to install new cabling in the form of galvanic twisted-pair or coaxial wires, or optical fibres. The alternative is to use existing cabling, such as power-lines and phone-lines.

Using existing cabling in the home is very convenient for end-users. For in-home networking via the phone-line, HomePNA[7] has become the de-facto standard, providing up to 10 Mbit/s (100 Mbit/s is expected). For power-line networking, low-bandwidth control using X10 [8], and (high) bandwidth data transfer using CEBus[9] and HomePlug[10] are the most prominent ones, offering from 10 kbit/s up to 14 Mbit/s.

New cabling requires an additional effort of installation, but has the advantage that premium-quality cabling can be chosen, dedicated to digital data-transport at high rates. The IEEE-1394a standard (also called Firewire and i.Link)[11] defines a serial bus

[7] HomePNA, http://www.homepna.org
[8] X10, http://www.x10.org
[9] CeBUS, http://www.cebus.org
[10] HomePlug, http://www.homeplug.org
[11] 1394 Trade Association, http://www.1394ta.org

that allows for data transfers up to 400 Mbit/s over a twisted-pair cable, and extension up to 3.2 Gbit/s using fibre is underway. Similarly, USB[12] defines a serial bus that allows for data transfers up to 480 Mbit/s over a twisted-pair cable, but using a master-slave protocol instead of the peer-to-peer protocol in IEEE-1394a. Both standards support hot plug-and-play and isochronous streaming, via centralised media access control, which are of significant importance for consumer-electronics applications. The disadvantage is that this sets a limit to the cable lengths between devices.

Another major player is the Ethernet, which has evolved via 10 Mbit/s Ethernet and 100 Mbit/s Fast Ethernet using twisted-pair cabling, into Gigabit Ethernet, providing 1 Gbit/s using fibre. Ethernet notably does not support isochronous streaming since it lacks centralised medium-access control. Also it does not support device discovery (plug-and-play). It is however widely used, also because of the low cost.

Currently there is no dominant wired networking standard for in the home, and networks are likely to be heterogeneous, incorporating multiple standards, both wired and wireless.

As opposed to wired networks, wireless systems are far easier to deploy. This is due to the smaller installation effort (no new wires), and due to a lower cost of the physical infrastructure, because the transmission medium is air. However, regulation by law and associated licensing fees may seriously affect the actual cost of the wireless connection. Additionally, the governmental regulations vary widely throughout the world. Especially for the license-free spectrum-bands, the issue of signal interference that limits usable bandwidth has to be solved.

For in-home networks, various technologies are deployed or being developed. Already widely deployed in Europe is the well-known DECT[13] technology, notably for voice communication. For services other than voice, new standards are emerging. With some overlap they can be divided into two categories: Wireless PANs and LANs.

Wireless personal-area networks (PANs) typically have a short range-of-use (10-100 meters), and are intended to set up connections between personal devices. They are close to the Short-Range Wireless concept. The most widely deployed standard in this class is Bluetooth . Its capability is providing 1 Mbit/s for few connected devices in a small network, called a piconet. Its range is between 10 and 100 meters depending on the transmission power. The used transmission-band for Bluetooth lies in the 2.4 GHz ISM band (license-fee).

The IEEE 802.15 standard is intended to go a step further. It integrates the Bluetooth standard and harmonizes it with the IEEE 802 family, such that it is IP and Ethernet compatible. The objectives are a high-bit rate solution (IEEE 802.15.3) providing up to 20 Mbit/s, and a low bit-rate one (IEEE 802.15.4, also known as ZigBee).

The HomeRF standard, like Bluetooth, also works in the 2.4 GHz ISM band. From an initial maximum data rate of 1.6 Mbit/s, it has been extended to 10 Mbit/s. HomeRF has a range of 50 meters at this speed. It is not interoperable with its strongest competitor, IEEE 802.11b, however.

---

[12] USB, http://www.usb.org
[13] DECT, http://www.dectweb.com/DECTForum

Wireless LANs have a broader application area: their purpose is to provide a wireless connection for networked devices like laptops or even handheld devices, not restricted to one person. The IEEE 802.11 series of standards are leading in this area. Two bands are being considered: The IEEE 802.11b (WiFi) standard uses the 2.4 GHz band, and the IEEE 802.11a standard the 5 GHz band. Notably the 802.11b standard is gaining market share. Capabilities of 802.11 are to provide up to 54 Mbit/s over 300 meters distance. The ETSI Hiperlan2 standard has now been merged with 802.11a, giving some features such as power control and QoS.

In conclusion the following challenges can be seen for home networks:

- to handle the heterogeneity, which requires bridging solutions, or a common network abstraction layer
- to support isochronous data-transfer and plug-and-play on top of Ethernet (and IP)
- to solve interference and regulation issues on power-line and phone-line networking
- to increase the bandwidth, to support future application needs
- to deal with governmental regulations that vary widely throughout the world, and prevent interference, especially in the license-free spectrum bands, to ensure optimal network performance
- to minimise the power consumption for mobile devices: since wireless networks enable mobile applications, their success relies on the duration and limited weight of the devices batteries. One of the requirements driving the development of Bluetooth was to have low-cost, low power consumption devices
- to enable the seamless integration of new devices. This involves interoperability for both low level protocols (plug-and-play devices) as well as higher-level functionality

5 Technology Development

The individual technical aspects have been covered in detail in specialized NGN-I working groups. Many of these working groups map onto the topics discussed in the proceeding chapters of this paper.

A summary of some of the key technologies for NGNs would include:

- middleware and distributed systems (to enable Service Provider - Network Provider separation)
- IP: IPv6, broadband, QoS, security, mobile and wireless
- multi-domain network management (for seamless roaming and QoS support)
- seamless interworking between core and access networks
- micro and opto-electronics
- trust and confidence enabling tools

- cross-media content
- multi-modal and adaptive interfaces
- multi-lingual dialogue mode
- embedded intelligence

6 Conclusion

Looking at the "big picture", global networks are becoming increasingly complex, in terms of the technologies, the interoperability (eg. fixed/wireless, but also with legacy networks), the services, and the management. Technical solutions offered by manufacturers can generally be relied upon to fulfil the purpose for which they were designed, but it becomes increasingly difficult to select the best solution for every situation, bearing in mind also evolutionary strategies and the regulatory environment. By pooling the knowledge and experiences from experts, network providers (for example) can better plan their deployment, and manufacturers can better understand the needs of the providers.

At the level of services, there is also much to clarify regarding the way in which new services can be introduced quickly and reliably.

At a more technical level, there is the topical issue of how to support end-to-end QoS across concatenated domains that recognise different QoS schemes. Other concerns, that are common to many players in the telecommunications marketplace are: broadband wireless technologies, fixed/wireless integration, optics, etc.

Many of these topics are interrelated, and the implications can only be fully appreciated when discussed in a multi-disciplinary group.

The NGN-I project had the mission to:

- gather together the (global) experts in the field of NGN, to: predict, plan for, and shape the future of networking
- provide the opportunity for discussions, in order to encourage the worldwide deployment of NGN. Topics included (but were not limited to):
  - emerging technologies
  - legacy networks
  - convergence
  - interoperability
  - regulation
  - standards
  - applications
  - services
  - business practices
- achieve consensus (where realistic). This can be beneficial in (for example) the area of standards, where competing products will lead to lower prices
- identify new products
- understand interoperability issues
- generally raise awareness of next generation networks

This paper has publicised several related aspects that have been collated into a Roadmap for NGN. These aspects include the "top-down" requirements from end-users and services, as well as the "bottom-up" capabilities of future technologies.

Acknowledgements. This paper is based heavily on the valued technical contributions from the NGN-I project members, who have been guided professionally by the leaders of the working groups. The inputs have then been edited by well-qualified representatives of the world communications industry, and the outputs have been co-ordinated by a core team of respected colleagues. The efforts were mainly funded by the Commission of the European Union (IST Programme) and the Swiss Bundesamt fuer Bildung und Wissenschaft. Without all this combined support this study and evaluation would not have been possible.

An IP QoS Architecture for 4G Networks

Janusz Gozdecki[1], Piotr Pacyna[1], Victor Marques[2], Rui L. Aguiar[3], Carlos Garcia[4], Jose Ignacio Moreno[4], Christophe Beaujean[5], Eric Melin[5], and Marco Liebsch[6]

[1] AGH University of Technology, Kraków, Poland
{pacyna,gozdecki}@kt.agh.edu.pl
[2] Portugal Telecom Inovação, 3810-106 Aveiro Portugal
victor-m-marques@ptinovacao.pt
[3] Instituto de Telecomunicações/Universidade de Aveiro, 3810 Aveiro, Portugal
ruilaa@det.ua.pt
[4] Universidad Carlos III de Madrid, Spain
{cgarcia,jmoreno}@it.uc3m.es
[5] Motorola Labs, Paris, France
Christophe.Beaujean@crm.mot.com, erik@motorola.com
[6] NEC Laboratories, Heidelberg, Germany
marco.liebsch@ccrle.nec.de

Abstract. This paper describes an architecture for differentiation of Quality of Service in heterogeneous wireless-wired networks. This architecture applies an "all-IP" paradigm, with embedded mobility of users. The architecture allows for multiple types of access networks, and enables user roaming between different operator domains. The architecture is able to provide quality of service per-user and per-service An integrated service and resource management approach is presented based on the cooperative association between Quality of Service Brokers and Authentication, Authorisation, Accounting and Charging systems. The different phases of QoS-operation are discussed. The overall QoS concepts are presented with some relevant enhancements that address specifically voice services. In particular, EF simulations results are discussed in this context.

1 Introduction

Availability of the network services anywhere, at anytime, can be one of the key factors that attract individuals and institutions to the new network infrastructures, stimulate the development of telecommunications, and propel economies. This bold idea has already made its way into the telecommunication community bringing new requirements for network design, and envisioning a change of the current model of providing services to customers. The emerging new communications paradigm assumes a user to be able to access services independently of her or his location, in an almost transparent way, with the terminal being able to pick the preferred access technology at current location (ad-hoc, wired, wireless LAN, or cellular), and move between technologies seamlessly i.e. without noticeable disruption.

Unified, secure, multi-service, and multiple-operator network architectures are now being developed in a context commonly referenced to as networks Beyond-3G or, alternatively, 4G networks [1]. The 4G concept supports the provisioning of multiple

W. Burakowski et al. (Eds.): Art-QoS 2003, LNCS 2698, pp. 18–28, 2003.

types of services, ranging from simple network access to complex multimedia virtual reality, including voice communication services, which are themselves a challenge in packet-based mobile communications environments.

Due to the heterogeneity of the access technologies, the Internet Protocol version 6 (IPv6) is being targeted as the common denominator across multiple access technologies, and make the solution basically independent of the underlying technology - and therefore future-proof. However, fitting such important concepts as support for Quality of Service (QoS), Authentication, Authorisation, Accounting and Charging (AAAC) and mobility into the native Internet architecture poses numerous difficulties and is a real challenge.

Therefore, the primary target of this paper is to present a solution for QoS support in mobile environments[1]. In order to do so, we make frequent references to the problem of integration of QoS, AAAC and mobility. In the course of the paper we discuss the methods that let us create and exploit the intrinsic associations between the service level agreements expressed in user profiles, and the network control mechanisms capable to monitor network usage per service and per user, in order to provide these services while the user moves and the terminal changes access technologies. The proposed architecture supports network services, in a secure and auditable way. Both user-to-network interfaces and inter-operator interfaces are defined, so that multiple service providers can interoperate. The architecture is able to support multimedia services, and has been further optimised for voice services. Voice services are now among the most demanding in terms of network design, imposing hard limits on network performance. In order to handle these services we will use the Expedited Forward (EF) concept of the differentiated services framework.

In the next section we briefly describe the network environment. Section 3 describes the overall QoS architecture, while section 4 details the signalling flow of end-to-end QoS support in the architecture and presents a simulation study that allows an optimised configuration of the access routers. Finally section 5 recaps our key conclusions.

2 An All-IP 4G Network Architecture

The overall 4G architecture discussed in this paper is IPv6-based, supporting seamless mobility between different access technologies. Mobility is a substantial problem in such environment, because inter-technology handovers have to be supported. In our case, we targeted Ethernet (802.3) for wired access; Wi-Fi (802.11b) for wireless LAN access; and W-CDMA - the radio interface of UMTS - for cellular access (Fig. 1). With this diversity, mobility cannot be simply handled by the lower layers, but needs to be implemented at the network layer. An "IPv6-based" mechanism has to be used for interworking, and no technology-internal mechanisms for handover, neither on the wireless LAN nor on other technology, can be used. So, in fact no mobility mechanisms are supported in the W-CDMA cells, but instead the same IP protocol supports the movement between cells. Similarly, the 802.11 nodes are only in BSS modes, and will not create an ESS: IPv6 mobility will handle handover between cells.

[1] The concepts that are presented in this paper have been developed and tested in controlled environments in the IST project Moby Dick [2] and are currently being refined.

The users/terminals may handover between any of these technologies without breaking their network connection, and sustaining voice connections. The users can further roam between administrative domains, being able to use their contracted services across domains if only appropriate agreements between those domains exist. The service providers are be able to keep track of the services being used by their costumers, both inside their own network, and while roaming. This is essential, e.g. for voice calls charging.

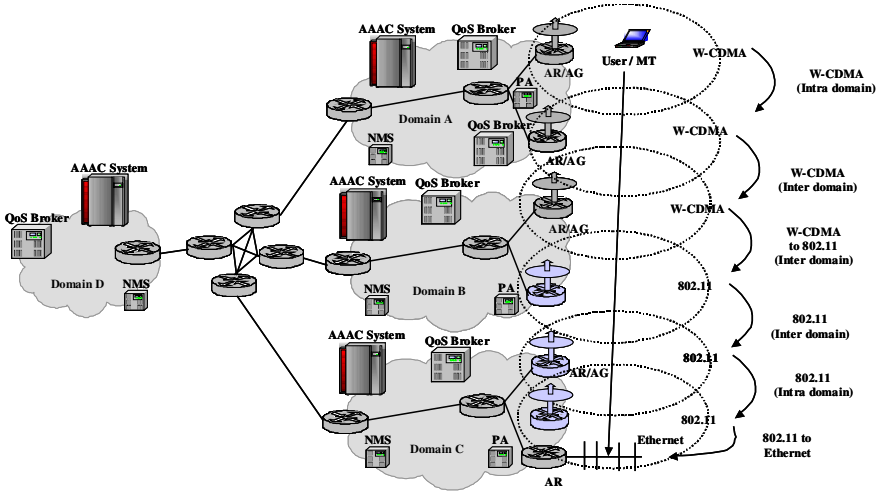


Fig. 1. General Network architecture

Figure 1 depicts the conceptual network architecture, illustrating some of the handover possibilities in such network with a moving user. Four administrative domains are shown in the figure with different types of access technologies. Each administrative domain is managed by an AAAC system. At least one network access control entity, the QoS Broker, is required per domain. Due to the requirements of full service control by the provider, all the handovers are explicitly handled by the management infrastructure through IP-based protocols, even when they are intra-technology, such as between two different Access Points in 802.11, or between two different Radio Network Controllers in WCDMA. All network resources are managed by the network provider, while the user only controls its local network, terminal, and applications.

Summarising figure 1, the key entities are:

- A user - a person or company with a service level agreement (SLA) contracted with an operator for a specific set of services. Our architecture is concerned with user mobility, meaning that access is granted to users, not to specific terminals.
- A MT (Mobile Terminal) - a terminal from where the user accesses services. Our network concept supports terminal portability, which means that a terminal may be shared among several users, although not at the same time.
- AR (Access Router) - the point of attachment to the network, which takes the name of RG (Radio Gateway) - for wireless access (WCDMA or 802.11).
- PA (Paging Agent) - entity responsible for locating the MT when it is in "idle mode" while there are packets to be delivered to it [4].

- QoS Broker - entity responsible of managing one or more ARs/AGs, controlling user access and access rights according to the information provided by the AAAC System.
- AAAC System - the Authentication, Authorization, Accounting and Charging System, responsible for service level management (including accounting and charging). In this paper, for simplicity, metering entities are considered an integral part of this AAAC system.
- NMS (Network Management System) - the entity responsible for managing and guaranteeing availability of resources in the Core Network, and overall network management and control.

This network is capable of supporting multiple functions:

- inter-operator information interchange for multiple-operator scenarios;
- confidentiality both of user traffic and of the network control information;
- mobility of users across multiple terminals;
- mobility of terminals across multiple technologies;
- QoS levels guaranties to traffic flows (aggregates), using, e.g. the EF Per Hop Behaviour (PHB);
- monitoring and measurement functions, to collect information about network and service usage;
- paging across multiple networks to ensure continuous accessibility of users.

Simple implementations of the above functions, including management aspects, have been done with the IPv6 protocol stack in Linux. The implementation relies on MIPL (Mobile IP for Linux). Other network and stack entities required for seamless operation of terminals in this heterogeneous environment have also been developed. QoS and AAAC sub-systems are responsible of serving a user according to his service contract. They operate at the network level and at the service level respectively, and employ a differentiated services approach for QoS. Fast MIP extension [3] and security (IPSec) have also been developed and integrated in the network.

3 Providing Quality of Service

The design principle for QoS architecture was to have a structure which allows for a potentially scalable system that can maintain contracted levels of QoS. Eventually, especially if able to provide an equivalent to the Universal Telephone Service, it could possibly replace today's telecommunications networks. Therefore, no specific network services should be presumed nor precluded, though the architecture should be optimised for a representative set of network services. Also, no special charging models should be imposed by the AAAC system, and the overall architecture must be able to support very restrictive network resource usage.

In terms of services, applications that use VoIP, video streaming, web, e-mail access and file transfer have completely different prerequisites, and the network should be able to differentiate their service. The scalability concerns favour a differentiated services (DiffServ) approach [5]. This approach is laid on the assumption to control the requests at the borders of the network, and that end-to-end QoS assurance is achieved by a concatenation of multiple managed entities. With such requirements, network resource control must be under the control of the network

service provider. It has to be able to control every resource, and to grant or deny user and service access. This requirement calls for flexible and robust explicit connections admission control (CAC) mechanisms at the network edge, able to take fast decisions on user requests.

3.1 Service and Network Management in Mobile Networks

Our approach for 4G networks and to service provisioning is based on the separation of service and network management entities. In our proposal we define a service layer, which has its own interoperation mechanisms across different administrative domains (and can be mapped to the service provider concept), and a network layer, which has its own interoperation mechanism between network domains. An administrative domain may be composed of one or more technology domains. Service definitions are handled inside administrative domains and service translation is done between administrative domains [6].

Each domain has an entity responsible for handling user service aspects (the AAAC system), and at least one entity handling the network resource management aspects at the access level (the QoS Broker). The AAAC system is the central point for Authentication, Authorization and Accounting. When a mobile user enters the network, the AAAC is supposed to authenticate him. Upon successful authentication, the AAAC sends to the QoS Broker the relevant QoS policy information based on the SLA of the user, derived from his profile. From then, it is assumed that the AAAC has delegated resource-related management tied to a particular user to the QoS Broker.

However, two different network types have to be considered in terms of QoS: the core and the access. In the differentiated services approach, the core is basically managed per aggregate based on the network services, and not by user services. In that sense, core management is decoupled from the access. We assume that the Core Network is managed as the ISPs manage it nowadays or with some new management techniques that might emerge in the future (e.g. aggregation techniques). As a result, the core will have installed the capabilities required to support a voice-call, e.g..

On the other hand, on the access network, the complexity of CAC can be very large, due to the potentially complex criteria and different policies. The QoS broker issues the commands to control both ARs and RGs, configuring e.g. an EF service. The QoS Broker is thus the entity that interfaces between the user-service level and the network-service level.

3.2 Implicit "Session" Signalling

In this architecture, each network service being offered in the network is associated to a different DSCP code. This way, every packet has the information needed to the network entities to correctly forward, account, and differentiate service delivered to different packets. After registering (with the AAAC system) a user application can “signal” the intention of using a service by sending packets marked with appropriate DSCP. These packets are sent in a regular way in wired access networks, or over a shared uplink channel used for signalling in W-CDMA. This way of requesting services corresponds to implicit signalling, user-dependent, as the QoS Broker will be aware of the semantics of each DSCP code per each user (although typically there

will be no variation on the meaning of DSCP codes between users). Thus QoS Broker has the relevant information for mapping user-service requests into network resources requirements and based on this information configures an access router.

A novel concept of "session" is implemented: the concept of a "session" is here associated with the usage of specific network resources, and not explicitly with specific traffic micro-flows. This process is further detailed in section 4.

Table 1. Example: Network Services

Service		Relative Priority	Service parameters	Typical Usage Description
Name	Class			
SIG	AF41	2a	Unspecified Signalling	(network usage)
S1	EF	1	Peak BW: 32 kbit/s	Real time services
S2	AF21	2b	CIR: 256 kbit/s	Priority (urgent) data transfer
S3	AF1*	2c	Three drop precedences (kbps): AF11 – 64 AF12 – 128 AF13 – 256	Olympic service (better then BE:streaming, ftp, etc)
S4	BE	3	Peak bit rate: 32 kbit/s	Best Effort (BE)
S5	BE	3	Peak bit rate: 64 kbit/s	Best effort
S6	BE	3	Peak bit rate: 256 kbit/s	Best effort
S7	Special Service Requesting AAAC Contact for specific network characteristics (DSCP, bw, etc)			

3.3 Network Services Offer

Services will be ofered a the network operator independently on the user applications, but will be flexible enough to support user applications

Each offered network service will be implemented with one of the three basic DiffServ per-hop behaviours (EF, AF, or BE), with associated bandwidth characteristics. Table 1 lists the network services used in the tests. The network services include support for voice communications (e.g. via S1) and data transfer services. Delay, delay jitter and packet loss rate are among the possible parameters to include in the future, but no specific control mechanisms for these parameters are currently used. The services may also be unidirectional or bi-directional. In fact, the QoS architecture can support any type of network service, where the only limit is the level of management complexity expressed in terms of complexity of interaction between the QoS Brokers, the AAAC systems and the AR that the network provider is willing to support.

Users will then subscribe to service level agreements consisting of different offerings. The operator may have a portfolio of packages composed by different criteria and targeting different groups of customers. An "Inexpensive service" can be supported at the network layer through S1, and S4 services (Table 1); and "Exclusive Pack", can be composed of S1, S2, S3, and S6. The technical translation of this "service pack" into network level services is the "network view of the user profile". (NVUP). The NVUP structure is not visible to the user, but impacts the way the services will be provided to the user, by the network.

4 End-to-End QoS Support

Given the concepts described in section 3, the entities developed in the project can support end-to-end QoS, without explicit reservations at the setup time. Three distinct situations arise in the QoS architecture: i) registration, when a user may only use network resources after authentication and authorization, ii) service authorisation, when the user has to be authorised to use specific services; and iii) handover - when there is a need to re-allocate resources from one AR to another.

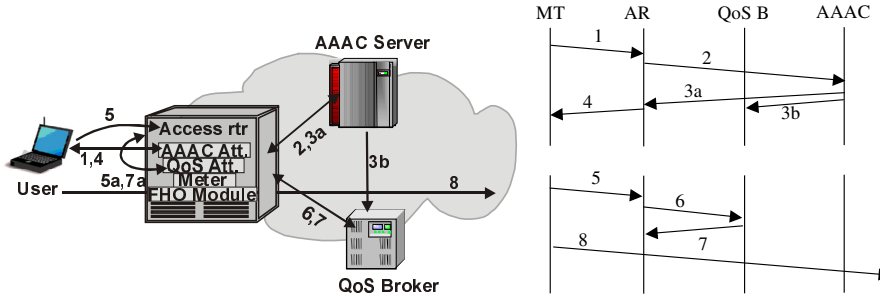


Fig. 2. Support for QoS - registration and service authorisation

4.1 Registration and Authorisation

The Registration process (see Fig. 2.) is initiated after a Care of Address (CoA) is acquired by the MT via stateless auto-configuration, avoiding Duplicate Address Detection (DAD) by using unique layer-2 identifiers [7] to create the Interface Identifier part of the IPv6 address. However, getting a CoA does not entitle the user to use resources, besides registration messages and emergency calls. The MT has to start the authentication process by exchanging the authentication information with the AAAC through the AR. Upon a successful authentication, the AAAC System will push the NVUP (network view of the User Profile) to both the QoS Broker and the MT, via the AR. Messages 1 to 4 on Fig. 2. detail this process.

The same picture shows how each network service is authorized (messages 5 to 8). The packets sent from the MT with a specific DSCP implicit signal the request of a particular service, such as a voice call (supported by network service S1, as in Table 1). If the requested service does not match any policy already set in the AR (that is, the user has not established a voice call before, e.g.), the QoS attendant/manager at the AR interacts with the QoS Broker that analyses the request and authorises the service or not, based on the User NVUP (Network View of the User Profile) and on the availability of resources. This authorisation corresponds to a configuration of the AR (via COPS [10]) with the appropriate policy for that user and that service (e.g. allowing the packets marked as “belonging” to voice call to go through, and configuring the proper scheduler parameters, as we will see in section 4.3). After that, packets with authorised profile will be let into the network and non-conformant packets will restart the authorization process once more, or will be discarded.

4.2 Handover with QoS Guarantees

One of the difficult problems of IP mobility is assuring a constant level of QoS. User mobility is assured in our network by means of fast handover techniques in conjunction with context transfer between network elements (ARs - old and new - and QoS Brokers).

When the quality of the radio signal in the MT to the current AR (called "old AR", AR1) drops, the terminal will start a handover procedure to a neighbouring AR (called "new AR", AR2) with better signal and from which it has received a beacon signal with the network prefix advertisement. This handover has to be completed without user perception, when making a voice call, e.g.. For achieving this, the MT will build its new care-of-address and will start the handover negotiation through the current AR, while still maintaining its current traffic. This AR will forward the handover request to both the new AR and to the QoS Broker. The two QoS Brokers (old and new) exchange context transfer information relative to the user's NVUP and the set of services currently in use by the MT. The new QoS Broker will use this information to verify the resources availability at the new AR and, in a positive case, configures the new AR to accept the handover. The MT is then informed that the necessary resources are available at the new AR and may then perform the Layer 2 handover. During this last phase, both ARs are bicasting, to minimize packet loss. The detailed messaging is presented in the next figure.

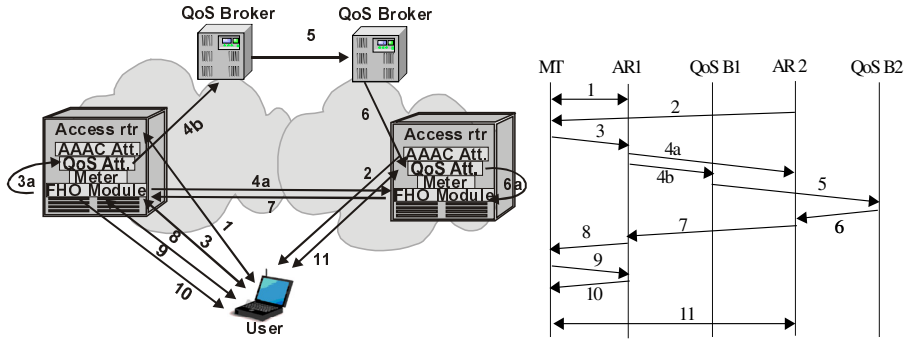


Fig. 3. End-to-End QoS Support – Handover with QoS

4.3 EF PHB Resource Provisioning

Building an all-IP architecture based on a Differentiated Services introduces a problem of how to create per-domain services for transport of traffic aggregates with a given QoS. Per-domain services support data exchange by mixing traffic of different applications, therefore different aggregates are required to support delay-sensitive traffic, delay tolerant traffic, inelastic, elastic, as well as network maintenance traffic (e.g. SNMP, DNS, COPS, AAAC etc.).

As applications generate traffic of different characteristics in terms of data rates, level of burstiness, packet size distribution and because the operator needs to protect the infrastructure against congestion, it is very important that aggregate scheduling will be accompanied by:

- per-user rate limitation performed in the ingress routers (ARs) based on user profile,
- dimensioning and configuration of network resources to allow for a wide range of user needs and services,
- resource management for edge-to-edge QoS.

Deterministic control of the edge-to-edge delay for delay-sensitive applications can be based on mathematical formulation of a delay bound for aggregate scheduling proposed in [8] and [9] – and this is the case of voice calls, usually considered the most delay-sensitive user application. The deterministic upper-bounding of edge-to-edge delay is possible, provided that the resource utilization factor for all links in a DiffServ domain is controlled and does not exceed pre-calculated values. Based on [9] one can calculate utilization factors that will not be overdraft, and thus guarantee that the target delay will not be exceeded.

To maintain resource utilization in the entire domain, the QoS Broker is expected to know the demand, current utilisation factors of all links based on incoming calls parameters or on measurements, and on additional information such as traffic load matrix. The real data traffic is provided by monitoring functions in the network, while traffic matrixes are induced on historical profiling (and with varying degrees of complexity). The QoS Broker will then use this knowledge for admission control and resource provisioning. The mathematical formulations have the disadvantage of relying on the worst-case scenario, which leads to substantial over dimensioning.

For defining simpler heuristics for application in the QoS Broker we conducted simulations in order to get insight into the delay issue, and how it varied in function of different schedulers, and respective parameters. We evaluated a set of per-hop and per-domain behaviours supporting the typical services defined in Moby Dick [11]. The following figures show the comparison of Strict Priority (PRI), Strict Priority with rate limitation (PRIs), Weighted Fair Queuing (WFQ) and Stochastic Fair Queuing (SFQ) scheduling algorithms that were considered to serve typical traffic classes. Figures 4 and 5 present edge-to-edge average and maximum queuing delays as a function of number of hops.

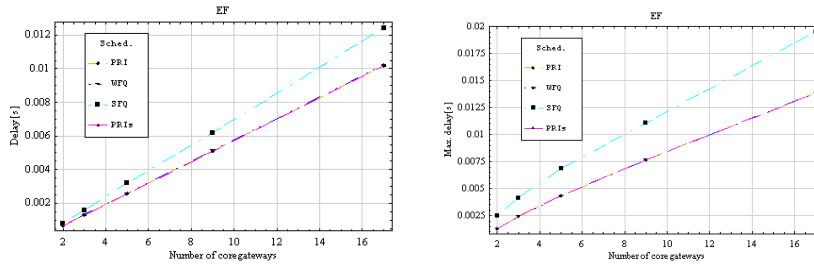


Fig. 4. Edge-to-edge average queuing delay and 99,5 percentile of maximum queuing delays of VoIP for EF aggregate (EF load=12% of link capacity, link load: 100%, link rate: 10 Mbit/s)

The basic evaluation criteria was the queuing delay and the delay jitter of EF PDB for flow S1. The SFQ algorithm exhibits the worst performance of all schedulers, especially for medium and high traffic loads on a link. A better performance exhibits the SFQ algorithm at a very low load, but it applies to average delays only. PRI, PRIs and WFQ algorithms produce comparable results. For the Moby Dick architecture we

are now considering to recommend PRIs, due to its simplicity when compared to WFQ.

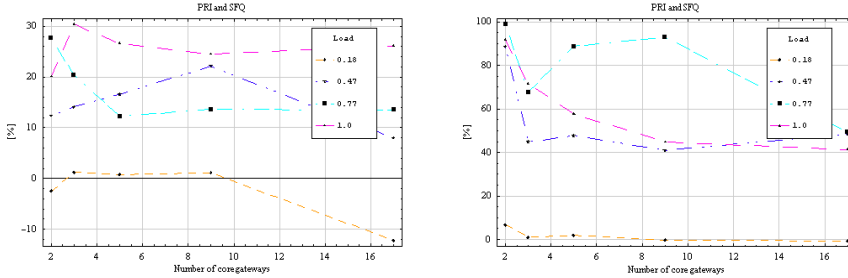


Fig. 5. Left: comparison of average delay for PRI and SFQ schedulers, Right: comparison of 99.5% quantile of delay for PRI and SFQ schedulers

Notice that in the simulations, a Time Sliding Window (TSW) algorithm was used for rate limitation. The main role of rate limitation was to prevent lower priority classes from being affected by higher priority classes, and is applied to all PHBs in each node. Here, we draw attention to the fact that it is very important to protect a class carrying network maintenance traffic (SIG traffic, Table 1), because this traffic plays an important role in maintaining the network infrastructure, but does not have the highest priority. When the traffic does not exceed the configured rate, the performance of PRI and PRIs is the same since the TSW does not affect traffic characteristics. The SP algorithm also fits into the Moby Dick concept of service classes.

The PRIs limitation has yet another advantage – rate limitation does not have influence on traffic characteristics when traffic level remains within limits, and the limits can be dynamically changed without inducing abrupt delay shift. For WFQ and SFQ algorithms dynamic change of bandwidth assigned for service class changes the service rate for this class, and can cause a transient increase of delay jitter. Thus this seems to be the preferable approach to support real-time services, such as voice calls.

5 Conclusion

We presented an architecture for supporting end-to-end QoS. This QoS architecture is able to support multi-service, multi-operator environments, handling complex multimedia services, with per user and per service differentiation, and integrating mobility and AAAC aspects. The main elements in our architecture are the MT, the AR and the QoS Brokers. We discussed the simple interoperation between these elements and depicted the overall QoS concept. With our approach, very little restrictions are imposed on the service offering. This architecture is currently being evolved for large testing in field trials across Madrid and Stuttgart.

Being an architecture specially targeted to support real time communications over packet networks, the network elements configuration must be well dissected. The simulation study summarized in the paper was a valuable input to the QoS Broker implementation and policies design, providing simple heuristics to properly configure

the access routers to achieve the best possible performance. The schedulers configuration on the core routers was also determined through the results of this simulation study.

This architecture still has some shortcomings, though, mostly due to its diffserv-orientation. Each domain has to implement its own plan for mapping between network service and a DSCP, and thus, for inter domain service provision, it is essential a service/DSCP mapping between neighbouring domains. Furthermore, an adequate middleware function is required in the MT, to optimally mark the packets generated by the applications and issue the proper service requests, which requires extensions in current protocol stacks.

Nevertheless, our proposal facilitates the deployment of multiple service provision models, as it decouples the notion of service (associated with the user contract) from the network management tasks. It seems to provide a simple, flexible, QoS architecture able to support multimedia service provision for future 4G networks.

Acknowledgements. We would like to acknowledge the European Commission support and all partners in the Moby Dick consortium [12]. The support of everyone involved in the Madrid trials is also acknowledged.

References

1. Pereira, J.M et al.: Fourth Generation: Now it is Personal! Proceedings of PIMRC 2000, London, Sep 2000
2. Einsiedler, H. et al.: The Moby Dick Project: A Mobile Heterogeneous All-IP Architecture. ATAMS 2001, Krakow, Poland, (http://www.ist-mobydick.org)
3. Dommety, G. (ed.): Fast Handovers in Mobile IPv6. Internet Draft, work in progress, <draft-ietf-mobileip-fast-mipv6-3.txt>, July 2001
4. Liebsch, M. et al: Paging Concept for IP based Networks. Internet Draft, draft-renker-paging-ipv6-01.txt, September 2001
5. Black, D., Blake, S., Carlson, M., Davies, E., Wang, Z., Weiss, W.: An Architecture for Differentiated Services. IETF RFC 2475, December 1998
6. Thi Mai Trang Nguyen et al: COPS-SLS: a Service Level Negotiation Protocol for the Internet. IEEE Communications Magazine , Vol. 40 No. 5 , May 2002 , pp. 158–165
7. Bagnulo, M., et al: Random generation of interface identifiers. Internet Draft, draft-soto-mobileip-random-iids-00.txt, January 2002
8. Charny, A., Le Boudec, J.-Y.: Delay Bounds in a Network with Aggregate Scheduling. Proceedings of QOFIS, Berlin, October 2000
9. Yuming Jiang: Delay Bounds for a Network of Guaranteed Rate Servers with FIFO Aggregation. Proceedings of ICCC 2002, New York, May 2002
10. Durham, D. et al. The COPS (Common Open Policy Protocol). Internet Engineering Task Force. RFC 2748, Jan 2000
11. Marques V., Aguiar R., Pacyna P., Gozdecki J., Beaujean Ch., Chaher N., García C., Moreno J.I., Einsiedler H.: An architecture supporting end-to-end QoS with user mobility for systems beyond 3rd generation. IST Mobile & Wireless Telecommunications Summit 2002, Thessaloniki, Greece, June 17-19, 2002, pp. 858–862
12. http://www.ist-mobydick.org

Integration of Mobility-, QoS-, and CAC-Management for Adaptive Mobile Applications

Daniel Prokopp[1], Michael Matthes[1], Oswald Drobnik[1], and Udo Krieger[1,2]

[1] Telematics & Distributed Systems, Institute of Computer Science,
J.W. Goethe-University, D-60054 Frankfurt, Germany
{prokopp,matthes,drobnik}@tm.informatik.uni-frankfurt.de
[2] T-Systems, Technologiezentrum, Am Kavalleriesand 3, D-64295 Darmstadt
udo.krieger@ieee.org

Abstract. We sketch a QoS framework for adaptive mobile multi-media environments that integrates mobility- and quality-of-service management with new connection admission control (CAC) schemes. Refering to measurement-based CAC procedures applied in mobile access networks, we discuss their extension to adaptive mobile applications and describe the integration of corresponding resource reservation and CAC policies in our framework. Furthermore, we evaluate the performance of several CAC schemes by a simulation study.

1 Introduction

Currently, the integration of 3rd generation (3G) wireless networks into high-speed next generation networks (NGNs) based on the IP technology and its corresponding resource reservation and QoS-management mechanisms are some of the most challenging tasks of network design and engineering. These 3G networks like UMTS or IEEE802.11a/b compatible wireless LANs (WLANs) are realized on top of new WCDMA or DSSS transmission technology. The fast convergence of these wireless and wired IP networks coincides with the planned deployment of new integrated location-aware multi-media and interactive Web services. The latter are independent of the realized mobility patterns of the terminals and users and derived from a unified personal mobility concept. At present, there are different proposals to realize such new mobile quality-of-service (QoS) architectures which integrate both wired core networks and wireless segments at the edges. In the IST-project Moby Dick, for instance, a Diffserv approach is extended to the wireless segments. Recently, an alternative methodology has been discussed that is derived from the Intserv service model and its resource management and signaling protocol RSVP (cf. [10], [11]). Since nowadays the closely related MPLS paradigm with RSVP-signaling and tunneling components is used in the core network, we follow the corresponding resource-management concept derived from a mobile RSVP (MRSVP) approach (cf. [12], [20]).

Using the currently available inexpensive IEEE802.11b compatible radio access network (RAN) technology and a Mobile IP based data communication,

W. Burakowski et al. (Eds.): Art-QoS 2003, LNCS 2698, pp. 29–48, 2003.

we have implemented a new integrated mobility and quality-of-service management concept for mobile adaptive multi-media environments. Refering to measurement-based connection admission control (CAC) schemes applied in radio access networks, we sketch their extension to adaptive mobile applications and the integration of the corresponding resource reservation and CAC policies into our developed framework. Furthermore, we describe several CAC schemes and evaluate their performance by a simulation study. The paper is organized as follows. In Section 2 we discuss mobility and QoS management as well as measurement-based CAC in mobile networks. Section 3 is devoted to the integration of micro-mobility, resource management and admission control policies. In Section 4 we present the results on the performance comparison of the integrated resource reservation and adapted CAC strategies. Finally, some findings are summarized.

2 Mobility- and QoS-Management in Mobile Networks

2.1 Mobility Management in IP Networks

Considering multi-media and advanced Web communication of mobile hosts (MHs), we can distinguish two different mobility scenarios:

- *Macro-mobility*: MHs move among different address domains of IP subnetworks with changes of the address space.
- *Micro-mobility*: MHs move within the same address domain.

Looking at macro-mobility issues, changing the access point (AP) or base station (BS) of the radio access network (RAN) may result in a new RAN operator. This type of movement is called *inter-domain* mobility. Such a change of the RAN operator causes, in general, a change of the IP address of the moving MH, since different Internet (subnetwork) domains are normally managed in distinct RANs. To guarantee the connectivity of an MH, the latter is assigned a local address of the related domain. The corresponding mobility and address management functions are provided by enhanced variants of the IPv4 protocol such as Mobile IP (MIP) or by an integral part of IPv6 not studied here (cf. [16]).

Normally, an MH moving within its home network is registered at a home agent with its permanent address called *home address*. Considering the delivery of datagrams to an MH moving outside its home network, apart from its fixed home address a second temporary local address, called *care-of-address*, is additionally assigned to the MH by the Mobile IP protocol after its handover to a new domain. It is provided by a corresponding mobility management agent, called foreign agent, and registered in its data base together with the other routing information. Then it is used to forward the incoming datagrams to the corresponding MH residing outside of its home network (cf. Fig. 1). Mobile applications identify the mobile clients by the home address. During the movement of an MH address translation is provided in a transparent manner by changing the care-of-address and its registration using a communication over tunnels between

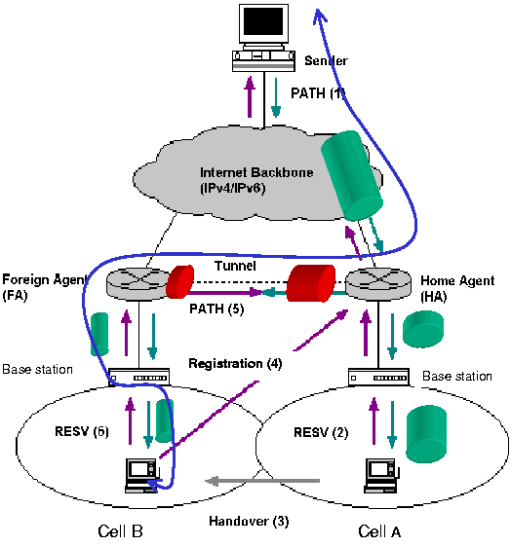


Fig. 1. Roaming and packet forwarding with Mobile IP

the involved home and foreign agents (cf. Fig. 1). However, a large overhead is required to guarantee a transparent communication among the peer instances resulting in a considerable performance degradation during the handover process. Therefore, specialized procedures have been developed to improve the latter processes by effective detection and realization of fast handoff control with QoS and security support (cf. [19], see Fig. 1).

Micro-mobility captures a scenario in a micro-cellular network where an MH detaches from its current AP, is handed over to a new radio cell and attaches to a new AP while residing within the management domain of the same RAN operator (cf. [17,18,19]). This scenario is called *intra-domain* mobility. The MH keeps its IP address assigned after a handoff in the domain of a RAN. The connectivity of the MH is guaranteed within a domain applying specific routing mechanisms (cf. [17,18,19]). Since the mobility management and corresponding registration procedures are performed within the same domain, the required location updates and signaling of address binding information of an MH can be efficiently realized compared to the inter-domain mechanisms used by Mobile IP.

2.2 QoS Management in Mobile Networks

IP communication in wireless segments of advanced mobile networks like the RAN of an IEEE802.11a WLAN with 54 Mbps is substantially different from current inter-networking in the high-speed backbone. Normally, the protocol functions of the error control and MAC sub-layers of the data link layer (DLL)

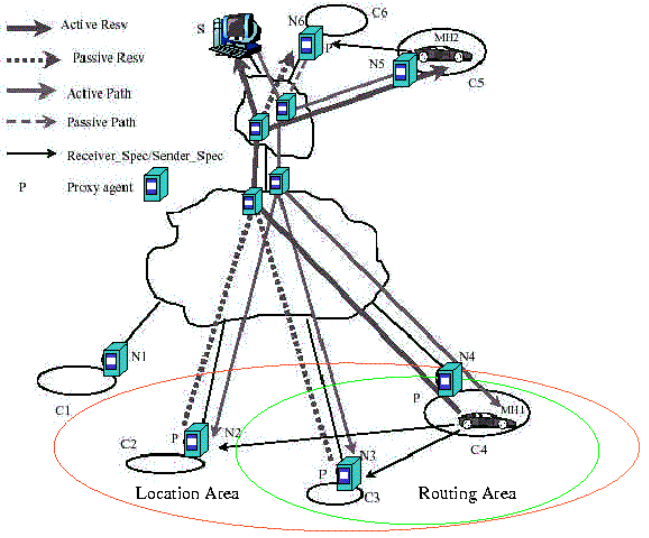


Fig. 2. Active and passive reservation of resources in the RAN

are decoupled of those procedures invoked at upper layers during data transfer. Moreover, distributed environments of contemporary IP networks do not offer any integrated controls that support the information exchange between the network layer and its lower layers. The latter, however, is a common approach during handoffs in 2+ and 3rd generation networks such as GPRS and UMTS where the radio resource management performs corresponding tasks.

To provide an efficient solution of micro-mobility management issues, optimized handoff and QoS management procedures are required (cf. [2,12]). It is the objective to minimize the disruption phase of connectivity while an MH detaches from its current AP and registers at the mobility agents after attaching to a new AP. In an advanced mobile network the corresponding handoff and registration processes affect several layers of the protocol stack (cf. Fig. 3). First, the movement detection functions at the MAC layer of the DLL are invoked to determine the required change of the AP and to identify the destination of a handoff. To maintain the connection and the effective flow of datagrams, registration and control instances at the upper layers are involved. For instance, by protocols like Hawai, cellular IP or hierarchical Mobile IP in a way similar to Mobile IP appropriate IP address and routing information is generated and routing agents are notified to guarantee a seamless flow of data after completion of a handoff (cf. [17,18,19]). Furthermore, the resources of the transferred connections required in the new cell have to be re-negotiated and the flow rates have to be adjusted. Then the associated mobile applications can adapt to the resource utilization within the new cell (cf. [2,12], see Fig. 3).

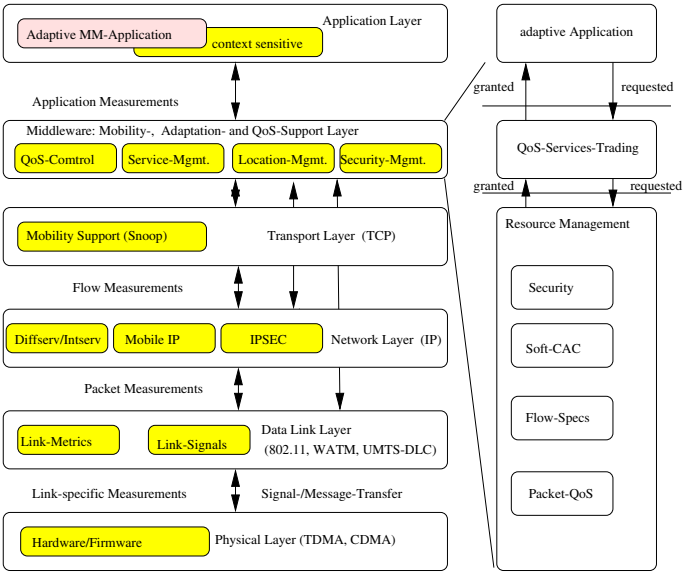


Fig. 3. Protocol stack of an NGN

After a handoff among radio cells the following situations can arise that result in quite different effects on the resource allocation and seizure by active connections of adaptive mobile applications:

- In the visited cell more resources are available than in the left. Hence, the MH can seize this additional capacity, distribute it among the active connections and trigger the invoked applications to increase their sending rates accordingly.
- In the new cell less resources are available than before. In this case, the MH has to reduce its consumption and to notify the active applications on the required adaptation processes.
- After a handoff no changes of the resource allocations are required.

In all cases an information exchange between the MH and a resource management component in the RAN about the required and granted resources and QoS objectives is necessary. Depending on the previous status of the resource allocation to the active connections of an MH corresponding adaptation processes have to be invoked at the transport and application layers (cf. [2,12], see Fig. 3).

2.3 Measurement-Based Adaptive Connection Admission Control

Considering mobile users with their mobile hosts (MHs) roaming within an advanced mobile network, the corresponding connections have different QoS requirements. The latter are characterized in terms of a quadruple: the minimal and maximal required bandwidth, a maximal end-to-end (E2E) delay and an

upper bound on the packet loss. Based on a chosen service model, normally we distinguish between those connections belonging to service class C1 with stringent QoS guarantees, i.e. guaranteed or premium service, a class C2 with soft requirements, i.e. predictive, assured or elastic service, and the best effort class C3 without any QoS requirements.

To maintain the QoS guarantees of active connections of roaming MHs during handover processes and to check whether new connections of fresh and handoff calls can be established without violating the guarantees of existing ones within a specific radio cell, a connection admission control (CAC) scheme constitutes a necessary ingredient of the resource management system (cf. [5,6,7,8,9,14,15, 21,22]). This decision process has to distinguish between both the connections of the sketched three service classes and the type of a call, i.e. whether it is a fresh call just triggering a new connection set-up in a cell or an already active connection requiring further connectivity in a neighbor cell after a handoff. Since the latter process may fail due to a lack of resources or a violation of existing QoS guarantees, a constraint on the corresponding dropping probability can be formulated as additional QoS criterion of a connection. For fresh calls the blocking probability of a connection is the basic QoS term apart from delay and loss constraints for real-time traffic of the premium class C1 and apart from the loss constraints for the elastic traffic of class C2. Furthermore, the utilization of the bandwidth offered by a cell constitutes the basic efficiency metric that has to be maximized by the network operator.

To achieve these performance objectives in the dynamic environment of a RAN, the corresponding CAC components comprise three building blocks: means to handle the traffic specification including the QoS requirements, a bandwidth reservation and admission control policy (ACP) and related decision algorithms as well as measurement functions to monitor the actual load and utilization of the resources managed by a specific resource management entity (RMC) of a RAN (cf. [21,22], see Fig. 4).

In the following we focus on the relationship of micro-mobility management and the corresponding CAC functionality. Regarding the QoS demands at the call level we restrict ourselves to the blocking and dropping probabilities in a single cell since the concatenation (i.e. convolution) of the latter performance indices provides a convenient tool to determine the E2E performance. Therefore, local and distributed control architectures are under consideration as ACP if we consider a target cell and its immediate neighbors.

The simplest ACP disregards the service classes and traffic types as well as the mobility (i.e. handoff) patterns and accepts fresh or handoff calls if there is enough capacity C available in a cell. For that purpose, a continuous monitoring of the load and utilization is required yielding measurement-based CAC schemes. Due to the random movement of the MHs dynamic ACPs are more appropriate policies. Since best-effort connections can be dropped without any constraints, it is sufficient to study the impact of micro-mobility on connections of the premium and elastic classes C1, C2. Since real-time traffic is transported in C1 and non-real-time in C2, we assume that only handoff calls of C1 specify

a maximal dropping probability $\mathbb{P}\{\text{drop}\}$. The latter has to be guaranteed by the ACP during handoff to the new cell whereas C2 handoff calls do not require such treatment due to the elasticity and delay-insensitivity of the corresponding sources. The actual realization of $\mathbb{P}\{\text{drop}\}$ is estimated by the ratio of rejected and the overall number of occuring handoff calls of the corresponding class.

To achieve the required protection of handoff calls compared to fresh calls, two elements of a protection strategy can be applied based on resource pooling and bandwidth reservation for handoff calls: reservation of a bandwidth portion in a specific handoff buffer and different threshold mechanisms (cf. [21,22]). They extend the concepts previously known as guarded channel schemes. Considering the simplest threshold schemes, bandwidth resources (BW) are reserved for calls potentially handed over to a cell. The reserved bandwidth is reflected by a threshold value. The latter is determined by the difference of the BW available in the cell and the required BW reserved for handoffs (i.e. a reservation index). A fresh call is accepted if the sum of the BW occupied by active calls and the applying fresh one does not exceed the threshold level, whereas a handoff call is allowed to exceed the threshold as long as enough capacity is available in the cell. In the case of a buffer scheme the sum of the BW occupied by active calls, the applying fresh one and the junk of bandwidth reserved for handoff calls is not allowed to exceed the cell capacity (cf. [5], see Fig. 2).

3 Integrating Micro-Mobility and Adaptive Connection Admission Controls

3.1 Integrated Mobility-, Resource-, and CAC-Management

In our approach we assume that mobility-aware multi-media and real-time applications are capable to adapt their required bandwidth by means of an advanced coding technology to a certain extent according to control signals emitted by data-stream and resource management components of our new QoS architecture (cf. [2,12]). To satisfy the diverse requirements of co-existing adaptive and standard applications, the developed programming model uses an abstraction with three layers: the standard and adaptive applications, the resource-management and QoS-adaptation level called QoS-services-trading, and the integrated connection and flow layer (see Figs. 3, 4). The latter is called network interface. It provides an abstraction of the transport and network layers and their corresponding signaling and data flows at the mobile nodes and the nodes of the access infrastructure such as the APs. Data flows of the applications are transported across these components. Moreover, the control functionality of these lower layers, e.g. reservation agents and bandwidth broker, traffic classifier, meter and shaper, packet scheduling and marking as well as buffer management, that is implemented in a distributed manner at the network elements to handle the QoS requirements at the connection, flow and packet levels can be adjusted by the network interface.

It is the goal of this mobile QoS (MQoS) management system to allocate, monitor, and manage the resources of the wireless network elements and to distribute their capacities in a fair manner among the competing applications and the associated data flows of their connections. The basic components performing these tasks are called data-stream management at the MNs and resource management (RMC) at the nodes of the access infrastructure (see Fig. 4). They handle the signaling of the requirements, the resource allocation, adaptation and management processes. To get the information required for the adaptation processes, they are supported by a monitoring component called environment monitor (EMC). It is an abstraction of those components already available in mobile terminals of second generation networks, e.g. the control processes monitoring the signal-strength, noise, BER and PDU-loss along the forward and backward paths of data flows over the air interface, and the movement-detection and location-identification processes. By these means the location of a terminal and the potential change of a micro-cell and its association to an AP due to a horizontal, or even vertical, handoff can be identified. Hence, the monitoring component can also provide control information and an interface for location-management and service-management components at higher layers below the applications if location-dependent services are supported (see Fig. 3).

Considering the actual actors at the distributed control instances residing at the session, flow and packet levels of the protocol stack of the MNs and the active nodes of the access network, appropriate control commandos and their parameters are passed by means of an intermediate mediator component called network-interface control. It translates the invocation messages of the management component into corresponding control actions that are transferred to the network-interface. Changes of the allocations, e.g. the addition of new bandwidth after the handoff of a flow to a new AP at the old interface and the reduction of the latter at the new interface among all competing flows due to the fair-share concept, are signalled by the data- and resource-management and translated into control actions of the network interface by the network-interface control.

Evaluating the experience gained in a new WLAN testbed (cf. [2,12,13]), we have realized that a re-negotiation of the resources after the completion of handoff procedures at the physical, MAC and IP layers consumes too much time. Therefore, a measurement-based CAC functionality is incorporated in our MQoS framework as connection access control (CON) component in the RAN (cf. Fig. 4). For this purpose, we have extended the existing ACPs to incorporate the adaptivity of advanced mobility-aware multi-media and real-time applications, e.g. voice services with AMR codecs and video applications with variable bit-rate codecs based on MPEG4 etc. This means that the corresponding handoff calls of C1 can adapt their sending rate or bandwidth requirements based on notification signals that are dynamically generated by the RMC and CON during a handoff process. The corresponding architecture is depicted in Fig. 4. This integration of micro-mobility management, bandwidth reservation and adaptive admission control proceeds as follows. In each cell a variable proportion of resources is assigned in a pro-active manner as bandwidth buffer to handle handoff calls arising

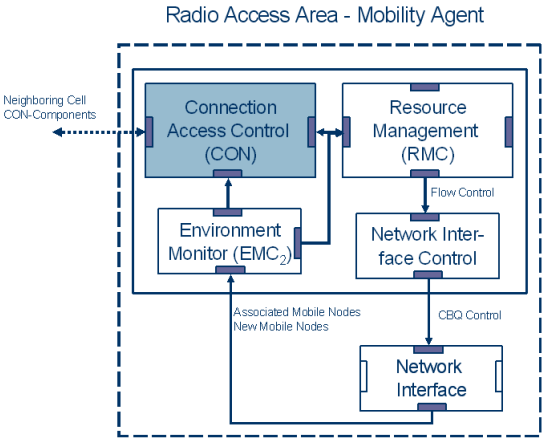


Fig. 4. Integration of mobility and resource management as well as connection admission control

from neighbor cells. The reservation processes are coordinated by the interworking of RMC, EMC and CON evaluating the micro-mobility patterns arising to and from as well as among neighbor cells (cf. [3,4,22,23]). For this purpose, the corresponding instances communicate by control messages with each other and the CON has interfaces to EMC, RMC and other CONs. The RMC determines the requested and granted resources for each call and CON, which implements the ACP, reserves an appropriate number of BW units for a potential handoff call in the neighbor cells.

To implement this process in a way similar to the mobile RSVP model, each connection of a service class is characterized by a bandwidth interval of minimal and claimed BW units (up to a maximal value) that are required to realize a call with sufficient service quality. Dropping of a handoff call is prevented or, at least, relieved by reserving an appropriate number of BW units in potential target cells surronding the current residence of an MH before a handoff. To avoid the wasteful passive reservation of bandwidth, the movement of MHs is anticipated and a portion of the actually demanded BW units is reserved for all potential calls and put into a BW pool called handoff BW buffer. It is exclusively used for handoff calls. To prevent inefficient BW use and to avoid high blocking probabilities of fresh calls, its size is limited.

Moreover, it is dynamically controlled depending on the experienced dropping probability. If the dropping probability that is monitored in a cell by the EMC and CON components exceeds a prescribed QoS level, more resources are granted in advance for future handoff calls in the cell. If a fresh call arrives and cannot be admitted due to a lack of capacity subject to these BW reservations applying a specified ACP, it is blocked. If a handoff call demands admission and

the required BW units are not available, then a two stage process is initiated. First, its requirements are reduced to the specified minimal value and admission control is applied again based on the specified ACP. Only if it fails, the handoff call is dropped. Thereby, the congestion of handoff calls is relieved exploiting the elasticity of the sources belonging to the premium class.

3.2 Admission Control Policies for Adaptive Mobile Applications

Considering the ACP for calls arising from adaptive mobile applications, we generalize the schemes discussed in [5]. We assume that class C1 comprises MHs running delay-sensitive applications whereas C2 includes delay-insensitive connections of elastic services, e.g. arising from http and plain tcp data transport. Regarding the performance targets of micro-mobility management C1 calls are characterized by maximal dropping probabilities $\mathbb{P}\{\text{drop}\}$ during handoff while no QoS objectives are specified for C2 calls.

Prediction of Mobility. The prediction of local movements of MHs between cells uses a list of handoff patterns observed in the past (cf. [3,4,5,22,23]). They are stored in an appropriate data structure. It comprises a list of quadrupels $Q = (t, k, i, d)$ related to a previous handoff event into the current cell 0 from a predecessor cell k. At time t after the dwell time d in the current cell 0 the MH has been handed over into the successor cell i. Regarding a new call arriving at cell 0 the tasks include the computation of the transition probabilities (t.p.s) p_h and the epochs t of its corresponding handoff events to successor cells i.

For the j-th connection $C_{0,j}$ handed over from a predecessor cell k into the current cell 0 a histogram-type estimate of the corresponding movement-dependent t.p.s can be calculated as follows:

$$p_h(C_{0,j} \to i) = \frac{\#\text{ quadrupels with predecessor cell } k \text{ and successor cell } i}{\#\text{ quadrupels with predecessor cell } k} \quad (1)$$

Regarding the conditional average dwell-times $t_h(C_{0,j} \to i)$ of the j-th call $C_{0,j}$ coming from cell k before handoff into cell i a similar empirical mean is used as estimate:

$$t_h(C_{0,j} \to i) = \sum_{l=1}^{|Q(k,i)|} d^l / |Q(k,i)| \quad (2)$$

Here $Q(k,i)$ denotes the set of all quadrupels with predecessor cell k and successor i and d^l the dwell time of the l-th quadrupel. To limit the computational effort, the order of the list should not exceed an upper bound $Nquad$ and outdated quadrupels are substituted by recent ones.

Bandwidth Reservation and Admission Control Strategies. Considering the passive resource management for calls of class C1 by CON, after arrival and admission to a cell 0 a portion b_{pre} of the demanded BW is reserved in the BW pool of C1-handoff calls in its neighbor cells. In cell i this passive BW reservation

of a call $C_{0,j}$ admitted in cell 0 is determined by the product of the seized BW $b(C_{0,j})$ and the t.p. $p_h(C_{0,j} \rightarrow i)$ to the successor cell i:

$$b_{pre} = b(C_{0,j}) * p_h(C_{0,j} \rightarrow i) \tag{3}$$

Together with the expected mean dwell time $t_h(C_{0,j} \rightarrow i)$ as time-to-activate (TTA) value, this amount b_{pre} is signalled by CON to cell i. Receiving the passive reservation message, the RMC of cell i is notified by its CON. It calculates the reservation epoch by $t = t_0 + t_h(C_{0,j} \rightarrow i) - T$ where t_0 is the actual time of a reservation receipt and T denotes a correction term. The latter adjusts the mobility predictions and is dynamically updated.

Following [7] the procedure quoted here has been incorporated in the ACP approach to adjust the corresponding correction T of the reservation epoch that influences a passive reservation. Here the variable STEP_SIZE controls the speed of the adaptation. It is required to correct errors of the mobility predictions that yield over- or under-estimates of the handoff BW and, hence, a performance degradation of the admission policy. A positive T will increase the reserved passive BW if the dropping level is exceeded. If one is in a regime far below this level, $T < 0$ shall reduce temporary the current portion of passive reservations.

```
SP =  1/P(Drop) ; LP = SP;
SH = 0; SHD = 0; LH = 0; LHD = 0;
T = 0;
IF (MH handed over)
{
    SH++; LH++;
    IF (MH connection dropped)
    {
        SHD++; LHD++;
        IF (LHD > 1)
        {
            LP += SP;
            T += STEP_SIZE;
        }
    }
    IF (SH == SP)
    {
        IF(SHD < 1)
            T -= STEP_SIZE;
        SH = 0; SHD = 0;
        IF (LH == LP)
        {
            LH = 0; LHD = 0;
            LP = SP;
        }
    }
}
```

Storing the capacity requests and TTA values of passive reservations in a list and traversing the latter periodically, CON and RMC of a RAN can determine the actual number of active reservations in a cell i. Activating a passive reservation, the reservation index B_i of a cell i is increased by the corresponding number b_{pre} of BW units. It determines the number of BW units in the handoff buffer serving the calls handed over from other cells.

If the connection $C_{0,j}$ leaves the cell 0, CON notifies the neighbor cells to release the corresponding reservations. If a reservation has not been activated, it is simply deleted from the list. Otherwise, the reservation index is decremented accordingly.

CON has the task to implement an ACP that guarantees a specified QoS level for a fresh or handoff call of classes C1 and C2, respectively. In our approach we use a scheme with this sketched distributed information state and a local

decision process. Regarding fresh calls of C2 and handoff calls of either type the acceptance decision is only determined by the utilization status of a considered cell 0. For fresh calls of C1 it is additionally determined by the state of the neighbor cells. We denote the reservation index of this target cell 0 with capacity $C(0)$ by B_0. Let C_0 be the set of all connection indices i in cell 0 with bandwidth demands $b(i) = b(C_{0,i})$ and let b_{new} be the BW demand of a new connection. Then the ACP is determined by the following rules:

1. admission of fresh calls of type C_1:
   - check $\sum_{i \in C_0} b(i) + b_{new} \leq C(0) - B_0$
   - check whether an overshooting of reservations exists in neigbhor cells (cf. [5])
2. admission of handoff calls of type C_1:
   - check $\sum_{i \in C_0} b(i) + b_{new} \leq C(0)$
3. admission of fresh and handoff calls of type C_2:
   - check $\sum_{i \in C_0} b(i) + b_{new} \leq C(0) - B_0$

Applied Adaptive Admission Control Policies. To identify the most appropriate ACP subject to a micro-mobility regime, several known bandwidth reservation and admission control schemes have been adapted to the new context of adaptive mobile connections arising from the premium and elastic services classes C1 and C2. There are three basic BW reservation schemes inherited from the class library of the tool Cellular Network Simulator (CNS) (cf. [1]):

Scheme A: Neither passive reservations nor CAC are applied. Fresh calls (FCs) are accepted as long as enough BW is available, whereas handoff calls (HOCs) are admitted if the minimal BW requirement can be satisfied.

Scheme B: It is similar to A. However, if BW is missing for a C1-HOC, it is grasped from the contingent of active C2 calls. Only the minimal requirement of a C1-HOC is seized. The reduction is uniformly distributed among all C2 calls.

UBB: This scheme applies passive BW reservations. A C1-FC is accepted if in the current and all neighbor cells there is enough BW applying the ACP therein. The maximal BW of all C1 calls is reserved in the neighbor cells.

A C1-HOC is admitted if enough BW can be provided to satisfy its minimal requirement in the current cell and all passive reservations in neighbor cells are granted. In contrast to C1-FCs or C2 calls, a C1-HOC is allowed and forced to seize a portion of the reserved BW in the handoff buffer. If the latter exceeds the capacity of the buffer partially or completely, the BW can be taken from the common resource pool if possible. If the call leaves the cell, the seized BW is put to the common pool and not added to the handoff pool. The latter is changed by future reservation or deletion requests.

These basic schemes have been combined with the following adjusted variants of some proposed adaptive distributed ACPs (cf. [5,6,7,15]):

UBB_adapt: It is similar to UBB, but applies a dynamic adjustment of the BW pool (cf. [15]). If a prescribed dropping level is exceeded or the utilization of the handoff pool exceeds a threshold, the latter is increased. If these control parameters fall below some thresholds, it is reduced.

Prob_Thres: For each C1 call a random portion of BW is reserved in neighbor cells and reduces the threshold levels (cf. [6]). A C1-FC is blocked if the modified threshold value cannot be granted in a neighbor cell. A C1-HOC can seize the overall available BW. Fresh calls and C2-HOCs are only admitted if the sum of their BW demand and the already seized BW does not exceed the threshold.

AC2: It is a threshold-based scheme with similar ACP like Prob_Thres regarding C2-FCs and C2-HOCs (cf. [7]). For C1-FCs and HOCs no passive reservations are performed in neighbor cells. It is only checked whether the seized BW does not exceed the thresholds there. In this case, the cell is too crowded and C1-calls are rejected.

AC3: It is similar to AC2, but additionally thresholds are adjusted between neighbor cells to increase inter-cell fairness (cf. [7]).

4 Performance Analysis of Adaptive CAC Schemes

4.1 An Enhanced Cellular Network Simulator (CNS)

Using the existing Java class library [1], we have enhanced the corresponding *Cellular Network Simulator (CNS)*. The related tool incorporates the sketched ACP variants of the bandwidth reservation and CAC schemes that have been adapted to adaptive mobile applications of classes C1 and C2, respectively. To enable a comprehensive investigation of these policies and their ability to guarantee both a maximal dropping probability of C1-HOCs and a high utilization of the bandwidth provided by a micro-cellular network subject to different handoff scenarios, the tool has been further enhanced by a unique graphical user interface (see Fig. 5). The latter offers the specification of all relevant network, load and simulation parameters. By these means it is possible to specify the network structure (called *Network Specification*) in terms of its size, i.e. as a grid of N micro-cells with a common capacity of C BW units each. By a *Mobile Host Specification* the parameters of both service classes C1, C2 can be determined. A connection of each class is given in terms of a minimal and a demanded BW level. Furthermore, the movement pattern of the MHs can be set. The attribute DIRECTED characterizes those MHs which select a random direction on the cell grid at connection set-up whereas RANDOM governs MHs selecting new cells at handoff events in a random manner. The grid is wrapped around, i.e. reaching its edge a new cell is selected at random. The *Simulation Specifications* allow to control the simulation runs and the used load model in terms of the arrival rate λ, the mean call durations μ^{-1}, and the mean dwell time γ^{-1} in a cell.

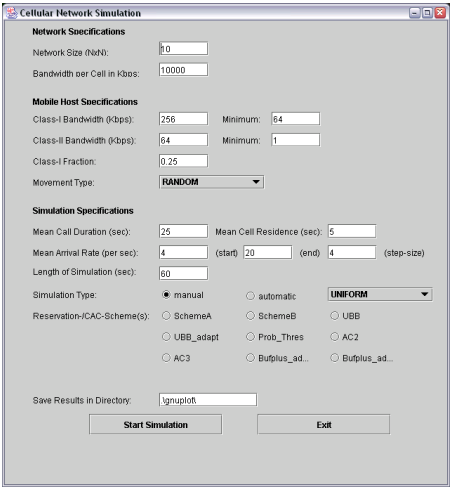


Fig. 5. GUI of a CNS enhancement

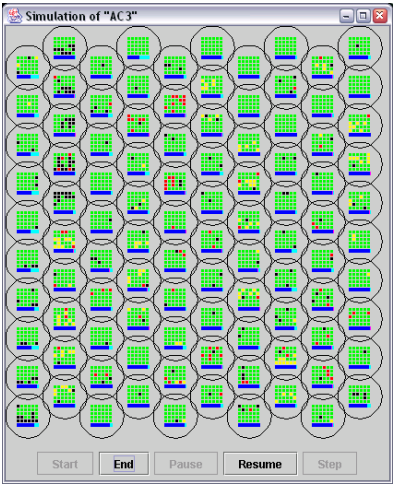


Fig. 6. Results of a simulation of a 10×10 grid

The arrival of MHs to a cell are governed by a Poisson distribution, and the call and dwell times are independent exponentially distributed. Furthermore, three different load scenarios can be studied:

- UNIFORM: For each cell the mean arrival rate is identical.
- SINGLE_SOURCE: New calls are always initiated in the same cell. Movement patterns can be tracked easily to trace a run.
- BURSTS: A target cell and its neighbors are the active zone of the network. After the expiration of a *Burst Cycle Timer* all MHs move simultaneously into the neighbor cells.

After the simulation runs all observed performance metrics, i.e. captured statistics of ACTIVE-, IDLE-, DROPPED- and BLOCKED calls, can be depicted and evaluated (cf. Fig. 6).

4.2 Performance Results of a Comparison

To investigate the performance of the developed ACPs and to determine which scheme should be implemented in the MQoS framework, a simulation study has been executed by the enhanced CNS tool. It uses a grid with 25 cells wrapped around at the edges yielding 6 neighbors of each cell.

The load is specified in terms of the demand of BW units (BU). C1 calls require 4 BU, C2 1 BU. Each cell can carry $C = 100$ BU. Two load scenarios are studied: either an equal number of C1 and C2 calls ($F_{C1} = 0.5$) is generated in a run or only C1 calls ($F_{C1} = 1$) are generated.

Two mobility patterns are studied: high mobility of MHs yielding on average 3.5 handoffs and low mobility resulting in 1.5 handoffs per call. The average call

duration is $\mu^{-1} = 120$ sec, the average dwell time in a cell $\gamma^{-1} = 34$ sec for high mobility and $\gamma^{-1} = 80$ sec for low mobility. The maximal utilization of a cell is determined by:

$$L = (4 * F_{C1} + 1 * (1 - F_{C1})) * \lambda * \mu^{-1}/C \tag{4}$$

It is the objective of the ACP simulation to compare the achieved QoS metrics, i.e. the mean blocking probabilities of calls and dropping probabilities of C1 handoff calls and the gained average utilization of a cell as well as the coordination costs of the control algorithms for adaptive mobile applications.

First, the impact of the sketched adapted local admission control schemes has been studied. Typical results for a scenario with load bursts show that AC2 is the worst scheme regarding the dropping and utilization objectives. The modifications exhibit a higher utilization. However, they cannot guarantee appropriate dropping levels. The second set of experiments compares the uncontrolled Scheme A with AC2 for a uniform load scenario. As shown in Figs. 7, 8 the utilization of Scheme A is only slightly higher than that achieved by AC2. However, Scheme A cannot guarantee the QoS objectives as expected. Hence, it is obvious

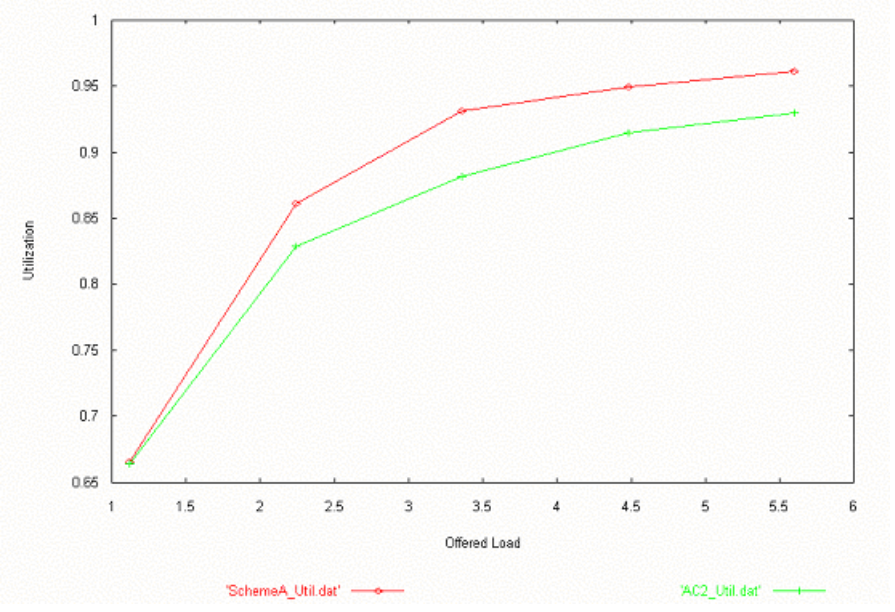


Fig. 7. Cell utilization for a uniform load scenario applying AC2 (lower curve) and Scheme A (upper curve)

that distributed ACPs are required to relieve congestion of handoff calls.

To evaluate further the benefits of those corresponding ACP proposals, moderately loaded cells with utilization $0.3 \leq \rho \leq 0.5$ are studied. Due to space

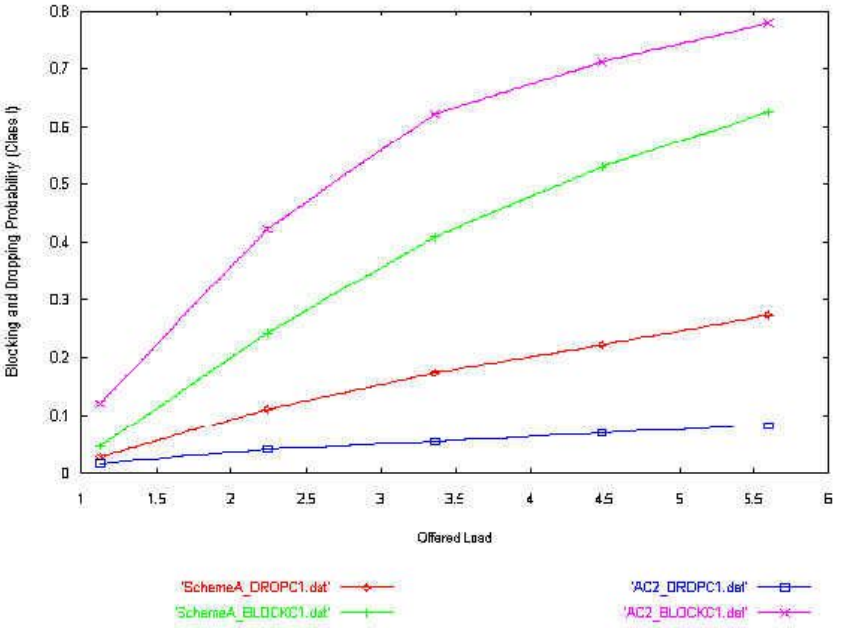


Fig. 8. Blocking and dropping probabilities of C1 calls given a uniform load scenario applying the AC2 and Scheme A

limitations only results of C1 calls are depicted here for the high mobility scenario. AC3 denotes the adapted scheme of Lee et al. [7], CHOI the variant of [5] and CHOI_agg_besserHoRes its enhanced version with reduced overhead developed here. While AC3 adjusts the reservation index B_i of a cell i mainly by the monitored dropping levels, and CHOI considers only BW requests of potential incoming handoff calls within a reservation window, CHOI_agg_besserHoRes represents a simplification of the latter. It works with the presented T-adjustment for the initial value STEP_SIZE = 1.

Focussing on the UNIFORM load model, a typical relationship of the dropping probabilities of C1 for different loads is shown in Fig. 9. We observe that all three schemes can guarantee the maximal dropping value of 0.1 and perform even better. Considering the average utilization of a cell, Fig. 10 illustrates that the proposed scheme CHOI_agg_besserHoRes outperforms the others. The reason is that it makes less passive reservations and distributes the saved BW among other calls. Therefore, the resulting dropping probability slightly increases as shown in Fig. 9. Finally, the communication overhead of a typical cell is depicted in Fig. 11. As expected CHOI experiences the highest and AC3 the least cost while the proposed scheme lies in between. The reason is that AC3 and CHOI_agg_besserHoRes require a test of the load status of all neighbor cells by the admission algorithm. If a call is completed or a handoff occurs, the latter scheme will cancel all pending reservations in these neighbor cells. Therefore, the communication overhead is twice as high as in AC3.

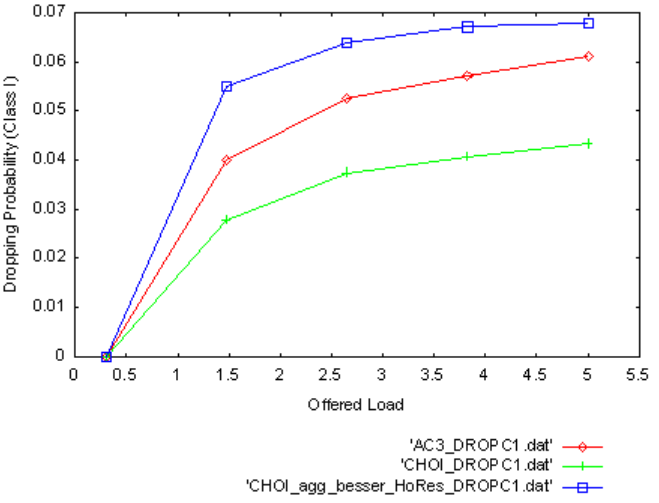


Fig. 9. Dropping probabilities for uniform load

In conclusion, we realize that our proposal of the distributed ACP CHOI_agg-_besserHoRes with passive resource reservation can be effectively implemented in our MQoS framework to improve the QoS management of adaptive mobile applications subject to micro-mobility regimes.

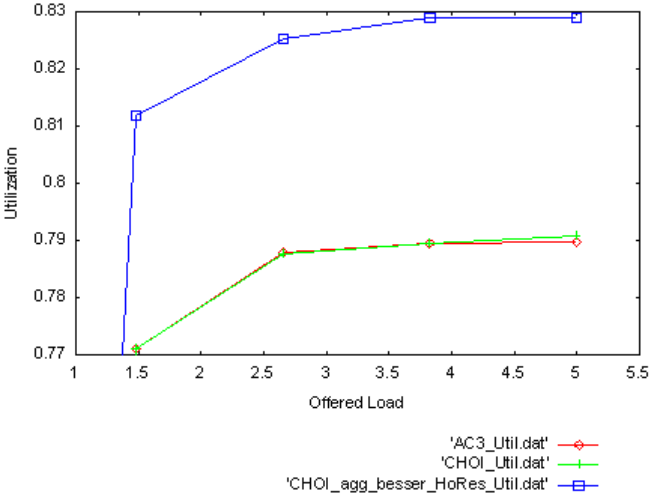


Fig. 10. Average utilization of a cell

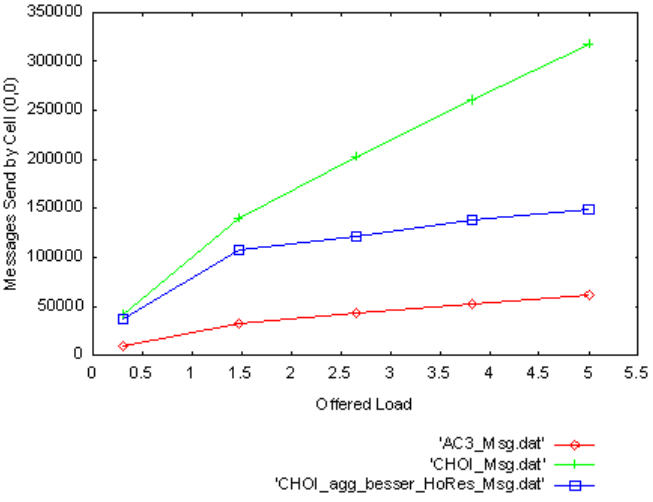


Fig. 11. Communication overhead

5 Conclusion

Currently, the integration of 3rd generation wireless networks such as WLANs and UMTS into high-speed IP core networks and its corresponding resource reservation and QoS-management mechanisms are some of the most challenging tasks of network design and engineering to support the deployment of new integrated location-aware multi-media and interactive Web services.

In our paper we have sketched a mobile QoS framework for adaptive mobile multi-media environments that integrates mobility- and quality-of-service management with new connection admission control (CAC) schemes. Refering to measurement-based CAC procedures applied in mobile access networks, we have discussed their extension to adaptive mobile applications and described the integration of the corresponding resource reservation and CAC policies into our framework. Furthermore, we have evaluated the performance of several CAC schemes by a simulation study. The results provide guidelines for further studies and an implementation in QoS architectures of next generation mobile networks.

Acknowledgment. The authors express their appreciation to J. Bachmann and T. Schönberger whose programming efforts constituted a substantial contribution to the project. Furthermore, U.R. Krieger wishes to express his sincerest gratitude to all those working at the library of T-Systems' Technologiezentrum Darmstadt, who supported him in gathering up-to-date scientific material throughout the past years. He acknowledges the "wisdom" of the trend-setting decision to close the library which reflects, from his personal point of view as a lecturer at J.W. Goethe-University, a mature level of scientific and cultural ignorance.

References

1. Possible Improvements to an Adaptive Bandwidth Reservation Scheme for High-Speed Multimedia Wireless Networks. *http://www.cs.odu.edu/~elkadi/mobile/final.html*, 1998.
2. J. Bachmann, M. Matthes, U.R. Krieger, and O. Drobnik. Mobility and QoS-management for adaptive applications. In *Proc. of 11. International World Wide Web Conference*, Hawai, May 2002.
3. P. Bahl, V. N. Padmanabhan, and A. Balachandran. Enhancements to the RADAR User Location and Tracking System. *Technical Report, MSR-TR-2000-12, University of Calfornia at San Diego and Microsoft Research*, December 2000.
4. J. Chan, S. Zhou, and A. Seneviratne. A QoS Adaptive Mobility Prediction Scheme for Wireless Networks. In *Proc. of Globecom'98*, 1998.
5. S. Choi and K. G. Shin. Predictive and adaptive bandwidth reservation for handoffs in QoS-sensitive cellular networks. In *Proc. of ACM SIGCOMM'98*, September 1998.
6. R. Hutchens and S. Singh. Bandwidth Reservation Strategies for Mobility Support of Wireless Connections with QoS Guarantees. In *Proc. of Twenty-Fifth Australian Computer Science Conference (ACSC2002)*, 2002.
7. J. Y. Lee, S. Bahk, and K. Park. Adaptive Admission Control with Inter-Cell Fairness in QoS-Sensitive Wireless Multimedia Networks. See *http://citeseer.nj.nec.com/479960.html*, 2003.
8. D. A. Levine, I. F. Akyildiz, and M. Naghshineh. A Resource Estimation and Call Admission Algorithm for Wireless Multimedia Networks Using the Shadow Cluster Concept. *IEEE/ACM Transactions on Networking (TON)*, vol. 5, no. 1, February 1997.
9. S. Lu and V. Bharghavan. Adaptive Resource Management Algorithms for Indoor Mobile Computing Environments. *ACM SIGCOMM Computer Communication Review*, vol. 26, no. 4, October 1996.
10. I. Mahadevan and K. M. Sivalingam. A Hierarchical Architecture for QoS Guarantees and Routing Wireless/Mobile Networks. *Journal of Parallel and Distributed Computing*, vol. 60(4), pp. 510–520, April 2000.
11. I. Mahadevan and K. M. Sivalingam. Architecture and Experimental Results for Quality of Service in Mobile Networks using RSVP and CBQ. *Wireless Networks*, vol. 6(3), pp. 221–234, June 2000.
12. M. Matthes, J. Bachmann, U.R. Krieger, and O. Drobnik. A QoS management system for adaptive applications supporting mobility and end-to-end guarantees. In *Proc. of the IASTED International Conference, Wireless and Optical Communicaions (WOC2002)*, Banff, July 2002.
13. M. Matthes, O. Drobnik, and U.R. Krieger. Auswirkung drahtloser Netzsegmente auf die Verkehrsgüte von TCP/IP-Verbindungen. *Kommunikation in Verteilten Systemen (KiVS)*, 12. Fachkonferenz der Gesellschaft für Informatik (GI), Fachgruppe "Kommunikation und Verteilte Systeme" (KuVS), Springer Verlag, Germany, February 2001.
14. M. Naghshineh and M. Schwartz. Distributed Call Admission Control in Mobile/Wireless Networks. *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 4, May 1996.
15. C. Oliveira, J. B. Kim, and T. Suda. An adaptive bandwidth reservation scheme for high-speed multimedia wireless networks. *IEEE Journal on Selected Areas in Communications*, vol. 16(6), August 1998.

16. C.E. Perkins. IP Mobility Support for IPv4. *IETF Working Group: Network, Request for Comments (RFC) 3344*, August 2002.
17. R. Ramjee, T. La Porta, S. Thuel, K. Varadhan, and S.Y. Wang. HAWAII: A Domain-Based Approach for Supporting Mobility in Wide-area Wireless Networks. In *Proc. of Seventh Annual International Conference on Network Protocols*, November 1999.
18. R. Ramjee, T. La Porta, S. Thuel, K. Varadhan, and S.Y. Wang. IP micro-mobility support using HAWAII. *IETF Internet Draft*, 1999.
19. P. Reinbold and O. Bonaventure. A Survey of IP Micro-Mobility Protocols. CiteSeer: Scientific Literature Digital Library, http://citeseer.nj.nec.com/ reinbold02survey.html, 2002.
20. A. K. Talukdar, B. R. Badrinath, and A. Acharya. Integrated Services Packet Networks with Mobile Hosts: Architecture and Performance. *Wireless Networks*, vol. 5(2), pp. 111–124, March 1999.
21. N. Tang, S. Tsui, and L. Wang. A Survey of Admission Control Algorithms. *Technical Report*, Course Project: A Survey of Admission Control Algorithms for Providing QoS over the Internet, Computer Science Department, University of California (UCLA), September 1998.
22. F. Yu and V.C.M. Leung. A framework of combining mobility management and connection admission control in wireless cellular networks. In *Proc. of IEEE ICC'01*, June 2001.
23. M. M. Zonoozi and P. Dassanayake. User mobility modeling and characterization of mobility patterns. *IEEE Journal on Selected Areas in Communications, 15(7)*, September 1997.

A Control Architecture for Quality of Service and Resource Allocation in Multiservice IP Networks

Raffaele Bolla, Franco Davoli, and Matteo Repetto

Department of Communications, Computer and Systems Science (DIST)
University of Genoa
Via Opera Pia 13, I-16145 Genova, Italy
{lelus,franco,repetto}@dist.unige.it

Abstract. A multiservice IP network based on the DiffServ paradigm is considered, composed by Edge Routers (ER) and Core Routers (CR), forming a domain that is supervised by a Bandwidth Broker (BB). The traffic in the network belongs to three basic categories: Expedited Forwarding (EF), Assured Forwarding (AF) and Best-Effort (BE). Consistently with the DiffServ environment, CRs only treat aggregate flows; on the other hand, ERs keep per-flow information (from external sources or other network Domains), and convey it to the BB, which knows at each time instant the number (and the bandwidth requirements) of flows in progress within the domain for both EF and AF traffic categories. A global strategy for admission control, bandwidth allocation and routing within the domain is introduced and discussed in the paper. The approach adopted is based on the combination of analytical and simulation models of traffic with service guarantees and of TCP aggregated traffic. The global scheme (under different traffic patterns) is investigated and the results of its application under different traffic loads are studied on a test network with a ns-2 simulation tool.

1 Introduction

Today's Internet [1] has a hierarchical structure, with Autonomous Systems (AS) at the top level, interconnected to each other to form the backbone of the net. The internal organization of ASs can be also hierarchical, reducing the number of peers in each level. This has many management benefits, such as simplifying IP addresses assignment to secondary Internet Service Providers (ISP), which also reduces IP routing table size, limiting control traffic to smaller areas, and so on. This is even simpler to obtain in IPv6 [2], where the increased size of the address field allows great flexibility in assigning and partitioning the whole space. As an example of how a huge network can affect performance, we cite inter-domain (among ASs) routing, where BGP often fails to assure route stability and routing convergence[3]-[5].

It is therefore reasonable to choose a hierarchical structure in global network control approaches, and to assume to have rather few nodes on each level. At the bottom level, nodes correspond to hosts; at higher levels, nodes correspond to aggregates of nodes at lower levels. Such architecture is very flexible: the same

W. Burakowski et al. (Eds.): Art-QoS 2003, LNCS 2698, pp. 49–63, 2003.

procedure can be used at every level, with the main difference lying on the aggregation size of the corresponding network traffic.

As regards Quality of Service (QoS) in the Internet [6], the two philosophies of *Integrated Services* (IntServ) and *Differentiated Services* (DiffServ) can be considered as rather complementary. IntServ tries to meet QoS requirements by reserving resources for every single flow along its path across the network, so that it can guarantee per-flow QoS. The RSVP signaling protocol [7], [8] is used for resource reservation, and the process may become very heavy to handle in the presence of many flows. On the other hand, DiffServ identifies only several QoS classes, with different packet treatment; it aims at satisfying common-class QoS requirements, by controlling network resources at the nodes (*Per Hop Behavior, PHB* [9]). A central controller, denoted as *Bandwidth Broker* (BB), can be dedicated to perform resource reservation, as well as to collect aggregate information for the upper levels.

In this paper, we assume the presence of a BB controlling a DiffServ network domain, and we introduce a control architecture with a two-level hierarchical structure, which acts on call (flow) admission control (CAC), bandwidth allocation to traffic classes, and routing. The aim is to minimize blocking of the "guaranteed" flows, while at the same time providing some resources also to best-effort traffic. As the main objective is to apply the control actions on-line, a computational structure is sought, which allows a relatively fast implementation of the overall strategy. In particular, a mix of analytical and simulation tools is applied jointly: "locally optimal" bandwidth partitions for the traffic classes are determined over each link (which set the parameters of the link schedulers in the CRs, the BE routing metrics and admission control thresholds); a "high level" simulation (involving no packet-level operation) is performed, to determine the per-link traffic flows that would be induced by the given allocation; then, a new analytical computation is carried out, followed by a simulation run, until the allocations do not significantly change. For what concerns the “high level simulation”, a simple model is proposed to describe the behavior of *aggregate* traffic composed by TCP flows. This model is tested to demonstrate that it gives an acceptable approximation of the performance index for this type of traffic.

We choose to consider three QoS classes: *Expedited Forwarding* (EF), *Assured Forwarding* (AF) and *Best Effort* (BE). EF is the maximum priority traffic, and it should experience very low loss, latency and jitter while traversing the network [10]; such a service appears to the endpoints like a point-to-point connection, or a leased line. Users requiring this kind of service are asked to declare their peak rate, and are classified in a limited number of classes, according to their rates.

AF is an intermediate class, with less pressing requirements on bandwidth, delay and jitter [11]. AF traffic is served on a link with lower priority with respect to EF. An AF user declares its mean and maximum transmission rate. The mean transmission rate is always guaranteed to users; better performance can be achieved if the network load is low. Also in this case users are classified in a fixed number of subclasses, based on their mean rate value. CAC is necessary to assure that there are enough resources for both EF and AF requirements. BE is the current Internet traffic, with no guarantees on throughput, delay and jitter.

The paper is organized as follows. We define the control architecture in Section 2. Section 3 deals with the flow model of TCP adopted. The control algorithm is detailed in Section 4. Section 5 reports numerical results and Section 6 the conclusions.

2 System Model and BB Architecture

As a basic issue, a suitable queuing discipline is needed to implement the PHB required by each class. This is not part of our control structure, which acts upon flows, so we do not go into the details of the possible implementations. We suppose different queues on each link, one for each kind of traffic.

Obviously, the performance experienced by the packets depends on the amount of traffic flows of each category admitted into the network, and on the presence of possible congestion avoidance mechanisms at the nodes [12, 13]. As regards the first issue, which is part of our control structure, we use the mechanisms of CAC and bandwidth allocation. Our aim is to control the bandwidth reserved for each class on every link. Given the peak rate for EF traffic, we know in advance how many connections can be accepted without exceeding the reserved bandwidth. A similar consideration holds for AF, except that mean rates are taken instead of peak rates. Briefly, we fix two bandwidth thresholds on each link: one for EF and the other for AF; the aggregate rates of flows traversing the link for each category can never exceed these thresholds. BE traffic can exploit all the unused link bandwidth at any time instant. This results in a *Complete Partitioning* [14] resource allocation strategy for both EF and AF, but not for BE, whose upper bound is not fixed, but depends dynamically on the bandwidth left free by other traffic flows. As complete partitioning may appear too rigid as a bandwidth allocation and CAC method, it is worth mentioning that the thresholds defining the partitions in our scheme will be made adaptive, following the variations in the bandwidth requests (the offered load) of the traffic categories.

As EF and AF flows require QoS, we think that this kind of traffic needs a fixed route from source to destination, which appears to them as a virtual leased line [10]. For this reason, from now on we will indicate these two categories as *Connection Oriented* (CO) traffic. The BB a priori fixes routing of CO connections, so that it can perform a CAC on the bandwidth available on each traversed link and, if necessary, deny access to the network. Thus, we have already identified a first task for our BB.

As regards the routing structure, in order to limit the scope of our architecture and the number of control issue involved, we have decided not to deal with QoS-routing for CO traffic. Since we wish to maximize the network throughput, we use a simple min-hop metrics for CO, to reduce the number of links traversed. In any case, the whole architecture is suitable for more advanced routing metrics [15], especially those based on aggregation and multi-parameter metrics [16]. For BE, instead, we choose a cost equal to the reciprocal of the residual available bandwidth left from CO connections. In this case, paths vary dynamically as network load changes over time.

In our model, both CO and BE traffic are generated at the ingress nodes of the network as aggregates: this means that the same node can generate more CO connections or BE flows. In such a way, we can easily view the node either as a host on the network or as a border router connected to a hierarchical low-level network.

Now, the issue is how to share link bandwidth among the three competitive classes adaptively, based on the threshold adjustment in the Complete Partitioning scheme. Obviously, CO traffic is critical, because it is thought of as a high revenue one. But we would like to give some form of “protection” to BE, too. To do this, we assign a cost to each class (as explained later in detail), and try to minimize some given cost function. Nodes collect data on the incoming traffic load; the BB gathers all these data

and computes a new allocation for all links. Finally, it sends the EF and AF thresholds to each node for its output links. This is the second task demanded to the BB.

To complete the description of the architecture, let us note that a third task of the BB is to collect information about domain ingoing/outgoing traffic from the nodes, and to act itself as a node for the BB of an upper level (if any).

The computational load on the BB, which may appear as scarcely scalable, can be reduced by a hierarchical organization structure: whenever the dimension of a domain becomes excessive in terms of computational effort and signaling, instead of further enlarging it, a new domain can be added, and the intra-domain resource allocation and routing can be managed by a higher level BB.

Global network allocation is not a simple task; so, to reduce complexity, we perform resource allocation separately for each link. On the other hand, bandwidth is a resource of the single link, and this somehow supports our choice. Note that our BE metric depends on allocation, but allocation in its turn (since it takes BE traffic into account) will depend on the paths selected by the routing algorithm. A congested link at a certain time instant can become an empty one after allocation, owing, for example, to the discovery of a new optimal route for BE traffic. Optimizing allocation separately on each link in the presence of such routing oscillations is very hard: changing bandwidth allocation results in modifying BE routing behavior. Our approach is to perform subsequent allocations until both allocation and routing possibly converge. For this purpose, we need a model for our network that drives the allocation choices. We have taken a hybrid approach, where a mix of analytical modeling and simulation is adopted.

As regards simulation, we distinguish two levels of detail. One, aimed at performance evaluation, attempts at modeling all aspects involved in the network with small time scale granularity (packet-level simulation). The other, which is used for control purposes, only takes into account aggregate traffic at the flow level.

Network simulation is done by means of the *network simulator* ns-2 [17]: we have added to the basic core all the elements required by our network architecture, such as the aggregate generators for both CO and BE traffic and the BB. An aggregate generator of CO traffic generates in parallel several EF or AF flows between two nodes (source and destination), referred to as peers. A BE generator works in a similar way. The CO flows dynamics is that of a bird-death process, and the BB keeps track of the offered load. Moreover, also the communication protocol between BB and CO peers has been implemented, so we can fully reproduce the behavior of our network taking into account the delay in the communication process between the nodes.

Let us now analyze in more detail how the BB works. We have two distinct phases: network simulation by a "fluid" model and allocation. The "fluid" model is needed by the allocation algorithm to simulate network (mainly BE routing) response to bandwidth allocation changes, as previously explained. We note again that in this case the model simulates the network at a flow level, in order to speed up the simulation.

The BE aggregate generators are used to produce several concurrent BE connections between two peers, but with this traffic we follow a quite different approach. We think of CO traffic as prevalently real-time, and therefore relying on RTP/UDP. Applications of such type usually transmit at constant bit rate, or at most adapt their rate dynamically (e.g., if they want to be "TCP-friendly"), but within a limited range. On the contrary, BE is the plain current Internet traffic, carried by TCP. TCP modeling is still an open issue in today's research, and it is a quite difficult task [18]. Some authors have proposed various models, even rather detailed [19]. Our aim

is to reproduce some peculiar characteristic of TCP *aggregate* behavior, without applying complex models; details on our approach are given in the next Section. Given the model and the sharing rules, on each link, we will determine the average bandwidth occupied by the BE traffic, to be used as one of the inputs to the allocation procedure.

The BB alternates simulation of the model and allocation steps. Through the model we generate network traffic and evaluate offered load on each link; successive allocations try to share the available link bandwidth among the different traffic classes. If allocation converges, network resources are optimized, and the BB can set new thresholds on all nodes. When variations on the incoming traffic are learned at the nodes, the BB is notified and starts a new allocation procedure.

3 Model of TCP Behavior

As already stressed, we adopt a very simple model of TCP (our goal is not to model the individual TCP flows, but rather to capture the aggregate behavior of connections between an ingress-egress router pair for control purpose). Basically, we take into account that connection durations in a simulation depend on network load. We believe it is more realistic to generate the amount of data to be transferred over a connection, instead of connection durations (sharing the same idea of other Internet researchers [18]). The effective duration of a connection is actually determined by the congestion control mechanisms of TCP and overall network conditions: in our architecture, the numbers of CO connections of every class determine the bandwidth left for BE traffic, and the TCP congestion control algorithm influences the way the remaining bandwidth is shared among the connections (or, better, among our aggregate flows). Because of this, we think of each BE peer as an infinite queue where session requests arrive; each session represents a given amount of data to be transferred. It can come from a network host (as a single user request) or from a lower-level network (as an aggregate of traffic). Inter-arrival times between sessions are exponentially distributed; the size p of each request follows a Pareto distribution, with shape and location parameters α and Δ, respectively:

$$f_p(P) = \alpha \Delta^{\alpha} P^{-(1+\alpha)} \tag{1}$$

Fixing $\alpha > 1$, the mean of this distribution is finite and equal to:

$$E\{P\} = \frac{\alpha \Delta}{\alpha - 1} \tag{2}$$

Such arrivals correspond to the amount of data to be transferred; sessions are served in parallel and the queue output rate is computed accordingly to network load, as outlined below.

In order to model the behavior of the aggregate flows of all peers in sharing the network bandwidth, so as to avoid a detailed simulation of the TCP characteristics, we suppose the flows to distribute according to a *max-min fair* rule [20]. Basically, this means that no aggregate flow between peers can increase its bandwidth share on a link at the expense of a flow with smaller demand traversing the same link. Given the

bandwidth available on each link for BE traffic, the max-min fair shares can be calculated with the algorithm presented in [20], whose basic idea consists of simultaneously increasing the rates of all yet unsaturated flows (i.e., those whose bottleneck link still has available bandwidth), until one link along the path (the bottleneck link) is filled. This means that all flows sharing that link cannot further increase, and the algorithm continues increasing all the others, until all flows cross at least one congested link and can no longer increase. It is worth noting that not all flows get the same share of a link, as flows belonging to different paths can experience different bottlenecks.

Two remarks are worth doing here. Firstly, though TCP connections, considered as individually composed by successive bursts, tend not to behave such as to achieve max-min fairness, nevertheless, by checking the results against detailed simulation we have found that aggregate TCP flows representing many connections along the path between peers actually do tend to distribute in this way. We will show some examples of this fact in the following. Secondly, it is true that this model neglects some important features of TCP, such as *slow start*, *error recovery*, etc. However, we recall that we are interested in simulating TCP data flowing across the network to calculate how long a connection takes to be served, and not in carefully modeling such protocol. So far, only a scenario with relatively low packet loss, due to congestion or Random Early Discard (RED) has been considered; high packet loss, which increases the size of traffic to be transferred, will be dealt with in future work.

As mentioned above, we give some examples to support the validation of our model. Since the unlimited-support continuous-time Pareto distribution (1) has raised several problems regarding stability of the simulations, we have preferred to use a truncated version with finite support. In our tests, the mean size of bursts (individual TCP connections) is chosen equal to 2 Mbits.

Fig. 1 shows the accuracy of our model, by comparison to ns-2 detailed simulation; several other tests varying the load of the peer, the link bandwidth and delay confirm this result. We have also observed that, at least in this simple scenario, the result is independent of the TCP version used, as shown in Fig. 2.

As a second example, we consider the simple network depicted in Fig. 3: two sources (peer 1 on node 3 and peer 2 on node 4) generate traffic toward the same destination (node 2). Link speeds are 25 Mbps for links 2 and 3, and 30 Mbps for link 1, respectively. Link 1 represents a bottleneck link for this network. The primary issue of this topology is to investigate the possible dependency of TCP behavior on different round trip time. Thus, links 1 and 2 are assigned a delay of 20 ms, while the delay of link 3 varies. We choose for link 3 several values (0.1 ms, 5 ms, 10 ms, 20 ms, 100 ms) that can be representative of different physical media (optical fiber, cable, wireless links, satellite links). Fig. 4 and Fig. 5 show no evidence of dependence on the RTT: this holds true also upon variations of the link bandwidth and the network load.

All the results in this section show that, although it is very simple, our flow model can be successfully used to represent the TCP aggregate behavior in a very fast and light (concerning CPU time) way.

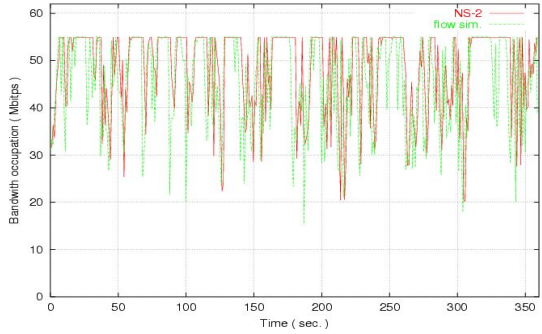


Fig. 1. Mean bandwidth from ns-2 and model. TCP load is 45 Mbps, link bandwidth 55 Mbps, and link delay 1 ms

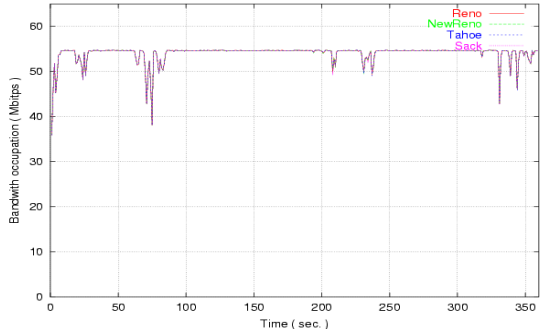


Fig. 2. Mean bandwidth with different TCP versions

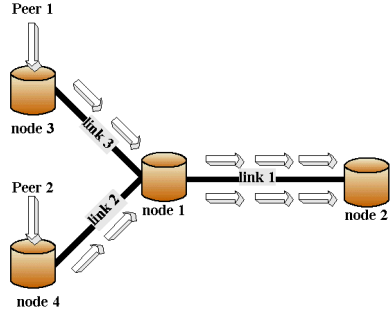


Fig. 3. The simple link bottleneck scenario

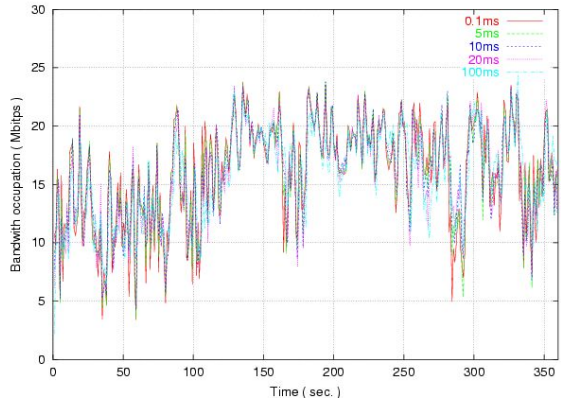


Fig. 4. Bandwidth occupation of peer 1 on link 1. Load of peer 1 is 15.6 Mbps

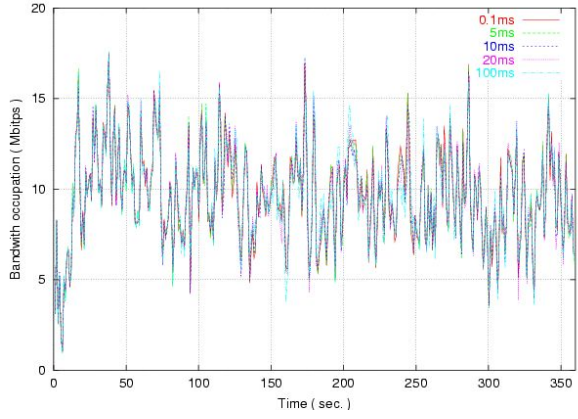


Fig. 5. Bandwidth occupation of peer 2 on link 1. Load of peer 2 is 9.6 Mbps

4 Allocation Algorithm

Allocating resources over the whole network is a very complex task, especially if many links are present. Furthermore, resources to be shared reside on each link. Thus, we choose to perform an independent optimization on every link, though computationally centralized at the BB.

As previously mentioned, the BB gets to know the input traffic matrix for each flow from the nodes. It runs a simplified simulation at the flow level (very fast) and collects data about offered load, in terms of number of flows trying to cross the link (CO) and average used bandwidth (BE) on each link. From this point on, it begins an independent allocation for each link.

We choose a bandwidth step *bw_step* equal to the minimum admissible rate for CO connections. Then, the algorithm starts calculating the costs for all possible bandwidth assignments, which are generated as follows:

- CO bandwidth is initially set to zero, then it is increased by *bw_step*, until the link bandwidth is reached;
- For each of these values, CO bandwidth is shared between EF and AF in a similar way: the EF threshold is set initially to zero and the AF threshold to the total CO assigned bandwidth. All possible assignments are generated incrementing EF and decreasing AF thresholds by *bw_step*.

Cost calculation

At the flow level, CO traffic looks like traditional circuit switched traffic: there are incoming connections of a given size (peak rate for EF and mean rate for AF), competing for a limited resource. Seen at each individual link independently, this is the so called *Stochastic Knapsack* problem, for which accurate analytical models are available; blocking probability can be calculated with an extended *Erlang Loss* formula [14].

Let K denote the number of connections of one CO class (EF, for example) and C the corresponding resource units. Within this class, connections of the k-th sub-class are distinguished by their size, b_k [bandwidth units], their arrival rate, λ_k, and their mean duration, $1/\mu_k$. Let us denote with n_k the number of active connections of class k and with $\mathcal{S}$ the space of admissible states:

$$S = \left\{ \mathbf{n} \in I^K : \mathbf{b} \cdot \mathbf{n} \le C \right\} \tag{3}$$

where I^K denotes the K-dimensional set of non-negative integers, $\mathbf{b}=(b_1,b_2,\ldots b_K)$ and $\mathbf{n}=(n_1,n_2,\ldots n_K)$.

Defining $\mathcal{S}_k$ as the subset of states in which the knapsack admits an arriving class-k object, that is

$$S_k = \left\{ \mathbf{n} \in S : \mathbf{b} \cdot \mathbf{n} \le C - b_k \right\} \tag{4}$$

the blocking probability for a class-k connection is:

$$B_k = 1 - \frac{\sum_{n \in \mathrm{S}_k} \prod_{j=1}^{K} \rho^{n_j} / n_j!}{\sum_{n \in \mathrm{S}} \prod_{j=1}^{K} \rho^{n_j} / n_j!} \tag{5}$$

where $\rho_j = \lambda_j / \mu_j$.

The blocking rate is computed separately for EF and AF, since each category has its own reserved bandwidth and incoming traffic, independent of the other.

A cost that is derived from the blocking probabilities of the different rate classes, and tends to equalize them, is given by

$$J_{CO} = \max_k \{B_k\} \tag{6}$$

This holds both for EF and for AF.

For BE traffic, we have a single aggregate: it is difficult to say if it has enough space or not, because TCP aims at occupying all the available bandwidth. A possibility consists of deciding that the bandwidth available to BE on a link can be considered sufficient if its average utilization by the BE traffic is within a given threshold.

To compute this, let *q(c)* be the probability to have *c* resources occupied in the system by EF and AF; *q(c)* can be computed through a recursive relation [14]. The average number of occupied resources (*UTIL*) results:

$$UTIL = \sum_{c=1}^{C} c \cdot q(c) \tag{7}$$

Thus, given EF and AF thresholds, the mean available bandwidth for BE is:

$$AVLB_{BE} = LINKBW - UTIL_{EF} - UTIL_{AF} \tag{8}$$

If *OCCBW* denotes the mean bandwidth occupied by BE traffic on the link, we could define the BE utilization ratio (UT) as:

$$UT = \frac{OCCBW}{AVLB_{BE}} \tag{9}$$

Note that, in principle, from the analytical and simulation model we could compute the average BE link utilization as follows. First, for each fixed combinations of values of *c* on each link and ensuing residual BE bandwidths, after finding the new BE routing, run a TCP flow level simulation, in the fashion described in the previous section, and compute the resulting utilization. In determining the residual bandwidth, we should take into account that the AF class has anyway priority over BE, e.g., by assigning it the peak rate of connections, as long as space permits. Then, all the values obtained in this way for BE utilizations should be averaged out with respect to the probabilities *q(c)*. In practice, this procedure would entail formidable computational difficulties. Thus, we have attempted the following approximate procedure, which exploits simulation not only to compute the distribution of BE flows, but also to generate a sample path of EF and AF calls, over which a time average is computed:

i) given a bandwidth allocation (EF and AF thresholds on all links), and basing on the knowledge of offered load, we generate birth and death events of all CO traffic in the network;

ii) between any two such events:
- the new BE routes, depending on the residual link bandwidths, are computed;
- the TCP flow model is run, by distributing the flows accordingly, and new BE offered loads resulting from the distribution are derived.

The BB keeps track of the BE mean occupied bandwidth on the links, during the simulation, and computes the value *OCCBW* by averaging over the whole period. Implicitly, in this procedure, we take into account the difference in time scales between the CO and BE traffic dynamics. Actually, events in the former are likely to be rather slow, thus allowing a significant redistribution of the TCP flows to take place. It is worth noting, anyway, that the length of the two "interlaced" flow-level simulations only depends on the numbers of events generated, and is completely independent of their actual durations. Summing up, each new value of the EF and AF thresholds induces new rates of BE traffic over the links, which in turn changes the

cost functions, and so on. We are therefore looking for a fixed point in this recursive procedure.

Returning to (9), we expect that connections on a link be given enough bandwidth if this ratio is less than a fixed value δ (i.e., the link is not completely saturated). In this case, we assign a null cost to the BE part; otherwise, we take the square of the difference between average utilization and threshold δ. Summarizing, we can compute:

$$G = \beta\left(\max\{0,\ UT - \delta\}\right)^2 \tag{10}$$

where β is a weight coefficient. This cost has a problem: it rises too fast above 1, while CO costs are upper bounded to 1. Varying β does not solve the problem; it is necessary to *smooth* the cost function after it reaches a given value. Thus, from (10) we get the BE cost function as:

$$J_{BE} = \min\{\phi + G/\Phi, G\} \tag{11}$$

Both parameters ϕ and Φ can be set to adjust the function's shape to our requirements. We found that Φ does not affect significantly our results, while they are more sensitive to variations of ϕ.

Optimization function

Several cost functions can be selected to optimize bandwidth sharing, depending on which results one is pursuing. Our aim is to equalize the costs of the classes, so that we can vary the system's fairness simply by changing some weight parameters:

$$J = \max\{\omega_{EF} J_{EF}, \omega_{AF} J_{AF}, \omega_{BE} J_{BE}\} \tag{12}$$

Equalization of costs results in minimizing (12):

$$\min_{T_{EF}, T_{AF}} \{J\} \tag{13}$$

where T_{EF} and T_{AF} are the thresholds for EF and AF traffic, respectively. This is replicated for each link.

5 Numerical Results

We have tested our algorithm by simulation on the network shown in Fig. 6; we have considered several active peers (see Table 1). The BB is located in node 1; node 2 is a critical point in this network. There are two bottleneck links, namely, 0-2 and 1-2. We use CO aggregate generators with λ = 0.6 flows per hour and μ = 0.4 h^{-1} (mean connection duration is 2.5 hours); for what regards the BE sessions, instead, they arrive with λ =1.3 bursts/s and the bursts have a mean size of 1 Mbit. The bandwidth is 2 Mbps for all links, except for links 0-2 and 1-2 (the bottlenecks), which have 7 Mbps. In all simulations, 95% confidence intervals have been computed (owing to the truncated Pareto distribution) to be within 5% of the estimated value.

We stressed our network starting with all EF peers listed in Table 1, plus BE1, BE2, AF1 and AF2. Then, we have increased the number of AF peers on each test by adding AF3, then AF4 and so on, up to AF11. We compare the results obtained with our allocation scheme with the situation of no allocation at all: in the latter case EF and AF flows can enter the system until there is free bandwidth on the links, in *Complete Sharing* fashion.

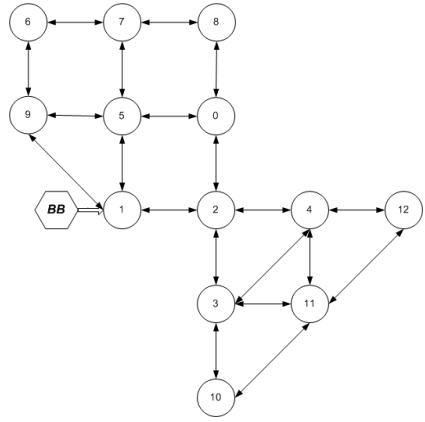


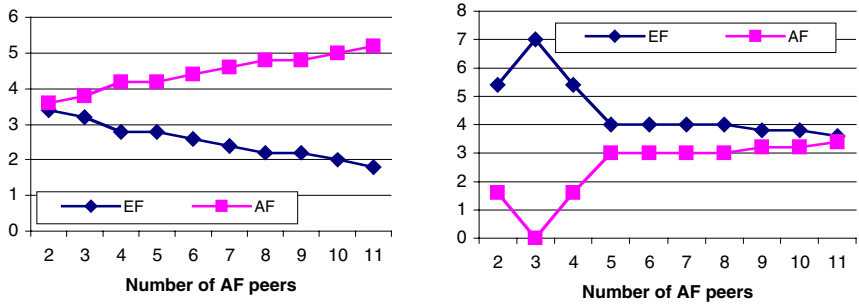
Fig. 6. The test network

Table 1. The peers in the test network

Traffic Generator	*Source Node*	*Destination Node*	*Flow Bandwidth (Kbps)*
EF1	5	4	200
EF2	0	3	200
EF3	6	11	200
EF4	9	10	200
EF5,AF4,AF9,AF10	7	3	200
AF1, AF6	5	3	200
AF2, AF7	0	3	200
AF3, AF8	8	11	200
AF5	6	12	200
AF11	5	5	200
BE1,BE3,BE5,BE7	0	4	-
BE2,BE4,BE6,BE8	9	3	-

Thus, in our tests AF load on the bottleneck links increases; correctly, also the reserved bandwidth on this link for AF increases (Fig. 7, allocated bandwidth on link 0-2 (A) and 1-2 (B)) and limits the global AF blocking probability (see Fig. 8, which

reports the global, i.e. averaged over all nodes, blocking probabilities). As a result of this, EF reserved bandwidth decreases (Fig. 7). We can observe that, without an evident general performance loss for the CO traffic (in terms of blocking probability, see Fig. 8) BE bandwidth on the bottleneck links significantly increases with respect to the situation of no allocation (Fig. 9 (A) and (B)): as already stated, assuring some form of protection also for BE traffic is one of our aims. The large difference in BE bandwidth allocated by the algorithm over the two bottleneck links in the case of few active AF peers are due to changes in routing behavior, which tend to have a smoother effect in the presence of many sources.



(A) (B)

Fig. 7. Allocated bandwidth on link 0-2 (A) and on link 1-2 (B)

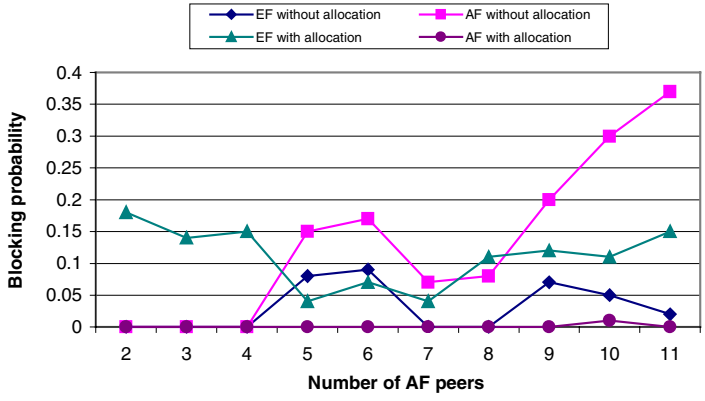
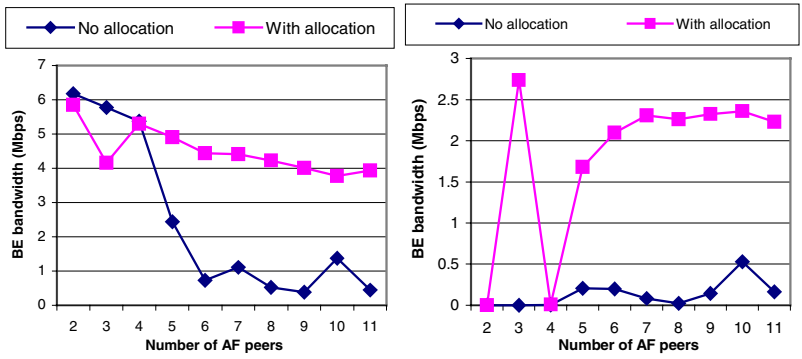


Fig. 8. Global blocking probabilities



(A) (B)

Fig. 9. BE mean occupied bandwidth on link 0-2 (A) and 1-2 (B)

6 Conclusions

We have introduced and analyzed a complete scheme for bandwidth allocation, CAC and routing in a multi-service IP network, operating in accordance with the DiffServ paradigm. The global strategy is based on a mix of analytical modeling and simulation, in order to find the optimum operating point for the network, under a given load pattern. The main objective of optimization is to obtain the capability to drive the network behavior toward a desired operating point, which takes into account the relative importance attributed in the DiffServ environment to the different service classes. Though further investigation is needed to assess the performance of the scheme, in terms of stability and convergence, and to compare it with other approaches, the results obtained so far show a very promising behavior. In particular, blocking of new flows for guaranteed bandwidth traffic can be kept at acceptable levels, without too much degradation in the performance of best effort traffic.

We have also outlined a simple model to describe the behavior of aggregate traffic composed by flows with congestion control (as the TCP one). Though the model is simple, we have demonstrated that it gives an acceptable approximation of the performance index for this type of traffic.

References

1. Telcordia Technologies: WWW document, http://www.netsizer.com
2. Deering, S., Hinden, R.: Internet Protocol Version 6 (IPv6) Specifications. RFC 2460, http://www.ietf.org/rfc/rfc2460.txt (1998)
3. Govindan, R., Reddy, A.: An Analysis of Internet Inter-Domain Topology and Route Stability. Proc. IEEE INFOCOM, Kobe, Japan (1997)

4. Labovitz, C., Ahuja, A., Bose, A., Jahanian, F.: Delayed Internet Routing Convergence. IEEE/ACM Trans. Networking, vol. 9, no. 3 (2001) 293–306
5 Paxson, V.: End-to-End Routing Behavior in the Internet. IEEE/ACM Trans. Networking, vol. 7, no.3 (1999) 277–292
6. Xiao, X., Ni, L. M.: Internet QoS: A Big Picture. IEEE Network (1999) 8–18
7. Braden, R., Zhang, L., Berson, S., Herzog, S., Jamin, S.: Resource ReSerVation Protocol – Version 1 Functional Specification. RFC 2205, http://www.ietf.org/rfc/rfc2205.txt (1997)
8. Hergoz, S.: RSVP Extensions for Policy Control. RFC 2750, http://www.ietf.org/rfc/rfc2750.txt (2000)
9. Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., Weiss, W.: An Architecture for Differentiated Services. RFC 2475, http://www.ietf.org/rfc/rfc2475.txt (1998)
10. Jacobson, V., Nichols, K., Poduri, K.: An Expedited Forwarding PHB. RFC 2598, http://www.ietf.org/rfc/rfc2598.txt (1999)
11. Heinanen, J., Finland, T., Baker, F., Weiss, W., Wroclawski, J.: Assured Forwarding PHB Group. RFC 2597 http://www.ietf.org/rfc/rfc2597.txt (1999)
12. Floyd, S., Jacobson, V.: Random Early Detection Gateways for Congestion Avoidance. IEEE/ACM Trans. Net., vol. 1, no. 4 (1993) 397–413
13. Aweya, J., Ouellette, M., Montuno, D. Y.: A Control Theoretic Approach to Active Queue Management. Computer Networks, vol. 36, issue 2–3 (2001) 203–35
14. Ross, K. W.: Multiservice Loss Models for Broadband Telecommunication Networks. Springer, Berlin (1995)
15. Chen, S., Nahrstedt K.: An Overview of Quality-of-Service Routing for the Next Generation High-Speed Networks: Problems and Solutions. IEEE Network, Special Issue on Transmission and Distribution of Digital Video, (1998)
16. Lui, K., Nahrstedt, K., Chen, S.: Hierarchical QoS Routing in Delay-Bandwidth Sensitive Networks. Proc. IEEE LCN 2000, Tampa FL (2000)
17. The Network Simulator – ns2. Documentation and source code from the home page: http://www.isi.edu/nsnam/ns/
18. Floyd, S., Paxson, V.: Difficulties in Simulating the Internet. IEEE/ACM Trans. Networking, vol. 9, no. 4 (2001)
19. Misra, V., Gong, W., Towsley, D.: Stochastic Differential Equation Modeling and Analysis of TCP-Windowsize Behavior. Proc. Performance '99, Istanbul, Turkey (1999)
20. Bertsekas, D., Gallager, R.: Data Networks. 2nd Ed., Prentice-Hall (1992)

Control Plane Architecture for QoS in OBS Networks Using Dynamic Wavelength Assignment

Sungchang Kim, JinSeek Choi, and Minho Kang

Optical Internetworking Lab, Information and Communications University,
58-4, Hwaam-Dong, Yuseong-gu, Daejeon, Korea
{pluto,jin,mhkang}@icu.ac.kr

Abstract. We address the control plane architecture of how to provide advance quality of service (QoS) in optical burst switching networks based on dynamic wavelength assignment. Unlike existing QoS guaranteeing scheme, such as buffer-based and offset-time based scheme, our proposed dynamic virtual lambda partitioning (DVLP) scheme does not mandate any buffer or extra offset time, but can achieve better QoS performance. This new DVLP scheme shares wavelength resources based on several different priority of classes in an efficient and QoS guaranteed manner. Also, hysteresis characteristic of DVLP is placed on robustness, meaning that each traffic classes with blocking probability conforming to the target value continue to receive the required QoS, despite the presence of misbehaving packets such as bursty arrival traffics. The performance results show that the proposed scheme effectively guarantees a target blocking probability of each traffic classes both in Poisson and Self-similar traffic environment.

1 Introduction

Increasing demands for transmission bandwidth driven by the growth of IP (Internet Protocol) based data traffic, especially real time multimedia services, give rise to dense wavelength division multiplexing (DWDM) technology which make possible to exploit the huge potential bandwidth of optical fibers. In addition to this advanced technology, it is natural to find ways to build the next-generation optical Internet architecture, which can transport IP packets directly over the optical layer without any opto-electro-optic (O/E/O) conversions.

Though optical packet switching technology can be attractive for all optical backbone networks, this technology has some technological limitations such as optical RAM and all optical processing. Presently, optical burst switching technology is under study as a promising solution for optical Internet backbone in the near future since OBS eliminate the electronic bottleneck at switching node with the help of no O/E/O conversion and guarantee the Quality of service (QoS) without any buffering [1][2][3].

In order to support today's mission-critical Internet traffic, optical Internet also has to support different traffic types based on their needs. However, until now there is

W. Burakowski et al. (Eds.): Art-QoS 2003, LNCS 2698, pp. 64–75, 2003.

little consideration of QoS in OBS networks. Existing QoS schemes for network layer are not readily achievable for WDM layer since the optical technology is not mature to support the optical buffering. Thus, in order to guarantee the QoS effectively in optical level, it is necessary to develop a new QoS scheme in optical layer, which should take into account the following considerations.

- Data information should be processed in all optical manner without E/O and O/E conversion at intermediate nodes.
- The upper levels of blocking probability and end to end delay should be supported.
- The hardware complexity such as processing time and implementation cost should be minimized.
- The QoS scheme should be efficiently scalable, reliable and available for WDM networks.

The focus of this paper is on the design of the control plane of OBS network and QoS guaranteeing algorithm based on dynamic wavelength assignment. The basic concept and general architecture of OBS core router is presented in Section 2 with data burst and control packet format and detailed description on the switch control unit. Subsequently, we proposed dynamic wavelength assignment algorithm for QoS performance using dynamic virtual lambda partitioning scheme in Section 3. Using this algorithm into the OBS network, Section 4 covers simulation and results, and some concluding remarks are made in Section 5.

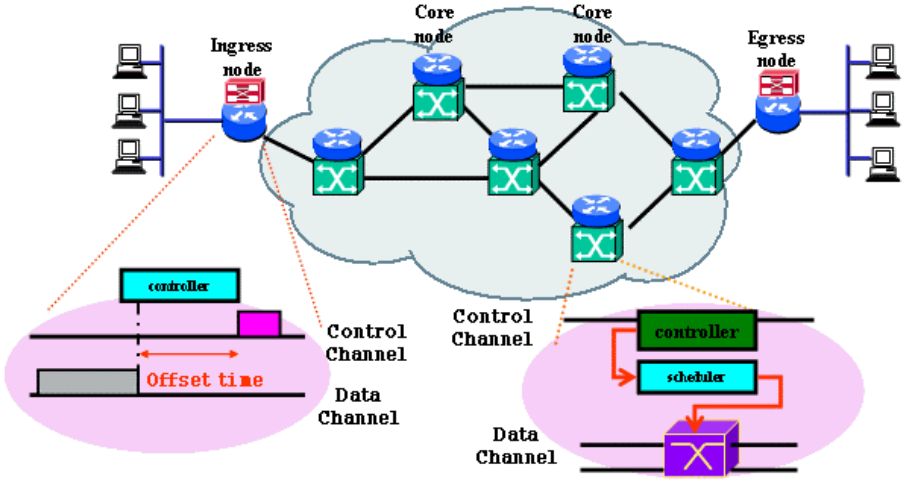


Fig. 1. An optical burst-switched network

2 Optical Burst Switching Network

OBS network consists of optical core routers and electronic edge routers connected by WDM links. Each burst in the OBS consists of a control header packet and a data

burst. The information of the data burst length and offset time is carried in the control header packet. Different from the conventional store-and-forward packet switching, the OBS uses separate wavelength channels for the data burst and its control header. The overview of OBS network is shown in Fig. 1, which consists of optical core routers and electronic edge routers connected by WDM links.

In an OBS network, packets are assembled into bursts at the network ingress and disassembled back into packets at the network egress. Therefore, the bandwidth is reserved at the burst level using a one-way process and a burst can cut through intermediate routers. The OBS network can be envisioned as two coupled overlay networks: a pure optical network transferring data bursts and a hybrid control network transferring control header packets. The control network is just a packet switched network, which controls the routing and scheduling of data bursts in the all optical network based on the information carried in their control header packets. This coupled overlay networks take advantage of both mature electronic control technologies and promising optical transport technologies.

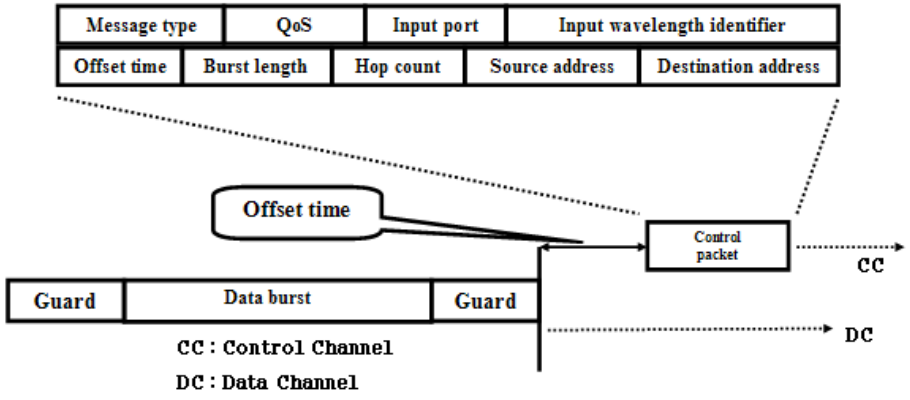


Fig. 2. Burst control packet format

Fig. 2 shows one example of control header packet format. The ***Message type*** indicates the header packet type. The message type field in each message uniquely determines the exact layout of the message. For the dynamic routing, there must be control messages such as link condition, header processing time and node condition to exchange information between each node so that it can discriminate the control header for the data burst and the control message packet for the maintenance of the network. The ***QoS*** indicates the following data burst's priority. With this and the offset time, core routers can select the shortest or the alternative light paths. It can be developed further to flow-based QoS. The ***Input port*** and the ***Input wavelength identifier*** are for the indication of the incoming data burst's port and wavelength. This provides switching information to the core router. The ***Offset time*** indicates the separation time between control header and data burst, which consists of the basic offset time and routing offset time. The ***Burst length*** indicates the length of data burst. The ***Offset time*** and the ***Burst length*** fields provide information for the core routers to reserve the

proper bandwidth and duration, and the ***Hop count*** is used to prevent bursts not to lose their destination router or the looping environment. ***The Source address*** and ***Destination address*** are addresses of the source and the destination.

2.1 Control Plain Architecture of OBS Network

The architecture of OBS core router using MEMS switch fabric is shown in Fig. 3, which mainly consists of switching control unit (SCU) and MEMS switch parts. Control channels are separated from data channels and terminated at the SCU. Control channels can be implemented either in-band or out-of-band signaling, which can be determined whether control channels and data channels are established within the same fiber. Typically in the case of in-band signaling, each input fiber has (K-*k*) channels for data burst, and *k* channels for control packets. On the other hand, in the case of out-of-band signaling, the control channel interface is independent from that of data channel, which means that the control networks can be implement by either electrical wire such as coaxial cable or fiber. The switch matrix which presented in Fig. 3 is MEMS technology based switching fabric, since MEMS switching fabric is considered as a promising technology can adopt next generation all optical switching matrix. The SCU electronically processes each incoming burst control packet (BCP) to reserve a bandwidth its corresponding data burst (DB) in advance while the data bursts are still remain in the form of optical signal in the MEMS fabric.

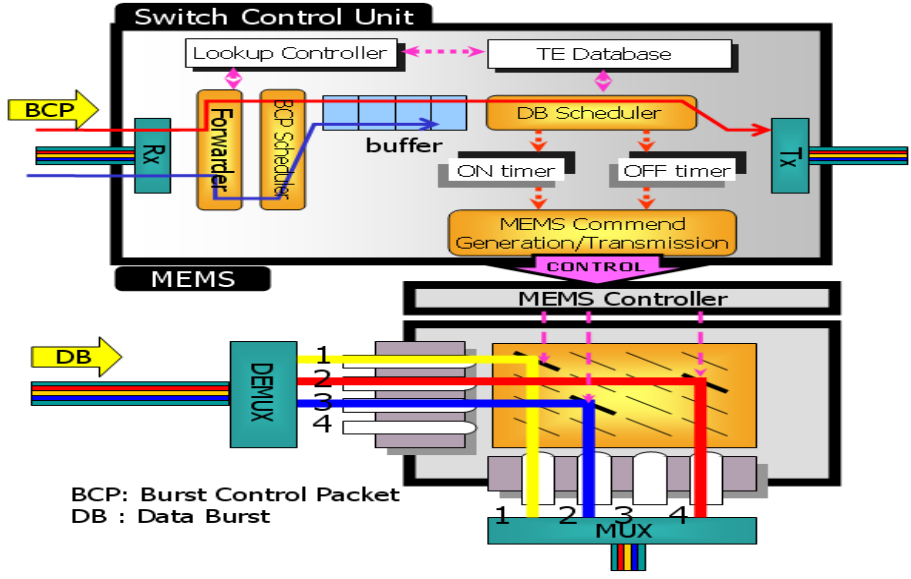


Fig. 3. OBS core node architecture

The detailed functional descriptions of SCU are described as follow. The Rx/Tx is 155Mbps ~ 622Mbps burst mode optical transceiver which is placed at input and out-

put interfaces. These perform optical to electrical or electrical to optical conversion, receiving and transmitting of asynchronous control packets which are usually fixed size, and L1 and L2 decapsulation functions. The forwarder performs the forwarding of BCP into the appropriate queue, which can be designed according to the destination and priority class, using lookup controller and TE database to decide on which outgoing control channel and priority class to satisfy the BCP.

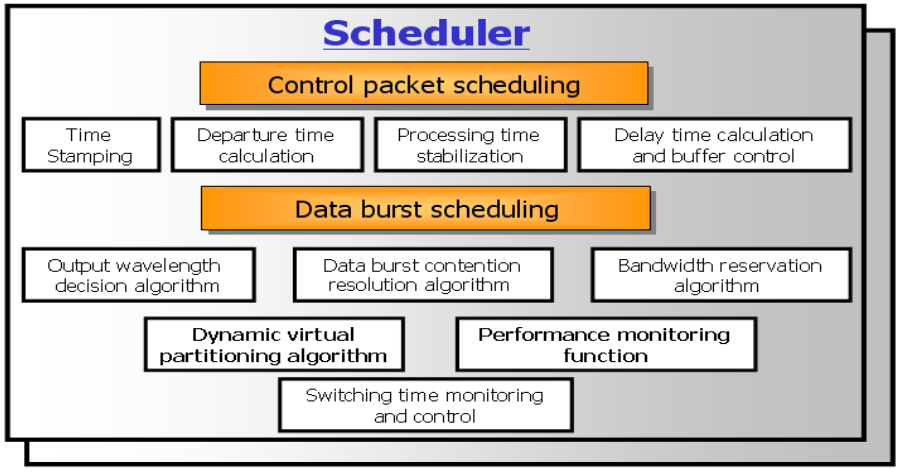


Fig. 4. Functional block diagram of scheduler

The scheduler in Fig. 4 is divided by two aspects. The one is control packet scheduling function and the other one is data burst scheduling function. The detailed functional block diagram of scheduler is presented in Fig 3. responsible for both the scheduling the MEMS switch of the data burst and the scheduling the transmission of its BCP.

The control packet scheduling function works as follows. Time stamping function first attaches a time-stamp to each arrival BCP, which records the arrival time of the associated data burst to the MEMS switching fabric. The departure time calculation function calculates the estimated departing time of control packet. If the expected departing time is exceed the upper bound of tolerable time, data burst will arrive earlier than control packet at the next hop. This phenomenon is called early arrival of data burst. In this situation, the scheduler may simply drop a control packet or assign a FDL to the data burst to make up additional delay time. Processing time stabilization function prevents large processing time variance which causes a serious increasing of end-to-end delay. Delay time calculation and buffer control function concerned about FDL control for data burst and buffer control for BCP. As we already know that OBS systems are not necessary for FDL, but limited number of FDL can significantly improve the network performance. The multiple buffer system for BCP can be implemented to support differentiated services operating with weighted fair queuing or preemption policies.

The first data burst scheduling function is output wavelength decision algorithm, which mainly searches for an idle outgoing data channel time slot to carry the data burst, making potential use of the FDLs to delay the data burst. Until now, various schemes are proposed as a direction of maximizing channel utilization. Data burst contention resolution algorithm also required to reduce the blocking probabilities, since the OBS network has inherently high blocking probabilities (bufferless concept). Subsequently, configuration monitoring and control function is carried out with the configuration information sent by the scheduler, the switch controller sends back an acknowledgement to the scheduler. The scheduler sends ON command to the switch controller based on arrival time of data burst and OFF command based on passing time of data burst. The scheduler then updates the state information of the data channel and control channel, modifies the BCP (e.g, the offset time and the data channel identifier). Our proposed dynamic virtual lambda partitioning (DVLP) algorithm is shown one of the main function in OBS control plane to support QoS performance. In order to guarantee a QoS performance, performance monitoring functions (e,g. blocking probability, channel utilization, and delay) are essential.

In order to implement practical OBS network, there are a lot of challenging issues to be solved. In respect of edge router, burst offset time management, burstification and burst assembly mechanism [4] are a critical issues. On the other hand, core routers need data burst and control header packet scheduling [5], protection and restoration mechanism and contention resolution scheme[6].

3 Dynamic Virtual Lambda Partitioning in OBS Network

In the view of OBS core routers, a burst control packet (BCP) arrives asynchronously from control channels of multiple input fibers. Processing of the BCP includes optical-electric conversion, address lookup to determine the output port, time and wavelength assignment, and switch control for possible reconfiguration of the switch. At the same instance, the scheduler feedbacks the information of the scheduling status to routing function block which increases the efficiency of the routing function. In the view of dynamic virtual partitioning, we focus on the scheduler, which is clearly an important function of resource sharing. In fact, OBS networks inherently have high blocking probability, which is mainly responsible for scheduler since it doesn't consider any buffering scheme in the intermediate nodes.

The dynamic virtual lambda partitioning (DVLP) is a scheme for sharing wavelength resources divided into several priority classes. All of the wavelengths within the fiber are dynamically partitioned depending upon QoS requirements. Furthermore, wavelength reservation policy of the each priority class is different. For example, high priority traffic can access the wavelength resources within own priority class as well as the resources in lower priority classes. The proposed algorithm guarantees a target blocking probability of each traffic classes and does not increase end-to-end delay without any extra offset time.

3.1 Description of DVLP Algorithm

We consider data and control channels of transmission rate R, a number of wavelengths in a fiber W, which is offered traffics from K classes. Class k traffics arrive as a Poisson process of rate λ_k, and independent exponentially distributed service times with mean $1/\mu_k$, that means data burst size is variable length. Define $W(i) = \{ [w_1, \cdots, w_k] \mid 1 \le w_1, \cdots, w_k \le W - K + 1, \sum_{n=1}^{K} w_n = W \}$ where $i = 1, 2, \cdots$, as the set of occupied wavelengths of each priority class when the i[th] BCP arrives at the scheduler and each class has to possess at least one wavelength. $B(i) = \{ [b_1, \cdots, b_k] \mid 0 \le b_1, \cdots, b_{N-1} \le 1 \}$ and $G = \{[g_1 \cdots, g_k] \mid 0 \le g_1 \cdots, g_k \le 1\}$, which represent the set of monitored blocking probability and the set of target blocking probability respectively.

DVLP has the following key ideas. First, at the time of design, each class is allocated a proper number of wavelengths, say, w_{k0} to class k, where $\sum_{n=1}^{K} w_{n0} = W$ to guarantee the QoS requirements, which the w_{k0} may be derived from traffic forecasts in conjunction with target blocking probabilities. These initial values are dynamically changed depending upon whether target blocking probabilities are satisfied or not along with time. This mechanism is implemented by a differentiation of monitored blocking probabilities which is easily calculated by comparing previous value. In order to reconfigure the number of wavelengths in each class, we adopt the threshold mechanism which has hysteresis characteristic. Using this mechanism, when the traffics are excessively fluctuated with time, the frequency of reconfiguration will be alleviated with less variance of time. Finally, the wavelength reservation rule of the each class is different depending upon its priority. Specifically, the high priority class traffics can reserve the bandwidth not only within its own wavelengths, but also the wavelengths in lower priority classes. For example, the class k traffics can access the wavelengths from the own class k to the lowest class 1.

The block diagram of our proposed DVLP algorithm is presented in Fig. 5. In this figure, we can define that T^u and T^o indicate occurrence number of the under-blocking and over-blocking probabilities respectively. Also, the th_u and th_o are the threshold values for under-blocking and over-blocking probabilities which can be determined to achieve hysteresis characteristic. In order to have a hysteresis charateristic, the th_o value sets up grater than th_u to alleviate the effects cased by bursty traffics. If the T^o is greater than th_o , this case is only occurred when the monitored blocking probability is over the target blocking probability, the higher class choose a wavelength which is randomly selected from the best effort class and vice versa.

When the BCP arrives at the scheduler, various reservation schemes can be applied such as first fit, round robin, and random. In our DVLP algorithm, we apply the LAUC-VP(Latest Available Unused Channel-Void Filling) [5] scheme. This scheme uses the void/gap, which is the unused channel capacity between the two data bursts in data channel. Using this scheme, we can achieve higher channel utilization and lower blocking probabilities than first fit or random scheme.

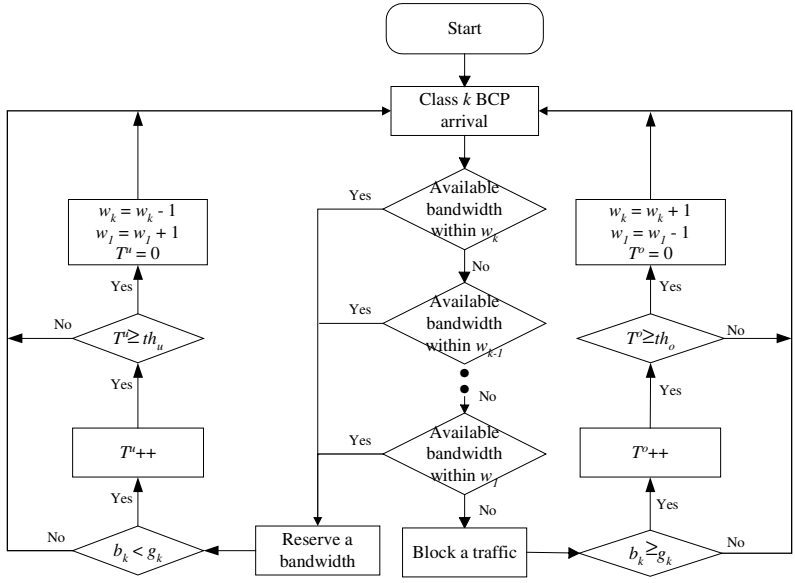


Fig. 5. Block diagram of DVLP algorithm

DVLP scheme is similar to the dynamic trunk reservation mechanism in TDM networks [7]. However, the performance of trunk reservation in TDM systems is different from that of the WDM system because of granularity of bandwidth reservation. Moreover, due to the delayed reservation, the statistical multiplexing gain of OBS will be another story compared with classical TDM networks.

4 Simulation and Results

We simulate the DVLP algorithm extensively using OBS simulator which made by java language. The simulation model is assumed that each node has 8×8 ports and that the priority of QoS defines three classes such as class 1, class 2 and class 3. Class 3 is the highest priority class and the class 1 is the lowest one. The scheduling mechanism for the data burst adopts the LAUC-VP scheme where 10 voids information can be stored in the scheduler because the memory space of scheduler is limited. After the simulation, we recognize that the 10 voids information for each channel is enough. We simulated that the distribution ratio of class 1, class 2 and class 3 are 70%, 20% and 10%, respectively. Additionally, we assume that the transmission rate of data and control channel is 10Gbps.

Basically, if the ingress edge router reduces the degree of self-similarity by certain traffic shaping function then exponential traffic affects the network performance. Unless the ingress edge router has any traffic shaping function, self-similar traffic affects the network performance. For this reason we chose the two different input traffics which one is Poisson and the other is self-similar traffic. We choose the Pareto

distribution which is simplest self-similar characteristic distribution with Hurst parameter H=0.9.

In order to decide the data burst length, we have to consider the several constraints. Usually, the link utilization of the OBS network will largely depend on the number of control channel, guard band period and traffic load. Let L_b be the average data burst length (in time unit), θ be the guard period of data burst, ρ be the offered load, and R and r be the data and control transmission rate. Then, the link utilization of the OBS network can be represented by

$$\eta = \frac{KR}{KR + kr} \times \frac{L_b - \theta}{L_b} \times \rho \tag{1}$$

Among the above factors, the guard band period θ can be significantly affected the channel utilization because of the technological limitations. For example, the switching time of the promising MEMS switch is in several *ms* order. Therefore the guard period θ of data bursts can be *ms* order. Let $\theta = 0.4ms$, in order to obtain 0.7 channel utilization, we can get the minimum burst length $L_b = 4.14ms$. Based on this constraint, we assume the data burst durations to be 4ms and offset time is $40\mu s$. Since, we do not use any buffer (or FDL), we analyze the blocking probability of classless OBS, say, completely sharing of wavelength resources, that can be modeled M/M/m/m systems [8] which is commonly called as the Erlang's B formula.

$$B(k, \rho) = \frac{m^k / k!}{\sum_{n=0}^{k} m^n / n!} \tag{2}$$

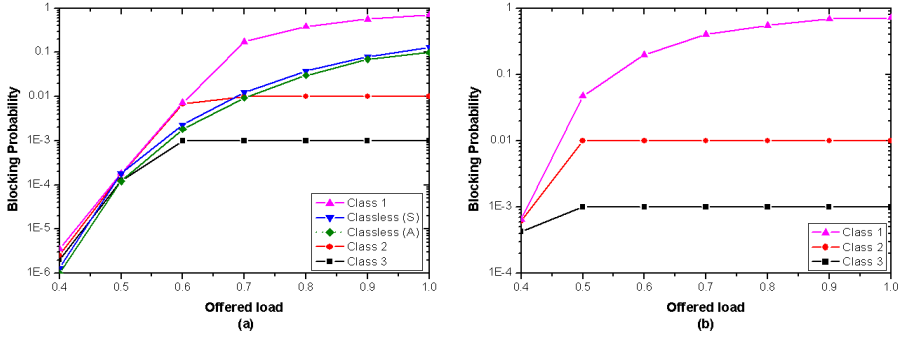


Fig. 6. (a) Average blocking probability of each class via offered load when Poisson traffic is applied. (b) Pareto traffic is applied

4.1 Performance Results

Fig 6(a) shows the blocking probabilities of individual priority classes as a function of the offered load when 32 wavelengths per port are used. The results of blocking

probabilities for classless OBS obtained from Erlang's B formula (represented A) are in good agreement with those from the simulation (represented S). We set the desired blocking probabilities of priority 3 class as 10^{-3}, priority 2 class as 10^{-2}, and priority 1 is the best effort service. As can be observed by comparing the blocking probabilities between priority 3 class and priority 1 class, service differentiation can be obtained by taking advantage of the DVLP algorithm. Fig. 6(b) indicates that the overall blocking probability when self-similar traffic is applied is higher than previous Poisson traffic case. However, we can see that service differentiation is also achieved between class 3 and class 1 very well.

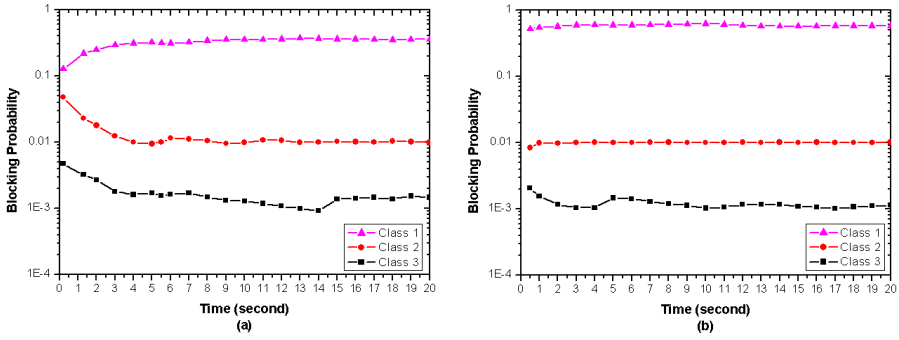


Fig. 7. (a) Average blocking probability of each class via time when the offered load is 0.8 and Poisson traffic applied. (b) Pareto traffic is applied

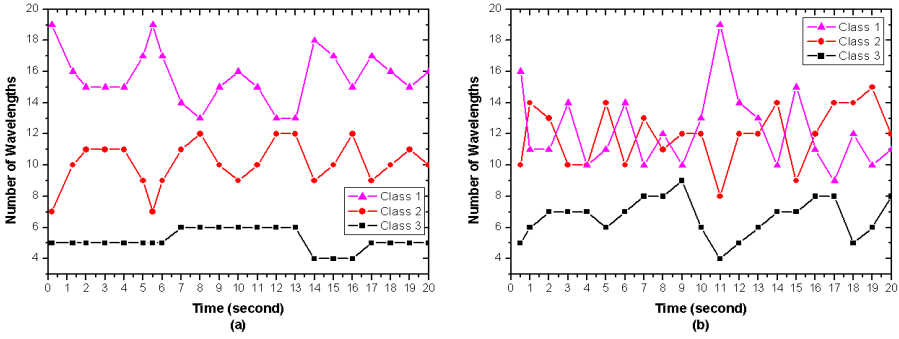


Fig. 8. (a) The number of wavelengths per class for 32 wavelengths per port when the offered load is 0.8 and Poisson traffic is applied. (b) Pareto traffic is applied

In addition to the service differentiation, Fig. 7(a) and (b) indicate that priority 3 and 2 classes show the desired blocking probabilities during the operating time when the offered load is 0.8 both Poisson and Pareto traffic cases respectively. However, price paid for the low blocking probability of priority 3 class is that the priority 1 class

has higher blocking probabilities than the classless case in the high offered load. This implies that the conservation law holds well. Thus we can regard the classless blocking probability is same as the average blocking probability of whole priority classes. Specifically, priority 3 guarantees the desired blocking probability after some settling time. This transient period can be existed to searching for the number of optimum wavelengths per group because the initial value w_{ko} is decided just after forecasting the future arrival rate and QoS requirements.

Fig. 8(a) and (b) show that the number of wavelengths dynamically varies by different priority classes when offered load is 0.8. The priority 3 and 2 classes minimally use the number of wavelengths while guaranteeing the desired blocking probability. As it can be seen in the case of Poisson traffic, the occupancy of wavelengths in priority 3 class is only from 4 to 6 wavelengths to guarantee the performance. On the other hand, in the case of Pareto traffic in Fig. 8 (b), the occupancy of wavelengths in priority class 3 is from 3 to 9 wavelengths. This implies that priority class 3 holds more wavelengths to reduce the blocking probability comparing to the Poisson traffic. Furthermore, you can identify that the fluctuation of the number of holding wavelengths in each classes is increased comparing to that of the Poisson traffic case. In this simulation, we found that the variation of the wavelength number is increased as the desired blocking probability is getting smaller. To compensate this variation, we need reconfiguration in such a way that the threshold values, th_u and th_o, have to set smaller than before because these can be react sensitively for guaranteeing lower blocking probability.

5 Conclusion

This paper considers a scheme for sharing a wavelength resources based on several different priority of classes in an efficient and QoS guaranteed. Hysteresis characteristic of dynamic partitioning is placed on robustness, meaning that each traffic classes with blocking probability conforming to the target value continue to receive the required QoS, despite the presence of misbehaving classes such as bursty arrival traffics. The algorithm we proposed is simple to implement for QoS performance in OBS networks.

Acknowledgement. This work was supported in part by the Korea Science and Engineering Foundation (KOSEF) through OIRC project.

References

1. Qiao, C.: Labled Optical burst switching for IP-over-WDM integration. *IEEE Communication Magazine*, Vol. 1, No. 9, September 2000, pp. 104–114

2. Wei. J.Y., Pastor. J.L., Ramamurthy, R.S, Tsai, Y: Just-in-time optical burst switching for multi-wavelength networks. Proceedings of 5th International Conference on Broadband Communications (BC'99), 1999, pp. 339–352
3. Qiao, C., Yoo, M.: Optical burst switching (OBS) – a new paradigm for an optical Internet. J of High Speed Networks, vol 8, no.1, pp. 68–84, 1999
4. An Ge, Callegati, F. and Lakshman S. Tamil: On Optical Burst Switching and Self-Similar Traffic. IEEE Communication Letters, Vol. 4, No. 3, March 2000, pp. 98–100
5. Xiong, Y., Vandenhoute, Hakki C.: Control Architecture in Optical Burst-Switched WDM Networks. IEEE JSAC vol.18, no.10, pp. 1838–1851, 2000
6. Sungchang Kim, Namook Kim, and Minho Kang: Contention Resolution for Optical Burst Switching Networks Using Alternative Routing. ICC 2002. IEEE International Conference on Communications, 2002, Volume: 5, pp. 2678–2681.
7. Sem C. Borst, and Debasis Mitra: Virtual Partitioning for Robust Resource Sharing: Computational Techniques for Heterogeneous Traffic. IEEE J. Selected Areas in Communications, vol.16, no. 5, pp. 668–678, June, 1998
8. Kleinrock, L.: Queueing Systems, volume 1: Theory. New York:Wiley Interscience, 1975

IP Services Market: Modelling, Research, and Reality

Piotr Arabas[1,2], Mariusz Kamola[1,2], and Krzysztof Malinowski[1,2]

[1] Warsaw University of Technology, Institute of Control and Computation Engineering, ul. Nowowiejska 15/19, 00-665 Warsaw, Poland
{P.Arabas, M.Kamola, K.Malinowski}@ia.pw.edu.pl
http://www.ia.pw.edu.pl
[2] Research and Academic Computer Network (NASK), ul. Wąwozowa 18, 02-796 Warsaw, Poland
{Piotr.Arabas, Mariusz.Kamola, Krzysztof.Malinowski}@nask.pl
http://www.nask.pl

Abstract. In the process of implementation and deployment of model-based tool for simulation, forecasting, decision support and optimisation, there discrepancies may emerge between the understanding of R&D team and the end user. Their origin is in the variety of optimisation problems a model is able to generate, depending on its parameters. A broad description of a market model for network services is given. Next, a case study follows that indicates a strategy for reduction of problems appearing while the product is handed over to the customer.

1 Introduction

The purpose of this paper is to present the outline of a model of network services market. Such model has been designed to be embedded in an optimisation solver to be then used for Network Services Provider (NSP) profit maximisation by setting service prices appropriately. Next, the authors focus on discrepancies that appeared between the model developer and the user regarding true nature of the simulation-based optimisation problem that emerged. Such discrepancies may result not from just imprecise problem definition but also from different notions of what actually can become the real obstacle for an optimisation task, as perceived by a development team with scientific background on one hand, and by the end user of the software on the other. Development and deployment of the simulation-optimisation module in the QOSIPS system will be the considered case study. It will be summarised with an account of how the resulting problems were solved in that particular case, and with a proposal of how to avoid such problems in the future.

QOSIPS is the acronym for "Quality of Service and Pricing Differentiation for IP Services" project whose main objectives were to develop innovative technologies for supporting QoS management, service differentiation and price setting of IP NSP's. The project was supposed to provide three major functionalities:

W. Burakowski et al. (Eds.): Art-QoS 2003, LNCS 2698, pp. 76–87, 2003.

accurate non-intrusive real-time QoS measurement of user's packets, service differentiation through traffic classification and use of QoS-oriented services, optimal composition and pricing of NSP services. One of the tasks within the Pricing Module workpackage was to develop a mathematical model of the IPN services (like VPN) offered to small and medium enterprises. After such a model is developed the model parameters are tuned, using either real life values (sales, utilisation etc.) or hypothetical scenarios, or just a rule of the thumb — depending on the available data. Once the model parameters are adjusted, this model can be used for prediction of market reaction to price changes, and even as a forecasting tool while introducing completely new products to the market. Embedding the market simulator in an optimisation routine allows the NSP to set the optimal product prices so that the profit (or other crucial parameters such as market share, or the mixture of them) is maximised.

The understanding of what the market model was expected to do was the same for all project partners, and resulted from careful investigation of the structure of the prevailing NSP tariffs. The modelling software closely followed its functional specification, was tested and accepted by the responsible parties in the project. Although the model details are commercially sensitive, some general outline will be given in Sect. 2. Next, the optimisation problem perception by the developer will be presented in Sect. 3, and confronted with that of the end user in Sect. 4. The conclusions are given in Sect. 5.

2 Market Model

The market modelling in QOSIPS was implemented to reflect the following phenomena:

- market segmentation,
- complexity of NSP offer (i.e. multi-component tariffs),
- usage-driven charging and cost schemes,
- customer activities (subscription, migration and churn),
- price-driven network utilisation,
- quality of service (QoS) degradation.

2.1 Market Segments

The market available to the NSP (and its competitors) can be divided in segments $S_1, S_2, ...$ Each segment is assumed to cluster potential customers having some distinct characteristics typical for them that may be identified. Market segmentation may be performed with respect to business type, budget level, usage profile and so on.

Market segmentation allows addressing ISP offer to the users, in particular to the market segments, basing upon common properties of the users grouped in a specific segment. Market segmentation is usually based on the customer features as follows: number of sites, number of roaming users, number of employees. The question arises, however, whether segmentation criteria presented

Segment name	Segmentation criteria: Type of business: Of low budget	Type of business: Of high budget	Usage pattern: Low data demand	Usage pattern: High data demand	Number of sites: 2	Number of sites: 3
S_1	√		√		√	
S_2	√			√		√
S_3		√	√	√	√	
S_4		√	√	√		√

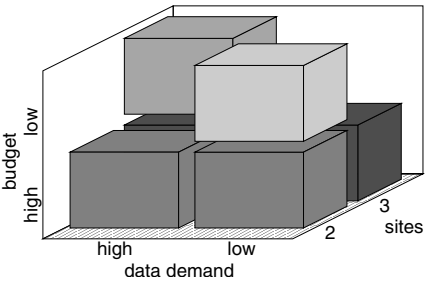


Fig. 1. Exemplary market segmentation presented by a table and graphically. Segment S_1 groups low-budget and low-data customers with two sites interconnected, and so on. The graphical representation splits the market represented by a large cube into smaller cubes of different colours that correspond segments — from the brightest (S_1) to the darkest (S_4). The three directions represent segmentation criteria. The two missing cubes represent the part of the market not being targeted.

above group potential customers properly. It seems that the number of sites or employees does not necessarily determine the traffic, usage or price sensitivity. Additional criteria that can give more appropriate market segmentation are the type of business, network/service usage pattern, type of application used and QoS requirements. A sample market segmentation is presented in Fig. 1.

Additionally, each market segment can be characterized by its size, i.e. the total number of current and potential customers belonging to it.

2.2 Tariffs and Their Elements

NSP offer consists of hardware, data links, applications, services, support options etc. bundled in packages. A package with its contents attributed by prices is called tariff. Let us denote tariffs by $T_1, T_2, ...$ A tariff contains universal elements (components) that can stand for hardware, application, services — depending on the context. They can be arranged freely in a tree-like structure to represent variety of NSPs' products. Besides, a tariff contains extra parameters as its expiry date and competitor prices (i.e. any known prices in competitive tariffs that influence demand for that particular tariff).

Any tariff element is allowed to be compound of sub-elements, thus providing the way to construct the tree-like structure mentioned earlier. Regardless of that, a subscription for elements can be obligatory, optional or subject to "one-of-many" scheme. Each element is attributed by a standard set of fees/costs that are charged/incurred when a client subscribes, changes, renounces or simply accepts for another month the NSP services. Therefore, there are monthly fee, monthly cost, installation fee, installation cost and so on. To reflect NSP incomes and expenses dependant on tariff element utilization, pricing schemes are introduced in the model.

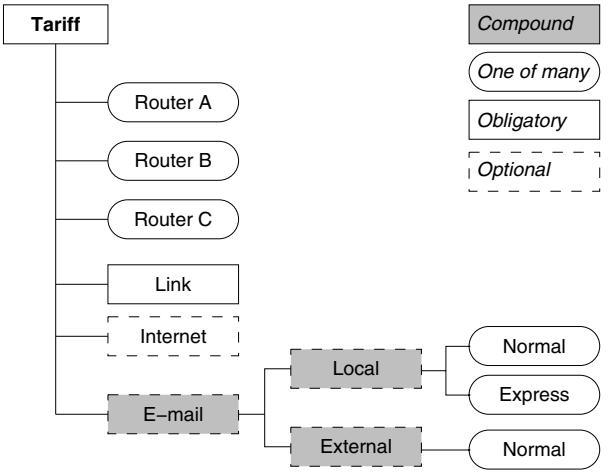


Fig. 2. An example of NSP offer represented as a tree of elements

Thanks to neutrality of elements, they can represent quantitative as well as qualitative diversity of the NSP offer. Let us look at Fig. 2 where elements serve equally well to describe hardware, data link, and services. The qualitative diversity (*Local/External* and *Normal/Express*) of mail is obtained in this case by extra branches hosting additional elements. The same could be done about e.g. the temporal service diversification (*Peak/Off-peak* hours, for example).

2.3 Usage-Based Schemes

Usage-based schemes are stepwise one-argument real-valued functions that are defined independently of other model entities. They are used to describe progressive pricing i.e. price that depends on some factor. Pricing schemes attribute tariff elements whose prices (and/or costs) are stepwise functions of that element utilization. Pricing schemes also serve in the calculation of compensation paid by NSP to the customer in case of not meeting QoS specification.

Let us denote usage based pricing schemes available in the system by $P_1, P_2, ...$ The definition of an exemplary pricing scheme and the graph of a corresponding stepwise function is presented in Fig. 3. This is, say, a charging scheme for data transfer. It defines that up to 6 MB (*Breakage level* column) the user is charged for every 2 MB (*Basic quantity*) by the amount of 2 (*Price per unit*). Having crossed the 6 MB boundary, the charging is as defined in the next line of the table.

2.4 Customer Activities

Tariffs can be offered to various segments. It is on the intersection of segment S_i and tariff T_j where customer behaviour can be measured because it takes form

Breakage level	Price per unit	Basic quantity
6	2	2
n/a	1	3

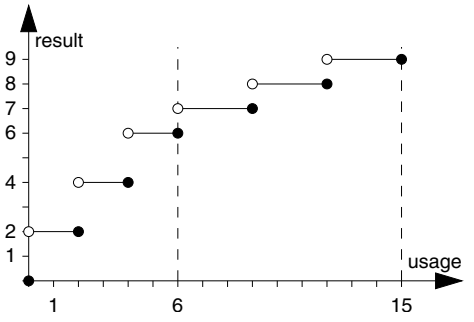


Fig. 3. Table defining a pricing scheme (left) and the corresponding graph (right)

of concrete numbers of new, migrating and churning customers, and of their distribution across the tariff elements.

Table 1. Sample assignment of tariffs to market segments

Market segments	Tariffs T_1	T_2	T_3	$\cdots$	T_n
S_1	$\surd$	$\surd$			
S_2		$\surd$	$\surd$		
S_3		$\surd$	$\surd$		
$\vdots$	$\vdots$			$\ddots$	$\vdots$
S_m					$\surd$

A single tariff may be addressed to multiple market segments, as it is presented in Table 1. With each tariff/segment intersection there is associated a state variable $x_{i,j}$ denoting the number of customers of tariff T_j in segment S_i. This number will change in time as new customers arrive, attracted by product prices, while the existing ones may stay, migrate or churn. The value of $x_{i,j}$ in month $t+1$ is determined by the following formula:

$$x_{x,j}(t+1) = x_{i,j}(t) + x_{i,j}^{new}(t+1) + x_{i,j}^{mig}(t+1) - x_{i,j}^{churn}(t+1)\ , \qquad (1)$$

where:

- $x_{i,j}(t)$ is the number of customers in the preceding month,
- $x_{i,j}^{new}(t+1)$ is the number of new customers,
- $x_{i,j}^{mig}(t+1)$ is the balance of migrating customers from and to other tariffs in the same market segment,
- $x_{i,j}^{churn}(t+1)$ is number of churning customers.

The number of new, migrating and churning customers are calculated using each time the widely known Cobb-Douglas formula (cf. [4,2]):

$$x(\mathbf{p}) = \alpha \prod_{k=1}^{\dim \mathbf{p}} p_k^{\beta_k} , \tag{2}$$

where $\mathbf{p} = (p_1, \ldots, p_k)$ is the vector of selected prices (own and competitors) or QoS metrics experienced in the previous month, and α, β_k are the function parameters: the scaling factor and sensitivities.

The distribution of $x_{i,j}$ across tariff elements is assumed constant (fixed). Such modelling simplification was introduced because of the amount of data too scarce to initialise many submodels that would be needed in case of exact modelling the number of customers for every sub-element.

2.5 Resource Utilisation

Utilization can be modelled for every tariff element, at the level of tariff/segment intersection. For this the purpose formula (2) is used but this time without previous QoS metrics as arguments. The meaning of the word "usage" depends on what actually the element is: e.g. for a data link it can be the total online time over a month, for an e-mail it can be the total volume of all mail transmitted etc.

2.6 QoS Degradation

Network traffic, represented by tariff elements utilization, may cause that the quality parameters of some services, guaranteed by provider in the contract, are not met. QoS guarantees depend on the nature of a particular application: for telnet it can be "transmission delay less than 10 millisecond", for video on demand it can be "packet loss rate less than 2%", and alike. In QOSIPS, one can associate with an element a number of QoS clauses like those above. If QoS requirements are not met, then a network provider must pay penalties dependent on the rate QoS degraded.

Let the QoS metric be the percent of data (or time) for which the QoS guarantee was not met. Therefore, a QoS metric can take values between 0 and 1, and obviously is determined by NSP infrastructure, customers' habits and locations. As there is no simple, precise and universal QoS modelling mechanism, rough formula has been put in the market model:

$$q(\mathbf{x}, \mathbf{u}) = \min\left(1, \max\left(0, a + b \sum_{k=1}^{\dim \mathbf{u}} x_k u_k\right)\right) , \tag{3}$$

which is but a linear function whose values are limited by 1 from above and by 0 from below. The function argument is what we have called *aggregated traffic*, i.e. the sum of products of indicated element utilizations u_k and the corresponding

number of users x_k for that element. This simple function has the advantage of having only two coefficients (a and b) and it is easy choose their values.

To compute the compensation for not meeting the QoS parameters, q is simply fed into one of progressive pricing schemes — the bigger QoS offence, the more money back to the customer. The employed stepwise functions allow for accommodating of the payback scheme to the individual nature of a subscriber. Adaptation of the compensations to a personal preferences of a user is an analogue for more complex methods of operational traffic control, e.g those based on the notion of utility function. For a discussion on the relations between pricing on the marketing and network operational level, see [6,7].

3 Development and Academic Model

In the process of the Mathematical Library Component development (MLC — the piece of the QOSIPS system responsible for forecasting and price optimisation), the R&D team produced many test problems to check the module against the existence of errors or to verify its compliance with the specification. For the purpose of this paper another simple market model has been created to demonstrate the crucial features of the MLC. It is composed of one tariff only, instead of the usual multitude of options (e.g. distribution across elements, tariff expiry dates, competitor prices); it models just the monthly volume of new and churning subscribers, the average network usage per subscriber, and the compensation paid when the NSP fails to provide the subscribers with appropriate quality of service.

Now it is time to mention the question of profit calculation that has not been touched while describing the model. The profit for each tariff element is calculated intuitively, by subtraction of monthly costs and QoS compensations from the income. The overall profit is the total of tariff element profits over a given period (6 months, on usual). Simple as it may seem, the profit calculation involves considerably difficult issues of putting some uncertainty in the model. That uncertainty initially did not exist, extinguished by modeling only means (of the number of customers, of the usage etc.) but it played an important role in calculations. Refinement of algorithms applied in MLC is still a subject under research.

In the considered simplified model there are only two prices that influence customer behaviour and determine the NSP profit. They are: the monthly fee and the price for transmitting a unit of data. There is also a credit for data, contained within the monthly fee. Usually, those two price variables are subject to additional constraints, i.e. they must lie within some interval determined either by the validity of the model or by market constraints (e.g. NSP own prices cannot differ too much from the competitive offer). Let us consider the first of them — the monthly fee. It is felt that normally the price that results in the highest profit must be neither too low nor too high. When low, there are many new customers and few leavers, but their number cannot compensate for losses due to the low price level. When the price is high, the customers will soon

desert to the competitors. Therefore, the optimal price must lie somewhere in the middle. The negated value of the NSP profit due to the recurrent fee in the academic model is drawn in Fig. 4a). From now on all the figures and tables will represent negated values of profit in order to convert the profit maximisation task into a standard objective function minimisation problem.

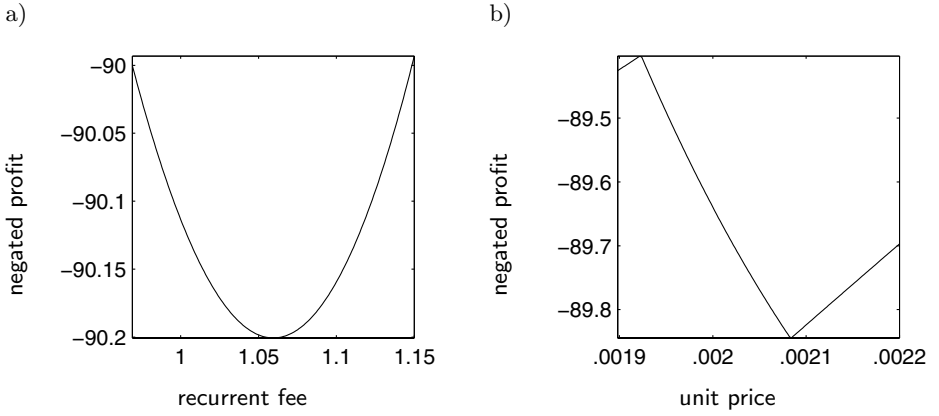


Fig. 4. a) Profit against the recurrent fee (for the unit price of 0.0022) **b)** Profit against the unit price (for the recurrent fee of 1.2)

There is also another price variable — the price of data unit. Like in the case of a monthly fee, raising the unit price discourages subscribers from increasing the volume of traffic beyond that covered by the fixed fee. Alternatively, lowering prices induces traffic volumes beyond what the network can transfer, and the NSP ends up paying high compensation to the users. Here again the optimum price lies somewhere in the middle, as shown in Fig. 4b). The contour plot of the performance function of both price variables is shown in Fig. 5.

In Fig. 5 one can easily observe irregularities in the level contours caused by activation of QoS paybacks that create additional, local optima. The development team expected them to be potentially the main cause of problems for the end user — especially when the starting point is away from the optimum. CRS2 and COMPLEX routines were employed for the task of optimisation, with modifications necessary to support extra constraints, whose description lies outside the scope of this paper. Also a proprietary solver developed by the QOSIPS project participants has been tested against this simple optimisation task.

– CRS2 is a global direct search method, described first in [3]. It maintains a set of points (randomly chosen at the beginning) and, by successive elimination of the worst ones and reflection of the best ones, slowly but steadily approaches the minimum. It performs best while far from the solution; in its proximity (i.e. where stronger assumptions as to the function shape can be made) it should be replaced by a more effective routine.

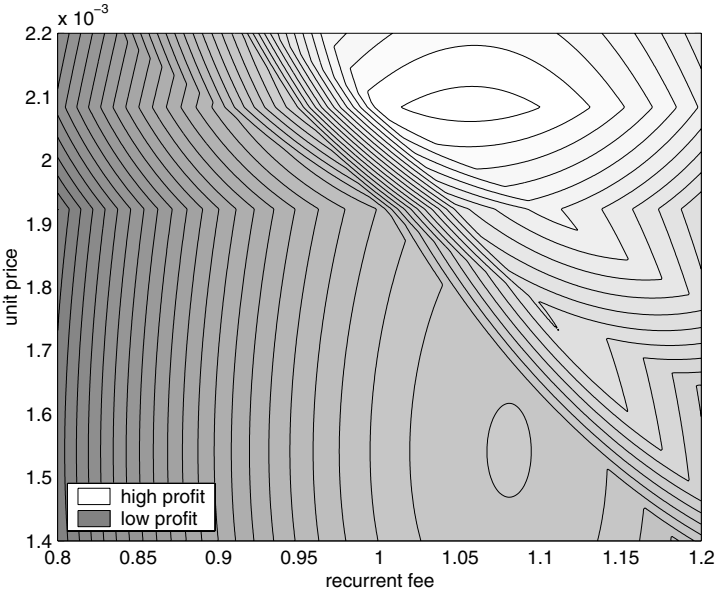


Fig. 5. Profit against the recurrent fee and the unit price

- COMPLEX, being similar in its workings to CRS2 (point pool, reflections), differs in that it has a smaller number of maintained points and converges less surely to global optimum. Instead, it supports complicated and implicitly defined constraints and performs better than CRS2 near the optimum. For the method details, see [1].
- Proprietary routine performs at each step the linearisation of the objective function and constraints, and solves the linear subproblem. Then, based on the past linear subproblems solutions, it computes the next point where the linear subproblem is solved. The routine is the result of QOSIPS participant's long-term experience in the field of optimisation: it incorporates amendments that improve the efficiency, prevent preliminary termination etc.

A series of price optimisations has been run for all the above solvers, from various starting points. The starting points are shown in Fig. 6a) and the optimum prices found by CRS2, COMPLEX and the proprietary solver are indicated in Figs. 6b-d), respectively. Axes descriptions have been omitted for clarity — they are the same as in Fig. 5. The best result for each solver has been put in Table 2.

The CRS2 and COMPLEX routines perform best — they yield optimal or almost optimal value of the performance function. They both converge globally, but CRS2 does not converge exactly to the minimum as the COMPLEX does — which corresponds to their general characteristics. On the other hand, the proprietary method solutions depend on the starting point — the method sticks sometimes at the local minimum at $[1.0731, 0.00154]$ yielding the profit value

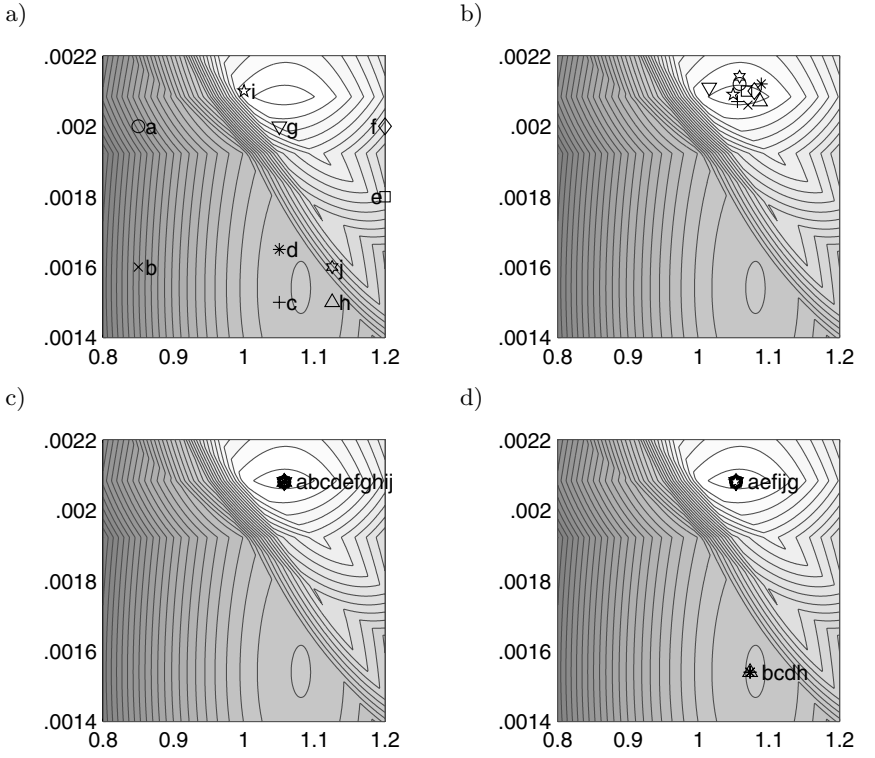


Fig. 6. a) Optimisation starting points (common for all tested solvers) **b)** Solutions found by CRS for the corresponding starting points **c)** Solutions found by COMPLEX for the corresponding starting points **d)** Solutions found by the proprietary routine for the corresponding starting points

88.85. The method does very well with gradient estimations even at the surface irregularities, but due to its local character it was not further considered for practical applications by the R&D team.

Table 2. Academic example — the best optimisation results for each tested solver

Routine	Recurrent fee	Unit price	Profit
CRS2	1.0494	0.00209	90.35
COMPLEX	1.0568	0.00208	90.37
proprietary	1.0528	0.00208	90.35

4 End User Model

The Academic team's perception of the potential problems in simulation and optimisation phases was not shared by the end user — the NSP and the team responsible for model tuning, deployment and field testing. Not having enough real life data concerning sales and network utilisation, they have constructed a model based on their long-term experience in product pricing in other fields. There, the following possible difficulties disappeared:

- **sharp edges of the performance function contours** caused by rapidly growing QoS paybacks — because the network QoS degradation model saturated slowly, thus facilitating gradient estimation;
- **multiple optima** — because model parameters made the performance function unimodal.

Two simple two-dimensional sections of the performance function surface are shown in Fig. 7a) and Fig. 7b). In this example there are 22 price variables being subject to the optimisation. The sections demonstrate that the function grows steadily, making the optimisation problem like as in linear programming, as the optima are always located in a corner of the feasible polyhedron set — the optimisation domain. In such circumstances the former winning routines like CRS2 or COMPLEX may promptly get stuck in a corner remote from the solution. On the other hand, the proprietary algorithm, once designed to handle similar pricing problems, performs well.

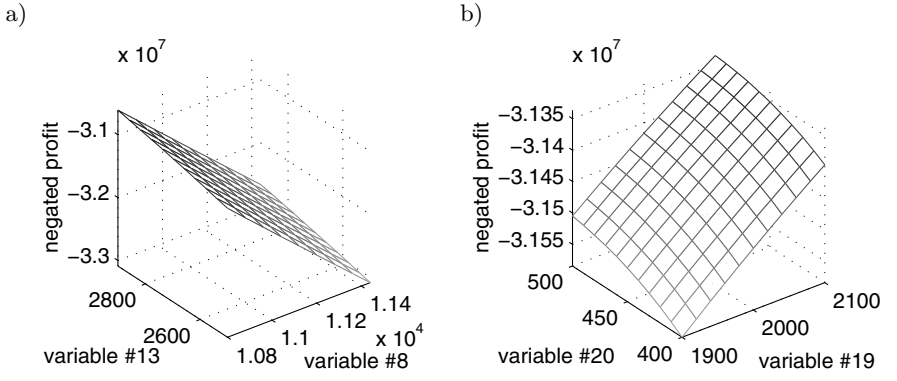


Fig. 7. a) Profit against variables #8 and #13 **b)** Profit against variables #19 and #20

There was another reason for CRS2 and COMPLEX failures — the way the linear equality constraints were defined by the user. The user was accustomed to introduce them by defining pairs of nonequality constraints, which was his habit for former products. The result was a zero-measure search domain with an almost

empty interior, unsupportable by the direct search methods. The proprietary algorithm, apt for this type of constraint definition, worked undisturbed.

5 Conclusions

As the case study shows, despite the model being defined and working properly, most of its behaviour towards the optimisation routine is defined by the model parameters. Correct (i.e. based on the realistic data) values of those parameters are usually not known to the R&D team that implements the model and plugs it into an optimisation solver. Moreover, they may not even be known for the end user, until the real data arrive.

The suggested strategy for R&D teams is then as follows:

1. Assess qualitatively what kinds of optimisation problems a given model may generate.
2. Consider a moderate set of well-tested optimisation algorithms appropriate for those problems.
3. Design the simulation/decision support module so that the user may choose which optimisation routine to utilise, and to use a solution found by one optimisation routine as a starting point for another.
4. Acquaint the user with the capabilities of the module, to change old habits that may decrease the effectiveness of the model and the optimisation solver.

Acknowledgments. This research was carried out within the IST Project 1999-20003 "Quality of Service and Pricing Differentiation for IP Services" (QOSIPS) of the 5th Framework Programme of the European Union.

The authors thank Xiao-Jun Zeng and Vicky Tsamadia from KSSG Ltd. for their fruitful cooperation in research, testing and in the application of the described software.

References

1. Box, M.J.: A new method of constrained optimisation and a comparison with other methods. The Computer Journal **8**:1 (1965) 42–52
2. Lilien, G.L., Kotler, P., Moorthy, K.S.: Marketing Models. Prentice Hall, Upper Saddle River (1992)
3. Price, W.L.: Global optimization algorithms for a CAD workstation. Journal of Optimisation Theory and Applications **55**:1 (1987) 133–146
4. Simon, H.: Price Management. Elsevier Science Publishers, Amsterdam (1989)
5. QOSIPS Deliverable 3.2.1: Research report on how pricing model, learning and optimization are to be implemented to satisfy requirements. (2001)
6. QOSIPS Deliverable 6.7.2: Pricing for IP networks and Services. (2001)
7. Arabas, P., Kamola, M., Malinowski, K., Małowidzki, M.: Pricing for IP Network and Services. Information Knowledge Systems Management **3**:2 (2003) (to appear)

Prototype Implementation for the Analysis of SIP, RSVP and COPS Interoperability

Tien Van Do, Barnabás Kálmán, Csaba Király, and Zsolt Pándi

Department of Telecommunications, Budapest University of Technology and Economics,
Pf. 91., 1521 Budapest, Hungary
{do,cskiraly,pandi}@hit.bme.hu
http://www.hit.bme.hu/

Abstract. The All-IP network concept with end-to-end QoS provisioning has received particular attention in 3GPP recently. This paper describes a prototype implementation for the analysis of the IP Multimedia Subsystem from the aspect of call control, resource reservation and network policing interoperability. The experiences based on a prototype implementation are reported to support the future implementations. It is worth emphasizing that the considered architecture is general enough to be applied in fixed IP networks as well.

1 Introduction

The future of mobile communications is just being defined in the framework of the 3rd Generation Partnership Project (3GPP), which organizes worldwide research on the wireless standardization. The standardization of the Universal Mobile Telecommunications System is still in progress, and the intentions of the standardization body are represented in the evolving standards [1], [2]. Presently 3GPP is aiming at the All-IP network concept as a final target. However, one of the most serious issues that still need to be resolved is end-to-end QoS (Quality of Service) provisioning. To provide QoS an appropriate networking technology must be selected with appropriate signaling protocols and mechanisms which implement the following major functions: call control, QoS architecture for resource reservation and network policing.

Recently, SIP proposed for signaling in IP networks has gained strength due to its flexibility and scalability [3,4]. SIP is used in various multimedia services and originates from the IP world, that is, it incorporates concepts and design patterns characteristic of well-known protocols applied in the Internet, such as the Hypertext Transfer Protocol (HTTP) [13-21].

As far as QoS architecture for resource reservation is regarded, the Internet Engineering Task Force (IETF) has elaborated two fundamental service architectures for QoS provisioning in IP networks: the Integrated Services (IntServ) architecture and the Differentiated Services (DiffServ) architecture ([5,6,7,8,9]). 3GPP standards propose the use of both architectures.

W. Burakowski et al. (Eds.): Art-QoS 2003, LNCS 2698, pp. 88–99, 2003.

Network policy control is inevitable in QoS provisioning. This is generally related to authentication, authorization and accounting (AAA), as well as resource management and call admission control. Service requests of users must be either accepted or refused based on several policy rules. Then this decision must be executed and adhered to at the appropriate network devices. The Common Open Policy Service (COPS) is a promising candidate protocol for communication on policy decisions between the so called Policy Decision Points (PDPs) and Policy Enforcement Points (PEPs) [10]. It has been defined so that it could be applied with RSVP, as well [11]. 3GPP standards propose COPS for the communication between PDPs and PEPs.

Although the functional components of the IMS and their operation have already been defined to a certain extent, the necessary interoperability of call control, resource reservation and network policing has not yet been covered. Therefore, it is necessary to have a prototype implementation to investigate the interoperability issues. This paper reports our prototype implementation for this purpose. Based on the prototype implementation further conclusions were drawn and reported in [22]

The rest of the paper is organized as follows. Section 2 gives an overview of the IMS architecture and considers some issues related to the prototype implementation. Section 3 discusses the prototype implementation. Section 4 reports the test of the prototype implementation. Finally Section 5 concludes the paper.

2 Interoperation of SIP, RSVP, and COPS in the IMS Architecture

A general scenario for the application layer signaling in the IMS (IP Multimedia Subsystem) architecture is illustrated in Fig. 1. According to [2,12] the most important functional elements in the IMS are the Gateway GPRS Support Node (GGSN), the Proxy-Call Session Control Function (P-CSCF) and the User Equipment (UE). Within the P-CSCF there are two fundamental functional elements: a local SIP proxy and a Policy Control Function (PCF).

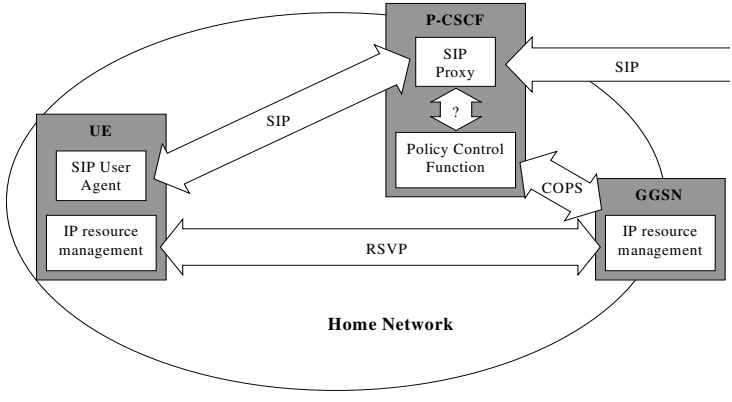


Fig. 1. Application layer signaling scenario in the IMS

Reference [2] discusses several possible scenarios where these functional elements have different capabilities. For example UEs may support RSVP signaling or DiffServ edge functions, but they do not have to have IP bearer service management functionality at all.

Assuming a scenario where UEs are RSVP-capable, that is, RSVP signaling is end-to-end and where GGSNs are not transparent forwarders of RSVP messages (scenario 4 in Annex A of [2]) the components of the architecture must have the following functionality. The *IP resource management function in the UE* is responsible for QoS requests using a suitable protocol (e.g. RSVP), whereas the *IP resource management function in the GGSN* must contain IP policy enforcement and DiffServ edge functionality. The *PCF* in the P-CSCF communicates with the GGSN through the Go interface, which is used for transmitting policing related data and policy decisions between the two entities. COPS is proposed for use in the Go interface. The P-CSCF also contains *SIP Proxy* functionality to be able to track current SIP calls and thus make appropriate policy decisions about resource reservation requests. However, the interface between the local SIP Proxy and the PCF is still undefined in the standards.

2.1 Policy Control Function

As it was already mentioned, the PCF collects all the necessary call parameters from SIP and RSVP signaling and decides whether the resource reservation request of the user may proceed. For making the policy decision the PCF must use a Policy Information Base (PIB), that contains identification information and service contract details for each user (user profile). Moreover, the PCF must have a predefined set of decision rules, and another database describing resource requirements of different codecs in terms of RSVP parameters.

2.2 SIP Proxy Server

The SIP proxy is primarily applied in the PCF to extract SIP session and SDP codec data from SIP call flows. This additional function is a relatively simple extension of the standardized SIP proxy function set.

Reference [15] defines two types of the SIP proxy server behavior, namely transaction stateful and transaction stateless proxies. The former should track transaction state (but not necessarily call state) whereas the latter does not keep state information. Since SIP and SDP specifications have been changing rapidly during the last year, the proposal for a stateless proxy is more preferable for the following reasons:

- The proxy does not generate SIP messages, so the SIP signaling can be end-to-end.
- A stateless proxy can simply forward messages without following transactions and dependencies between messages, thus implementation and future adoption according to the non-final standard is easier.
- Better scalability can be achieved with the stateless design.

However, there are some counterarguments to consider:

- The proxy can only perform syntactic check on SIP messages, but it cannot force the correct order of SIP messages. The latter could be implemented in the PCF but

it would violate our goal to separate responsibilities, since the PCF itself does not take part in SIP signaling.
- The proxy can not filter out repeated (retransmitted) SIP messages. Retransmitted SIP messages, however, should be identical, so in case of repeated SIP messages the stateless proxy sends repeated messages to the PCF as well, which can easily be recognized.
- The stateless proxy cannot support advanced functions like forking proxy mode. This restriction can be relaxed if necessary, since another general-purpose SIP Proxy can be chained after the stateless proxy.
- Implementing a SIP authentication scheme stronger than "HTTP Basic" authentication can be more complicated.

A stateful proxy does not have these problems, however, its implementation and maintenance requires far more efforts than that of a stateless proxy, due to the fact that changes in SIP and SDP standards must be followed.

2.3 Protocol between the SIP Proxy and the PCF

As it was previously noted, the communication protocol between the PCF and the SIP Proxy is not yet standardized, therefore we had to elaborate its details. In this paper we restrict ourselves to review some considerations and decisions without describing the protocol syntax.

The PCF has to receive enough information from the SIP Proxy to
- identify the user,
- recognize protocol messages belonging to the same session,
- calculate RSVP Envelopes based on media and codec information in the SDP offer/answer pair,
- couple COPS requests with the corresponding RSVP Envelope.

2.3.1 Protocol Messages and Statefulness of the SIP Proxy

Assuming that the SIP Proxy is stateless, it cannot gather information about the INVITE transaction or call state, thus a message must be sent to the PCF every time a SIP message contains any information necessary for the decision. These SIP messages are INVITE, 183, PRACK, COMET, 200 messages for call setup and BYE, CANCEL for ending the session.

When communicating with the PCF the protocol messages sent by the SIP proxy must contain data from the *to*, *from*, *call-id*, and *cseq* fields of the SIP message to facilitate user and call identification and authentication. At the same time, RSVP Envelope calculation requires the transmission of SDP body copied from the SIP message, as well. The latter provides enough information also to match COPS requests and RSVP Envelopes. To extract this information from the forwarded SIP messages the SIP Proxy has to do only minimal processing on SIP messages and it can handle SDP data transparently, which may simplify the proxy implementation even further.

A stateful proxy, on the other hand, may gather all the SIP and SDP information necessary for the decision and forward it in a single message. However, this delays the decision process significantly, as the PCF can only consult the PIB when it receives

the message from the SIP Proxy. Therefore, it is advantageous to forward different pieces of information as soon as they become available.

2.3.2 Transport Protocol

The transport protocol used for the transmission of the Proxy-PCF protocol messages must be reliable and it should be non-blocking to allow for the easy implementation of a stateless proxy. A persistent connection should be set up between the SIP Proxy and the PCF, thus the TCP protocol is a straightforward choice for this purpose. If, however, the Proxy and the PCF entities are co-located in the same network device, or they are components of the same software, then any other means of non-blocking and reliable data transfer might be suitable.

2.3.3 Enhancements via Feedback

The protocol messages mentioned so far are all uni-directional (the SIP Proxy notifies the PCF). All the necessary functionality can be implemented with this uni-directional protocol, although bi-directional communication would facilitate:

- the modification of SDP and SIP parameters in view of the user profile,
- user notification via SIP in the following cases:
  - The SIP session could be ended in case of an RSVP error, which is of particular importance if the error is between the UE and the GGSN, and the UE loses the connection with the GGSN (which may happen in mobile networks).
  - Authentication parameters could be forwarded to the UE.
  - Information calculated from the user profile and the SDP could be forwarded to the UE.
  - If the authentication fails the UE could be instantly notified, as opposed to the present situation, where it will only be notified about rejected PATH messages. (The latter results in superfluous delay, unnecessarily reserved resources and the fact that the other end will also begin to set up the RSVP session.)

Nevertheless, those opting for utilizing the advantages of bidirectional communication must be aware of the fact, that interacting in the SIP session this way breaks the end-to-end nature of SIP signaling and it is not conformant to the current SIP standard.

3 Prototype Implementation

In order to be able to analyze the IMS we implemented a SIP User Agent (UA), a SIP Proxy, and a COPS Policy Decision Point (PDP) software with the appropriate functionalities listed in Table 1. These requirements were derived from [1] and [2], and will be detailed later on. As the implementation of the GGSN a commercially available and widespread IP router was used.

The DiffServ functionality in the GGSN was not used due to the following reasons. Firstly, in the focus of interest there are the requirements for the Home Network and not the IP cloud beyond the GGSN. Moreover, [8] already demonstrated how RSVP

might be used over a DiffServ domain; therefore, in the prototype implementation it is irrelevant whether RSVP or DiffServ marking ensures QoS between GGSNs.

Table 1. Functional elements of the implementation

IMS function	Implementation	Remark
UE	SIP User Agent software	must support end-to-end SIP and RSVP signaling for call control and resource reservation
GGSN	Router	must contain RSVP and COPS PEP functionality
P-CSCF/ SIP Proxy	SIP Proxy software	must be able to provide the COPS PDP with session data
P-CSCF/ PCF	COPS PDP software	must be capable of making policy decisions based on SIP session information and a priori configuration data

3.1 User Agent

None of the freely available SIP state machine source code packages satisfied completely our expectations. Among their problems multiple thread usage, lack of complete SIP RFC compliance and lack of integrated RSVP support must be mentioned. The only available solution capable of supporting RSVP was VOCAL of Vovida [23] at the time of the implementation. However, RSVP support was not fully integrated into the state machine and the SIP RFC was also violated, because e.g. message resending was not implemented.

Therefore we decided to implement the whole UA from scratch. As a consequence, the subtasks of the implementation were the following:

- state machine improved with RSVP usage and COMET and PRACK messages,
- SIP parser,
- RTP flow handling.

We specified the state machine for the UA implementation (due to the limited space we will not report the state machine here). The Prolog language seemed to be a perfect choice for coding the state machine and the SIP parser due to the pattern match applied therein.

For network handling and the user interface, however, C++ proved to be a better selection.

Therefore the UA consists of two modules:

- the overall control and SIP message parsing is the task of a hidden Prolog module,
- while the communication is carried out by a C++ shell.

The two modules communicate with each other via short text messages sent through pipes. The user interface is a simple command-prompt like textual interface, where the messages of the UA are also shown.

3.2 SIP Proxy

The proxy is a stateless proxy with noop location service. It sends a small report of certain SIP messages to the PDP (Table 2). The address of the PDP must be set in the configuration file. The content of the report depends on the type of the message. The proxy assumes the INVITE message contains an SDP offer, the 183 message contains the SDP answer, and there are no more modifications of the session.

Route and Record-Route headers are ignored. Otherwise the proxy follows rfc2543bis-9. The optional loop detection is not implemented, the Max-Forwards mechanism is used to prevent loops.

Table 2. Messages between the SIP proxy and the PDP

SIP message	preprocessed message
INVITE	i,<From>,<To>,<Call-Id>,<IP>,<Port>
183	m,<From>,<To>,<Call-Id>,<IP>,<Port>,<Codecs>
CANCEL	b,<From>,<To>,<Call-Id>
BYE	b,<From>,<To>,<Call-Id>

3.3 PDP

The PDP is an event-driven OO program written in C++. The structure of the program is determined by its one-threaded design, the connections it should mountain and the pool of independent state machines handling client sessions. Fig. 2. shows the class diagram of the PDP.

As it can be seen from the figure, the PDP has five type of external connections. To support future improvement and changes (both architectural changes and changes in the protocols used) the PDP was decomposed to the following modules:

1. Framework for single thread operation
2. COPS module
3. Proxy connection module
4. User profile module
5. Codec parameter module
6. Event distribution
7. State machine (for each session)

The framework is able to handle several sockets and timers at the same time thus enabling the single threaded PDP to receive events from both the Proxy and the Edge Routers. It supports scheduling of inputs, reading from and writing to TCP streams or UDP datagrams, and accepting TCP connections.

Scheduling is performed by the World class. It receives file descriptor and timer registrations and notifies the appropriate object if the file descriptor is readable/writable or the timer is elapsed. The Porter class is responsible for handling incoming TCP connections, while FdReader and FdWriter classes handle established TCP connections and UDP sockets.

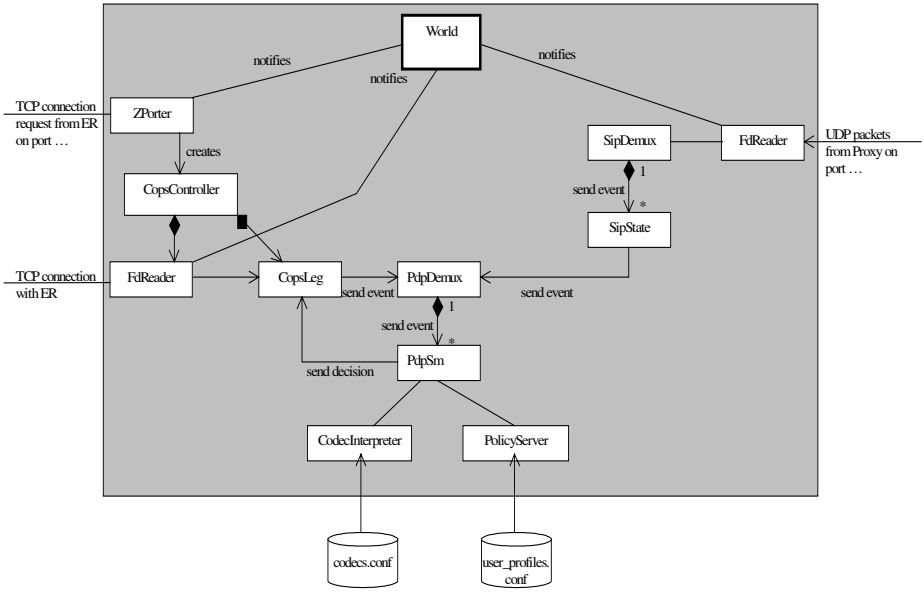


Fig. 2. PDP class diagram

In the PDP the framework is used for the following tasks:

1. The ZPorter object handles incoming TCP connection requests on the COPS port. For each incoming request a new CopsController object is created. This allows the PDP to connect to several Edge Routers at the same time.
2. A CopsController object creates an FdReader for reading from the TCP connection and a CopsLeg object for interpreting COPS messages.
3. The communication between the PDP and the Proxy uses UDP datagrams thus connection setup is not needed. An FdReader object reads the datagrams arriving to the pre-defined port.
4. These datagrams are handed to the SipDemux object for parsing and interpretation.

The PDP is capable of accepting connections of an arbitrary number of PEPs for client type 1 (as defined in [11]) and it maintains the connections by echoing the keepalive messages sent by the PEPs. For PEPs trying to register a different client type the PDP sends a client close message.

The PDP sends a positive decision in reply to any request that does not contain a PATH message, except for configuration requests, which are not supported. (See context object R-Type in [10].) PATH messages are checked whether they contain the mandatory and the necessary optional RSVP objects embedded in their client SI, and an appropriate answer is sent if not.

When all data is available for making the decision, the PDP sends a decision message to the PEP. However, the PEP will not always get a decision, as the presence of

necessary data cannot be guaranteed. In such cases the PDP should reply with a negative decision before the keepalive timer expires, nevertheless it has not been implemented yet.

As for shutting down the PDP, the PEPs can only perceive this event through the closing of the TCP connection.

The ZPorter, the CopsController, the CopsLeg and the FDReader objects are used in the PDP to implement COPS Usage functions. The ZPorter listens at port 3288 (assigned to COPS by the IANA) and accepts all incoming connections, for each of which it creates a CopsController instance. The CopsController exists as long as the TCP connection with the PEP is open. This object is in charge of creating object instances to handle COPS communication with the PEP. It creates an FDReader, assigns it to the fd of the TCP connection received from ZPorter, and links the FDReader to a CopsLeg. The FDReader is notified about incoming messages on the TCP connection and it simply forwards all the received data to the CopsLeg.

CopsLeg interprets incoming messages and handles COPS level dialogs. It recognizes COPS messages and processes them alone, except for messages that request a decision on a PATH message. All the necessary data is extracted from such messages, and then the PdpDemux is contacted for asking decision, which will finally result in contacting a PdpState. The CopsLeg is later on called by a PdpState when all the necessary data arrived and the decision is ready. In this case the CopsLeg is only used for constructing and sending the decision message.

The PdpDemux must be notified also when a delete request message arrives or when the PEP closes the TCP connection. In such cases the references to the CopsLeg object to be destroyed must be removed from all of the involved PdpStates.

For constructing and interpreting COPS messages the CopsLeg applies the COPS stack implemented by VOVIDA in VOCAL [23]. For RSVP message processing both original and adapted code were used from USC Information Sciences Institute's RSVP implementation.

4 Testing

In order to test the functionality a minimal test configuration depicted in Fig. 3 can be used. Since we do not intend to test the performance, only a simple network is needed. The configuration consists of two LANs connected through a backbone link with a capacity of 2 Mbits/s between two routers. Two User Agent instances were used during the tests, one on each LAN. The PDP was run on the same machine to decrease the number of computers allocated for the tests.

Table 3 lists the basic scenarios identified to test the functionality of the system.

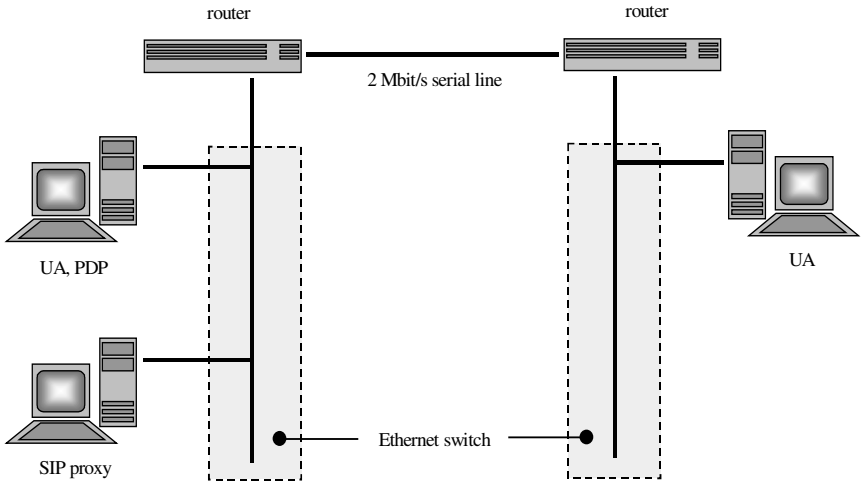


Fig. 3. Configuration used for tests

Table 3. Test scenarios

Test description	Expected outcome
outgoing call, the caller is known and he terminates the call	successful call setup and termination
outgoing call, the caller is known, the callee terminates the call	successful call setup and termination
too large resource allocation request for a known user (user profile violation)	the call is refused by PdpSm, the router is directed to reject the PATH messages
both the caller and the callee are unknown (no user profile available)	call is refused by PdpSm, the router is directed to reject the PATH messages
resource allocation request is refused due to lack of resources (or router settings)	call is accepted by PdpSm, but the router rejects the PATH messages
too large resource allocation request for a set of codecs (codec limit violation)	call is refused by PdpSm, the router is directed to reject the PATH messages

The performed tests proved that the system works correctly in the test scenarios.

5 Conclusions

This paper reports the experiences and proposal for the prototype implementation to investigate the interoperability of SIP, RSVP and COPS in the IMS architecture. Our work was guided by several RFCs and drafts, in fact, sometimes there were too many of them providing contradictory solutions for the same problems. Another difficulty

was the fact that internet drafts keep changing and some of those that we adhered to had been thoroughly reworked by the time the implementation was finished.

Nevertheless, a solution was proposed with a stateless SIP proxy. Based on our experiences we can conclude that tracking the SIP state in the PDP does not give extra information that is useful when making decisions; therefore, the decision about the SIP stateless/stateful PDP could be a function of the features of the SIP proxy.

The implemented PDP and SIP proxy uses a unidirectional communication, as implementing the listed advantages deriving from bidirectional communication was again out of the scope. There are, however, situations, as mentioned above, where SIP would be a more useful means for user notification than RSVP.

Acknowledgements. This work was supported by NOKIA Hungary. The authors would like to express their gratitude to György Wolfner of NOKIA for the valuable discussions. We would like to thank also Dóra Erös of NOKIA for her continous support during the course of this work.

References

1. 3GPP TS 23.228 (V5.5.0): IP Multimedia Subsystem (IMS). June 2002
2. 3GPP TS 23.207 (V5.4.0): End-to-End QoS Concept and Architecture. June 2002
3. Schulzrinne, H., Rosenberg, J.: The Session Initiation Protocol: Internet Centric Signaling. IEEE Communications Magazine, October 2000, p. 134
4. Canal, G., Cuda, A.: Why SIP will Pave the way towards NGN. In Proceedings of the 7 th ITU International Conference on Intelligence in Networks, October 2001
5. Braden, R. et al.: Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification. RFC 2205, September 1997
6. Herzog, S.: RSVP Extensions for Policy Control. RFC 2750, January 2000
7. Blake, S. et al.: An Architecture for Differentiated Service. RFC 2475, January 2000
8. Detti, A. et al.: Supporting RSVP in a Differentiated Service Domain: an Architectural Framework and a Scalability Analysis. International Conference on Communications, Vancouver (Canada), June 1999
9. Bernet, Y. et al.: A Framework for Integrated Services Operation over Diffserv Networks. RFC 2998, November 2000
10. Durham, D. et al.: The COPS (Common Open Policy Service) Protocol. RFC 2748, January 2000
11. Herzog, S. et al.: COPS Usage for RSVP. RFC 2749, January 2000
12. Lin, Y., Pang, A., Haung, Y., Chlamtac, I.: An All-IP Approach for UMTS Third-Generation Mobile Networks. IEEE Network, vol. 16, no. 5, September 2002
13. Rosenberg, J. et al.: An Offer/Answer Model with SDP. draft-rosenberg-mmusic-sdp-offer-answer-00 (formerly RFC2543 Appendix B), October 2001
14. Handley, M. et al.: SIP: Session Initiation Protocol. RFC 2543, March 1999
15. Handley, M. et al.: SIP: Session Initiation Protocol. draft-ietf-sip-rfc2543bis-05, October 2001

16. Rosenberg, J. et al.: SIP: Session Initiation Protocol. RFC 3261 (formerly draft-ietf-sip-rfc2543bis-09), June 2002
17. Camarillo, G. et al.: Integration of Resource Management and SIP. draft-ietf-sip-manyfolks-resource-02, February 2001
18. Johnston, A. et al.: SIP Call Flow Examples. draft-ietf-sip-call-flows-05, June 2001
19. Rosenberg, J. et al.: Reliability of Provisional Responses in the Session Initiation Protocol (SIP). RFC 3262, June 2002
20. Rosenberg, J. et al.: An Offer/Answer Model with the Session Description Protocol (SDP). RFC 3264 (formerly draft-rosenberg-mmusic-sdp-offer-answer-00), June 2002
21. Handley, M. et al.: SDP: Session Description Protocol. RFC 2327, April 1998
22. Cs. Király, Zs. Pándi, T. V. Do: Analysis of SIP, RSVP and COPS Interoperability. The 2nd international workshop on QoS in Multiservice IP networks (QOS-IP 2003), Milano, Italy
23. The VOVIDA Homepage: http://www.vovida.org

Reinforcement Learning as a Means of Dynamic Aggregate QoS Provisioning*

Nail Akar and Cem Sahin

Electrical and Electronics Engineering Dept., Bilkent University 06800 Bilkent, Ankara, Turkey
{akar,csahin}@ee.bilkent.edu.tr

Abstract. Dynamic capacity management (or dynamic provisioning) is the process of dynamically changing the capacity allocation (reservation) of a virtual path (or a pseudo-wire) established between two network end points. This process is based on certain criteria including instantaneous traffic load for the pseudo-wire, network utilization, hour of day, or day of week. Frequent adjustment of the capacity yields a scalability issue in the form of a significant amount of message distribution and processing (i.e., signaling) in the network elements involved in the capacity update process. We therefore use the term "signaling rate" for the number of capacity updates per unit time. On the other hand, if the capacity is adjusted once and for the highest loaded traffic conditions, a significant amount of bandwidth may be wasted depending on the actual traffic load. There is then a need for dynamic capacity management that takes into account the tradeoff between signaling scalability and bandwidth efficiency. In this paper, we introduce a Markov decision framework for an optimal capacity management scheme. Moreover, for problems with large sizes and for which the desired signaling rate is imposed as a constraint, we provide suboptimal schemes using reinforcement learning. Our numerical results demonstrate that the reinforcement learning schemes that we propose provide significantly better bandwidth efficiencies than the static allocation policy without violating the signaling rate requirements of the underlying network.

1 Introduction

In this paper, dynamic capacity management refers to the process of dynamically changing the capacity reservation of a VP (Virtual Path) set up between two network end points based on certain criteria including instantaneous traffic load for the virtual path, network utilization, hour of day, or day of week. We use the terms "virtual path" and "pseudo-wire" synonymously in this paper to define a generic path carrying aggregate traffic with Quality of Service (QoS) between two network end points. The route of the virtual path is fixed and the capacity allocated to it can dynamically be resized on-line (without a need for tearing it

* This work is supported by The Scientific and Technical Research Council of Turkey (TUBITAK) under grant EEEAG-101E048

W. Burakowski et al. (Eds.): Art-QoS 2003, LNCS 2698, pp. 100–114, 2003.

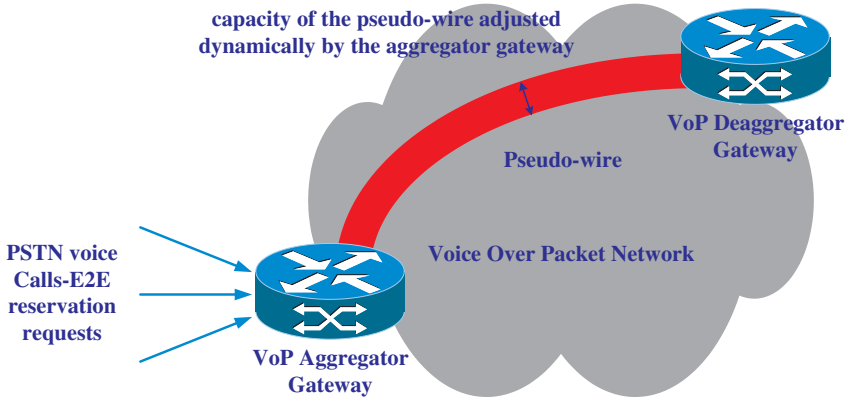


Fig. 1. E2E (End-to-End) reservations due to PSTN voice calls are aggregated into one single reservation through the voice over packet network

down and reestablishing it with a new capacity) using signaling. With this generic definition, multiple networking technologies can be accommodated; a virtual path may be an MPLS-TE (MultiProtocol Label Switching - Traffic Engineering) LSP (Label Switched Path) [6], an ATM (Asynchronous Transfer Mode) VP [1], or a single aggregate RSVP (Resource ReserVation Protocol) reservation [2]. The end points of the virtual path will then be LSRs (Label Switch Router), ATM switches, or RSVP-capable routers, respectively.

We are motivated in this paper by "voice over packet" networks where individual voice calls are aggregated into virtual paths in the packet-based network, although the methodology proposed in this paper is more general and is amenable to dynamic capacity management for non-voice scenarios as well. Figure 1 depicts a general "voice over packet" network. At the edge of the packet network, there are the voice over packet gateways which are interconnected to each other using virtual paths or pseudo-wires. The packet network may be an MPLS, or an ATM, or a pure IP network supporting dynamic aggregate reservations. In this scenario, end to end reservation requests that are initiated by PSTN (Public Switched Telephone Network) voice calls and which are destined to a particular voice over packet gateway arrive at the aggregator gateway. These reservations are then aggregated into a single dynamic reservation through the packet network. The destination gateway then deaggregates these reservations and forwards the requests back to the PSTN.

An aggregate of voice calls flows through the pseudo-wire in Figure 1. This enables possible aggregation of forwarding, scheduling, and classification state through the packet network, thus enhancing the scalability of core routers and switches. The capacity allocated to the aggregate should ideally track the actual aggregate traffic for optimal use of resources but this policy requires a substantial amount of signaling rates and it would not scale to large networks with rapidly changing traffic. For example, consider two "voice over packet" gateways inter-

connected to each other using a pseudo-wire. Calls from the PSTN are admitted into the pseudo-wire only when there is enough bandwidth and once admitted, traffic is packetized and forwarded from one gateway to the other in which it will be depacketized and forwarded back to the PSTN. Every time a new voice call arrives or an existing call terminates, the capacity of the pseudo-wire may be adjusted for optimal use of resources. This approach will be referred to as the SVC (Switched Virtual Circuit) approach throughout this paper since the messaging and signaling requirements of this approach will be very similar to the case where each voice call uses its own SVC as in SVC-based ATM networks. Another approach to engineer the pseudo-wire is through allocating capacity for the highest load over a long time window (e.g., 24-hour period). This approach would not suffer from signaling and message processing requirements since each capacity update would take place only once in a very long time window. Motivated by ATM networks, we call this approach the PVP (Permanent Virtual Path) approach. However, the downside of this approach is that the capacity may be vastly underutilized when the load is significantly lower than the allocated capacity, which is the peak load. In this case, this idle capacity would not be available to other aggregates that actually need it and this would lead to inefficient use of resources.

In this paper, we propose the DCM (Dynamic Capacity Management) approach with two different formulations. In the first formulation, we assign a cost for every capacity update (denoted by S) and a cost for allocated unit bandwidth per unit time (denoted by b). This formulation is amenable to solution using the traditional average cost "Markov decision" framework [16] which has been a popular paradigm for sequential decision making under uncertainty. Such problems can be solved by Dynamic Programming (DP) [16] which provides a suitable framework and algorithms to find optimal policies. Policy iteration and relative value iteration [16] are the most commonly used DP algorithms for average cost Markov decision problems. However, these algorithms become impractical when the underlying state-space of the Markov decision problem is large, leading to the so-called "curse of dimensionality". Recently, an adaptive control paradigm, the so-called "Reinforcement Learning" (RL) [15], [3] has attracted the attention of many researchers in the field of Markov decision processes. RL is based on a simulation scenario in which an agent learns by trial and error to choose actions that maximize the long-run reward it receives. RL methods are known to scale better than their DP counterparts [15]. In such problematic large sized problems, we show in this paper that reinforcement learning-based solutions are feasible for finding suboptimal dynamic capacity management policies in virtual path-based networks.

The second drawback of the Markov decision formulation is due to the practical limit to the number of capacity updates per unit time per pseudo-wire, a constraint which cannot be converted easily to a cost parameter per capacity update. For example, let us assume that the network nodes in the aggregation region can handle at most N capacity update requests per hour, which is the scalability requirement. Assuming that on the average there are I output in-

terfaces on every node and L pseudo-wires established on every such interface, an individual pseudo-wire may be resized on the average $N/(IL)$ times in every hour. With typical values of $N = 36000$ (10 capacity updates per second for an individual network node), I=16, and L=100, one can afford adjusting the capacity of each pseudo-wire 22.5 times in an hour. The goal of our second DCM formulation is to minimize the idle capacity between the allocated capacity and the actual bandwidth requirement over time while satisfying the scalability requirement, i.e., by resizing the capacity of the pseudo-wire less than 22.5 times per hour. We propose a novel reinforcement learning based-scheme to find suboptimal solutions to this constrained stochastic optimization problem.

There are several other techniques proposed in the literature to solve the dynamic capacity allocation problem. In [14], the capacity of the pseudo-wire is changed at regular intervals based on the QoS measured in the previous interval. A heuristic multiplicative increase multiplicative decrease algorithm in case of stationary bandwidth demand gives the amount of change. If the bandwidth demand exhibits a cyclic variation pattern, Kalman filtering is used to extract the new capacity requirement. In [8], blocking rates are calculated for the pseudo-wire using the Pointwise Stationary Fluid Flow Approximation (PSFFA) and capacity is updated based on these blocking rates. Their approach is mainly based on the principle that if the calculated blocking rate is much less than the desired blocking rate, then the capacity is decreased by a certain amount and it is increased otherwise.

The remainder of the article is organized as follows. In Section II, general QoS architectures including the aggregate reservations concept are reviewed and compared and contrasted with each other in terms of performance and scalability. The Markov decision framework for optimal aggregate reservations as well as a reinforcement learning approach for the two formulations are presented in Section III. Section IV provides numerical examples to demonstrate the efficacy of the proposed approach. The final section is devoted to conclusions and future work.

2 QoS Models

Several QoS architectures that are proposed by the IETF (Internet Engineering Task Force) for IP networks will now briefly be reviewed and how they relate to dynamic capacity management will then be presented.

2.1 Integrated Services

The integrated services architecture defines a set of extensions to the traditional best effort model of the Internet so as to provide end-to-end QoS commitments to certain applications with quantitative performance requirements [17], [13]. An explicit setup mechanism like RSVP will be used in the integrated services architecture to convey information to IP routers so that they can provide requested

services to flows that request them [18]. Upon receiving per-flow resource requirements through RSVP, the routers apply admission control to signaled requests. The routers also employ traffic control mechanisms to ensure that each admitted flow receives the requested service irrespective of other flows. These mechanisms include the maintenance of per-flow classification and scheduling states. One of the reasons that have impeded the wide-scale deployment of integrated services with RSVP is the excessive cost of per-flow state and per-flow processing that are required for integrated services.

The integrated services architecture is similar to the ATM SVC architecture in which ATM signaling is used to route a single call over an SVC that provides the QoS commitments of the associated call. The fundamental difference between the two architectures is that the former typically uses the traditional hop-by-hop IP routing paradigm whereas the latter uses the more sophisticated QoS source routing paradigm.

2.2 Differentiated Services

In contrast with the per-flow nature of integrated services, differentiated services (diffserv) networks classify packets into one of a small number of aggregated flows or "classes" based on the Diffserv Codepoint (DSCP) in the packet's IP header [12], [4]. This is known as Behavior Aggregate (BA) classification. At each diffserv router in a Diffserv Domain (DS domain), packets receive a Per Hop Behavior (PHB), which is dictated by the DSCP. Since diffserv is void of per-flow state and per-flow processing, it is generally known to scale well to large core networks. Differentiated services are extended across a DS domain boundary by establishing a Service Level Agreement (SLA) between an upstream network and a downstream DS domain. Traffic classification and conditioning functions (metering, shaping, policing, and remarking) are performed at this boundary to ensure that traffic entering the DS domain conforms to the rules specified in the Traffic Conditioning Agreement (TCA) which is derived from the SLA.

2.3 Aggregation of RSVP Reservations

In the integrated services architecture, each E2E reservation requires a significant amount of message exchange, computation, and memory resources in each router along the way. Reducing this burden to a more manageable level via the aggregation of E2E reservations into one single aggregate reservation is addressed by the IETF [2]. Although aggregation reduces the level of isolation between individual flows belonging to the aggregate, there is evidence that it may potentially have a positive impact on delay distributions if used properly [5] and aggregation is required for scalability purposes.

In the aggregation of E2E reservations, we have an aggregator router, an aggregation region, and a deaggregator. Aggregation is based on hiding the E2E RSVP messages from RSVP-capable routers inside the aggregation region. To achieve this, the IP protocol number in the E2E reservation's Path, PathTear, and ResvConf messages is changed by the aggregator router from RSVP (46) to

RSVP-E2E-IGNORE (134) upon entering the aggregation region, and restored to RSVP at the deaggregator point. Such messages are treated as normal IP datagrams inside the aggregation region and no state is stored. Aggregate Path messages are sent from the aggregator to the deaggregator using RSVP's normal IP protocol number. Aggregate RESV messages are then sent back from the deaggregator to the aggregator via which an aggregate reservation with some suitable capacity will be established between the aggregator and the deaggregator to carry the E2E flows that share the reservation. Such establishment of a smaller number of aggregate reservations on behalf of a larger number of E2E flows leads to a significant reduction in the amount of state to be stored and the amount of signaling messages exchanged in the aggregation region.

One fundamental question to answer related to aggregate reservations is on sizing the reservation for the aggregate. A variety of options exist for determining the capacity of the aggregate reservation, which presents a tradeoff between optimality and scalability. On one end (i.e., SVC approach), each time an underlying E2E reservation changes, the size of the reservation is changed accordingly but one advantage of aggregation, namely the reduction of message processing cost, is lost. On the other end (i.e., PVP approach), in anticipation of the worst-case token bucket parameters of individual E2E flows, a semipermanent reservation is made. Depending on the actual pattern of E2E reservation requests, the PVP approach, despite its simplicity, may lead to a significant waste of bandwidth. Therefore, a policy is required which maintains the amount of bandwidth required on a given aggregate reservation by taking account of the sum of the bandwidths of its underlying E2E reservations, while endeavoring to change it infrequently. If the traffic trend analysis suggests a significant probability that in the next interval of time the current aggregate reservation will be exhausted, then the aggregator router will have to predict the necessary bandwidth and request it by an aggregate Path message. Or similarly, if the traffic analysis suggests that the reserved amount will not be used efficiently by the future E2E reservations, some suitable portion of the aggregate reservation may be released. We call such a scheme a dynamic capacity management scheme.

Classification of the aggregate traffic is another issue that remains to be solved. IETF proposes that the aggregate traffic requiring a reservation may all be marked with a certain DSCP and the routers in the aggregation region will recognize the aggregate through this DSCP. This solves the traffic classification problem in a scalable manner.

Aggregation of RSVP reservations in IP networks is very similar in concept to the Virtual Path in ATM networks. In this framework, several ATM virtual circuits can be tunneled into one single ATM VP for manageability and scalability purposes. A Virtual Path Identifier (VPI) in the ATM cell header is used to classify the aggregate in the aggregation region (VP switches) and the Virtual Channel Identifier (VCI) is used for aggregation/deaggregation purposes. A VP can be resized through signaling or management.

3 Semi-Markov Decision Framework

A tool to obtain optimal capacity management policies for scalable aggregate reservations is the semi-Markov decision model [16]. This model concerns a dynamic system which at random points in time is observed and classified into a possible number of states. We consider a network as in Figure 1 that supports aggregate reservations. We assume E2E reservation requests are identical and they arrive at the aggregator according to a homogeneous Poisson process with rate λ. We also assume exponentially distributed holding times for each E2E reservation with mean $1/\mu$. In this model, each individual reservation request is identical (i.e., one unit), and we assume that there is an upper limit C_{max} units for the aggregate reservation. We suggest to set C_{max} to the minimum capacity required to achieve a desired blocking probability p. C_{max} is typically derived using $p = EB(C_{max}, \lambda/\mu)$ where EB represents the Erlang's B formula. This ensures that the E2E reservation requests will be rejected when the instantaneous aggregate reservation is exactly C_{max} units. In our simulation studies, we take $p = 0.01$. In this paper, we do not study the blocking probabilities when an attempt to increase the aggregate reservation is rejected by the network due to unavailability of bandwidth.

3.1 Formulation with Cost Parameters (S, b)

In this formulation, we assign a cost for every capacity update (S) and a cost for allocated unit bandwidth per unit time (b). Our goal is to minimize the average cost per unit time as opposed to the total cumulative discounted cost, because our problem has no meaningful discount criteria. We denote the set of possible states in our model by **S**:

$$\mathbf{S} = \{s | s = (s_a, s_r), 0 \leq s_a \leq C_{max}, \max(0, s_a - 1) \leq s_r \leq C_{max}\},$$

where s_a refers to the number of active calls using the pseudo-wire just after an event which is defined either as a call arrival or a call departure. The notation s_r denotes the amount of aggregate reservation before the event. For each $s = (s_a, s_r) \in S$, one has a possible action of reserving $s'_r, s_a \leq s'_r \leq C_{max}$ units of bandwidth until the next event. The time until the next decision epoch (state transition time) is a random variable denoted by τ_s that depends only on s_a and its average value is given by

$$\bar{\tau}_s = \frac{1}{\lambda + s_a \mu} \tag{1}$$

Two types of incremental costs are incurred when at state $s = (s_a, s_r)$ and action s'_r is chosen; first one is the cost of reserved bandwidth which is expressed as $b\tau_s s'_r$ where b is the cost parameter of reserved unit bandwidth per unit time. Secondly, since each reservation update requires message processing in the network elements, we also assume that a change in the reservation yields a fixed cost S. As described, at a decision epoch, the action s'_r (whether to update or

not and if an update decision is made, how much allocation/deallocation will be performed) is chosen at state (s_a, s_r), then the time until, and the state at, the next decision epoch depend only on the present state (s_a, s_r) and the subsequently chosen action s'_r, and are thus independent of the past history of the system. Upon the chosen action s'_r, the state will evolve to the next state $s' = (s'_a, s'_r)$ and s'_a will equal to either $(s_a + 1)$ or $(s_a - 1)$ according to whether the next event is a call arrival or departure. The probability of the next event being a call arrival or call departure is given as

$$p(s'_a \mid s_a) = \begin{cases} \frac{\lambda}{\lambda + s_a\mu}, & \text{for } s'_a = s_a + 1, \\ \frac{s_a\mu}{\lambda + s_a\mu} & \text{for } s'_a = s_a - 1. \end{cases}$$

This formulation fits very well into a semi-Markov decision model where the long-run average cost is taken as the optimality criterion. We propose the following two algorithms for this problem based on [16], [11], and [10].

Relative Value Iteration (RVI). Our approach is outlined below but we refer the reader to [16] for details. A data transformation is first used to convert the semi-Markov decision problem to a discrete-time Markov decision model with the same state space [16]. For this purpose, let $c_s(s'_r)$ denote the average cost until next state when the current state is $s = (s_a, s_r)$ and action s'_r is chosen. Also let $\tau_s(s'_r)$ denote the average state transition time and $p_{s,s'}(s'_r)$ denote the state transition probability from the initial state s to the next state s' when action s'_r is chosen. Average immediate costs and one-step transition probabilities of the converted Markov decision model are given as [16]:

$$\widetilde{c}_s(s'_r) = \frac{c_s(s'_r)}{\tau_s(s'_r)} \tag{2}$$

$$\widetilde{p}_{s,s'}(s'_r) = \frac{\tau}{\tau_s(s'_r)} p_{s,s'}(s'_r), s' \neq s \tag{3}$$

$$\widetilde{p}_{s,s'}(s'_r) = \frac{\tau}{\tau_s(s'_r)} p_{s,s'}(s'_r) + (1 - \frac{\tau}{\tau_s(s'_r)}), s' = s \tag{4}$$

where τ should be chosen to satisfy

$$0 < \tau \leq min_{(s,s'_r)} \tau_s(s'_r)$$

With this transformation, the relative value iteration algorithm is given as follows [16]:

Step 0 Select $V_0(s)$, $\forall s \in \mathbf{S}$, from $0 \leq V_0(s) \leq \min_{s'_r} \widetilde{c}_s(s'_r)$ and $n := 1$.

Step 1a Compute the function $V_n(s)$, $\forall s \in \mathbf{S}$, from the equation

$$V_n(s) = \min_{s'_r} \left[\widetilde{c}_s(s'_r) + \frac{\tau}{\tau_s(s'_r)} \sum_{s'} p_{s,s'}(s'_r) V_{n-1}(s') + (1 - \frac{\tau}{\tau_s(s'_r)}) V_{n-1}(s) \right] \tag{5}$$

Step 1b Perform the following for all $s \in \mathbf{S}$ where s_0 is a pre-specified reference state:

$$V_n(s) := V_n(s) - V_n(s_0) \tag{6}$$

Step 2 Compute the following values

$$M_n = \min_s (V_n(s) - V_{n-1}(s)),$$
$$m_n = \max_s (V_n(s) - V_{n-1}(s)). \tag{7}$$

The algorithm is stopped when the following convergence condition is satisfied

$$0 \leq (M_n - m_n) \leq \varepsilon m_n, \tag{8}$$

where ε is a pre-specified tolerance. This condition signals that there is no more significant change in the value function of the states $\{V_n(\cdot)\}$. If convergence condition is not satisfied, we let $n := n+1$ and we branch to ***Step 1a***. Otherwise, the optimal policy is obtained by choosing the argument that minimizes the right hand side of (5).

Asynchronous Relative Value Iteration (A-RVI). When the state space of the underlying Markov decision problem is large, dynamic programming algorithms will be intractable and we suggest to use reinforcement learning based algorithms in such cases to obtain optimal or sub-optimal solutions. In particular, we propose the asynchronous version of RVI, the so-called Asynchronous Relative Value Iteration (A-RVI) ([11], [10]) that uses simulation-based learning. At a single iteration, only the visited state's value is updated (single or asynchronous updating) instead of updating all the states' values (batch updating). A-RVI is given as follows:

Step 0 Initialize $V(s) = 0$, $\forall s \in \mathbf{S}$, $n := 1$, average cost $\rho = 0$ and fix a reference state s_0, that $V(s_0) = 0$ for all iterations. Select a random initial state and start simulation.

Step 1 Choose the best possible action from the information gathered so far using the following local minimization problem:

$$\min_{s'_r} \left[\widetilde{c}_s(s'_r) + \frac{\tau}{\tau_s(s'_r)} \sum_{s'} p_{s,s'}(s'_r) V(s') + (1 - \frac{\tau}{\tau_s(s'_r)}) V(s) \right] \tag{9}$$

Step 2 Carry out the best or another random exploratory action. Observe the incurring cost c_{inc} and next state s'. If best action is selected, perform the following updates:

$$V(s) := (1 - \kappa_n) V(s) + \kappa_n (c_{inc} - \rho + V(s'))$$
$$\rho := (1 - \kappa_n) \rho + \kappa_n (c_{inc} + V(s') - V(s))$$

Step 3 $n := n + 1$, $s := s'$. Stop if $n = max_{steps}$, else goto ***Step 1***.

The algorithm terminates with the stationary policy comprising the actions that minimize (9). κ_n is the learning rate which is forced to die with increasing number of iterations. Exploration is crucial in guaranteeing the convergence of

this algorithm and we suggest to use the ϵ-directed heuristic search which means that with some small probability ϵ, we choose an exploratory action (as opposed to the best possible action) at each iteration that would lead the process to the least visited state [11].

3.2 Formulation with the Signaling Rate Constraint

In the previous formulation with the two cost parameters S and b, there is no immediate mechanism to set these two parameters. We therefore suggest a revised formulation. In this new formulation, we introduce a desired signaling rate D (number of desired capacity updates per hour). Our goal is then to minimize the average aggregate reserved bandwidth subject to the constraint that the frequency of capacity updates is less than the desired rate D.

A generic leaky bucket counter is a counter that is incremented by unity each time an event occurs and that is periodically decremented by a fixed value. Such counters have successfully been used for usage parameter control in ATM networks [1] and traffic conditioning at the boundary of a diffserv domain [9]. We suggest to use a modified leaky bucket counter for the dynamic capacity management problem to regulate the actual signaling rate to the desired value. Let $X, 0 \leq X \leq B_{max}$ be the value of the counter where B_{max} is the size of the counter. The working principle of our modified leaky bucket counter is given as follows:

When a new capacity update request occurs, then

a) If $X < B_{max} - 1$, then the bucket counter is incremented by one,
b) If $X = B_{max}$, then the capacity update request will be rejected, and
c) If $X = B_{max} - 1$, then the new reserved capacity for the aggregate will be forced to be C_{max} and the counter will be incremented by one to B_{max}.

In the meantime, the counter is decremented every $3600/D$ seconds. The difference between the modified counter introduced above and the generic leaky bucket counter is the operation under the condition c). The motivation behind the operation c) is that if the aggregate reservation was not set to C_{max}, then in the worst case scenario, the blocking probability would have exceeded p until the next epoch when the counter will be decremented. With this choice, we upper bound the average blocking rate by p irrespective of the desired signaling rate. We also note that B_{max} is analogous to the maximum burst size in ATM networks and its role in this paper is to limit the number of successive capacity update requests. In our simulations, we fix $B_{max} = 10$ and leave a detailed study of the impact of B_{max} for future work.

With this methodology, the actual signaling rate will be regulated to the desired signaling rate D. We remove the cost parameter of signaling and the only cost in the formulation is incurred via b normalized to 1. In other words, our aim is to find the best capacity updating policy whose average aggregate bandwidth reservation is minimal without exceeding the desired signaling rate D. Our re-defined state space is as follows:

$$\mathbf{S} = \{s | s = (s_a, s_r, s_b), 0 \leq s_b \leq B_{max}\},$$

where s_a and s_r are as defined before and s_b refers to the value of the leaky bucket counter. We propose the following model-free reinforcement learning algorithm based on [7] which is given as follows:

Step 0 Initialize $Q(s, s'_r) = 0$, $\forall s \in \mathbf{S}$, $\forall s'_r \in [s_a, C_{max}]$, set $n := 1$, cumulative cost $c_{cum} = 0$, total time $T = 0$, average cost $\rho = 0$ and start simulation after selecting an initial starting state.

Step 1 Choose the best possible action by finding

$$arg \min_{s'_r} Q(s, s'_r) \tag{10}$$

Step 2 Carry out the best or another random exploratory action. Observe the incurring cost c_{inc}, state transition time τ_s and next state s'. Perform the following update

$$Q(s, s'_r) := (1 - \kappa_n)Q(s, s'_r) + \kappa_n(c_{inc} - \rho\tau_s + \min_{s'_r} Q(s', s'_r)) \tag{11}$$

If the best action is selected, perform the following updates:

$$\begin{aligned} c_{cum} &:= (1 - \varsigma_n)c_{cum} + \varsigma_n c_{inc} \\ T &:= (1 - \varsigma_n)T + \varsigma_n \tau_s \\ \rho &= \frac{c_{cum}}{T} \end{aligned}$$

Step 3 $n := n + 1$, $s := s'$. Stop if $n = max_{steps}$, else goto ***Step 1***.

The algorithm terminates with the stationary policy comprising the actions (10). κ_n and ς_n are learning rates which are forced to die with increasing number of iterations. Again, we used the ϵ-directed heuristic search during simulations.

4 Numerical Results

4.1 Results for a Given S/b Ratio

We verify our approach by comparing RVI and A-RVI with the two traditional reservation mechanisms, namely SVC and PVP. The problem parameters are chosen as $\lambda = 0.0493$ calls/sec., $\mu = 1/180$ sec., $C_{max} = 16$. We run ten different 12 hour simulations for different values of S/b, and average of these simulations are reported. Figure 2 shows the average performance metrics: average cost, average reserved bandwidth and average number of capacity updates using different methods. Irrespective of the cost ratio S/b, policies obtained via RVI and A-RVI give very close results for the average cost. However, there is a slight difference in the optimal policies found using RVI and A-RVI since the average reserved bandwidth and average number of capacity updates with the RVI and A-RVI policies are found to be different using simulations. When the ratio S/b approaches zero, the RVI and A-RVI policies give very close results to that of the SVC approach. This is expected since when the signaling cost is very low, SVC provide the most efficient bandwidth mechanism. On the other hand, when the

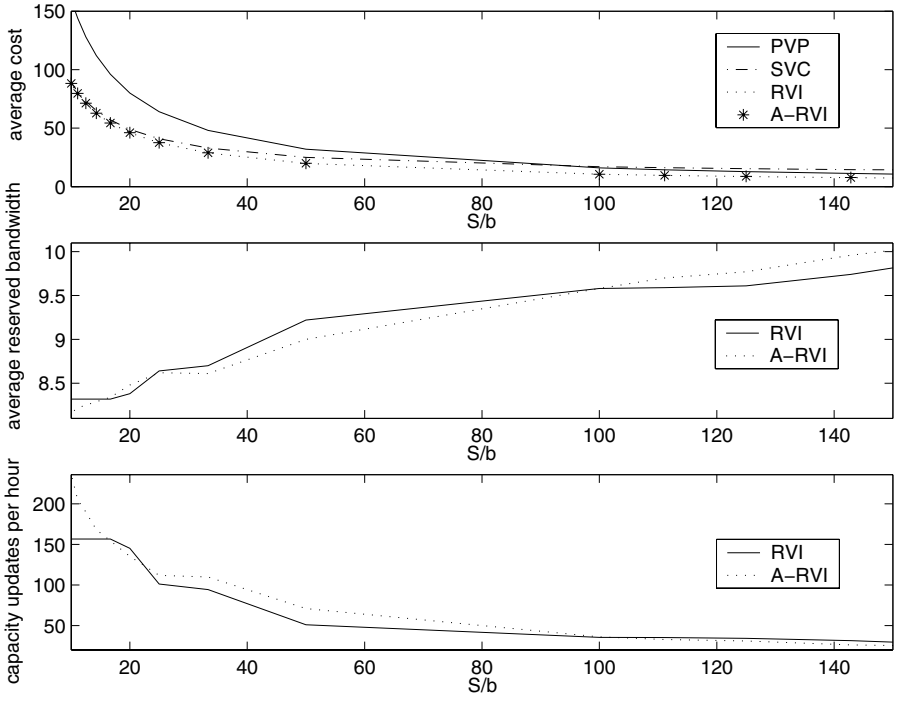


Fig. 2. Average cost, average reserved bandwidth, and average number of capacity updates using PVP, SVC, RVI, and A-RVI for the case $\lambda = 0.0493$ calls/sec., $\mu = 1/180$ sec., $C_{max} = 16$

ratio $S/b \to \infty$, RVI and A-RVI policies very much resemble the PVP approach. This is also intuitive since when the signaling cost is very high, the only option is allocating bandwidth for the aggregate for once in a very long period of time.

Table 1 shows the performance of A-RVI for a larger size problem where the RVI solution is numerically intractable. We take $C_{max} = 300$ and $\lambda = 1.5396$ calls/sec. This table demonstrates that with a suitable choice of the ratio S/b, one can limit the frequency of capacity updates in a dynamic capacity management scenario. Moreover, A-RVI consistently gives better results than both PVP and SVC in terms of the overall average cost.

4.2 Results for a Given Desired Signaling Rate (without Cost Parameters)

We tested our approach for different values of the desired signaling rate D. The problem parameters are chosen as $\lambda = 0.0493$ calls/sec., $\mu = 1/180$ sec., $C_{max} = 16$. Figure 3 depicts the average reserved bandwidth of our approach (denoted by DCM), in terms of capacity units which are obtained out of a 24 hour simulation. It is observed that when we decrease D, the average reserved

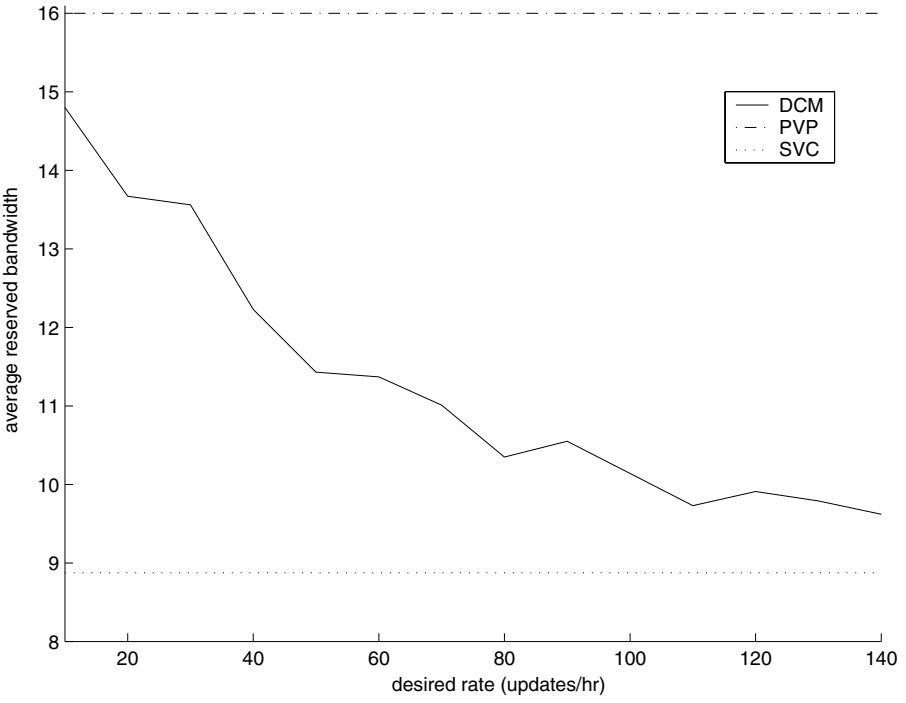


Fig. 3. Average reserved bandwidth with our approach for different values of D

bandwidth converges to that of the static PVP policy (i.e., C_{max} units). On the other hand, when D increases, the policy found by RL approaches the SVC approach as expected. We also note that the observed signaling rate in the simulations was within the 2% neighborhood of the desired signaling rate irrespective of D.

5 Conclusions

In this paper, we introduce a dynamic provisioning problem that arises in a number of aggregate reservation scenarios including virtual path based voice over packet backbones. The capacity provisioning problem is posed in two different formulations. In the first formulation, a cost is assigned to each capacity update as well a cost for reserved bandwidth and the goal is to minimize the average long run cost. This problem turns out to fit well into the semi-Markov decision framework and we propose dynamic programming and reinforcement learning solutions. We show that reinforcement learning solutions scale very well up to large sized problems and they provide close results to those of the dynamic programming approach for small sized problems. In the second formulation, we introduce a constraint on the number of capacity updates in unit time and we

Table 1. Performance results of the policy obtained via A-RVI for the case $C_{max} = 300$

	$S/b = 100$	$S/b = 50$	$S/b = 20$
A-RVI average cost	272.2	524.0	1277
SVC average cost	526.6	775.8	1523
PVP average cost	300	600	1500
A-RVI average reserved bandwidth	272	261	254
A-RVI # of capacity updates per hour	45	550	2418

seek the minimization of the long term average reserved bandwidth. We propose a reinforcement learning solution to this constrained stochastic optimization problem. Our results indicate a significant bandwidth efficiency with respect to a static PVP-type allocation while satisfying signaling rate constraints. As future work, we are considering extension of this work to multimedia networks with more general traffic characteristics.

References

1. "ATM User Network Interface (UNI)", ATM Forum Specification version 4.0, AF-UNI-4.0, July 1996.
2. F. Baker, C. Iturralde, F. Le Faucheur, B. Davie, "Aggregation of RSVP for IPv4 and IPv6 Reservations", RFC 3175, September 2001.
3. D. P. Bertsekas and J. N. Tsitsiklis, "Neuro-Dynamic Programming" Athena Scientific, Belmont, MA, 1996.
4. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services", RFC 2475, 1998.
5. D. Clark, S. Shenker, L. Zhang, "Supporting Real-time Applications in an Integrated Services Packet Network: Architecture and Mechanism", in Proc. SIGCOMM'92, September 1992.
6. B. Davie, Y. Rekhter, "MPLS: Technology and Applications", Morgan Kaufmann Publishers, 2000.
7. A. Gosavi, "A Convergent Reinforcement Learning Algorithm for Solving Markov and Semi-Markov Decision Problems Under Long-Run Average Cost", Accepted in the European Journal of Operational Research, 2001.
8. B. Groskinsky, D. Medhi, D. Tipper, "An Investigation of Adaptive Capacity Control Schemes in a Dynamic Traffic Environment", IEICE Trans. Commun., Vol. E00-A, No, 13, 2001.
9. J. Heinanen, R. Guerin, "A Two Rate Three Color Marker", RFC 2698, 1999.
10. A. Jalali, M.Ferguson. "Computationally efficient adaptive control algorithms for Markov chains", In Proceedings of the 28th. IEEE Conference on Decision and Control, pages 1283–1288, 1989.
11. S. Mahadevan, "Average Reward Reinforcement Learning: Foundations, Algorithms and Empirical Results", Machine Learning, 22, 159–196, 1996.
12. K. Nichols, S. Blake, F. Baker, D. Black, "Definition of the Differentiated Services Field (DS field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.
13. S. Shenker, C. Partridge, R. Guerin, "Specification of Guaranteed Quality of Service", RFC 2212, 1997.

14. S. Shiodam, H. Saito, H. Yokoi, “Sizing and Provisioning for Physical and Virtual Path Networks Using Self-sizing Capability”, IEICE Trans. Commun., Vol. E80-B, No. 2, February 1997.
15. R. S. Sutton and A. G. Barto, “Reinforcement Learning: An Introduction”, MIT Press, 1998.
16. H. C. Tijms, “Stochastic Models: An Algorithmic Approach”, John Wiley and Sons Ltd., 1994.
17. J. Wroclawski, “Specification of the Controlled-Load Network Element Service”, RFC 2211, 1997.
18. J. Wroclawski, “The Use of RSVP with IETF Integrated Services ”, RFC 2210, 1997.

Calculating End-to-End Queuing Delay for Real-Time Services on an IP Network

Robert E. Kooij[1], Olaf Østerbø[2], and J.C. van der Wal[3]

[1] TNO Telecom, St. Paulusstraat 4, 2264 XZ Leidschendam, The Netherlands
R.E.Kooij@telecom.tno.nl

[2] Telenor Research, Snarøyveien 30, N-1331 Fornebu, Norway
Olav-Norvald.Osterbo@telenor.com

[3] Independent Consultant, Kwekerijstraat 22, 2613 VE Delft, The Netherlands
Kees.vanderWal@net.hcc.nl

Abstract. A crucial factor for real-time (interactive) services is the end-to-end delay experienced by the application. The contribution resulting from the queuing delay induced by the network nodes is the most difficult to assess. First, it is a stochastic quantity which should be aggregated over many (possibly different) network nodes. Secondly, the queuing delay in a single node stems from two different mechanisms: one related to interference with other interactive flows and one related to interference with the ubiquitous best-effort data flows. Earlier work assessed these two components separately, leading to a 'worst case' result. This paper models both components and develops formulas to calculate exact results for the end-to-end queuing delay. Results are shown indicating an improvement up to 45% over the worst-case method. The formulae developed in this paper are expected to be useful in network dimensioning, in setting network performance requirements and in admission control mechanisms.

1 Introduction

Many consider the Internet Protocol (IP) to be an enabling technology for multi-service networks. However, the currently widely deployed Internet service provides a best effort service only. Delay and packet loss may be introduced at any node along the end-to-end route through the network, depending on that node's state of congestion. Especially when interactive voice and video are transported over the network concurrently with the traditional data, the network needs to implement specific actions in order to control (i.e. minimise) the packet loss and packet delay experienced by the interactive streams.

To that end, the Internet Engineering Task Force (IETF) has defined two approaches to support Quality of Service (QoS) in IP networks: the Integrated Services model [1] (IntServ) and the Differentiated Services [2] (DiffServ) model. Both approaches provide a framework to achieve (1) that for interactive and streaming services the packet loss in each node will be virtually zero and (2) that for interactive services the packet

W. Burakowski et al. (Eds.): Art-QoS 2003, LNCS 2698, pp. 115–126, 2003.

delay in each node is kept small. The solutions to achieve these objectives are not specified and thus implementation dependent. Subsection 1.2 describes the node model used in this paper.

1.1 Importance of Delay and Delay Assessment for Interactive Applications

For interactive services, it is crucial that the end-to-end delay as experienced by the service is kept small, such that the interactive nature of the application is not impeded. For example, according to [3] the end-to-end delay should not exceed 150 ms for telephony. There are several components in the end-to-end delay of the service: delay contributions by the *terminals* (such as packetisation delay and dejitter delay) and the delay incurred by the *network* to each IP packet. The latter delay component can be decomposed into a deterministic (e.g. propagation) and a stochastic (e.g. queuing) part. The deterministic delay is relatively easy to determine, queuing delay is more difficult to assess because it is a stochastic value which also depends on the congestion state in each of the network nodes.

Assessing the queuing delay is important for two reasons. First of all it contributes directly to the end-to-end delay for which strict delay bounds should be met in order to deliver voice with sufficient quality (e.g. 150 ms, see above). Secondly, in order to cope with variation in delay (often referred to as jitter) at the destination of every end-to-end route a dejittering buffer is implemented. The receiving terminal retains the first arriving packet of a flow in the dejittering buffer for some time before delivering packets to the decoder at the rate with which the packets were originally generated. This delay provides some slack for subsequent packets which happen to arrive 'late' due to queuing in the network nodes. In the event of an unexpectedly 'extremely late' arriving packet, the dejitter buffer will run empty and the receiver should feed some dummy information (e.g. silence) to the decoder instead. Dimensioning the initial delay at the receiver side involves a trade-off between delay and packet loss. A large initial delay will contribute largely to the end-to-end service delay but will keep the fraction of packets which arrive too late to be played-out to a small value. When the initial delay is chosen equal to the $(1-p)$-quantile of the end-to-end queuing delay, the value of p indicates the fraction of packets that can be expected to arrive too late for play-out

In other words, the effect of the queuing delay reflects twice in the end-to-end delay: once as the genuine contribution to the packet delay and once (in the format of a suitable quantile of its distribution) as the optimum (minimum) setting of the initial delay in the dejittering buffer. Finally, note that the queuing delay is one of the factors that can be influenced by proper traffic management. For example propagation delay cannot be very well controlled, given the distance to be covered, whereas the queuing delay for real-time traffic can be kept small by controlling the load on the queues (devoted to real-time traffic).

There is a need to predict the delay before large-scale deployment of real-time services over IP networks. As these services are not implemented yet on a sufficiently large scale to perform representative life measurements, we follow a theoretical ap-

proach for assessing the delay performance that could be obtained under certain traffic assumptions. It is well known that deterministic upper bounds for queuing delay are easily obtained, see [4], [5], [6], but these upper bounds lead to unrealistically high values. Instead, in this paper statistical "upper bounds" (i.e. quantiles) are derived for the queuing delay, leading to more realistic values, see also [7], [8], [9]. The formulae developed in this paper are invaluable tools in network dimensioning, in setting network performance requirements [10] and admission control mechanisms that are required to keep the load on those queues under control.

1.2 Node Model

A single traditional First-In-First-Out (FIFO) queue (per output interface) is not sufficient for supporting voice, video and data services on a single network (node), because data sources making use of the Transmission Control Protocol (TCP) tend to increase their sending rate until (a part of) the network is congested, resulting in excessive delays and packet loss. This paper uses a node model with two queues (per output interface) served by a non-pre-emptive Head-of-Line (HoL) scheduler illustrated in Fig. 1. Head-of-line queueing system. Real-time packets get absolute priority over data packets. Data packets are only served when the real-time queue is empty. In case DiffServ is used packets with different transport requirements can be distinguished (and classified into the correct queue) by the code-point in the header of the IP packet.

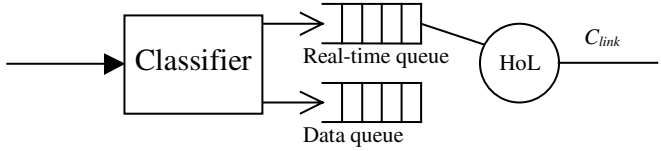


Fig. 1. Head-of-line queueing system

Most papers study the real-time queue in isolation, because the queuing delay of the real-time services and the queuing delay due to the residual service of the data packet are assumed to be statistically independent [11]. In [8], [9] the contribution of the data to the queueing delay is assessed as a worst-case estimate i.e. as the service time of one data packet. In this paper we will tighten the assessment of the queueing delay by incorporating the residual service of the data packet.

We will assume that the real-time service generates packets of constant length and at a constant rate (CBR traffic); each packet traverses a given number of hops (routers) to reach the receiving destination. The delay components in the different routers are assumed to be independent. For some specific cases this assumption will not be valid and therefore the resulting model will be approximate. There is, however, reason to believe that if the actual load from each stream is small compared to the total load on the network elements considered, this assumption will be quite reasonable, see [12], [13], [14].

2 Queueing Model

In order to calculate a quantile of the queuing delay we assume that all real time flows are Constant Bit Rate (CBR) flows generating packets with a constant length. However, as the traffic flows interfere with other flows, the CBR character of the flows disappears while traversing the network crossing multiple routers.
In [7] it is conjectured that flows that are initially CBR cannot be disturbed to a stream that has more burstiness than a Poisson stream, as long as it interferes only with flows that were originally CBR. A recent paper [15] presents analytical and simulation results that support this conjecture.
Hence, in order to calculate the delay we need to model each node as an M/D/1 queue with non-preemptive priority and then apply convolution to get the end-to-end waiting time distribution.
If we let W_k denote the waiting time in the k^{th} node for the real-time packets then the total delay may be written $W_{NP}^H = W_1 + ... + W_K$, and the Laplace-Stieltjes Transform (LST) of the sum is found as the product of the LSTs of the waiting times in the individual nodes

$$\tilde{W}_{NP}^H(s) = \prod_{k=1}^{K} \tilde{W}_k(s), \tag{1}$$

where $W_k(s)$ is the LST of the waiting time for the highest priority packets in a M/D/1 non-preemptive queueing model. In order to make the results more general we will use the LST of the waiting time for the highest priority packets in a M/G/1 non-preemptive queueing model given by the *Pollaczek-Khinchin* formula (with a slight modification); see for instance [16]:

$$\tilde{W}_k(s) = \frac{1-\rho_k^H}{1-\rho_k^H \hat{B}_k^H(s)}(1-p_k + p_k \hat{B}_k^L(s)), \tag{2}$$

where ρ_k^H and ρ_k^L are the loads and $\hat{B}_k^H(s)$ and $\hat{B}_k^L(s)$ are the LSTs of the remaining service times for high and low priority packets respectively and further $p_k = \rho_k^L/(1-\rho_k^H)$. Note that the relation between the LST of the remaining service times and the "ordinary" service times is given as $\hat{B}(s) = (1-\tilde{B}(s))/(sb)$ where $b = E[B]$ is the mean service time.
It may be convenient to relate the end-to-end waiting time distribution based on the LST (1) to the corresponding model without any priority. We may write $W_{NP}^H = W^T + B_{NP}^T$ where W^T and B_{NP}^T are independent and W^T represents the end-to-end queuing delay for the corresponding path (system) without any priority and B_{NP}^T is the extra delay due to the influence from the lower priority packets. Further the distri-

bution of W_{NP}^{H} may therefore be found by convoluting the distributions of W^{T} and B_{NP}^{T}. We may therefore re-write:

$$\tilde{W}_{NP}^{H}(s) = \tilde{W}^{T}(s)\tilde{B}_{NP}^{T}(s), \tag{3}$$

with

$$\tilde{W}^{T}(s) = \prod_{k=1}^{K} \frac{1-\rho_k^H}{1-\rho_k^H \hat{B}_k^H(s)} \tag{4}$$

and

$$\tilde{B}_{NP}^{T}(s) = \prod_{k=1}^{K} (1 - p_k + p_k \hat{B}_k^L(s)). \tag{5}$$

The expressions above will also apply for saturated system. In this case there will always be low priority packets present in the low priority queue and this correspond to the case with $p_k = 1$ (or $\rho_k^H + \rho_k^L = 1$).

In the Appendix it is explained how to find (invert) the DF of the end-to-end waiting time $W_{NP}^{H}(t) = P\{W_{NP}^{H} \le t\}$ based on the LST (1) and (2) or (3), (4) and (5).
In this paper we will only consider the case when all the nodes are identical, that is the network is homogeneous in the sense that all links have equal capacity and load. The case of a heterogeneous network is treated in [17] but there the data queues are not taken into account.

3 Comparison of the Methods

In this section we will compute quantiles of end-to-end delay for several scenarios. We will compare the outcome of the exact method as discussed in Section 2 with the so called worst-case (WC) method, introduced in Section 1. Note that the worst-case method assesses end-to-end delay quantiles by taking the sum of the end-to-end quantile for K M/D/1 queues (contribution of real-time queue) and K times the service time of a data packet (contribution of data queue), see [9], [10]. We will consider several scenarios by varying the following parameters: the number of hops, the quantiles, the ratio of packet sizes of best effort traffic and real-time traffic and the load due to real-time traffic. In addition we will assume that there will always be low priority packets present in the low priority queue (corresponding to the case $p_k = 1$ in Section 2).

In Tables 1-2 we give (1-*p*)-quantiles of the queuing delay expressed in number of real-time packets where $p = 10^{-3}$.
Within each table the load due to real-time traffic and the number of hops (denoted by *K*) are varied. The difference between each of the tables lies in the different ratios

of packet sizes of best effort data traffic and real-time traffic. If we denote by P_{be} and P_{rt} the packet size (in byte) of best effort data packets and real-time packets respectively, then

Table 1. (1-10-3)-quantiles of queueing delay with Pbe = 5Prt
Table 2. (1-10-3)-quantiles of queueing delay with Pbe = 10 Prt

correspond to the case $P_{be} = 5P_{rt}$ and $P_{be} = 10P_{rt}$ respectively.
Note that in order to express the delay in seconds the numbers in the tables should be multiplied with the service time of an real-time packet, i.e. $8P_{rt}/C$, where C denotes the link capacity (in bit/s).

Table 1. (1-10-3)-quantiles of queueing delay with Pbe = 5Prt

load	40%		50%		60%	
	WC	exact	WC	exact	WC	exact
1 hop	8.94	7.65	10.16	8.70	11.96	10.31
2 hops	15.06	12.31	16.70	13.61	19.09	15.64
4 hops	26.84	20.80	29.12	22.48	32.52	25.20
8 hops	49.75	36.21	53.15	38.66	58.24	42.65
16 hops	94.73	64.59	100.12	68.59	108.08	75.13

Table 2. (1-10^{-3})-quantiles of queueing delay with $P_{be} = 10\ P_{rt}$

load	40%		50%		60%	
	WC	exact	WC	exact	WC	exact
1 hop	13.94	12.22	15.16	13.15	16.96	14.59
2 hops	25.06	21.31	26.70	22.34	29.09	24.00
4 hops	46.84	38.03	49.12	39.21	52.52	41.23
8 hops	89.75	67.46	93.15	69.33	98.24	72.39
16 hops	174.73	121.10	180.12	124.43	188.08	129.69

In Tables 3-4 we present (1-*p*)-quantiles of the queuing delay expressed in number of real-time packets where $p = 10^{-5}$.

Table 3. (1-10^{-5})-quantiles of queueing delay with $P_{be} = 5\ P_{rt}$

load	40%		50%		60%	
	WC	exact	WC	exact	WC	exact
1 hop	11.79	10.49	13.83	12.37	16.82	15.17
2 hops	18.17	15.50	20.67	17.57	24.37	20.96
4 hops	30.29	24.56	33.59	27.20	38.41	31.33
8 hops	53.72	41.25	58.31	44.65	65.04	50.15
16 hops	99.43	71.77	106.23	76.70	116.24	84.72

Table 4. $(1\text{-}10^{-5})$-quantiles of queueing delay with $P_{be} = 10\ P_{rt}$

load	40%		50%		60%	
	WC	exact	WC	exact	WC	exact
1 hop	16.79	15.07	18.83	16.82	21.82	19.45
2 hops	28.17	24.54	30.67	26.47	34.37	29.51
4 hops	50.29	42.37	53.59	44.45	58.41	47.83
8 hops	93.72	75.11	98.31	77.55	105.04	81.68
16 hops	179.43	133.49	186.23	137.25	196.24	143.42

All exact results in Tables 1-4 have been validated through simulation and they all fall within 95% confidence intervals.
Tables 1-4 lead to the following observations:

- The *worst-case method* produces reasonably accurate results only for the single hop case. The relative difference becomes quite considerable (up to 45%) for cases with a larger number of hops ($K = 16$) and in particular for cases where the best-effort packets are much larger than the real-time packets ($P_{be} = 10P_{rt}$).
- The *exact method* has performed in a stable way and has produced accurate results. The gain in accuracy compared to the worst-case method (-32%) is considerable, in particular for a large number of hops and for real-time packets much smaller than the best-effort packets.
- Relative errors for the worst-case estimations of $(1\text{-}p)$-quantiles get slightly better for smaller p.
- Relative errors for the worst-case estimations are rather insensitive with respect to the load due to real-time traffic.

4 Example Application of the Methods

In this section we will apply, as an example case, both the exact and the worst-case method for the calculation of delay in an access scenario and a core scenario.
For the access scenario we assume real-time packets of size 150 byte. Data packets are assumed to be 1500 byte. The packets traverse two links of 256 kbit/s. Figure 2 shows (1-p)-quantiles of the delay over the two links for varying values of the load.

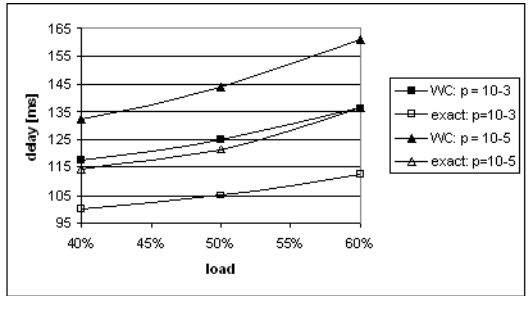


Fig. 2. Queueing delay over 2 access links: $P_{be}=10P_{rt}$ =1500 byte, C = 256 kbit/s

For the core scenario we assume real-time packets of size 300 byte. Data packets are assumed to be 1500 byte. The packets traverse 16 links of 155 Mbit/s. Figure 3 shows (1-p)-quantiles of the delay over the 16 links for varying values of the load.

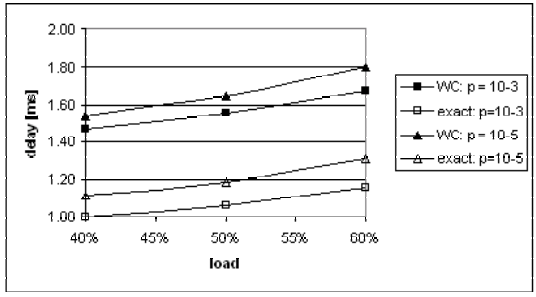


Fig. 3. Queueing delay over 16 core links: $P_{be}=5P_{rt}$ =1500 byte, C = 155 Mbit/s

From Figures 2-3 and other scenario's that are not reported here we draw the following conclusions:

- In the example access scenario the worst-case method seriously overestimates the quantile values for queueing delay. This overestimation can be in the order of tens of milliseconds.
- In the example core network scenario the overestimation due to the worst-case method will not be more than 0.5 ms.

5 Discussion and Conclusions

We have discussed two methods to assess the end-to-end queuing delay experienced by real-time IP datagrams which get non-preemptive priority over data packets. The first method, referred to as the *worst-case method*, quantifies the contributions in a tandem of nodes due to interference with real-time traffic and due to interference with data traffic separately and then aggregates them to arrive at an upper bound for quantile values of the end-to-end queuing delay. The second method, which is referred to as the *exact method,* considers a single node as a queuing system that is formed by a convolution of the waiting time in the real-time queue and the residual service time of the data queue. Both the worst-case and the exact method make use of convolutions to evaluate the result in multiple nodes.

We have applied the methods under a number of simplifying conditions: all nodes are identical, all packet sizes are identical and the waiting times in consecutive nodes are independent. For heterogeneous nodes our methods can also be applied although the complexity increases significantly. For non-deterministic packet size distributions in general it is not possible to use our method because there is no explicit expression for the waiting time distribution, such as (11). However, in this case one can use approximative methods, see [17]. Simulations for some "real" network scenarios, see [18],

show that for moderate loads (say up to 60%) the independence assumption leads to accurate results.

It is concluded that the exact method provides results which are significantly smaller than the worst-case method.

For network scenarios with only low bit rates the improvement due to the exact method can be in the order of tens of milliseconds, and hence using the exact method instead of the worst-case method is very relevant

For network scenarios with only high bit rates the improvement due to the exact method will be very small, in the order of 1 ms, and in such cases the worst-case method suffices.

The formulae developed in this paper are expected to be useful in network dimensioning, in setting network performance requirements and in admission control mechanisms.

References

1. Braden, R., Clark, D. and Shenker, S.: Integrated Services in the Internet Architecture: and Overview. RFC 1633, (1994)
2. Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z. and Weiss, W.: An Architecture for Differentiated Service. RFC 2475, (1998)
3. ITU-T Recommendation G.114: One-way transmission time. May, (2002)
4. Charny, A. and Le Boudec, J.-Y.: Delay Bounds in a Network with Aggregate Scheduling. Proceedings of First COST 263 International Workshop, pp. 1–13, QofIS 2000, Berlin, Germany, (2000)
5. Parekh, A. and Gallager, R.: A generalized processor sharing approach to flow control in integrated services networks: The single node case. IEEE/ACM Transactions on Networking, Vol. 1, (1993), pp. 344–357
6. Parekh, A. and Gallager, R.: A generalized processor sharing approach to flow control in integrated services networks: The multiple node case. IEEE/ACM Transactions on Networking, Vol. 2, (1994), pp. 137–150
7. Brichet, F., Massoulié, L. and Roberts, J.W.: Stochastic Ordering and the Notion of Negligible CDV. Proceedings of ITC 1, Washington (USA), (1997), pp. 1433–1444
8. Mandjes, M.R.H., Van Der Wal, J.C., Kooij, R.E. and Bastiaansen, H.J.M.: End-to-end Delay models for Interactive Services on a Large-Scale IP Network. Proceedings of the 7th workshop on performance modelling and evaluation of ATM & IP networks (IFIP99), (1999)
9. De Vleeschauwer, D., Petit, G.H., Wittevrongel, S., Steyaert, B. and Bruneel, H.: An Accurate Closed-Form Formula to Calculate the Dejittering Delay in Packetised Voice Transport. Proceedings of the IFIP-TC6 / European Commission International Conference NETWORKING 2000 Paris, (2000), pp. 374–385
10. ITU-T: Network performance objectives for IP-based services. ITU-T Recommendation Y. 1541, May (2002) and Appendix X, November (2002) http://www.itu.int/rec/recommendation.asp?type=folders&lang=e&parent=T-REC-Y.1541
11. Kleinrock, L.: Queuing Systems Volume 1: Theory. John Wiley & Sons, New York, (1975)

12. Kruskal, C.P., Snir, M. and Weiss, A.: On the Distribution of Delays in Buffered Multistage Interconnection Networks for Uniform and Nonuniform Traffic. Proceedings of the International Conference on Parallel Processing, (1984), pp. 215–219
13. Lau, W.C. and Li, S.Q.: Traffic Distortion and Inter-source Cross-correlation in High-speed Integrated Networks. Computer Networks and ISDN Systems, Vol. 29, (1997), pp. 811–830
14. Van den Berg, J.L., Lavrijsen, C.P.H.M. and De Vleeschauwer, D.: End-to-end Delays in ATM Networks: Theory and Practice. Proceedings of the ATM Developments '95 Conference, Rennes (France), 29-30 March (1995)
15. Bonald, T., Proutière, A. and Roberts, J.W.: Statistical Performance Guarantees for Streaming Flows using Expedited Forwarding. Proceedings of INFOCOM 2001, Volume 2, Anchorage (AL), USA, (2001), pp. 1104–1112
16. Kleinrock L.: Queueing Systems, Volume II: Computer Applications. New York, John Wiley & Sons, (1976)
17. De Vleeschauwer, D., Büchli, M.J.C., Van Moffaert, A., Kooij, R.E.: End-to-end queueing delay assessment in multi-service IP networks. Journal of Statistical Computation and Simulation, Vol. 72(10), (2002), pp. 803–824
18. Østerbø, O., Models for Calculating End-to-End delay in Packet Networks. Paper submitted to ITC-18, (2003)

Appendix: Exact Results When All the Nodes Are Identical

For the case where all the nodes are statistically independent it is possible to carry the analysis significant further without introducing any approximations. This is due to the fact that it is possible to obtain the LST of the convolution through partial derivatives of the load for the LST of the waiting times in a single M/G/1 queue. For instance, if we let

$$\tilde{W}(s,\rho) = \frac{1-\rho}{1-\rho\hat{B}^H(s)} \tag{6}$$

be the LST for the waiting time for one single queue then simple partial derivative with respect to the parameter ρ yields:

$$\tilde{W}^T(s,\rho) = \left(\tilde{W}(s,\rho)\right)^K = \frac{(1-\rho)^K}{(K-1)!}\frac{\partial^{K-1}}{\partial\rho^{K-1}}\left\{\frac{\rho^{K-1}}{1-\rho}\tilde{W}(s,\rho)\right\}. \tag{7}$$

The same result will also apply for the DF (and also PDF) of the convolution, hence

$$W^T(t,\rho) = \frac{(1-\rho)^K}{(K-1)!}\frac{\partial^{K-1}}{\partial\rho^{K-1}}\left\{\frac{\rho^{K-1}}{1-\rho}W(t,\rho)\right\}, \tag{8}$$

where $W(t,\rho)$ denotes the DF of the waiting time in a M/G/1 queue with load ρ. Similar and more extended results may be found in a separate paper [18]. The DF of the end-to-end queueing delay therefore be written as the sum obtained by inverting (3)-(5):

$$W_{NP}^H(t) = (1-p)^K W^T(t,\rho^H) + \sum_{r=1}^{K} b_r(p,K) W^T(t,\rho^H)(*)\hat{b}^L(t)^{*(r)}. \tag{9}$$

where $b_r(p,K)=\binom{K}{r}p^r(1-p)^{K-r}$ is the binominal probability with parameters $p=\frac{\rho^L}{1-\rho^H}$ and K and $\hat{b}^L(t)^{*(r)}$ is the r-fold convolution of the PDF of the remaining service times for the low priority packets ($(*)$ denotes convolution). The expression (9) represents a general expression without any specific assumptions on the actual service time distributions. To carry the analysis any further specific choices on the service time distributions therefore has to be made.

In the following we shall assume that both the high and low priority packets have constant service times given by b^H and b^L respectively. In this case we have

$$W(t,\rho^H)=q(\frac{t}{b^H},\rho^H), \tag{10}$$

where

$$q(x,\rho)=(1-\rho)\sum_{k=0}^{\lfloor x\rfloor}\frac{[\rho(k-x)]^k}{k!}e^{-\rho(k-x)} . \tag{11}$$

is the DF of the waiting time in a M/D/1 queue with service times scaled to unity. The K-fold convolution of $W(t,\rho^H)$ is found from relation (8) by differentiation:

$$W^T(t,\rho^H)=q^K(\frac{t}{b^H},\rho^H), \tag{12}$$

where

$$q^K(x,\rho)=(1-\rho)^K\sum_{k=0}^{\lfloor x\rfloor}\sum_{l=0}^{K-1}\frac{(-1)^l}{l!k!}\binom{K+k-1}{K-l-1}(\rho(k-x))^{k+l}e^{-\rho(k-x)} . \tag{13}$$

Further we have that the remaining service times for the low priority packets are uniformly distributed over the interval $(0,b^L)$, and we find the r-time convolution $\hat{b}^L(t)^{*(r)}$ on the following form:

$$\hat{b}^L(t)^{*(r)}=\frac{r}{(b^L)^r}\sum_{m=0}^{r}\frac{(-1)^m}{m!(r-m)!}H(t-mb^L)(t-mb^L)^{r-1}, \tag{14}$$

where $H(x)$ is the unit step function. By a rather lengthy evaluation the convolutions $W^T(t,\rho^H)(*)\hat{b}^L(t)^{*(r)}$ may be found by applying the relation (8). By collecting the evaluations we finally get:

$$W_{NP}^H(t)=(1-p)^K q(\frac{t}{b^H},\rho^H)+$$
$$\sum_{r=1}^{K}b_r(p,K)\left(\frac{b^H}{b^L}\right)^r\sum_{k=0}^{\lfloor\frac{t}{b^H}\rfloor}\sum_{m=0}^{\lfloor\frac{t-kb^H}{b^L}\rfloor}(-1)^m\binom{r}{m}\binom{k+r-1}{r-1}q^{K,r-1}(\frac{t-kb^H-mb^L}{b^H},\rho^H) \tag{15}$$

where the auxiliary functions $q^{K,i}(x,\rho)$; $i=1,2...,K-2$ is defined through:

$$q^{K,j}(x,\rho)=\frac{(1-\rho)^K}{(K-1)!}\frac{\partial^{K-1}}{\partial\rho^{K-1}}\left\{\rho^{K-i-2}\frac{q(x,\rho)}{1-\rho}\right\}=$$

$$\frac{(1-\rho)^K}{\rho^{i+1}}\sum_{k=0}^{\lfloor x\rfloor}\sum_{l=0}^{K-1}\frac{(-1)^l}{l!k!}\binom{K+k-i-2}{K-l-1}(\rho(k-x))^{k+l}e^{-\rho(k-x)}.$$

For $i=K-1$ we must add an extra term to get $q^{K,K-1}(x,\rho)$:

$$q^{K,K-1}(x,\rho)=\frac{(1-\rho)^K}{(K-1)!}\frac{\partial^{K-1}}{\partial\rho^{K-1}}\left\{\rho^{-1}\frac{q(x,\rho)}{1-\rho}\right\}+(-1)^K\left(\frac{1-\rho}{\rho}\right)^K=$$

$$\frac{(1-\rho)^K}{\rho^K}\left(\sum_{k=0}^{\lfloor x\rfloor}\sum_{l=0}^{K-1}\frac{(-1)^l}{l!k!}\binom{k-1}{K-l-1}(\rho(k-x))^{k+l}e^{-\rho(k-x)}+(-1)^K\right).$$

Note that in the expression above we define the binominal coefficient as $\binom{n}{m}=\frac{n(n-1)...(n-m+1)}{m!}$ also allowing for negative n and implying $\binom{n}{m}=0$ for $m>n$. The expression (15) for the DF of the end-to-end queueing delay gives stable numerical results for at least up to $K=20$ identical nodes. The numerical accuracy depends heavily on the fact that the auxiliary functions $q^{K,i}(x,\rho)$ may be calculated by introducing "local" variables (see [18]) and thereby avoiding summations of alternating series.

Substantial simplification can be obtained for special choices of the parameters. If the service times for the low priority packets is exactly an integer times the service times of the low priority packets (that is $b^L=lb^H$ with integer l), and the queueing system is saturated, that is $\rho^H+\rho^L=1$, then it can be shown that:

$$W_{NP}^H(t)=\sum_{k=0}^{\left\lfloor\frac{t}{b^H}\right\rfloor}c_K(k,l)q^{K,K-1}(\frac{t}{b^H}-k,\rho^H),$$

where

$$c_r(k,l)=l^{-r}\sum_{m=0}^{\min\left\{r,\left\lfloor\frac{k}{l}\right\rfloor\right\}}(-1)^m\binom{r}{m}\binom{r+k-lm-1}{r-1}.$$

Admission Control Method Based on Effective Delay for Flows Using EF PHB

Marcin Narloch and Sylwester Kaczmarek

Gdansk University of Technology, Faculty of Electronics, Telecommunications and Informatics, Narutowicza11/12, 80-952 Gdansk, Poland
{narloch,kasyl}@eti.pg.gda.pl

Abstract. Admission Control Method for flows using EF PHB is presented. The method is based on the concept of effective delay for EF PHB flows. Effective delay is a notion which concentrates in one single measure statistical upper bound of packet delay, related probability for that bound and state of the node (queue) in the path of the stream for which performance guarantees must be fulfilled. Presented numerical and simulation results demonstrate accuracy and usefulness of the proposed method for providing statistical guarantees in networks with aggregate scheduling.

1 Introduction

Traditional IP network supported only single service known as Best Effort (BE), which is characterized by complete absence of any Quality of Service (QoS) guarantees. Though IP network makes attempts to transfer packets from source to destination on best effort rule, there are no guarantees when or even whether packets would be delivered. The problem of possible losses is solved by retransmissions controlled by TCP from higher layer than IP. That model of providing services is sufficient for the elastic traffic from applications such as www, ftp, telnet which can tolerate arbitrarily large delays. On the other hand there exist many applications for which that approach of providing services can not be accepted. Particularly that regards to services with precisely defined delay and delay variance dependencies which are characteristic for circuit switched networks.

Rapidly increasing tendencies to provide services typical for traditional telecommunication networks in the Internet, rise new challenges for realizing services with guaranteed QoS in IP based networks. That results in evolution of the Internet towards a multiservice network. Within Internet Engineering Task Force (IETF) two models were proposed for providing services with QoS: Integrated Services (IntServ) [4] and Differentiated Services (DiffServ) [2,22]. Due to the problems with scalability of IntServ in the core of the network, currently research attention is mainly focused on DiffServ. A particular field of interest is a problem of providing real-time services for streaming flows using Expedited Forwarding Per Hop Behavior (EF PHB) [6,8,18] in a DiffServ network with aggregate scheduling. It seems that success of IP as a multiservice network strongly depends on providing solution for that problem.

In the paper we propose Admission Control Method for flows using EF PHB. The method allows to verify whether providing end-to-end statistical guarantees for EF

W. Burakowski et al. (Eds.): Art-QoS 2003, LNCS 2698, pp. 127–138, 2003.

flows is possible, if current state of the flow path and QoS requirements are given. The method is based on the proposed concept of *effective delay*. *Effective delay* is a notion which concentrates in one single measure statistical upper bound of packet delay, related probability for that bound and state of the node (queue) in the path of the stream for which performance guarantees must be fulfilled.

The paper is organized as follows. In Section 2 previous work on the subject is presented. Section 3 presents a model of evaluated network. Methods for end-to-end packet delay probability calculation are presented in Section 4. Among presented methods are: approximation based on Erlang-n distribution, Gaussian approximation and methods based on Large Deviation approach, i.e. bounds based on Chernoff Theorem and Bahadur-Rao Theorem. All presented methods provide possibility to evaluate statistical guarantees for packet delays of flows using EF PHB, but with different degree of accuracy. Also in Section 4 concept of *effective delay* for stream using EF PHB is presented. Proposed Admission Control method based on *effective delay* concept for EF streams is evaluated and compared with simulation results in Section 5. Section 6 concludes the paper.

2 Related Work

There exist two distinct approaches to analysis of delay guarantees in the networks with aggregate scheduling. First approach derived from the context of Integrated Services architecture [25] and represented by [1,7] is based on deterministic bounds on performance guarantees with worst case assumptions for end-to-end delays. Analysis in [7] shows that deterministic bound on the delay lead to very low level of the utilization for a network with flow aggregation. The level is the order of $1/(n-1)$ in general case, where n is the number of nodes the observed flow passes through. Such a small result for allowed utilization indicates, that deterministic approach to the analysis of performance guarantees can not be applied in practice.

Second approach relies on the statistical performance guarantees for flows using EF PHB. It allows larger level of utilization at the cost that DiffServ network assures only certain packet loss ratio and guarantees that probability of packet delay in the network exceeding certain value is smaller than certain level (statistical upper bound guarantees). That methodology is analogous to description of QoS for ATM CBR service. Also the approach based on statistical performance guarantees appeared in the proposed standards for Service Level Specification (SLS) for DiffServ [16,24].

Thorough overview of all recent advances in Internet quality of service, including deterministic and statistical guarantees, can be found in [13] (see also references therein). One of the most recent attempts to explore statistical gain was made in [26] where Large Deviations Theorems were applied to results obtained by the use of network calculus. However we would like to point out that above result is limited to single node case. Thus suggested in [26] end-to-end delay calculation, which was obtained by summing bounds evaluated for single nodes in isolation, still seems to be conservative approach.

In the development of the *effective delay* concept we follow alternative approach to statistical guarantees based on the Better than Poisson/Negligible Jitter (NJ) conjecture presented in [3]. That approach is the extension of the negligible Cell Delay Variation notion presented for ATM network in [5] to the case of IP environment with

variable packet lengths. That approach allows radical simplification of the traffic management function because worst case traffic inside a network can be modeled as Poisson stream of MTU size packets. Moreover that approach is consistent with formulation of packet delay performance guarantees in the SLS specification and allows realization of the real-time services with larger network utilization than methods based on worst case, deterministic bound on the delay. Thus we focus on statistical approaches and assume for further development that NJ conjecture is valid [3,5,23].

3 Evaluated Model

We considered a network (similar to presented in [3]) consisted of n non-preemptive priority queues arranged in tandem, serving in high priority (EF) queue observed (tagged) CBR stream T passing through all nodes and interfering Poisson MTU packet sized cross EF-traffic CT_k passing k-th queue (Figure 1).

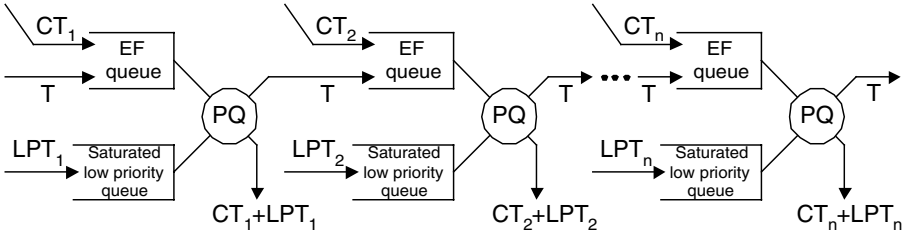


Fig. 1. Analyzed network of priority queues arranged in tandem

In order to model worst case of the influence of the lower priority, non-EF traffic on EF streams performance guarantees in priority queue, we considered in nodes queues with multiple and exhaustive vacations with constant vacation time equal to MTU packet transmission time [12]. In that model arrivals and services have the same characteristics as in ordinary $M/D_{MTU}/1$ queue, but in queue with vacations when high priority (EF) queue is empty, server takes vacation instead being idle waiting for EF packet to arrive for service. If queue server finds EF packets when returning from vacation, it serves them until EF queue becomes empty (exhaustive discipline) and than it takes next vacation. If there is no packet in high priority queue after returning from vacation, server takes another vacation (multiple vacation discipline). That allows modeling the real system with priority queuing and link transmitting non-EF, low priority packets wherever there is no EF packets to transmit. This is also the worst case approach with respect to EF stream performance guarantees because we assumed that link is saturated and there is always MTU sized low priority packet in node to send.

Let us assume that queues belong to J different classes, where each class is characterized by similar value of offered load ρ_{CT} in every queue of the particular class. The number of queues in j-th class is denoted by n_j and the total number of queues in tandem equals $n=\sum n_j$. We assumed that offered load ρ_T of observed traffic is relatively small in comparison with cross traffic offered load ρ_{CT} in every queue. We also

assumed independence of queues in particular nodes, which greatly simplifies the whole evaluation. For example we do not have to consider the effects of distribution of queues from particular class within a chain. It should be noted that independence conjecture is reported in [3] as conservative, however it seems that those assumptions are valid in the case of the core network (DiffServ network).

4 Methods for Packet Delay Probability Calculation and Concept of Effective Delay

Before we present Admission Control method based on the *effective delay* concept, we briefly present methods for analytical evaluation of probabilities of end-to-end delays experienced by packets from CBR flow in the presence of cross traffic in the network. In [3] method based on approximation of queuing delay of EF packet by Erlang-n distribution. That approach is based on presented in [23] in the context of ATM network approximation of queue size distribution $P(Q>x) \approx k \cdot exp(-r \cdot x)$, with simple extension to priority queue. Erlang-n approximation is limited to the homogenous case only and unfortunately that approach cannot be used directly to evaluate packet delay distribution for the heterogeneous case, which is typical for any practical network scenario.

If we assume that n is sufficiently large, limit theorems can be applied for the evaluation of packet delay probabilities. With that assumption the most natural approach is the application of Central Limit Theorem and Gaussian approximation of delay distribution. It is simple extension of the model presented in [17] in the context of CBR service in ATM to the case of variable length IP packets and queues with vacations. In that method, Gaussian distribution of packet waiting times has mean and variance respectively:

$$\mu = \sum_{j=1}^{J} n_j \cdot \mu_j \,,\ \sigma^2 = \sum_{j=1}^{J} n_j \cdot {\sigma_j}^2 \,, \tag{1}$$

where μ_j and σ_j^2 are respectively mean and variance of waiting time in $M/D_{MTU}/1$ queue with vacation of j-th class. Appropriate formulas for μ_j and σ_j^2 can be obtained with the aid of respective derivatives of moment generating functions $M(\theta)=E(exp(\theta X))$ of waiting time for queues with vacations.

Moment generating function $M_j(\theta)$ for $M/D_{MTU}/1$ queue with vacation can be obtained by stochastic decomposition property described in [14,15] and generalized in [10,11]. Stochastic decomposition property allows to consider the waiting time in the M/GI/1 queue with vacations as the sum of two independent components, one distributed as the waiting time in the ordinary queue in the corresponding M/GI/1 queue without vacations and the other as the equilibrium residual time in a vacation. Thus moment generating function $M(\theta)$ for $M/D_{MTU}/1$ queue with vacation can be calculated as follows:

$$M(\theta) = \frac{U(\theta)-1}{u\theta} M_{M/D/1}(\theta) \,, \tag{2}$$

where $M_{M/D/1}$ is moment generating function in ordinary $M/D_{MTU}/1$ queue and $U(\theta)$ denotes moment generating function for the vacation time. In the considered case of constant vacation time equal to MTU packet transmission time: $U(\theta)=exp(\theta u)$ where $u=MTU/C$.

In next two approaches based on limit theorems and Theory of Large Deviations [9] we explore the fact, that delay values for the probabilities of interest [19] are largely deviated from the mean. Evaluation of end-to-end delays in the network based on Large Deviation approach appeared previously in [20]. However, presented in [20] method to simplify numerical calculation can not be directly extended to the case of queues with vacations. Moreover authors of [20] focused on slightly different, general aspects influencing QoS in Internet. From practical point of view we are interested only in evaluation of probability that end-to-end packet queuing delay X exceeds certain value D. Thus we can write quality of service requirements as:

$$P(X > D) \le L \ , \tag{3}$$

where L is a small number, i.e. $L \in \langle 10^{-2}, 10^{-6} \rangle$ [19]. In order to derive waiting time probability distribution approximation based on Chernoff Theorem [9] is used:

$$\log P\left(\sum_{j=1}^{J} \sum_{i=1}^{n_j} X_{ji} \ge D \right) \le -F(\theta^*) \ , \tag{4}$$

where X_{ji} are independent random variables denoting packet waiting time in i-th queue of j-th class and

$$X = \sum_{j=1}^{J} \sum_{i=1}^{n_j} X_{ji} \ . \tag{5}$$

Large deviations rate function $F(\theta^*)$ is defined as:

$$F(\theta^*) = \sup_{\theta \ge 0} F(\theta) \text{ and } F(\theta) = \theta \cdot D - \sum_{j=1}^{J} n_j \log M_j(\theta) \ . \tag{6}$$

The following condition describes region of application for presented method:

$$\sum_{i=1}^{J} n_j E(X_{ji}) < D \ . \tag{7}$$

In order to compute desired probability we have to find θ^* for which supremum of $F(\theta)$ is attained taking into consideration distribution function and moment generating function $M_j(\theta)$ of packet waiting time in used queuing model. Thus θ^* is positive root of the equation with derivative of $F(\theta^*)$:

$$F'(\theta^*) = 0 \text{, where } F'(\theta^*) = D - \sum_{j=1}^{J} n_j \frac{M_j'(\theta)}{M_j(\theta)} \ . \tag{8}$$

Refinement of the previous approximation method based on Chernoff bound is approach based on Bahadur-Rao Theorem (Local Limit Theorem) [9]. Probability that packet waiting time exceed certain value D can be approximated by the expression:

$$P\left(\sum_{j=1}^{J}\sum_{i=1}^{n_j} X_{ji} \geq D\right) \approx \frac{e^{-F(\theta^*)}}{\sqrt{2\pi}\cdot\theta^*\cdot\sigma(\theta^*)} \ , \tag{9}$$

where:

$$\sigma^2 = \sum_{j=1}^{J} n_j \frac{M_j''(\theta)\cdot M_j(\theta) - \left(M_j(\theta)\right)^2}{M_j{}^2(\theta)} \ . \tag{10}$$

Formulas presented above become simpler for homogenous case with queues belonging only to the single class, i.e. J=1. Moreover we can obtain n_{max}, which is such value that for all $n \leq n_{max}$ quality of service condition (3) is satisfied. In heterogeneous case with multiple classes, i.e. J>1, we can define *set of admissible nodes (queues) A* for which quality of service condition is satisfied:

$$A = \left\{\tilde{n} : P(X > D) \leq L\right\} \ , \tag{11}$$

where vector $\tilde{n}$ is described by J-tuples $\tilde{n} = (n_1, n_2, \ldots, n_J)$.

In Figure 2 we present boundaries of the *set of admissible nodes* for exemplary case of simplest heterogeneity with J=2 classes of nodes with $\rho_{CT,1}$=0.1, $\rho_{CT,2}$=0.3. That value of J was only chosen for the sake of presentation clarity. The respective points on the boundaries were calculated with the aid of Bahadur-Rao approximation, which seems to be the only solution to obtain accurate values of probability estimate (3) for heterogeneous case. We found boundaries of the *set of admissible nodes* for delay bounds D=1.0·10^{-3} and D=2.0·10^{-3} and respective values of delay probability L=10^{-2}÷10^{-6}. In Figure 2 it can be seen that the boundaries of the *set of admissible nodes* are highly linear. Thus we propose a method for linear development of that boundaries and describe $\tilde{n}$ as:

$$\tilde{n} : \sum_{j=1}^{J} n_j \cdot ed_j \leq D \ , \tag{12}$$

where

$$ed_j = \frac{D}{n_{\max,j}} \tag{13}$$

and $n_{max,j}$ is the maximum number of nodes for which quality of service condition (3) is satisfied where only nodes of *j*-th class are considered. We define ed_j as *effective delay* of *j*-th class of nodes.

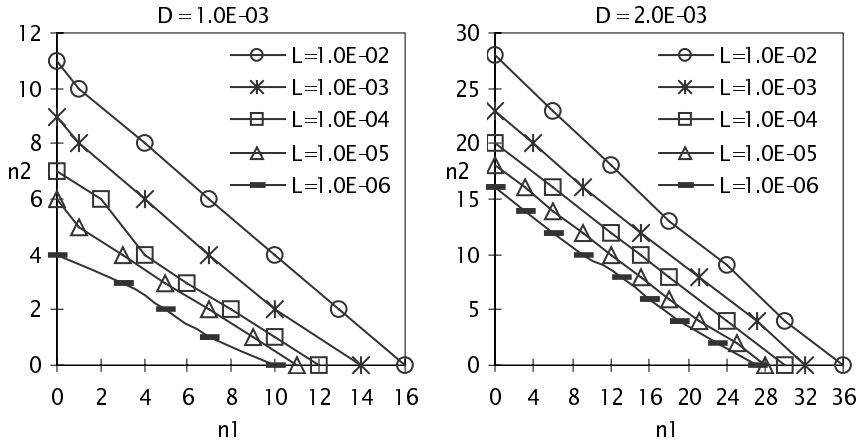


Fig. 2. Boundaries of s*ets of admissible nodes* for different delay bounds

That approach to calculation of the *set of admissible nodes* is accurate, simpler and relatively easier to perform than approximation based on calculation of respective probabilities for the heterogeneous case. Moreover in that approach we can apply any method of delay probability calculation, regardless of whether the method is general as Bahadur-Rao approximation, or limited to homogenous case as Erlang-n approximation. With that approach the only criterions for application particular method are complexity and accuracy. Last subject plays important role in all applications of the *effective delay*, because values of n_{max} obtained by particular probability approximation method to the same case strongly depend on accuracy of the method itself [21] (see also section with numerical results). Values of n_{max} can be overly optimistic as in the case of Gaussian approximation, which is optimistic for values of probability smaller than 10^{-2}, or overly pessimistic as in the case of probability approximation based on Chernoff bound. Thus, in the analysis we apply method providing the most accurate results of probability calculations, it is the method based on Bahadur-Rao approximation.

Presented framework can be used in Admission Control function for flows using EF PHB. The procedure works as follows:

- having statistical upper bound for queuing delay *D* with probability condition *L*,
- calculate the *effective delay* for every node along the EF stream path with formula (13),
- check whether condition (12) is satisfied, i.e. whether we are within the boundaries of *set of admissible nodes*,
- admit flow in case of the positive answer (statistical performance guarantees can be fulfilled).

Proposed algorithm can be directly applied to the network where centralized server (Bandwidth Broker - BB [22]) performs all admission control decisions for the DS domain. In that case network topology, routing and state of the domain including utilization on any link are known to the BB. Moreover, NJ conjecture makes that task relatively simple with only one parameter per link to collect. Concept of the *effective delay* can also be used in decentralized approach to admission control function [24], where admission decisions are taken within given resource amount by Admission Control Agents (ACA), closely related to particular Edge Routers. In that case *effective delay* can be used in the decisions performed by centralized Resource Control Agent (RCA) to optimize the global distribution of the resources for ACAs. Traffic description used in calculations of *effective delays* should then consider maximum of potentially admitted traffic by given distribution of resources to ACAs. In a fully distributed admission control function, where decisions are made in the nodes separately, a signaling request with given flow traffic parameters and its QoS parameters D, L must follow the path of stream. Also that signaling request must transport extra parameter which is the sum of *effective delays* of nodes (links) visited by the request (accumulated *effective delay*). In that case the admission control procedure must be slightly modified. Calculation of *effective delay* in the node is done after receiving signaling request, than the accumulated *effective delay* is updated and compared with D. Flow is admitted and request is passed forward if sum of *effective delays* remains smaller than D, indicating that statistical performance guarantees can be provided.

5 Numerical Results

In order to verify the accuracy of the presented methods, to examine proposed Admission Control algorithm and to verify the applicability of *effective delay* notion we considered the network of 5 nodes in the first scenario and the network of 15 nodes in the second case with heterogeneous J=5 classes regarding to queue offered load ρ_{CT}. The value of offered load in j-th class equals $0.1 \cdot j$, thus we cover the range of loads from 0.1 to 0.5 with step 0.1. Each interconnecting link has bandwidth 150 Mb/s and buffers for EF streams of size 20 packets. The observed traffic consisted of one CBR flow with rate 1.5 Mb/s (offered load ρ_T =0.01) and packet size equal to 100 bytes. Lower priority packets of size MTU=1500 bytes filled remaining link capacity in every node of the network.

In Figures 3 and 4 we present computation and simulation results in order to verify real possibilities of providing QoS guarantees in evaluated network and present framework for evaluation accuracy of proposed Admission Control method. However we would like to emphasize that proposed Admission Control method based on *effective delay* concept does not demand any direct delay probability calculations for the heterogeneous case. Also in that figures Chernoff and Gaussian approximations are presented only for sake of comparisons, because we focused on the delay probability calculations done by Bahadur-Rao approximation. In order to provide possibility of comparison between calculations and simulations we split packet delay into two components: the deterministic service time equal to $n \cdot \tau$ (τ is observed CBR flow packet service time) and the stochastic waiting time in queues modeled as a chain of n $M/D_{MTU}/1$ queues with vacations. Having parameters described above we calculated

probability that waiting time exceeds certain value D and consequently probability that end-to-end delay exceeds value $D+n\cdot\tau$ was easily obtained (see Figures 3 and 4).

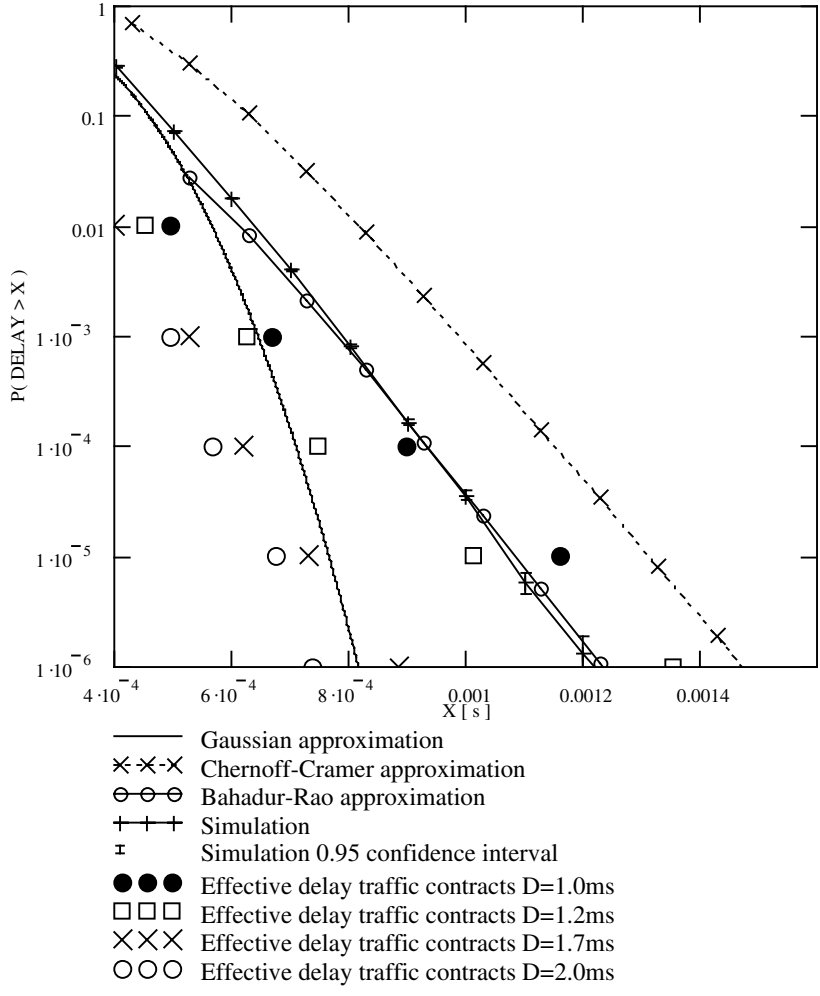


Fig. 3. Packet delay distribution of 100B packet CBR flow and effective delay representation of traffic contracts in heterogeneous network of n=5 nodes

In Figures 3 and 4 we also present results for calculations of individual traffic contract with the requested queuing delay D with probability L. We consider four different cases of delay $D_1=1.0\cdot10^{-3}$, $D_2=1.2\cdot10^{-3}$, $D_3=1.7\cdot10^{-3}$ and $D_4=2.0\cdot10^{-3}$ together with different probabilities L varied in the range from 10^{-2} to 10^{-6}. Each unique combination of parameters D and L represents different abstract user quality of service requirements and is indicated by the single point in Figures 3 and 4 with coordinates calcu-

lated as follows. The *x*-axis value was equal to the total *effective delay* for that particular traffic contract in the evaluated network calculated by formula:

$$ed_{tot} = \sum_{j=1}^{J} n_j \cdot ed_j \tag{14}$$

and *y*-axis value was equal to the delay probability requirement *L*.

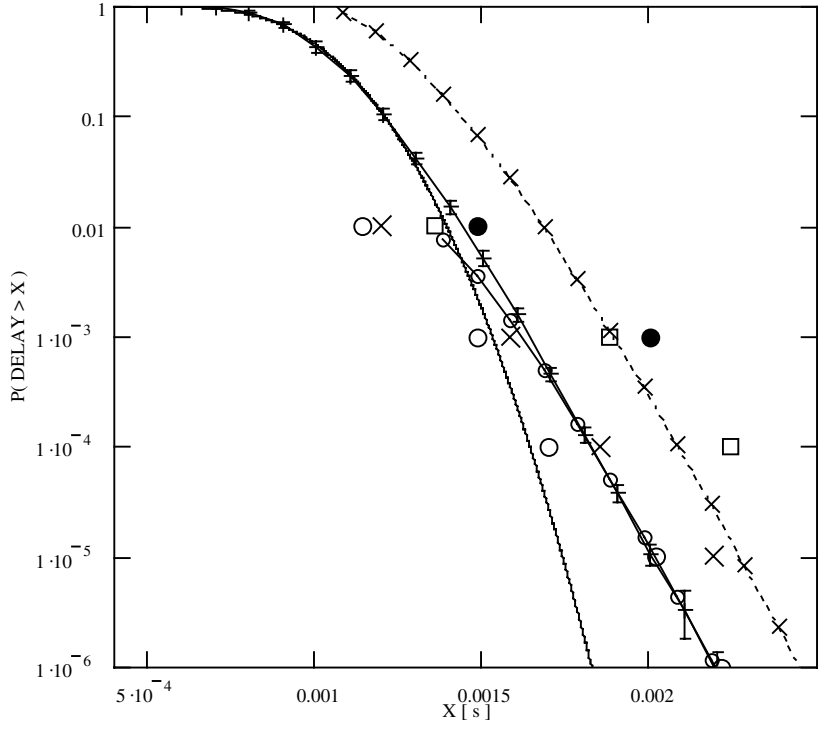


Fig. 4. Packet delay distribution of 100B packet CBR flow and effective delay representation of traffic contracts in heterogeneous network of n=15 nodes

Table 1 presents values of the lowest probability requirement *L* for which delay bound can be fulfilled for two different cases of evaluated networks. For example, in the case of 15 node network, according to direct calculations results for quality of service requirements for delay upper bound D_1 can not be fulfilled for probabilities lower than $1.7 \cdot 10^{-2}$, because for the value of delay D_1 probability equals that value, which is indicated by appropriate point on the delay distribution curve. Similarly for the delay upper bound D_2 performance guarantees can not be fulfilled for probabilities lower than $1.4 \cdot 10^{-2}$, in case of D_3 for $1.6 \cdot 10^{-4}$ and D_4 for $4.3 \cdot 10^{-6}$. That results in the position of points representing traffic contracts computed with aid of *effective delay* ap-

proximation. In all cases where performance bounds for EF stream can be guaranteed by network, condition (12) is satisfied. It can be seen in Figures 3 and 4 that for all those points (contracts) ed_{tot} values are smaller than delay value for respective points on the delay distribution curve (for the same probability value L).

Table 1. Direct calculations of lowest probability for which delay bound D can be provided

Delay bound D	5 node case	15 node case
$1.0 \cdot 10^{-3}$	$2.3 \cdot 10^{-5}$	$1.7 \cdot 10^{-2}$
$1.2 \cdot 10^{-3}$	$1.1 \cdot 10^{-5}$	$1.4 \cdot 10^{-2}$
$1.7 \cdot 10^{-3}$	$4.3 \cdot 10^{-10}$	$1.6 \cdot 10^{-4}$
$2.0 \cdot 10^{-3}$	$3.9 \cdot 10^{-12}$	$4.3 \cdot 10^{-6}$

6 Conclusions

Presented method of the admission control for EF flows seems to be robust and simple. It allows verification of possibility to provide statistical guarantees in the network with aggregate scheduling. The method is based on the concept of *effective delay* which itself is attractive and promising. The development of that concept is similar to the development of effective bandwidth for unbuffered resource [23]. Moreover, it seems that usefulness of *effective delay* plays the same role as the usefulness of effective bandwidth.

Effective delay concentrates in one single measure statistical upper bound of packet delay, related probability for that bound and state of the node (queue) in the path of the stream for which performance guarantees must be fulfilled. As can be seen, generalized concept of *effective delay* is independent of the particular method for evaluation of statistical bounds for delay probabilities. However precision of calculations for particular method influences accuracy of *effective delay* values and consequently accuracy of admission control function.

Proposed admission control method should be extended to include other types of queue models in the EF stream path. For example, evaluated network model can be extended to include the node with nD/D/1 queue with vacation which more precisely model ingress edge router in DiffServ network than M/D/1 queue with vacation.

References

1. Bennett, J.C.R., Benson, K., Charny, A., Courtney, W.F., Le Boudec, J.-Y.: Delay Jitter Bounds and Packet Scale Rate Guarantee for Expedited Forwarding. Proceedings of Infocom 2001, Anchorage, USA (2001)
2. Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., Weiss, W.: An Architecture for Differentiated Services. RFC 2475 (1998)
3. Bonald, T., Proutiere, A., Roberts, J.W.: Statistical Performance Guarantees for Streaming Flows using Expedited Forwarding. Proceedings of Infocom 2001, Anchorage, USA (2001)
4. Braden, R., Clark, D., Shenker, S.: Integrated Services in Internet Architecture: an Overview. RFC 1633 (1994)

5. Brichet F., Massoulie L., Roberts J.: Stochastic Ordering and the Notion of Negligible CDV. In: Ramaswami, V., Wirth, P.E. (eds.): Teletraffic Contributions for the Information Age. Proceedings of the 15th ITC, Elsevier Amsterdam Lausanne New York Oxford Shannon Tokyo (1997) 1433–1444
6. Charny, A. et al.: Supplemental Information for the New Definition of the EF PHB (Expedited Forwarding Per-Hop Behavior). RFC 3247 (2002)
7. Charny, A., Le Boudec J.-Y.: Delay Bounds in a Network with Aggregate Scheduling. In: Crowcroft, J., Roberts, J., Smirnov M. I., (eds.): Quality of Future Internet Services. Proceedings of First COST 263 International Workshop, QofIS2000. Lecture Notes in Computer Science, Vol. 1922. Springer-Verlag, Berlin Heidelberg New York (2000) 1–13
8. Davie, B. et al.: An Expedited Forwarding PHB (Per-Hop Behavior). RFC 3246, (2002)
9. Dembo, A., Zeitouni, O.: Large Deviations Techniques and Applications. Jones and Bartlet Publishers, Boston London (1992)
10. Doshi, B.T.: Generalizations of the Stochastic Decomposition Results for Single Server Queues with Vacations. Comm. Statistic. Stochastic Models, Vol. 6, No. 2 (1990) 307–333
11. Doshi, B.T.: A Note on Stochastic Decomposition in a GI/G/1 Queue with Vacations or Set-up Times. Journal of Applied. Probability, Vol. 22 (1985) 419–428.
12. Doshi B. T.: Queuing Systems with Vacations - A Survey. Queuing Systems Theory and Applications. Vol. 1 (1986) 29–66
13. Firoiu, V., Le Boudec, J.-Y., Towsley, D., Zhang, Z.-L.: Theories and Models for Internet Quality of Service. Proceedings of the IEEE, Vol. 90, No. 9 (2002) 1565–1591
14. Fuhrmann, S.W.: A Note on The M/G/1 Queue with Server Vacation. Operations Research, Vol. 32 (1984) 1368–1373
15. Fuhrmann, S. W., Cooper, R. B.: Stochastic Decompositions in a M/G/1 Queue with Generalized Vacation. Operations Research, Vol. 33, No. 5 (1985) 1117–1129
16. Goderis, D. et al.: Service Level Specification Semantics, Parameters and Negotiation Requirements. Internet-Draft: draft-tequila-sls-01.txt, work in progress (2001)
17. Grossglauser M., Keshav S.: On CBR Service (extended version). Proceedings of Infocom 1996 (1996)
18. Jacobson, V., Nichols, K., Poduri K.: An Expedited Forwarding PHB. RFC 2598 (1999)
19. Karam, M., Tobagi, F.: Analysis of the Delay and Jitter of Voice Traffic Over the Internet. Proceedings of Infocom 2001 (2001)
20. Mandjes, M., van der Wal, K., Kooij, R., Bastiaansen H.: End-to-end Delay Analysis for Interactive Services on a Large-scale IP Network. Proceedings of the 7th IFIP Workshop on Performance Modeling and Evaluation of ATM/IP networks, Antwerp, Netherlands (1999)
21. Narloch, N., Kaczmarek, S.: Methods for Evaluation Packet Delay Distribution of Flows Using Expedited Forwarding PHB. Proceedings of 2nd Polish-German Teletraffic Symposium PGTS2002, Gdansk (2002) 85–94
22. Nichols, K. et al.: A Two-bit Differentiated Services Architecture for the Internet. RFC 2638 (1999)
23. Roberts, J., Mocci, U., Virtamo, J. (eds.): Broadband Network Teletraffic. Performance Evaluation and Design of Broadband Multiservice Networks. (Final Report of COST 242), Springer Verlag, Heidelberg (1996)
24. Salsano, S. et al.: Definition and Usage of SLS in the AQUILA Consortium. Internet Draft: draft-salsano-aquila-sls-00.txt, work in progress (2000)
25. Shenker, S., Partridge, C., Guerin, R.: Specification of Guaranteed Quality of Service. RFC 2212 (1997)
26. Vojnovic, M., Le Boudec, J.-Y.: Stochastic Analysis of Some Expedited Forwarding Networks. Proceedings of Infocom 2002, New-York (2002)

QoS Provisioning for VoIP Traffic by Deploying Admission Control

Hung Tuan Tran[1], Thomas Ziegler[1], and Fabio Ricciato[2]

[1] Telecommunications Research Center Vienna (ftw.)
Donaucity Strasse 1, 1220, Vienna, Austria
{tran,ziegler}@ftw.at
[2] INFO-COM Department, Univ. of Rome "La Sapienza"
Via Eudossiana 18, 00184 Rome, Italy
ricciato@coritel.it

Abstract. In this paper we propose a model-based admission control scheme for maintaining QoS of voice traffic over DiffServ networks. This CAC approach implies two main components. The first one is the application of the NJ (Negligible Jitter) conjecture extended to the case of heterogenous variable bit rate voice sources. The second one is the analysis of a finite queueing system with exhaustive service and multiple server vacations, which is motivated by the strict priority scheduling deployed in DiffServ-capable routers. Extensive analytical and simulation results are investigated to assess the applicability of the CAC proposal.

1 Introduction

The pure best-effort nature of services provided over the worldwide Internet today seems to be insufficient in many cases, as the demand for quality of service (QoS) increasingly emerges. One of the most popular application over the Internet that could be offered with guaranteed QoS is real-time, particularly streaming voice application. For this real time application (often referred to as Voice over IP), strict end-to-end delay, jitter and loss should be delivered (e.g. less than 150ms end-to-end delay and 1% packet loss rate).

Over the last years, the challenge of providing QoS has stimulated a huge amount of research efforts resulting in a number of potential QoS architectures like IntServ [4], DiffServ [5], DPS [8]. Up to now, the DiffServ concept seems to be the most prominent due to its salient scalability and feasibility. However, because QoS is only delivered to a few classes of traffic inside the DiffServ domain, we need connection admission control (CAC) to ensure the QoS level of each real-time traffic flow. Basically, the task of CAC mechanisms is that given the required specifications in terms of loss or/and delay or/and jitter of voice flows and an amount of available resources, to decide whether a new voice flow can be admitted with guaranteed QoS and (at the same time) without destroying the QoS of other, currently active voice flows.

In this paper we propose a two-stage CAC approach for voice traffic flows with loss and delay requirements. We extend the use of the recently divulged negligible

W. Burakowski et al. (Eds.): Art-QoS 2003, LNCS 2698, pp. 139–153, 2003.

jitter conjecture [3] to the case of heterogeneous traffic flow by considering the rate envelope multiplexing [7] principle. Afterwards, in order to make admission control decisions we perform analysis of a *finite* M/D/1/K queue with exhaustive service and multiple server vacation. This queue is introduced to mimic the priority scheduling applied in the DiffSserv architecture for real-time voice and best-effort traffic. Moreover, we build up simulation runs reflecting realistic network scenarios to examine and verify the merit of the proposed model-based CAC scheme.

The paper is organised as follows. In Section 2 we describe the proposed two-stage CAC mechanism for a single bottleneck link case. Several aspects are presented including the motivations, the basic operation, the theoretical background and the realization model. The computational procedures for relevant performance parameters are also explained in details in this section. We present a possible extension of the proposed CAC mechanism to a network environment in Section 3. In order to evaluate the practical applicability of the model used in the CAC scheme, we perform both analytical and simulation analysis. Section 4 contains the description of analytical and simulation scenarios, the obtained results and their discussions. Finally, Section 5 ends the paper.

2 A Two-Stage CAC Scheme for the Single Bottleneck Link Case

2.1 Informal Description

The idea of our two-stage CAC is originally inspired by the work of [3]. The authors state therein the so called NJ (negligible jitter) conjecture that provides a useful methodology for traffic engineering of EF (Expedited Forwarding) traffic in DiffServ networks. The NJ conjecture sketches that if

- the network realizes priority queueing for EF traffic at each router,
- EF flows have negligible jitter at the ingress router with respect to a Poisson process with MTU packet size, and
- at every multiplexing stage (router) within the network the sum of input rates is less than the service rate,

then throughout the network the EF flows behave better than a Poisson stream with MTU packet size, i.e the EF flows can be replaced by a Poisson process for dimensioning rules. The effect of jitter then is ignored while applying engineering rules to the flows. The statistical bound for the queue length (and so for the queueing delay) therein is provided by the expression

$$P(Q > x) \leq \begin{cases} 1 & x < x_{min} \\ ke^{-rx/MTU} & x \geq x_{min} \end{cases}, \tag{1}$$

where Q is the queue length, r is a root of the equation $\rho(e^r - 1) - r = 0$, $k = \dfrac{1-\rho}{\rho^2 e^r - \rho}$, $x_{min} = \dfrac{-MTU}{r} \log \dfrac{1}{k}$ and ρ is the utilization of the assimilated

infinite queue. The NJ conjecture has been affirmed with both theoretical arguments based on considerations of stochastic ordering and with simulation results. However, only the case when the EF traffic is generated by CBR sources with constant packet size was dealt in details, leaving the case of VBR traffic with different packet size touched at the mentioning level.

Suppose now that we have to perform admission control for *heterogenous* voice flows at the output link of a given node in the network. We split the CAC functionalities into two modules in sequence. The first module ensures that the rate of the aggregate voice traffic only exceeds the capacity of the output link with a small probability (called rate overloading probability). Moreover, whenever rate overload takes place, it discards the excess traffic. Our motivation of introducing such the first module is to make the NJ conjecture applicable (see the third prerequisite of the original NJ conjecture) and thus to facilitate further analysis needed in the second module for loss and delay related requirements of the aggregate voice traffic with respect to the presence of the best-effort traffic. Consequently, when a new voice flow requesting admission arrives, a two-stage CAC procedure is performed as follows.

- Decision 1: if the new flow does not make the rate overloading probability increase beyond the predefined one (denoted by $e^{-\gamma}$), it is admitted by the first module and is passed to the second module. Otherwise, the flow is rejected and the CAC procedure ends.
- Decision 2: if the new flow is admitted to the second module, a numerical analysis is done to check further delay and loss metrics of the voice flows. The computation procedure for the assimilated model (see later in Section 2.2) is achieved to check the delay metric $d_{current}$ and the buffer overflow probability $p_{current}$ taking into account also the load produced by the new flow. Let d_{req} and p_{req} be the delay and loss requirements of the voice flows. If $d_{req} > d_{current}$ and $p_{req} > p_{current} + e^{-\gamma}$, the new flow is accepted, otherwise it is rejected.

From engineering aspects, the rate envelope multiplexing (REM) concept [7] is appropriate to be applied for the first module. The REM module ensures that the rate overload probability determined as $\frac{E(\Lambda_t - C)^+}{E(\Lambda_t)}$, where C is the output link capacity, Λ_t is the aggregate input rate and $(.)^+ = max(0, .)$, is below the predefined threshold $e^{-\gamma}$. For the second module, an analytical model is needed. The application of the NJ conjecture and the consideration of the priority scheduling scheme deployed in the DiffServ architecture will lead to the analysis task of a *finite* $M/D/1/K$ with exhaustive service and multiple server vacations.

2.2 Realization

Our approach to realize the two-stage CAC scheme is shown in Fig. 1. The REM module in this realization is done with effective bandwidth based multiplexing. The effective bandwidths of voice flows are derived from the Chernoff inequality

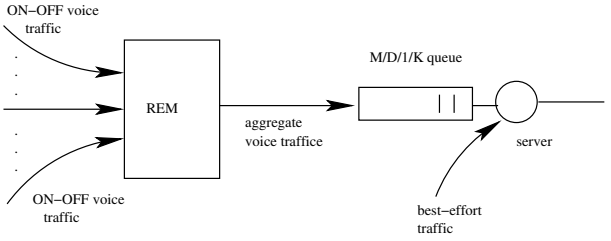


Fig. 1. A possible realization of the CAC approach

[7]. The reason for this choice is because the Chernoff bound approach can give a tight upper bound on the rate overload probability. Moreover, it is capable to deal with multiple classes of ON-OFF input traffic sources, which is typically the case of voice traffic stemming from different codecs. For the details on the implementation of the REM module, we refer the interested readers to [9].

It is assured by the REM module that the aggregate input rate only exceeds the rate of the output link with a predefined and sufficiently small probability. Thus, owing to the statement of the NJ conjecture, we may now assimilate the aggregate voice traffic with a Poisson process with the same load and with constant packet size. We adopt a queueing model where voice packets are fed into a *finite buffer* and served by a server representing the output link. Note that with the consideration of the finite queue, we will be able to examine both delay and loss metrics of voice traffic. Due to the presence of best effort traffic and to the non-preemptive priority scheduling, the operation of the server is considered in a manner of exhaustive service and multiple vacation scenario. That is the server delivers voice packets in the finite buffer until it becomes empty. At the finishing instant of the service, if the server finds the queue empty, it takes vacation. If there are still no packets in the queue when the server returns from its vacation, it takes another vacation and so on. The vacations of the server correspond to the situation when the output link is occupied by best effort packets. The assumption of multiple vacation implies that the offered load of the best effort traffic is sufficiently high to immediately utilize the link capacity whenever no voice packet is present. As a worst-case assumption, the vacation time is assumed to be the time needed for transmission of a best effort packet with MTU size. In effects, we obtain a finite M/D/1/K queue with exhaustive service and multiple server vacation.

We solve the steady state distribution of this queue based on the work available from [6] and accomplish some further derivations which enable the calculation of important statistical measures. We are particularly interested in queueing performance parameters related to the delay metric. The rationale behind this is that the queueing delay is the sole variant component contributing to the end-to-end delay of a given voice connection. Moreover, since it is the sole variable component, the jitter behavior of the voice traffic is basically determined by the queueing delay. Specifically, the following parameters are considered:

– *Mean queueing delay*: This parameter is calculable by means of Little's law

$$d_{avg} = \frac{\text{Mean queue length}}{\lambda(1 - p_{loss})}, \tag{2}$$

where λ is the mean packet arrival rate and p_{loss} is the packet loss probability calculated by expression (6).

– *p-percentile of queueing delay*: Although the mean queueing delay is definitely of interest, the p-percentile of delay could be a better parameter to characterize the end-to-end delay behavior of the voice connection, because it provides statistical information on the upper-bound of jitter. Having the delay value d of the p-percentile, the probability that the queueing delay is larger than d is at most $1 - p$, i.e.

$$P(delay \geq d) \leq 1 - p. \tag{3}$$

In general, the p-percentile of queueing delay can be determined exactly, if the density function of the queueing time is known. Unfortunately, the density function in the time domain is unavailable. Therefore, we propose here the solution derived from the Laplace domain analysis. In [6], the LST (Laplace-Stieltjes transform) of the queueing time distribution (which is identical to the Laplace transform of the density function) is given as

$$W(s) = S(s)^K \sum_{j=0}^{K} \pi_j \left(\frac{\lambda}{\lambda - s}\right)^{K+1-j} + \frac{\lambda\pi_0}{1 - h_0} \frac{\left[1 - \left(\frac{\lambda S(s)}{\lambda - s}\right)^{K+1}\right](V(s) - 1)}{\lambda - s - \lambda S(s)}, \tag{4}$$

where π_j is the steady state probability that j voice packets are left in the system at a departure epoch, $S(s)$ and $V(s)$ are LSTs of the service and vacation time, respectively. To transform back this expression into the time domain is very hard, if not impossible. Thus, in order to obtain the p-percentile for a given p, we resort to an approximation using the Chernoff inequality in probabilistic theory, which gives

$$P(delay \geq d) \leq \inf_s \frac{W(-s)}{e^{ds}}. \tag{5}$$

To find the infimum of the right hand $G(s) = \dfrac{W(-s)}{e^{ds}}$ of the inequality (5) we have to solve the transcendental equation $\dfrac{dG(s)}{ds} = 0$. As one can recognize from equation (4), the expression for $G(s)$ is quite complicate and so is its derivation. The best suitable numerical method we are aware of in this case is the Secant method [1]. In fact, this method has been chosen to be implemented in our work. Another observation is that the computation can only be done in such a way, that the value d must be given as an input parameter, based on which the value $p = 1 - \inf_s G(s)$ is calculated. This is in contrast with the original intention according to which p should be

1. Set an appropriately, small initial d, and set $p = 0$
2. $WHILE$ $(p < p_{need})$
   3.1. $d = d + d_{step}$
   3.1. Find $s^* = \arg\inf_s G(s, d)$
   3.2. Compute $p = 1 - G(s^*, d)$
3. Return d and p

Fig. 2. Calculation procedure for the delay percentile

given as an input parameter and d is the output parameter. However, this contradiction can be resolved by a computational procedure presented in Fig. 2, where d_{step} stands for the increment with a sufficient granularity to reach the required p. It should be pointed out, however, that to make the complexity reasonably low, a binary search may be involved instead of the linear search shown in Fig. 2. This is particularly useful, when new flows arrive frequently, leading to frequent executions of the calculation procedure to conduct admissibility decisions.

– *Packet loss probability due to buffer overflow*: From [6]

$$p_{loss} = 1 - \frac{(1 - h_0)\lambda^{-1}}{E(V)\pi_0 + E(S)(1 - h_0)}, \tag{6}$$

where $E(V)$, $E(S)$ are the mean service time and mean vacation time, h_0 is the probability that no voice packet arrival occur during a vacation time V, π_0 is the steady state probability that no voice packet is left in the system at a departure epoch.

3 Extension of the CAC Scheme to the Case of Multiple Bottleneck Links

The most straightforward extension of the proposed CAC mechanism to the case of multiple node in a network environment can be described as follows. Whenever a new flow arrives at the network ingress, its path through the network is defined. We assume that the path of the flow traverses n nodes and the individual behavior observed at each node is independent. All packets of the flow will follow this path. The new flow declares to the network control entity its mean and peak rate, as well as its end-to-end loss and statistical end-to-end delay requirement as $P_{loss} < \epsilon_l$, $P(delay < d) \geq \epsilon_d$.

If the congestion state along the path is considered evenly distributed between its nodes, then at each node the statistical delay bound the flow requires could be the original end-to-end statistical one divided by the total number of nodes in the flow's path and so is the loss requirement. More precisely, at each node we have to ensure that $P_{loss,node} < \frac{\epsilon_l}{n}$, and $P(delay < d/n) \geq (\epsilon_d)^{1/n}$ by the two-stage CAC mechanism proposed in the previous section.

The CAC decisions are done in a hop-by-hop manner, for instance, on the way back, i.e. the last node in the path makes its decision first. If it can accept

the new flow, the next CAC decision is performed at the next upward node and so on. If the CAC decision at any internal node results in rejection then the edge node eventually learns about it and the whole CAC procedure ends.

Note that at each node, the CAC procedure requires further the knowledge about the number of streaming flows currently multiplexed at the output link in question, as well as the traffic description of each streaming flow (mean and peak rate). However, to gain better scalability, flows can be grouped into classes according to certain rules and then it could be sufficient to store only the mean and peak rate of the traffic class the flow belongs to.

4 Analytical and Simulative Discussions

Our goal in this section is to investigate the relation between performance (delay and loss) metrics acquired with the proposed analytical model and with the corresponding simulation scenarios. This will shed light on the merit of the proposed analytical model, based on which CAC decisions are conducted.

In our previous work [9], we have come to the conclusion that the proposed CAC model gives a good prediction on both delay and loss metrics if the mean packet size approach is followed. That is in the assimilated Poisson process, the packet size is set to the average of all the packet sizes currently being in the system, instead of MTU size suggested by the original NJ conjecture. However, the investigation therein relates only to the case of a single bottleneck link. In this paper, we extend the assessment to the case of multiple bottleneck links and produce extensive analytical and simulative comparative results. Because of the before-mentioned observation, this time we only deal with the mean packet size approach.

4.1 Simulation Scenarios

We consider voice streaming traffic in the following manner. With the application of voice activities detection, the voice traffic typically has ON-OFF nature. During the ON period, voice packets of identical length are generated periodically, while no packets are produced over the OFF period. The ON-OFF voice sources have the mean time of ON period 0.35s and the mean time of OFF period 0.65s. The packet length and the periodicity of packet generation depend on the type of a voice codec applied at the source. Basically, in our analysis we take three types of voice codecs into consideration: G723.1 (type 1), G726 (type 2) and G729 (type 3). The codec characteristics are summarized in Table 1.

For simulation investigations, we use the network topology shown in Fig. 3. The target aggregate voice traffic is supposed to traverse from host h_1 to host h_{2n+2} along a transmission path compromising several routers R_1, R_2, ..., R_n, and consequently several bottleneck links between the routers. To mimic the real-life traffic multiplexing situation and to stay in accordance with the developed analytical model, at each internal router there are two kinds of cross traffic. Namely, voice cross traffic and Web-like cross traffic representing the presence of

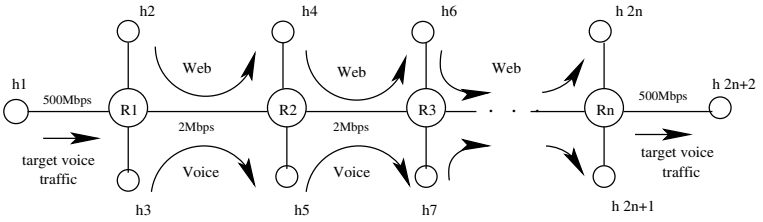


Fig. 3. The topology used in simulation for the test of the CAC approach

the best-effort traffic are considered. The Web-like traffic is generated according to the model of [2]. To achieve a reasonably high load, 500 Web sessions are simultaneously established. If not stated otherwise, we construct the worst case in simulation by setting the size of the web-packets to the value of MTU (1500 bytes). Each voice and Web-like cross traffic flow, however, resorts to only one bottleneck link of the target transmission path. That is, each voice cross flow goes from host h_{2i+1} to host h_{2i+3}, and each Web-like flow goes from host h_{2i} to host h_{2i+2} $(i = 1, 2...)$.

All the access links between a given host and a given router have a capacity of 500Mbps and 5ms propagation delay. All the bottleneck links have identical capacity and the same propagation delay of 10ms. The number of bottleneck links in the transmission path and their concrete capacity are subject to change accordingly to the investigation aim. At each output link of the routers along the transmission path, non-preemptive priority scheduling for voice and best-effort traffic is implemented. A high priority queue is set up for the aggregate voice traffic and the low priority one is for the best-effort traffic. In all cases, the buffer capacity for voice traffic is measured in the number of packets with size identical to the average size of all packet types being present in the system. The reported results are concerned with the 99-th percentile delay and packet loss probability. In order to collect correct stationary results, each simulation run lasts for 2500 seconds in simulation time.

4.2 Discussions

Homogeneous Voice Sources

Firstly, we restrict our attention to the case when only one type of voice sources is present in the network. That is all the target and the cross voice flows belong to the same source type, which is chosen to be G723.1 in our case. We fix the number of voice cross flows at each internal node along the path to be 50 and

Table 1. Characteristics of some voice codecs

Codec	Packet size (payload + 40 byte header)	Packet interval-time during ON period	Peak rate during ON period
G723.1	64 byte	30.5ms	16.78Kbps
G726	120 byte	20ms	48Kbps
G729	60 byte	20ms	24Kbps

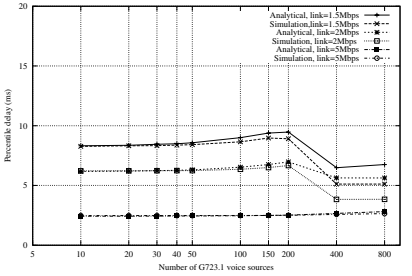


Fig. 4. Percentile delay evolution in case of 1 bottleneck link with different capacities. The buffer size is set to 15 packets

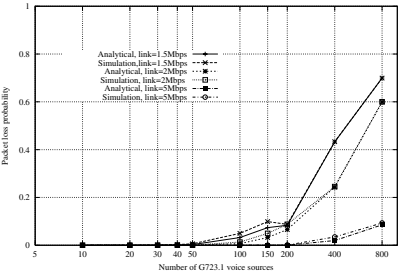


Fig. 5. Packet loss behavior in case of 1 bottleneck link with different capacity. The buffer size is set to 15 packets

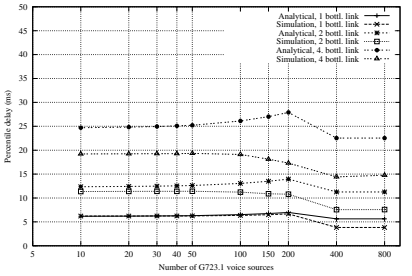


Fig. 6. Percentile delay evolution for different number of bottleneck links with capacity of 2Mbps. The buffer size is set to 15 packets

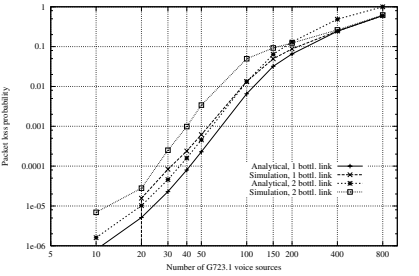


Fig. 7. Packet loss behavior for different number of bottleneck links with capacity of 2Mbps. The buffer size is set to 15 packets

check the behaviour of loss and delay parameters while varying the number of target flows.

From Fig. 4, we observe that concerning the delay percentile, the analytical results provide very tight upperbounds for those obtained with simulation. The closeness only becomes somewhat looser when the offered load is extremely heavy (e.g. with 400 and more voice flows) and the link has relatively low speed (1.5Mbps or 2Mbps). This deterioration is due to the approximation applied during the computation of the analytical results which has been outlined in Section 2.2. It is interesting to note that at such a very high load of voice traffic, the delay percentile tends to become constant and even smaller than that observed in low load situations. This is due to the effect of best effort traffic. When the load of voice traffic is low, the high priority queue for voice packets may be empty, allowing best effort packets to have a chance to occupy the link. Consequently, an arriving voice packet has to wait not only for the service completion of other voice packets in front of it in the queue, but also for the service completion of the best effort packet currently in service (because of the non-preemptive discipline). When the load of voice traffic is very high, the high priority queue is practically

never empty, so that the delay of a given voice packet is solely the time to complete service of voice packets in front of it in the queue. Given that the length of best-effort packets (1500 byte) is considerably large compared to that of voice packets (64 byte), the beforementioned delay evolution is experienced.

Regarding the packet loss probability, it is well remarkable from Fig. 5 that when the simulated scenario gives zero packet loss probability, so does the analytical analysis. When loss events begin to show up, simulation results are a little bit greater than those computed with the analytical model under moderate loads (e.g. up to 200 voice flows in case of 1.5Mbps bottleneck link). This is because with the assimilated Poisson process in our analytical model, we are less able to capture the losses caused be simultaneous arrivals of voice packets from different sources. However, for heavy loads, simulation and analytical results are exactly identical. Note that the remarks above remain in place with the increase of the bottleneck link capacity, as shown in the figures.

The impact of the number of bottleneck links on the simulative-analytical result relation is demonstrated in Fig. 6 and Fig. 7. In Fig. 6, one can see again that the delay curves of simulation and analytical model match each other quite well, particularly when there is only 1 bottleneck link. The delay bound provided by the analytical model becomes coarser as more bottleneck links are involved in the path. The reason behind this is that in the analytical model we have adopted a conservative assumption of independence between the internal nodes of the path, leading to the summation of delay metric observed at each node.

Concerning the packet loss probability, we can see in Fig. 7 (in this figure, for better visibility, we skip the cases of more than 2 bottleneck links) that the shape of the curves of simulation and analytical model are similar. The analytical model, however, underestimates the loss observed with simulation.

The next figures show the effect of the buffer size on the shape of the interested performance parameters. Figure 8 and Fig. 9 allow to draw the same conclusions about the close relation between the delay percentiles retrieved with simulation and with the analytical model for different buffer sizes. For the packet loss probability, Fig. 10 and Fig. 11 imply the fact that in case of small buffer size (around of 15 packets), the analytical model slightly underestimates the results from simulation. This tendency, however, disappears for larger buffer sizes (more than 30 packets), when the two kinds of results perfectly match each other. Note that when there are no loss events (e.g. when the buffer size is set to 50 or 100 packets), zero packet loss rate is obtained in both simulation and analytical model (in the figures, the vertical line means that the loss rate drops to zero).

Heterogeneous Voice Sources

We now consider the case when three types of voice flows are mixed at each internal node along the path. Cross voice flows are chosen to be either G726 or G729 type. At each internal node, new voice flows of type G726 and G729 join the target G723.1 aggregate traffic. From Fig. 12 and Fig. 14, the familiar phenomena is again observed about the quite good delay inter-relation. The underestimation of packet loss rate in case of small buffer size is still captured

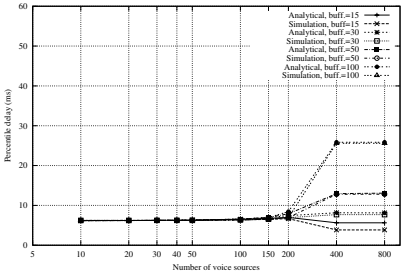


Fig. 8. Delay evolution for different buffer sizes. The transmission path consists of 1 bottleneck link with 2Mbps capacity

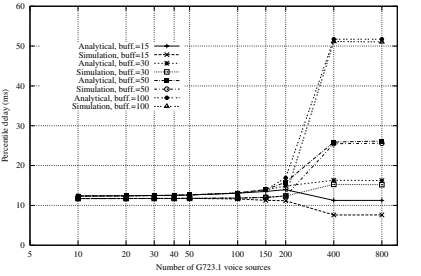


Fig. 9. Delay evolution for different buffer sizes. The transmission path consists of 2 bottleneck links with 2Mbps capacity

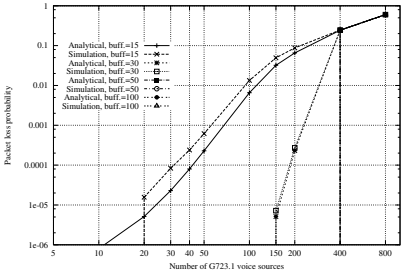


Fig. 10. Packet loss behaviour for different buffer sizes. The transmission path consists of 1 bottleneck link with 2Mbps capacity

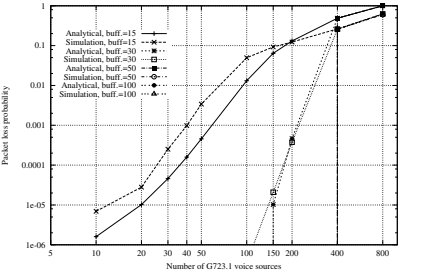


Fig. 11. Packet loss behavior for different buffer sizes. The transmission path consists of 2 bottleneck links with 2Mbps capacity

in Fig. 13 and Fig. 15 (in this figure, when the buffer size is 50 packets, no loss event is detected by both simulation and analytical results).

We have also performed the analysis with several other source combinations at each internal node and we observe no significant change in the relation concerning delay and loss evolution between simulation and analytical results.

Implication of the Proposed Analytical Model

The obtained results have indicated that when we use the mean packet size in the assimilated Poisson model, a tight upper bound for the delay is obtainable. Thus, from the aspect of delay metric, one can safely rely on the analytical model in order to make admission decisions. However, for small buffer sizes (around of 15 packets) the loss rate probability is underestimated. The latter fact, however, does not limit the applicability of the analytical model to CAC decisions due to the following reasons.

We should keep in mind that the simulation scenarios with all best effort packets of 1500 bytes represent the worst case from the aspect of the voice traffic. This is of course not the case in reality, where the average packet size of

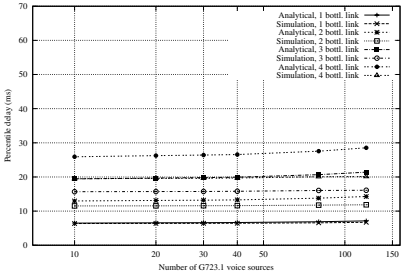


Fig. 12. Delay evolution for different numbers of bottleneck links with 2Mbps capacity. At each internal node, 25 G726-type and 20 G729-type fresh flows join to the output link. The buffer size is set to 15 packets

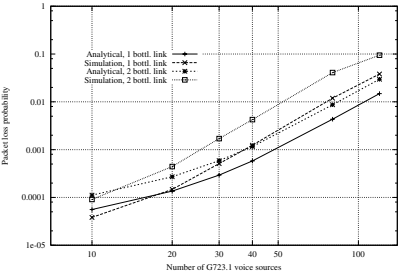


Fig. 13. Packet loss behavior in case of 1 or 2 bottleneck links with 2Mbps capacity. At each internal node, 25 G726-type and 20 G729-type fresh flows join to the output link. The buffer size is set to 15 packets

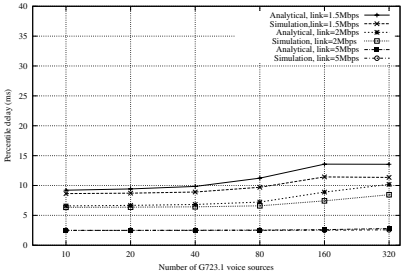


Fig. 14. Delay evolution in case of 1 bottleneck link with different capacities. At each internal node, 15 G726-type and 50 G729-type fresh flows join to the output link. The buffer size is set to 30 packets

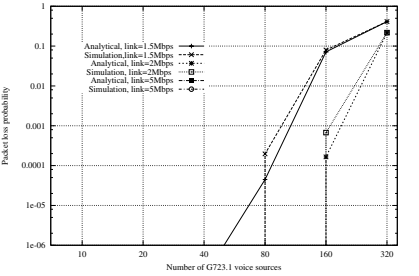


Fig. 15. Packet loss behavior in case of 1 bottleneck link with different capacities. At each internal node, 15 G726-type and 50 G729-type fresh flows join to the output link. The buffer size is set to 30 packets

the best effort traffic is typically smaller, e.g. in a range of 500 bytes. With this consideration, the analytical results safely give upperbounds for both delay and loss results as demonstrated in Figs. 16, 17, 18, 19, and thus can be involved to conduct admissibility decisions from both loss and delay aspects. As a side remark, we note that even in the worst case when every best effort packet has a maximum size of 1500 bytes, we have an another option to increase the packet size of the assimilated model to get upper-bound for the packet loss probability as well. Apparently, this manipulation also introduces larger delay metrics. By increasing the packet size with a proper scaling factor we can reach the situation when both delay and loss are bounded by those provided by the analytical results.

Finally, we present an example of the admission region conducted with the proposed analytical model. Recall that whenever a new target voice flow intends

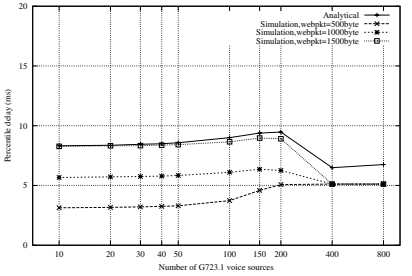


Fig. 16. Effect of web-packet size on the evolution of percentile delay. The bottleneck link's capacity is 1.5Mbps, the buffer size is 15 packets

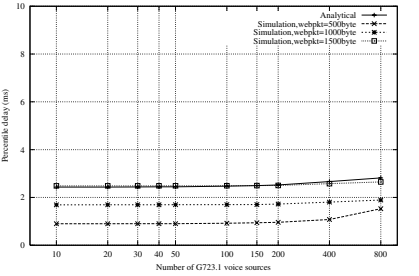


Fig. 17. Effect of web-packet size on the evolution of percentile delay. The bottleneck link's capacity is 5Mbps. The buffer size is 15 packets

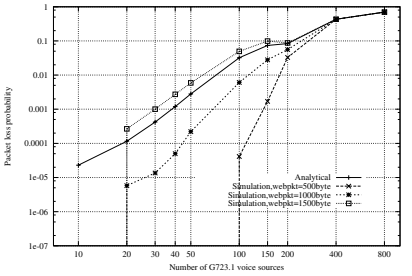


Fig. 18. Effect of web-packet size on the evolution of packet loss rate. The bottleneck link's capacity is 1.5Mbps. The buffer size is 15 packets

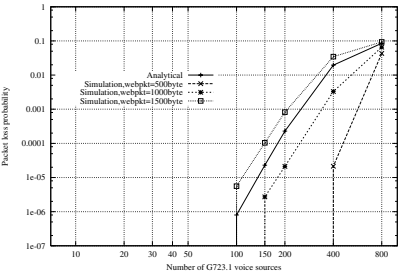


Fig. 19. Effect of web packet-size on the evolution of packet loss rate. The bottleneck link's capacity is 5Mbps. The buffer size is 15 packets

to join the aggregate traffic from the near-end node toward the far-end one, a CAC decision must be performed at each internal node considering also the traffic load the new flow would contribute. On the other hand, at each internal node of the transmission path, whenever a new voice flow belonging to the cross traffic is generated, admission control must also be performed. In both cases, the QoS criteria (loss and delay requirement) of this CAC decision is basically defined and influenced by the target traffic, or more precisely by the length of the transmission path the target traffic traverses. This is because the per-node loss and delay requirements are derived from the end-to-end ones in an adequate way as described earlier in Section 3. Table 2 illustrates the corresponding relation between the end-to-end and per-node requirements[1].

[1] Recall that we currently consider the case when congestion is evenly distributed among the internal nodes. A more general situation when each node along the target path has it own congestion state (but together they deduce the required end-to-end requirements) is a subject of further work.

Table 2. Illustrative relation between end-to-end and per-node delay requirements

Per-node		2 bottleneck link		3 bottleneck link		4 bottleneck link	
Percentile	Delay value	Percentile	Delay value	Percentile	Delay value	Percentile	Delay value
99 %	d	98.01 %	2d	97.03 %	3d	96.06%	4d
99.9 %	d	99.80 %	2d	99.70 %	3d	99.60%	4d

Figure 20 shows the per-node CAC region when the link capacity is 2Mbps, the 99-th delay percentile for each node is required to be $7.2ms$ and the end-to-end loss requirement is 10^{-3}. When the buffer size is set to 15 packets at each internal node, the prolongation of the transmission path results in a smaller per-node admissible region. This is because the per-node loss requirement becomes stricter with the length of the transmission path. However, if the buffer size is set to 50 packets, the analytical model gives a negligible packet loss rate, which is in a range of 10^{-6} (see Table 3). Thus, in this case the value of 10^{-3} for the end-to-end loss requirement indeed does not have effect on the admissible region, because it is automatically fulfilled even for a number of bottleneck links. The require percentile delay then is the sole crucial factor deciding the form of the CAC region. In fact, we get a larger per-node admissible region than the one related to the case of buffer of 15 packets.

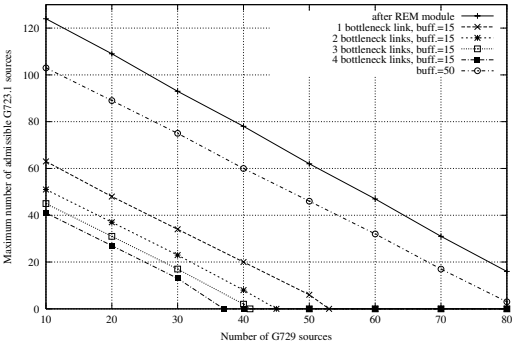


Fig. 20. The per-node CAC region with predefined loss and delay requirements (#G726=25)

5 Conclusion

We have proposed a CAC scheme for provisioning QoS to voice traffic over DiffServ networks. This scheme is based on the extended NJ conjecture and the analysis of an $M/D/1/K$ queue with multiple server vacations. The criteria for admission of voice traffic covers both end-to-end loss and delay requirement.

The model of the scheme is indeed useful to conduct admission control for voice flows because it predicts very well delay metrics observed in the network.

Table 3. Per-node CAC regions, 1 bottleneck link with 2Mbps capacity, the 99th delay percentile is 7.2ms, buffer size is 50 packets

#G729	#G726	max. #G723.1 after the 1st module	max #G723.1 after the 2nd module	$P_{loss,anal}$	$P_{loss,sim.}$
10	25	124	103	6.57e-6	0
20	25	109	89	6.57e-6	0
30	25	93	75	6.57e-6	0
40	25	78	60	6.57e-6	0
50	25	62	46	6.57e-6	0
60	25	47	32	6.57e-6	0
70	25	31	17	6.57e-6	0
80	25	16	3	6.57e-6	0

It also provides a good loss metric when not too small buffer sizes are employed. It should be kept in mind that the analytical model will anyway do conservative dimensioning due to the implication of the worst-case aspects related to the heavy load and the packet length of the background traffic. However, this is needed to assure quality for voice traffic.

There are some issues requiring further work. For example, to increase the utilization of the REM module (by using another approach rather than the Chernoff bound based effective bandwidth) or to incorporate measurement results into the CAC decisions are research challenges left for the near future.

References

1. Numerical Recipes in C- Book on-line. *http://www.library.cornell.edu/nr/bookcpdf.html*.
2. P. Barford and M. E. Crovella. Generating representative Workloads for Network and Server Performance Evaluation. In *Proceedings of ACM SIGMETRICS*, pages 151–160, 1998.
3. T. Bonald, A. Proutiere, and J. W. Roberts. Statistical Performance Guarantees for Streaming Flows using Expedited Forwarding. In *Proceedings of IEEE INFOCOM*, volume 2, pages 1104–1112, 2001.
4. R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: an Overview. RFC1633, June 1994.
5. S. Blake et al. An Architecture for Differentiated Services. RFC2745, Dec 1998.
6. A. Frey and Y. Takahashi. A note on an M/GI/1/N queue with vacation time and exhaustive service discipline. *Operations Research Letter*, 21(2):95–100, 1997.
7. J. Roberts, U. Mocci, and J. Virtamo, editors. *Broadband Network Traffic: Final report of Action COST 242*. 2nd Edition, Springer-Verlag, 1996.
8. I. Stoica and H. Zhang. Providing Guaranteed Services without per Flow Management. In *Proceedings of SIGCOMM*, pages 81–94, 1999.
9. H. T. Tran and T. Ziegler. Engineering Solution of a CAC Mechanism for Voice Traffic over IP Networks. TD(02)037, COST 279, September 2002.

Overview of the Project AQUILA (IST-1999-10077)

Bert F. Koch[1] and Heinrich Hussmann[2]

[1] Siemens AG/Project Management Consultant, Munich, Germany
bert.koch@t-online.de
[2] Technische Universität Dresden, Germany

1 Objectives

AQUILA defines, evaluates, and implements an enhanced architecture for QoS in the Internet. Existing approaches e.g. Differentiated Services, Integrated Services and label switching technologies have been exploited and significantly enhanced, contributing to international standardisation. The architecture has been designed to be cost-effective and scalable. It introduces a software layer for distributed and adaptive resource control and facilitates migration from existing networks and end-user applications. Technical solutions have been verified by experiments and trials, including QoS-enhanced on-line multimedia services.

The key objectives of the project are:

1. To enable **dynamic end-to-end QoS** provisioning in IP networks for QoS sensitive applications e.g. Internet telephony, premium web surfing and video streaming. Static resource assignments have been considered as well as dynamic resource control.
2. To continuously analyse **market situations** and **technological trends** for QoS solutions and to exploit the results of the project creating applicable business plans based on the user and service provider requirements.
3. To design a **QoS architecture** including an extra layer for resource control for scalable QoS control and to facilitate migration from existing networks. The Differentiated Services architecture for IP networks has been enhanced introducing dynamic resource and admission control.
4. To **implement prototypes** of the QoS architecture as well as QoS based end-user services and tools in order to validate the technical approach of the solution design.

2 Architecture

The project assumes the DiffServ architecture as the most promising starting point for its work. The project develops extensions of this architecture in order to avoid the

W. Burakowski et al. (Eds.): Art-QoS 2003, LNCS 2698, pp. 154–164, 2003.

statically fixed pre-allocation of resources to users. Dynamic adaptation of resource allocation to user requests is enabled in a way that keeps the overall architecture scalable to very large networks.

2.1 Resource Control Layer (RCL)

The Resource Control Layer (RCL) is an overlay network on top of the DiffServ core network (see Fig. 1). The Resource Control Layer provides an abstraction of the underlying layers. The RCL mainly has three tasks, which are assigned to different logical entities:

- to monitor, control and distribute the resources in the network by the Resource Control Agent (RCA).
- to control access to the network by performing policy control and admission control by the Admission Control Agent (ACA).
- to offer an interface of this QoS infrastructure to applications by the End-user Application Toolkit (EAT).

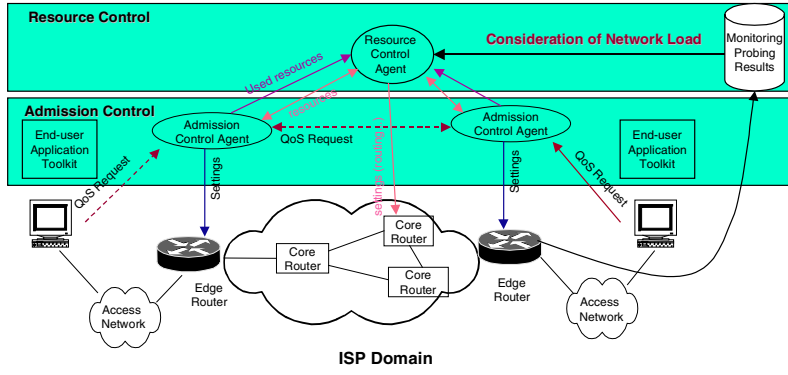


Fig. 1. AQUILA Resource Control Layer

2.2 Resource Control Agent (RCA)

A node in the Resource Control Layer is called a Resource Control Agent and represents a portion of the IP network, which internally has the same QoS control mechanisms. An RCA is a generalisation of the concept of the Bandwidth Broker in the DiffServ architecture. RCAs are logical units that run on several physical configurations, e.g. one server per RCA or several RCAs co-located on one server. The QoS control mechanisms used in the underlying network are of varying nature, e.g. in some part the routers may not even support DiffServ - which means that there

is only a trivial best-effort QoS control - while in other parts they may be DiffServ capable. Moreover, some parts of the network may allow dynamic reconfiguration of resources, e.g. by adding ATM connections, others may have a more or less fixed configuration, e.g. pure SDH or WDM sub-networks. Another reason for the introduction of separate RCAs is that sub-networks are domains managed by different operators.

A Resource Control Agent is able to observe and in some sense to influence the actual configuration in the network portion it represents. Configuration parameters may describe the fraction of a network connection devoted to a specific DiffServ traffic class or the existence of a virtual connection (in ATM networks) with a specified bandwidth.

2.3 Admission Control Agent (ACA)

A DiffServ network can only provide Quality of Service, if it is accompanied by an admission control, which limits the amount of traffic in each DiffServ class. The AQUILA architecture uses a local admission control located in the Admission Control Agent, which is associated with the ingress and egress edge router or border router. To enable the ACA to answer the admission control question without interaction with a central instance, the RCA will locate objects representing some share of the network resources nearby the ACA. Resources are assigned to these objects proactively.

Admission control can be performed either at the ingress or at the egress or at both, depending on the reservation style.

The ACA will just allocate and de-allocate resources from its associated share. The ACA is not involved in the mechanisms used by the RCA to provide this resource share, to extend and to reduce it.

Resources are handled separately for incoming traffic (ingress) and for outgoing traffic (egress). The following description of resource distribution applies to both.

Resource distribution is performed by the RCA in a hierarchical manner using so-called *Resource Pools*. For this purpose it is assumed, that the DiffServ domain is structured into a backbone network, which interconnects several sub-areas. Each sub-area injects traffic only at a few points into the backbone network. This structuring may be repeated on several levels of hierarchy.

2.4 End-User Application Toolkit (EAT)

The End-user Application Toolkit (EAT) aims to provide access to end-user applications to QoS features. The EAT is a middleware between the end-user applications (Basic Internet Applications and Complex Internet Services) and the AQUILA network infrastructure.

The EAT supports two major kinds of (Internet) applications:

- Legacy Applications that are in fact QoS-unaware and that cannot be modified in order to directly access the EAT or any other QoS infrastructure. The most of existing Internet applications are legacy ones.

- QoS-aware Applications that can themselves request for QoS, by using an API, for example (EAT-based Applications use the EAT API), or by using signalling protocols such as RSVP and SIP.

Internet applications, however, have also to be distinguished with regard to their *complexity*. In AQUILA, we make a distinction between Basic Internet Applications and Complex Internet Services. They have to be supported in different ways: Whereas Basic Internet Applications are often legacy ones which cannot directly use the EAT, Complex Internet Services can be QoS-aware or even EAT-based although they consist of basic applications.

Generally, the EAT provides – at the control plane – a set of application interfaces in order to support the wide range of different applications (Fig. 2):

- ***Legacy applications*** do not interact with the EAT. QoS reservations must therefore be made manually. For that reason, the EAT offers some **Graphical User Interfaces (GUIs)** for manual reservation requests (see below).
- For some ***specific legacy applications*** that dynamically negotiate data port numbers or rely on signalling protocols, special **Protocol Gateways (Proxies)** (e.g. for H.323, SIP) enable the selective processing of the application's control plane information by forwarding QoS-relevant data to the EAT Manager in order to initiate QoS requests. The Proxy Framework is flexible and extensible in order to include additional Proxies (e.g. for RSVP) later on.
- For ***QoS-aware, EAT-based applications***, an **Application Programming Interface (API)** provides interfaces and methods for login, reservation requests and releases, etc. This proprietary API is accessible via CORBA and provides the full AQUILA functionality. The **EAT Manager** directly implements the API in order to manage user access and reservations. (The EAT Manager is the main part of the EAT and controls the whole process. It also acts as mediator between the other EAT components and towards the ACA.)

Due to the fact that the EAT is fully transparent for legacy applications – even if they are supported by a Proxy – QoS reservations must be performed in a different way. For that reason, the EAT provides a set of GUIs in form of Web pages (the so-called **AQUILA Portal**), in which an end-user can *manually* request for QoS reservations. Moreover, the so-called AQUILA Portal offers among other things two different reservation modes: an advanced one for end-users that have knowledge about the technical details of an AQUILA request, and a regular one for end-users that are not familiar with AQUILA.

In order to support the regular reservation mode, an additional application "interface" is provided, the so-called **Application Profile** methodology. Application Profiles contain reservation "schemes" with technical parameters mapped to well understandable QoS metaphors. The **Converter** is the component which takes care of the mapping/converting of the technical parameters of the profiles and the (by the end-user subscribed) network services into the QoS metaphors corresponding to the application in use.

Note that the regular reservation mode is not necessarily part of the AQUILA Portal. In fact, Application Profiles are usable via the EAT API and can therefore be called by every Complex Internet Service that wants to make use of the AQUILA QoS capabilities. In that way, such an Internet service may offer its own regular

reservation mode, by showing the QoS metaphors from the proper Application Profiles of its basic applications/plug-ins.

The following figure gives an overview on the above mentioned interfaces and components of the EAT, and how they interact:

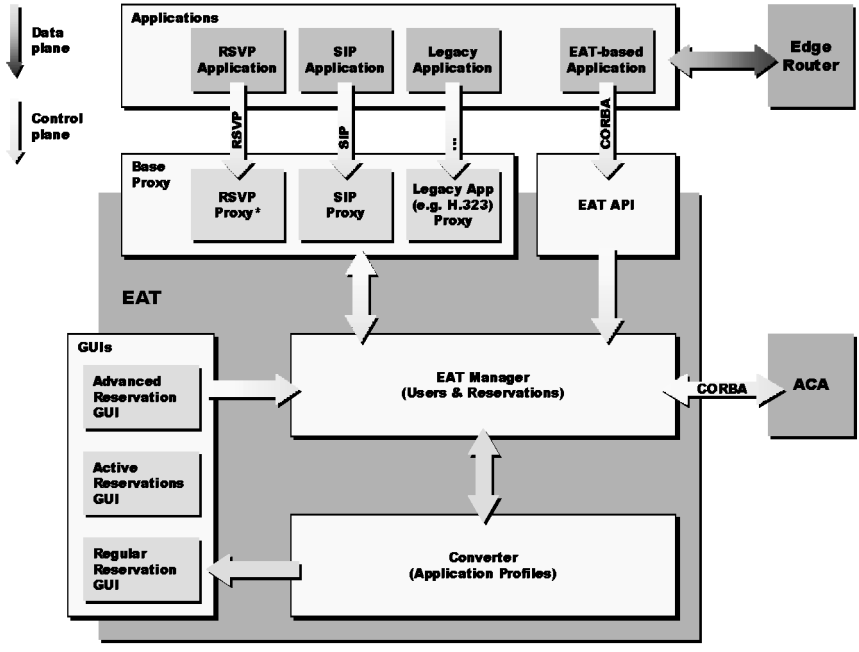


Fig. 2. EAT's basic building blocks and application interfaces

2.5 Distributed QoS Measurement

Advanced measurement and monitoring is a base to ensure and supervise QoS and performance parameters of applications over QoS-based IP networks and to verify them and to optimise their behaviour. To ensure that the developed architecture and its implementation is practically useable for providing application-specific QoS demands, a distributed QoS measurement infrastructure with several application-like traffic generators was developed. Measurements are able to simulate real users with different behaviours simultaneously within a test laboratory trial to prove the stability of the proposed architecture before integrating it into the more complex and expensive field trial with real user scenarios. The distributed QoS measurement enables the project to evaluate specific network and application profiles which cannot be realised in a field trial.

Distributed QoS measurement introduces new functionality for the analysis of applications and protocols like automation of test scenarios with different protocol parameters and network configurations ("tuning"). It is based on distributed measurement agents within different kind of networking components controlled

remotely by an operator via a user-friendly graphical interface. This allows specification and execution of measurement test suites in different modes (multiplexed, point-to-point, point-to-multipoint, multicast), traffic characteristics, QoS measurement requirements and resource reservation parameters dependent on the applications. These can be stored in a measurement information database together with network device parameters and the measurement results. The calibration curves for admission control can be calculated from this database.

More information about the project can be found in [1].

2.6 Service Management

The AQUILA project aims at the *dynamic* provision of Quality of Services features for end-users over the existing Internet. For that, the AQUILA network offers different **network services** with different *predefined* QoS characteristics to the customers of the network and implements them internally by different **traffic classes**. Network services can be seen as products provided by the QoS-enabled AQUILA network and designed for typical application requirements.

The AQUILA project have defined four network services that are in fact four manageable premium transport options beside best effort for IP user traffic:

- **PCBR (Premium Constant Bit Rate)**, designed to serve a constant bit rate traffic. Examples of applications: voice trunking and virtual leased lines. This service should support circuit emulation and meets hard QoS requirements with respect to packet loss ratio (not greater than 10^{-8}) and packet delay (not greater than 150 ms, low jitter).
- **PVBR (Premium Variable Bit Rate)**, designed to provide effective transfer of streaming flows of variable bit rate type. The traffic description of a flow has two parameters to declare, the Sustainable Rate (SR) and Peak Rate (PR). Policing assumes double token bucket. For the purpose of admission control algorithm, the notion of effective bandwidth (evaluated on the basis of SR, PR and dedicated for this service link capacity) is used.
- **PMM (Premium Multi-Media)**, designed to support greedy and adaptive applications that require some minimum bandwidth to be delivered with a high probability. Although the PMM service is not primarily targeted for applications using TCP, but there is optimisation regarding the TCP transport protocol.
- **PMC (Premium Mission Critical)**, designed to support non-greedy applications. The sending behaviour may be very bursty. This requires a very low loss and low delay service to be delivered with a high probability. Throughput is of no primary concern for PMC applications. There is an optimisation regarding the TCP transport protocol.

The network services and their detailed characteristics are defined by the network operator. Their goal of them is to provide a few specific offerings from the network operator to the customer, which are relatively easy to understand, suitable for a specific group of applications, and maintainable in large scale networks.

The network services are store as XML data (based on a common Document Type Definition) on a central directory server. The **QoS Management Tool (QMTool)** provides for network operators access to the network services. It is able to retrieve the

XML data from the server, to modify the entries in order to adapt the network service parameters, and finally to store the adapted entries on the server. (Even new network services can be created in this way.)

Customers can subscribe network services in order to have the policy to request for them for their applications. More specifically, customers initiate via the **End-user Application Toolkit (EAT)** QoS requests by firstly selecting a network service, which can be seen as *predefined SLS*, and secondly by giving additional data for the chosen **Service Level Specification (SLS)** such as:

- Scope, indicates the typology of the ongoing reservation with reference to the end-points of the traffic flows
- Flow identification, focuses on the association between packets and SLSs
- Traffic description, describes the traffic relevant to the reservation
- Performance guarantees, describes additional QoS requirements the customer and the commitment of the network operator to fulfil theses requirements
- Service schedule, provides the information related to the start and the duration of the service

The AQUILA notation is a practical utilisation of the notation, developed by the TEQUILA project and follows recommendations of the CADENUS project.

3 Resource Management

In chapter 2.1 we explained the AQUILA approach of the Resource Control Layer. Here we will focus on two aspects: the Resource Pools, and the design and implementation of BGRP for the Inter-domain approach of QoS provisioning in IP networks.

3.1 Resource Pools

Resource distribution is performed on a per DiffServ class basis. In the first trial, there was no dynamic reconfiguration of DiffServ classes. So, the resources of each class could be handled separately and independently of each other. This per class distribution however is not appropriate for edge devices, which are connected via small bandwidth links to the core network.

Resources are handled separately for incoming traffic (ingress) and for outgoing traffic (egress). The following description of resource distribution applies to both.

Resource distribution is performed by the Resource Control Agent (RCA) in a hierarchical manner using so called **Resource Pools**. For this purpose it is assumed, that the DiffServ domain is structured into a backbone network, which interconnects several sub-areas. Each sub-area injects traffic only at a few points into the backbone network. As described later, this structuring may be repeated on several levels of hierarchy.

When considering the resources in the backbone network, all traffic coming from or going to one sub-area can be handled together. So it is reasonable to assign a specific amount of bandwidth (incoming and outgoing separately) to each sub-area.

Depending on the topology of the backbone network, it may be useful to add some degree of dynamic to this distribution. The RCA may assign a larger bandwidth to one specific sub-area, when the bandwidth is reduced in other sub-areas. This dynamics may be described by the following formulas:

$$r_i \leq R_i \tag{1}$$

$$\sum_i r_i \leq R \tag{2}$$

where r_i is the resource limit actually assigned to ACA_i and R_i is an upper bound for this value. R is the overall limit of all resources distributed to all Admission Control Agents (ACAs). These formulas express the following behaviour:

- The bandwidth assigned to each lower level entity r_i must not exceed an individual limit for this entity R_i. This limit R_i reflects the linkage of the lower level entity (e.g. sub-area) to the upper level entity (e.g. core network).
- The sum of the bandwidth assigned to all lower level entities must not exceed an overall limit R.

Depending on the values chosen for R_i and R, a more or less dynamic behaviour can be achieved.

Please note, that describing bandwidth with a single value (bits per second) is not sufficient in all cases. The characteristics of the traffic have to be taken into account. This may lead to an "effective bandwidth" formula, which is specific for each traffic class. It may also be necessary to describe bandwidth with a much more complex data structure, for which "addition" and "comparison" may be defined as rather complicated operations.

Resource shares are completely managed by the RCA. The resource share object itself is responsible to manage its resources and to check, whether a new bandwidth allocation request fits into the available bandwidth. If the amount of available bandwidth crosses some low-water-mark, the resource share object may precautionary request more resources from the resource pool. On the other hand, the resource share object will return unused resources to the pool.

Within a sub-area, there may be further subordinated sub-areas, which could be handled similar. Each resource share r_i assigned to a sub-area can be handled again as a resource pool R, which is distributed in a similar way among the sub-areas. Finally, resources can be used by ACAs as "consumable ResourceShare".

The depth of this hierarchical structure may be chosen as needed. It is also possible to mix several degrees of hierarchy, e.g. to break down the structure near edge routers more deeply than the structure of border routers, which are likely to be directly connected to the backbone.

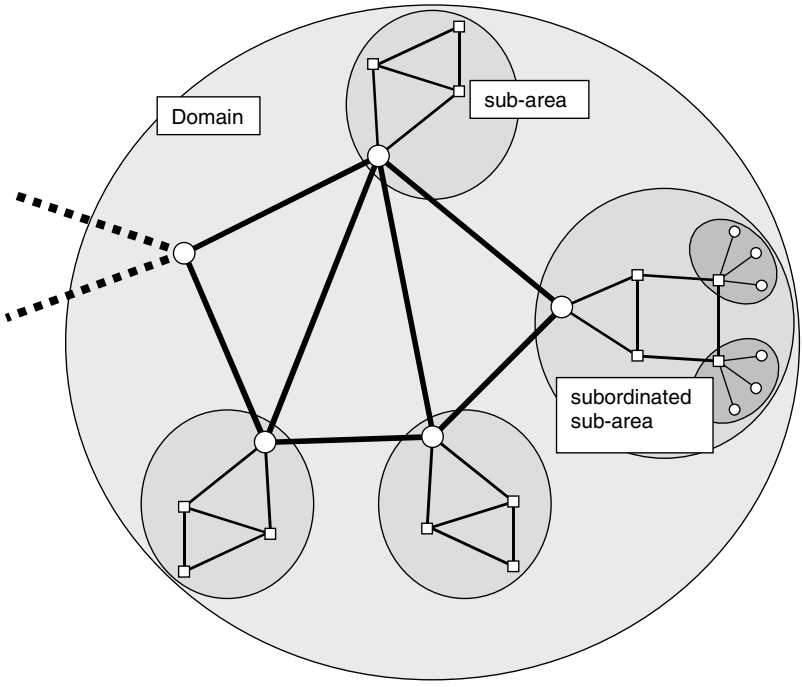


Fig. 3. Hierarchical resource pools

The figure above illustrates this. It shows an example domain, which contains four sub-areas and one border router. In one of the sub-areas, the further division into subordinated sub-areas is illustrated.

Obviously, the ability to structure a domain like this strongly depends on the topology. In the access area of a network however it is likely, that tree-like structures exist, which enable the definition of such a structure.

3.2 Border Gateway Routing Protocol (BGRP) Appliance

BGRP is a framework for scalable resource control [2]. It assumes, that BGP is used for routing between domains (autonomous systems, AS).

The basic idea of BGRP is the aggregation of reservations along the sink trees formed by the BGP routing protocol. It is a characteristic of the BGP routing protocol to forward all packets for the same destination AS to the same next hop AS. This property guarantees the formation of a sink tree for each destination AS. All traffic destined for the same AS travels along the branches of this tree towards the root.

Similar to the QBone approach, some kind of “bandwidth broker” is established in each domain. However, not just a single entity is responsible for the whole domain. Instead, a BGRP agent is associated with each border router. Reservations for the same destination AS are aggregated at each BGRP agent. This has the following implications:

- The number of simultaneous active reservations at each domain cannot exceed the number of autonomous systems in the Internet.
- The source and destination addresses cannot be carried in the reservation requests between domains, because of the aggregation mechanism.

However, the aggregation mechanism does not automatically reduce the number of signalling messages. Each request may still travel end-to-end. Additional damping is necessary, e.g. by reserving additional resources in advance or by deferred release of resources.

In summary, the BGRP framework provides a possible approach to a scalable inter-domain architecture. However, the following issues have to be solved:

- Introduction of a damping mechanism as described above. The authors of [2] make some proposals here. However, also the experiences from the resource pools used for the AQUILA intra-domain resource allocation are well suited to address this topic.
- Because BGRP messages not always travel all the way to the destination domain, the problem of QoS signalling within the last domain towards the destination host has to be solved.
- BGRP is still a framework only. The detailed information exchange between BGRP bandwidth brokers as well as the interaction with the intra-domain resource control has to be specified.

3.3 Inter-domain Requirements

An architecture for the AQUILA inter-domain resource control has to fulfil the following requirements:

- Scalability
  When high quality services will be established in the Internet world-wide, the number of individual resource reservations will grow rapidly. The architecture must be able to cope with that.
- Works with multiple intra-domain resource control mechanisms
  Operators should be free to use any resource control mechanism within their domain. The AQUILA intra-domain approach is just one possible example. An interface must be defined and standardised, through which the inter-domain resource control interacts with the domain specific QoS mechanisms.
- Edge-to-edge QoS guarantee
  The architecture must be able to support a certain level of QoS guarantee from the ingress edge of the source domain to the egress edge of the destination domain.
- Stepwise deployment
  It must be possible to deploy the architecture in the Internet step by step. An architecture, where any modification or enhancement has to be installed in each AS, is not acceptable.

3.4 Inter-domain Architecture

In order to fulfil the requirements listed above, an architecture according to the BGRP framework will be chosen. However, a number of extensions and enhancements have to be added to make a running implementation out of the framework.

Also, ideas and mechanisms developed for the intra-domain resource control will also influence the AQUILA inter-domain architecture.

This chapter specifies the general architecture for the AQUILA inter-domain resource control, where the next chapter addresses detailed aspects.

The following picture gives a rough overview of the architecture and depicts the basic interactions between the intra- and inter-domain resource control layer in the source, intermediate and destination domain.

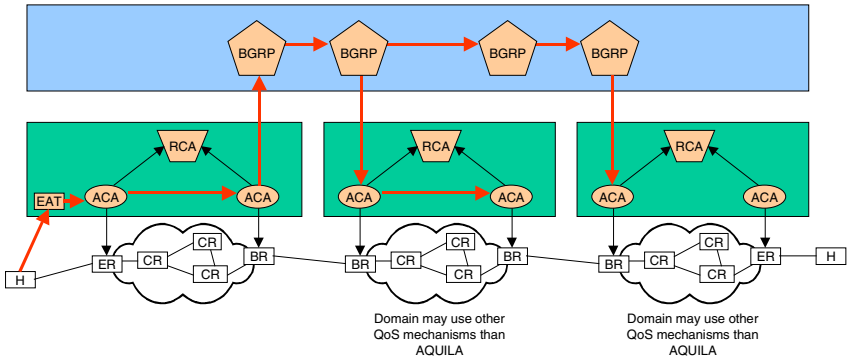


Fig. 4. General inter-domain architecture and message flow

A so-called BGRP agent is associated with each border router. These agents interact with the AQUILA intra-domain resource control layer in the following way:

- Inter-domain resource requests are initiated by the ACA associated with the egress border router of the initiating domain and sent to the corresponding BGRP agent.
- BGRP agents associated with ingress border routers use the ingress ACA to establish intra-domain resource reservations.

Further information on the AQUILA project can be found at [1].

References

1. AQUILA, Project Home Page, URL: http://www.ist-aquila.org/, AQUILA consortium, Febraury 2003
2. Salsano, S. et. Al: Inter-domain QoS Signaling: the BGRP Plus Architecture. Internet Draft, May 2002

Application Support by QoS Middleware*

Falk Kemmel[2], Sotiris Maniatis[1], Anne Thomas[2], and Charilaos Tsetsekas[1]

[1]National Technical University of Athens, School of Electrical and Computer Engineering, GR-157 73 Athens
{sotos,htset}@telecom.ntua.gr

[2]Technische Universität Dresden, Fakultät Informatik, Institut SMT, D-01062 Dresden
{Falk.Kemmel,Anne.Thomas}@inf.tu-dresden.de

Abstract. A QoS middleware provides the Quality of Service support required by many QoS sensitive applications. The QoS middleware is between a host and a QoS-enabled infrastructure and hides the complexity of the extra functionality behind a set of APIs. This article presents the AQUILA QoS middleware, a QoS API to implement QoS-aware applications, a QoS Portal based on application specifications and Proxies to provide legacy non QoS-aware applications with QoS.

1 Introduction

Quality of Service (QoS) in the Internet becomes more important. Several technologies have been developed that provide QoS to modern Internet applications, having very specific QoS requirements. Streaming applications, for example, need a certain level of throughput where interactive applications have strict requirements for delay and jitter. One of the most notable QoS approaches is the DiffServ architecture, which supports relative prioritization of IP traffic [4]. DiffServ, however, has two major handicaps: the lack of *QoS guarantees*, and the complicated accessibility for end-user applications.

The IST project AQUILA [2] has developed an architecture that extends the DiffServ approach by adding a new layer on the top of DiffServ that acts as distributed bandwidth broker [8]. This so-called Resource Control Layer (RCL) shares resources over the core network, takes care of admission and resource control and allows hard *QoS reservations* on specific, predefined service level specifications (network services) for applications and end-users. It also provides easy access by introducing a special QoS middleware, the End-user Application Toolkit (EAT).

The EAT is the component of the RCL, which builds a bridge between the RCL and the applications and the end-users of it. The EAT is responsible for providing different kinds of interfaces towards applications and end-users, to forward any QoS request to the RCL, and to map between the different views on QoS. The EAT aims to support a wide range of existing and newly developed applications on different levels

* This work was partially funded by the European Union under contract number IST-1999-10077 “AQUILA”.

W. Burakowski et al. (Eds.): Art-QoS 2003, LNCS 2698, pp. 165–176, 2003.

of QoS abstraction. Consequently it provides flexible interfaces for non QoS-aware, legacy applications and their users, as well as for the developers of new, QoS-aware applications.

After presenting the general architecture of the EAT, these interfaces are described in more detail in the section 2 of this paper. Section 3 discusses some related projects in order to estimate our solution. The paper finishes with a short conclusion and outlook.

2 A Middleware for QoS-Enabled Networks

The EAT is a QoS middleware that aims to fill the gap between the applications of the end-users as well as the QoS-enabled AQUILA network. This flexible middleware provides QoS to a wide range of applications including non QoS-aware, legacy applications as well as newly developed QoS-aware ones. For that reason it offers a set of different interfaces towards – on the one hand – the applications and end-users, such as an Application Programming Interface (API), a Graphical User Interface (GUI – here also called *QoS Portal*), and a set of Protocol Gateways (or *Proxies*). On the other hand, the EAT acts as a front-end tool of the AQUILA RCL. Therefore the EAT interfaces the Admission Control Agent (ACA) in order to communicate QoS requests to the RCL.

Figure 1 depicts these interfaces as well as the basic building blocks of the EAT and the control flow in between. Legacy applications do not directly interact with the EAT, since they usually cannot be recompiled in order to use the EAT's API. Therefore, legacy applications are supported in two different ways: The EAT's QoS Portal offers a GUI for manual reservations by end-users. Different Proxies allow the automatic or half-automatic support of legacy applications that rely on special signaling protocols such as H.323 (e.g. NetMeeting) and SIP (e.g. VoIP applications).

The EAT, moreover, supports also the development of new, QoS-aware applications, by providing an API. This API forwards any request to the component for user and reservation management, which then handles the user accounts, the end-user's QoS requests and sessions, etc. However, it does not establish a reservation within the network. This is done by the ACA on EAT's demand.

For the legacy applications, the EAT manages a set of pre-defined Application Profiles that store application-dependent QoS characteristics and that support the generation of user-friendly forms for QoS requests. The Converter is the component that reads the profiles in order to handle both the abstract QoS level for the end-user and the concrete one for AQUILA.

More details about the above mentioned components can be found in the following chapters.

3 Application Development Support

The EAT provides a QoS API towards QoS-aware applications such as Internet portals and services that want to provide their basic applications with QoS. It also

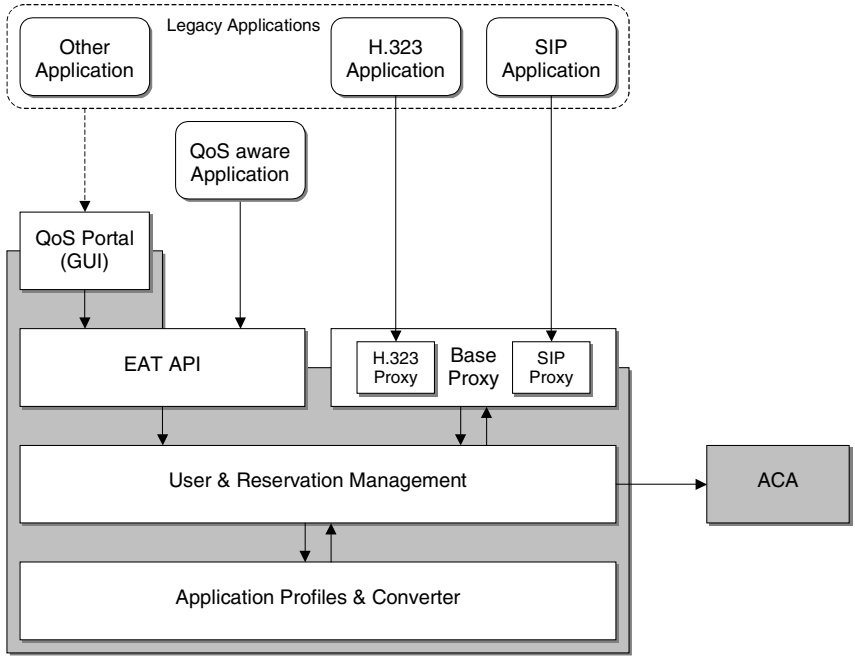


Fig. 1. Basic building blocks of the EAT

aims to assist the developers of new, QoS-aware applications by offering high-level QoS functions for:

- User authentication: End-users (i.e. the AQUILA customers) need an account for the AQUILA RCL in order to request for QoS.
- Service retrieval: The RCL offers a set of pre-defined Service Level Specifications (in AQUILA: *network services*). Customers may have contracts (Service Level Agreements) to make use of some or all of them.
- Application Profile retrieval: Customers can ask for existing Application Profiles (see chapter 4.1).
- Reservation request & release: Customers can request for QoS reservations at different levels of abstraction. They can also ask for unidirectional or bidirectional reservation units or even for multidimensional reservation groups. Finally, they can release the reservations in order to de-allocate the network resources.
- Accounting information retrieval: Customers can ask for the accounting information (such as volume, duration) of their QoS sessions useful for later charging and billing, for example.

The following figure depicts the essential classes and operations of the API.

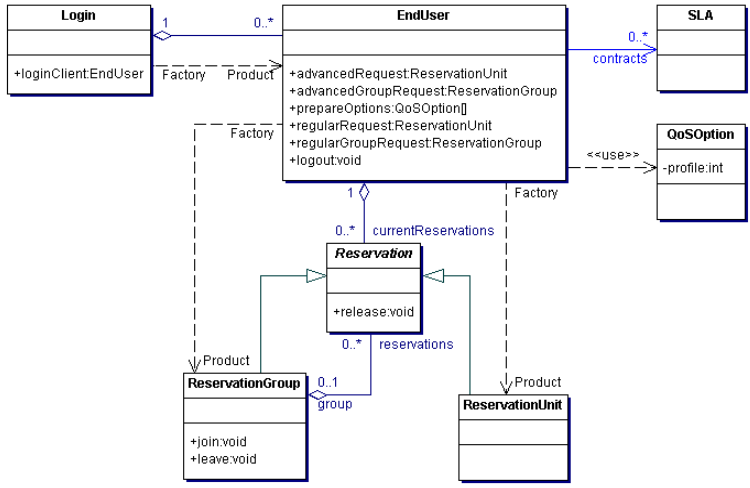


Fig. 2. Extract of the API's class diagram

An end-user must firstly login (with account name and password) in order to get access to the API. The newly created EndUser object is now the client's agent for the retrieval of SLAs, for reservation requests, etc.

Reservation requests can be done at two different levels: The advanced level, on the one hand, assumes that all the technical details of the traffic and QoS requirements of the application are known. The regular level, on the other hand, uses so-called QoS options, which are stored in the Application Profiles, to hide these technical characteristics.

An advanced request can directly be submitted after login. If successful, either a new reservation unit is created, or a reservation group is established which in turn consists of one or more units. Such a unit represents the ongoing reservation within the RCL network.

Before a regular request can be submitted, the available QoS options must be prepared, since the request is based on one of them. QoS options belong to Application Profiles, which can – in a first step – be retrieved by using another interface (not shown in the figure above). After selecting one of the existing Application Profiles, the QoS options can be prepared for the regular request. The EAT only returns those options for which a contract exists. Once having the available options, a regular reservation can be requested including flow-specific parameters but without any traffic and QoS parameters.

The EAT API is a remote, platform-independent API based on CORBA and implemented in Java. By using this API, several clients can simultaneously access the EAT. Even several sessions per customer are possible. The QoS Portal (see figures 3 and 4) directly uses the EAT API.

4 Legacy Application Support

The current Internet delivers best effort support for Internet traffic and it is not foreseeable when a QoS-aware public network will be available. This fact implies that there are only few APIs (Windows 2000, etc.) for QoS technologies. At the time being there is no necessity for application developers to implement QoS-aware applications that will not be supported by the network infrastructure. One of the consequences is that when QoS-aware networks will come up, a high majority of legacy applications will be available. Some of them will be non QoS-aware, others will implement codecs and signaling protocols (like SIP, H.323 etc.). In the first case a QoS request is only possible manually via an in-between GUI called a QoS Portal, in the second case the information produced by the codecs is interpreted and used by Protocol Gateways to produce automatically a QoS request.

4.1 Manual Support

The non QoS-aware legacy applications can neither request for QoS nor fetch the QoS offers of the network. In order to provide such applications with QoS a third instance – e.g. a QoS Portal – between the application and the network is needed. The aims of such a QoS Portal are: 1) to enable the identification of the application used, 2) to present to the end-user the different possible quality levels or options, and 3) to enable the request for QoS on behalf of the application toward the network.

Beside using the application the end-user has to handle the portal. The portal is a web interface where the end-user registers in a first step (see Fig. 3) which application he is using and selects in a second step (see Fig. 4) the quality level he wants for his current session.

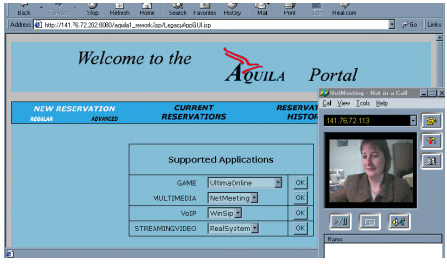


Fig. 3. QoS Portal and NetMeeting application. First step: manual registration of the application

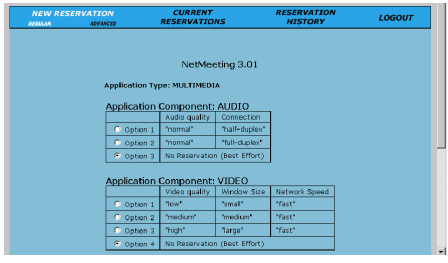


Fig. 4. QoS Portal. Second step: selection of the quality levels for the two service components *audio* and *video* of NetMeeting

As the applications are designed and optimized to use the best-effort network and does not implement the APIs of the QoS-enabled network there is no connection and information transfer possible between the application and the network. Because the applications are not reprogrammed or changed in order to implement these networks APIs such a manual reservation via a third party is necessary. The portal is responsible for converting application specific information into the network APIs syntax and vice versa presenting network specific information in end-user metaphors.

The portal is based on a *Converter* that uses a repository created with the *Application Profile*, a high-level application specification. It contains QoS information of applications and their expression at end-user, application and network levels.

Application Profile. The high-level application specification profile is a syntax designed to create a repository for applications and their QoS information. The repository serves the conversion between end-user, application and network levels with the aim to request for QoS at a QoS-enabled network.

The high-level application (e.g. complex multi-media application like a video conference) specification includes the high-level specification of: 1) the decomposition of the *application* into *service components* (e.g. basic multi-media services like audio or video) and their minimal *QoS requirements* for an optimal/smooth execution, 2) the implemented different quality *options* with the generic description of the produced traffic: the *traffic specification,* and the end-user metaphors: the *session characteristics*, 3) the implementation dependent *AQUILA traffic specification* for the mapping toward a concrete QoS-enabled network.

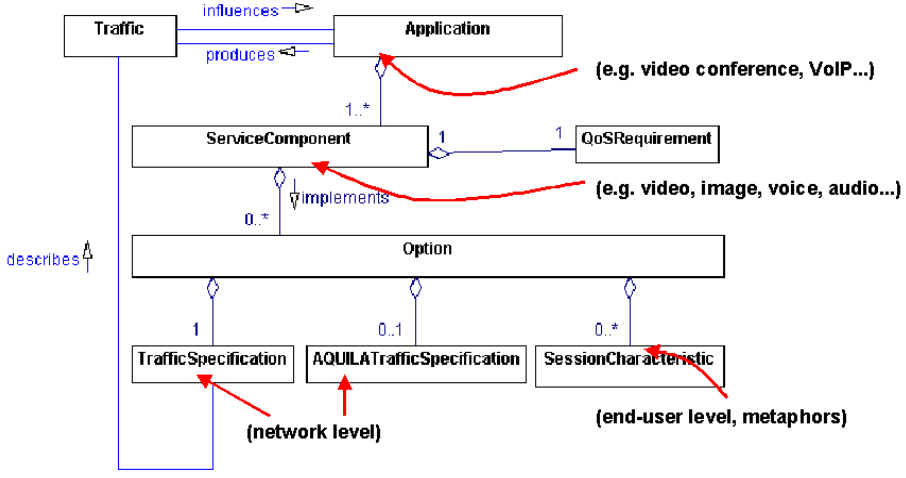


Fig. 5. UML diagram of the elements of the application specification

The syntax is a DTD [5], in this form it gives the rules to for creating the repository in the XML [19] format. In the following an example of the content of an Application Profile for NetMeeting for its video service component:

```
<ServiceComponentProfile name="NetMeeting_3.01_Video"
serviceComponent="VIDEO">
  <QoSRequirement>
    <maxDelay unit="ms" requirement="high">1200</maxDelay>
    <maxJitter unit="ms" requirement="low">120</maxJitter>
    <maxLoss unit="percent" requirement="medium">10</maxLoss>
    <bandwidth unit="kbps" requirement="high"></bandwidth>
    <ordering requirement="true"/>
  </QoSRequirement>
```

```
<Option optionID="1" description="video low quality scenario">
  <SessionCharacteristic>
    <name>video quality</name>
    <semanticalGroup type="UserFriendly" language="en">
      <description>Video quality</description>
      <qualifier>very low quality (28.8kBit/s)</qualifier>
    </semanticalGroup>...
  </SessionCharacteristic>
  <TrafficSpecification>
    <type type="elastic"/>
    <duration value="longLiving"/>
    <adaptivity value="false"/>
    <burstiness value="true"/>
   <packetSize qualitatively="medium" variability="variable">...
    <bitRate qualitatively="high" variability="variable"> ...
    <flow value="greedy"/>
  </TrafficSpecification>
  <AQUILASpecification>
    <serviceID value="PVBR"/>
    <BSP unit="bytes">2000</BSP>
    <BSS unit="bytes">2048</BSS>
    <minPU unit="bytes">60</minPU>
    <maxPS unit="bytes">1500</maxPS>
    <PR unit="bit/s">28800</PR>
    <SR unit="bit/s">19200</SR>
  </AQUILASpecification>
</Option>
```

It is necessary for each application to create a profile. This has to be done from scratch by analyzing the application and its behavior.

Converter. The automatic preparation of the reservation request towards the network is actually performed by an internal module of the portal architecture, which is called the Converter. Actually, the purpose of the Converter is twofold:

- Firstly, to prepare the available QoS options pertaining to the particular Service Level Agreements (SLAs) of the logged-in end-user. In other words, the Converter parses the Application Profile and filters out only the QoS options that are intended for the specific end-user. The end-user is therefore significantly assisted in his effort to select the most appropriate QoS option, without having to examine a vast amount of alternatives.
- Secondly, after the end-user has selected the desired level of QoS Fig. 4, the Converter maps the end-user selections to concrete, implementation-dependent traffic specifications, and forwards them to the module appropriate for the admission control.

The existence of the *AQUILA specification* in the Application Profile can make both operations straightforward. However, typically, the two procedures described above are much more complicated. The first one – the preparation of the available QoS options – implies that the Converter is able to match the *QoS requirement* of each *service component* with the existing network services and the actual subscriptions (or SLAs) of the end-user. In order to accomplish this, a matching algorithm is used that isolates the service components that correspond to the network services that the end-user is allowed to employ.

The second operation – the preparation of the concrete traffic request – makes also use of a complex mapping algorithm that converts the *traffic specification* of each selected *option* to the implementation-dependent traffic syntax.

Apart from the described functionality for the support of the non QoS-aware legacy applications, the Converter plays also a principal role for legacy applications that use codecs or protocols (like SIP or H.323), which carry a lot of significant QoS-related information. For example, a SIP application exchanges SIP signaling messages that may convey all needed QoS data. The QoS Portal architecture, after ripping off all necessary QoS information with the aid of the Proxies or Protocol Gateways (see also section 4.2), forwards the data to the Converter. The Converter in turn makes use of the complex mapping algorithm, mentioned above, to convert this data to the implementation-dependent traffic syntax.

In brief, the Converter is a vital part of the automatic operation of the QoS Portal for the legacy application support. Its operation does not only heavily depend on the actual content of the Application Profiles, but it is also significantly assisted by the nature of the structured XML syntax, which alleviates intense operations like parsing, comparing, and mapping.

4.2 Automatic Support by the Proxies/Protocol Gateways

The edges of the network, where the end-user packet flows enter the core network are the points where marking, policing and traffic shaping takes place, in order to make sure that flows comply to the traffic specification agreed during admission control. Since the total number of flows passing from an edge router is relatively small, policing and shaping can take place at flow granularity, i.e. managing each flow separately. For the Multi-Field (MF) classification and marking of those end-user flows at the edge routers, not only the source and destination IP addresses, but also the TCP/UDP port numbers of the connection are required.

However, the port numbers are not usually a priori known for many end-user flows. Instead, they are negotiated during the call setup. Applications, especially multimedia ones, usually open data connections with the use of connection setup protocols. Example protocols are the Session Initiation Protocol (SIP) and the H.323 protocol suite for the establishment of IP telephony calls.

Proxies or Protocol Gateways [15] are used in order to take part in this protocol exchange. Proxies intercept the signaling messages, translate them and extract their content. This content includes information about the source and destination addresses and TCP/UDP ports. Moreover, in the case of multimedia applications, the exchanged content provides also information about the coding schemes used for audio and video communication. All this information is assembled and forwarded to the EAT's component for user & reservation management.

The Proxy Framework provided by the End-user Application Toolkit consists of a set of Application-level Proxies or Protocol Gateways. A Protocol Gateway is provided for each popular connection setup protocol. Protocol Gateways are instantiated and controlled by a central entity, the Proxy Manager. The Proxy Manager is also responsible for the communication with the other EAT components.

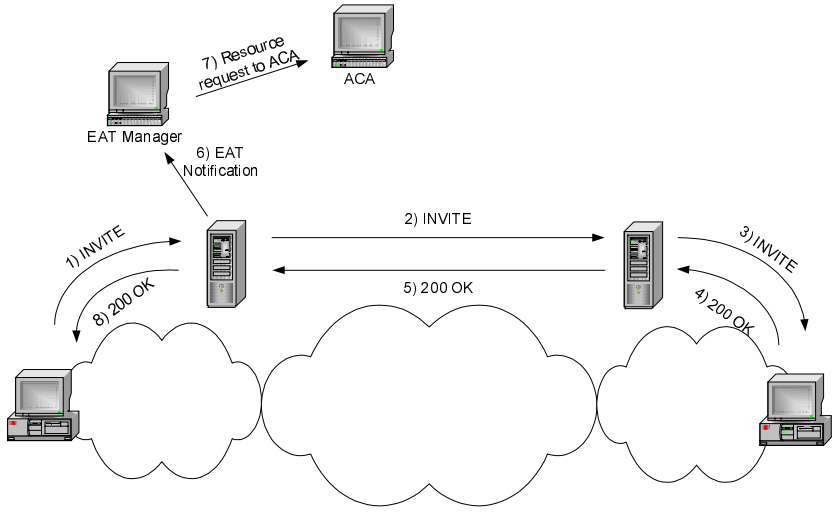


Fig. 6. Example Proxy operation for the support of QoS for SIP calls

The Protocol Gateway approach is followed also in the specification of an RSVP daemon for the transparent support of QoS for RSVP-enabled applications [9] in the AQUILA architecture. In this way, RSVP can be used as another QoS signaling mechanism, besides the AQUILA's CORBA-based signaling. The RSVP daemon module resides at the edges of the network, usually being a part of the edge router. In a manner similar to the IntServ-DiffServ interoperability model [3], the RSVP daemon forwards transparently the RSVP messages within the DiffServ core network and intercepts them at its edges. The content of the PATH and RESV messages is extracted and consequently passed to the EAT for the establishment of reservation by the Resource Control Layer.

5 Related Work and Evaluation of our Solution

5.1 APIs

The EAT API is a quite specific QoS API foreseen for QoS requests etc. in AQUILA networks only. Nevertheless, it contains some new interesting concepts (e.g. bidirectional reservations, reservation groups) that "extend" other existing QoS APIs:

The RSVP API (RAPI) [10] is a specific, low-level QoS API for RSVP going to be standardized by the Open Group. The API provides a set of C operations and structures for RSVP messages. Whereas this API is used by many higher-level APIs, the EAT API does not deal with RSVP.

The proprietary Windows Sockets 2 API [17] provides QoS facilities on a more generic level, protocol independent level. It is based on the Generic QoS (GQoS) API from Intel. However, RSVP plays also a central role in this API.

A more abstract QoS API is proposed by the Internet 2 QoS working group [11]. This API is able to use different underlying QoS APIs such as RAPI and Winsock2. It provides quite similar functions as the EAT API, e.g. concerning authentication, QoS requests (binding) and releases (unbinding), as well as the use of profiles.

The so-called QoSockets concept is part of the QoSME framework [16]. The QoSockets also form an API, which abstracts and unifies QoS negotiation, allows the specification of QoS constraints, and automatically monitors QoS performance statistics.

The ACE QoS API (AQoSA) [13] is the QoS API of the Adaptive Communication Environment (ACE). Similarly to the Internet 2 proposal, AQoSA forms an abstraction layer on top of RAPI, and GQoS. Utilizing some design patterns this API combines the capabilities of middleware (TAO) and QoS architectures.

5.2 Application Specification

In [6] Jin and Nahrstedt have studied in detail and classified QoS specification languages using a three layers partitioning: end-user, application and resource layers. The AQUILA approach [14] follows a similar layering. It covers those three layers and the specification syntax *Application Profile* enables the mapping between them.

The XML-based HQML specification language [18] assists the application developer to develop and deploy quality support for multimedia applications. At end-user layer HQML provides tags to specify qualitative QoS criteria, user focus of attention, and pricing information. At application level HQML provides tags to specify application level QoS parameters, and at system resource level tags to specifies different system resource requirements. Our work distinguishes itself by providing a special support to legacy applications.

The CQML modeling language [1] is a general-purpose language for specifying QoS. It enables the definition of QoS languages. The QoS a component provides can be specified independently of how the support is to be implemented and without affecting the specification of its functional properties. In comparison with HQML it does not provide a repository of pre-defined “standard” constructs but is highly reusable.

5.3 Proxies/Protocol Gateways

The concept of Proxies that make resource reservations on behalf of applications has been proposed in the literature in various ways. Proxies may be located inside the user terminal and run as a background process, or they may reside in the network in dedicated systems. Moreover, they may operate in a similar way to Web proxies (i.e. to serve as the end-user’s gateway to the network), or they may act as transparent proxies or sniffers that snoop on the end-user’s traffic and automatically perform proxy operations.

In the AQUILA approach, a Proxy residing at the edges of the core network may leverage both proxy and sniffing operations to launch reservations on behalf of the end-user applications.

In [18] proxies in the middleware are used that encapsulate the QoS functionality of the network, thus leaving applications unaware of the underlying complexity. In

order for these applications to reserve resources in the network, they have to invoke the services of the proxy, who formulates and sends requests on their behalf.

The same approach is followed in [7], where proxies are implemented as CORBA stubs that are downloaded on demand to the client and used to leverage the QoS functionality of the specific network.

A variant of the sniffer approach is used in [12]. An intermediate layer is present in the terminal's protocol stack that detects the launching of applications that may require QoS. A policy server is used to decide whether QoS will be provided for the application in question and, as a result, the operating system's packet marking mechanisms are activated.

6 Conclusion – Outlook

This paper has presented a QoS middleware for the support of QoS for legacy and non-legacy applications. The QoS middleware – called End-user Application Toolkit (EAT) in terms of the RCL architecture – resides at the access of the network and provides mechanisms for the presentation of network services to end-users and the formulation of reservation requests to the QoS network.

The EAT provides a QoS API for application developers to use, in order to implement new QoS-aware applications that directly make use of the capabilities of the QoS-enabled network. Moreover, the EAT supports legacy applications in two ways; it offers Proxies and Protocol Gateways to support legacy QoS-aware applications that use some kind of control protocol, like H.323, SIP or RSVP. It also offers a QoS Portal towards end-users so that they can manually setup a reservation for their application's sessions. To this end, the EAT makes use of the concept of Application Profiles and the Converter to define high level abstractions for QoS parameters and their mapping to low level (network-level) parameters, in an effort to facilitate QoS selection by the end-users.

Future research can be directed towards the definition of dynamic Application Profiles that are adapted on a periodic basis, in order to perfect their representation of the application behavior. Combining a QoS measurements infrastructure and learning mechanisms, such as Neural Networks, the Application Profiles may be updated, providing QoS and traffic parameters that are more close to the reality.

Finally, the issue of billing for QoS has not been tackled in the context of the AQUILA project. It would be interesting for the Application Profiles to accompany each QoS selection to the end-user with the respective pricing.

References

1. Aagedal: Quality of service support in development of distributed systems. PhD thesis, University of Oslo, march 2001
2. AQUILA – Adaptive Resource Control for QoS Using an IP-based Layered Architecture. http://www.ist-aquila.org/
3. Bernet et al.: A Framework for Integrated Services Operation over Diffserv Networks. IETF, RFC 2998
4. Blake et al.: An Architecture for Differentiated Services. IETF, RFC 2475

5. DTD – Document Type Definition. http://www.w3.org/XML/1998/06/xmlspec-report-v20.htm
6. Jingwen, Nahrstedt: Classification and Comparison of QoS Specification Languages for Distributed Multimedia Applications. Technical Report UIUCDCS-R-2002-2302/UILU-ENG-2002-1745, Department of Computer Science, University of Illinois at Urbana-Champaign, November 2002
7. Koster, Kramp: Structuring QoS-Supporting Services with Smart Proxies, In IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing. Hudson River Valley, USA, April 2000.
8. Nichols et al.: A Two-bit Differentiated Services Architecture for the Internet. IETF, RFC 2638
9. Nikolouzou, Tsetsekas, Maniatis, Venieris: RSVP as a User Signalling Protocol in a Multi-Layer Bandwidth Broker Architecture. In Proceedings of SPIE ITCom '01, Denver, USA, August 2001
10. Resource Reservation Setup Protocol API (RAPI), Open Group Technical Standard. http://www.opengroup.org/products/publications/catalog/c809.htm
11. Riddle, Adamson: A Quality of Service API Proposal. http://qos.internet2.edu/may98Workshop/html/apiprop.html
12. Roscoe, Bowen: Script-driven Packet Marking for Quality of Service Support in Legacy Applications. Proceedings of SPIE Conference on Multimedia Computing and Networking 2000, January 2000
13. Schmidt et al: Developing Next-generation Distributed Applications with QoS-enabled DPE Middleware. http://www.cs.wustl.edu/~schmidt/PDF/tao_qos.pdf
14. Thomas: Supplying legacy applications with QoS: a description syntax at application, end-user and network level. IASTED conference on Software Engineering and Applications (SEA 2002), November 4-6, 2002, MIT, Cambridge, USA
15. Tsetsekas, Maniatis, Venieris: Supporting QoS for Legacy Applications. In Proceedings of ICN '01, Colmar, France, July 2001
16. Wang: QoSME: Quality of Service Management Environment. http://www.cs.columbia.edu/dcc/qosockets/
17. Windows Sockets 2 Application Programming Interface. Microsoft, ftp://ftp.microsoft.com/bussys/winsock/winsock2/WSAPI22.DOC
18. Xiaohui et al.: An XML-based Quality of Service Enabling Language for the Web. Journal of Visual Language and Computing (JVLC), special issue on Multimedia Languages for the Web, vol. 13, num. 1, pp. 61–95, Academic Press, February 2002
19. XML – Extensible Markup Language. http://www.w3.org/XML/

BGRP Plus: Quiet Grafting Mechanisms for Providing a Scalable End-to-End QoS Solution

Eugenia Nikolouzou[1], Petros Sampatakos[1], Lila Dimopoulou[1], Stefano Salsano[2], and Iakovos S. Venieris[1]

[1] National Technical University of Athens,
Department of Electrical and Computer Engineering,
9 Heroon Polytechniou str, 157 73, Athens, Greece
Telephone: +30 10 772 2551, FAX: +30 10 772 2534
{enik,psampa,lila}@telecom.ntua.gr
ivenieri@cc.ece.ntua.gr

[2] DIE, University of Rome ("Tor Vergata")
Stefano.Salsano@uniroma2.it

Abstract. The provisioning of quality of service guarantees over a region spanning multiple administrative domains is nowadays one of the most basic and crucial issues. Towards this end, this paper introduces a scalable inter-domain resource control architecture for DiffServ networks. The proposed architecture is based on the BGRP framework. Scalability issues are elaborated, while our discussion mainly extends to "quiet grafting mechanisms", which succeed in significantly limiting the signaling load and efficiently handling the resources reserved between domains. Extended simulation scenarios verify the significance and efficiency of the proposed "quiet grafting mechanisms".

1 Introduction

Quality of Service provision is one of the main points of focus for the next generation Internet architectures. The existence of various QoS technologies necessitates the need to endow this diverse environment with individual inter-domain mechanisms for the provision of end-to-end QoS to users. The protocols and mechanisms of the current Internet technology seem to be insufficient for delivering the traffic of the arising and demanding multimedia applications with the appropriate Quality of Service (QoS) characteristics, and thus enhanced mechanisms have to be deployed to provide a QoS-enabled Internet infrastructure.

The current effort is basically focused on two distinct approaches: the Integrated Services (IntServ) [1] and the Differentiated Services (DiffServ) [2], [3]. The first approach was the first significant step for the introduction of QoS in the Internet. IntServ uses the Resource Reservation Protocol (RSVP) for the explicit set-up of reservation state on each network node along the path from the sender to the receiver. However, the constant exchange of RSVP messages, as well as the need for separate reservation establishment for each flow, has raised scalability concerns. The main problem is that the amount of state information stored in each router increases proportionally with the number of flows. This places a huge storage and processing

W. Burakowski et al. (Eds.): Art-QoS 2003, LNCS 2698, pp. 177–188, 2003.

overhead especially on the backbone routers. Therefore, the support of the IntServ/RSVP framework imposes many scalability limitations especially when the Internet core network is considered, which is traversed by millions of flows.
In contrast to the per-flow orientation of RSVP, DiffServ architecture classifies packets into a small number of aggregated flows, based on the DiffServ Code-Point (DSCP) in the packet's IP header. The primary benefit of DiffServ is scalability, since it eliminates the need for per-flow state and pre-flow processing and therefore scales well to large-scale networks. Moreover, the concept of the Bandwidth Broker (BB), which has been introduced from the early stages of the DiffServ model [4], is responsible for performing policy-based admission control, managing network resources, configuring specific network nodes, among others.
Nevertheless, as witnessed by the activity of the IETF NSIS WG [5], the support of dynamic QoS over a number of different domains is still an open issue. QoS signaling capabilities are indeed needed to extend the provisioning of QoS in IP networks from a static model towards a dynamic one. A fundamental issue in the definition of an inter-domain QoS model is scalability, since the motivation is to support QoS services on the scale of the global Internet.
On providing a more scalable and easy deployable solution for the inter-domain resource control mechanism, the architecture proposed in this paper, namely the "BGRP Plus" (BGRPP) architecture [6], originates from the Border Gateway Resource Protocol (BGRP) framework [7]. The architecture proposes a solution to the scaling problem, by providing sink tree based aggregation for resource reservations over a network of DiffServ domains. The aggregated reservations are negotiated between the so-called BGRPP agents, which are deployed at each BGP-capable border router of each DiffServ domain. By aggregating the reservations according to the sink trees created by the BGP routing protocol [8],[9], the number of reservations and thus the amount of state information stored in the network can be reduced.
However, aggregation of reservations is just the first step towards scalability. To limit the signaling load and the processing power required by the BGRPP agents, it is also necessary to reduce the number of signaling messages. Mechanisms for the early response to reservation messages, in [7] called "quiet grafting", are proposed in this paper. They focus on reducing the path that a signaling message has to travel along a region composed of a number of DiffServ domains. The main objective of the paper is to state the significance of the quiet grafting mechanisms, which are evaluated through a number of simulation scenarios.
The remainder of this paper is organized as follows. Section II gives an overview of the architectural principles of BGRPP by emphasizing on its messages. Section III focuses on the quiet grafting mechanisms and provides an analysis of the proposed solutions. We continue with Section IV, where the results of the simulations carried out for evaluating the proposed enhancements of the protocol are presented. Finally the conclusions are given in Section V.

2 Architectural Principles

2.1 BGRP Concepts

When resource reservation mechanisms will be deployed in a large-scale environment, it is imperative that scalability issues are taken into utter consideration.

The problem particularly arises in large transit domains where the number of simultaneous reservations processed at each domain's routers may become extremely high and thus prohibitive with regard to the CPU processing, memory and link bandwidth requirements. More precisely, the number of reservations grows like $O(N^2)$ with N the number of autonomous systems (AS) in the internet. It is therefore evident that individual handling of each flow is not considered to be a viable solution when applicable to inter-domain QoS reservation schemes. In essence, the need for aggregation is made apparent, no matter the granularity, for addressing effectively the aforementioned constraints.

The BGRP approach [7] proposes the aggregation of reservations on the basis of the destination domain. This functionality is closely related to the property of the BGP routing protocol that enables the creation of sink trees while domains trace their route towards a particular domain. Consequently, reservations are aggregated along the sink trees created by the BGP protocol, limiting to a great extent the number of active reservations maintained at the routers to a factor of O(N).

Accordingly, the BGRPP protocol operates between BGP-capable border routers of each DiffServ domain, namely the BGRPP agents. On providing the desired communication, three messages are mainly used in the BGRPP framework as described in [6]: the PROBE, GRAFT and REFRESH messages. Figure 1 depicts a reference network where the BGRPP framework can be applied.

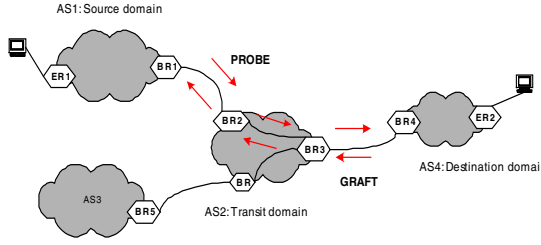


Fig. 1. A reference network

In order to form an inter-domain resource reservation request, the BGRPP agent of the source domain creates a PROBE message, where the required amount of resources and the destination address are specified. However, it is not evident, to which sink tree this reservation will belong. So the PROBE message is forwarded between BGRPP agents hop-by-hop along the BGP route until it reaches the destination domain. On its way towards the destination domain, the path of autonomous domains traversed by the PROBE message is recorded within it and resource availability within each domain is checked upon. Taken into consideration that adequate resources can be provided by the BGRPP agents forming the end-to-end path, the PROBE message reaches its destination. Obviously, the last BGRPP agent that corresponds to the root of the sink tree can assign a sink tree identifier to the reservation, which uniquely identifies the sink tree the reservation belongs to. Then, a GRAFT message is generated containing this identifier. The GRAFT message travels back to the source along the recorded path. Each BGRPP agent belonging to this path, after processing the GRAFT message, aggregates the reservation with the existing ones pertaining to the same sink tree and reserves the requested resources.

Finally, REFRESH messages are exchanged regularly between BGRPP agents with the aim of preserving the reservation state established at the corresponding routers.
In this way, BGRPP mainly addresses the scalability problem in terms of the. However, the enhancements proposed in the rest of the paper aim at limiting the path length traversed by the PROBE and GRAFT messages, the state information kept at the border routers and the REFRESH messages' overhead, alleviating to a greater extent the scalability problem.

3 Quiet Grafting

3.1 Requirements

The BGRPP is an inter-domain protocol, which tries to tackle the scalability problems by reducing the amount of state information for each resource reservation as well as the processing of resource reservation messages. Therefore, the need for memory and CPU usage in each router is considerably reduced.
However, in order a resource reservation request to be successfully established in the network, each BGRPP agent forming the end-to-end path should check the resource availability. Moreover, the sink tree that a reservation request belongs to is identified at the root of the sink tree. Consequently, signaling messages still have to travel the full path from the source to destination increasing the signaling overhead and utilizing a significant amount of bandwidth. Aiming at reducing the signaling overhead, the quiet grafting mechanisms are introduced.
The quiet grafting mechanisms should provide an intermediate BGRPP agent with the necessary functionality to successfully answer a PROBE message, before the latter arrives at the destination domain. Towards this end, the following conditions should be met:

1. The BGRPP agent must be able to identify the sink tree, to which the reservation belongs.
2. The BGRPP agent must have pre-reserved resources for this sink tree, so that he can guarantee that the resources are available on the path from the current point to the destination domain.
3. As the last BGRPP agent may no longer be informed about a new reservation, the BGRPP agent must provide means to contact the destination domain, so that resources can also be reserved on the non-BGRPP-controlled path from BR4 to ED2 (Figure 1).

The following sections describe the proposed solutions, which aim at providing an efficient solution to the above requirements.

3.2 NLRI Labeling for Sink Tree Identification

The potential for grafting a new reservation onto an existing sink tree before it reaches the destination domain necessitates the existence of a mechanism that enables the classification of a new reservation to the corresponding tree. The network layer reachability information (NLRI) labeling is considered adequate for serving this purpose due to the property of the BGP protocol that bases its route selection on the

shortest path, thus rendering it unique. Therefore, if the destination IP address of a reservation request belongs to a certain NLRI label, it is certain that there will be a single route (corresponding to a single sink tree) to this destination. Thus, unless there appears a BGP route change, the NLRI labeling suffices to guarantee the identification of a sink tree among those a single border router belongs to. The NLRI information is propagated back from the root of the sink tree in GRAFT messages and this information is stored in each BGRPP agent, which processes the message.
This proves to be of great importance since a reservation request (PROBE message) can be satisfied earlier (answered with a GRAFT message) without having to travel all the way towards the root of the sink tree as long as the latter has been discovered. Successful sink tree identification is a precondition in order to perform successful quiet grafting as described in next sections.

3.3 Signaling in the Last Domain

The quiet grafting mechanism inhibits the forwarding of signaling messages to the destination domain, since these are already answered at some intermediate stage. This has as impact that the last domain is unaware of a new reservation or a release request.
Specifically, resource reservations are carried out or pre-reserved resources are used up to the ingress border router of the destination domain. Therefore, only the ingress border router of the last domain has reserved resources, while no resource reservation is performed on the path from the ingress border router to the egress edge router, which is connected to the destination host. In order to enable though edge-to-edge QoS-aware services, it is necessary to reserve resources within the last domain, which is achieved by allowing for a direct communication between the initiating and the last domain.
In particular, each domain should provide a standardized interface to its intra-domain resource control entity. Therefore, enhancing the information carried by the GRAFT message can solve the problem of signaling in the last domain. It is actually proposed to back–propagate a reference to this interface as well as the IP address of the ingress border router of the last domain within the GRAFT message so that this information is stored at each intermediate BGRPP agent.
In this way, the source intra-domain resource control entity retrieves the reference of the destination's intra-domain entity, which is responsible for reserving resources within the last domain. Consequently, a direct communication between the two domains is achieved and then, the initiating domain can explicitly request the resources in the destination domain, if necessary.

3.4 Resource Cushion Algorithm

When a BGRPP agent receives a REFRESH message indicating a smaller amount of resources than currently reserved and decides not to further forward this message downstream towards the root of the sink tree, then it allocates resources downstream, which are not in use upstream. These resources, which are reserved downstream but not upstream of a BGRPP agent due to retained REFRESH messages, are called resource cushions. A resource cushion for a specific BGRPP agent is tied to a sink

tree. A BGRPP agent may build resource cushions for all of its sink trees. Building resource cushions has as an impact the reduction of the signaling load, since retained REFRESH messages reduce the signaling load of downstream domains. The use of resource cushions for arriving reservation requests further reduces the signaling load of downstream domains, when reservation requests are not forwarded but served from resource cushion immediately.
A BGRPP agent uses two parameters to control the size of its resource cushions. These parameters are:

- RBS: Release block size
- RP: Retain period

Both parameters together define a virtual release rate RR = RBS/RP. Whenever a resource cushion exceeds the RBS, a release timer is activated which runs down during the duration of a single RP. If a resource cushion shrinks to a size below RBS during a running release timer, then the release timer is cancelled. In the case where the release timer runs out without being cancelled, then the corresponding resource cushion is decreased by RBS and a REFRESH message releasing this amount of resources is generated and sent downstream to the next BGRPP agent on the sink tree. Thus, resource cushions are reduced with the release rate RR unless they are used to serve arriving resource requests. RBS and RP are the parameters that control the release rate RR.
In this way released resources are not immediately forwarded towards the sink of the tree, but are used to form resource cushions. Additionally, those retained resources are released step-wise improving in this way the performance of the network.

4 Simulations

4.1 Proof of Concept of Early Identification

Our main goal is to evaluate the effectiveness of the quiet grafting mechanism in terms of the number of hops that a reservation request has to traverse in order to be accommodated into its corresponding sink tree. It is assumed that resources are always available, and therefore the effectiveness of the NLRI information distribution will be solely examined.
The topology for carrying out the simulations has been defined after taking into account the actual topology of the Internet. More precisely, the number of the AS domains that an inter-domain reservation can possibly traverse will not exceed the 9 [10]. In addition, for each transit AS domain, there will be at least two border routers (see Figure 1) that will forward the reservation request to the border router of the adjacent AS domain downstream. Given the fact that there will be one BGRPP agent corresponding to a border router of the AS domain, we can deduce that the path being traversed by a reservation request will consist of a maximum number of 18 BGRPP agents.
Hence, the simulations performed are based on sink tree topologies that vary in depth (4 to 14) in order that they reflect the actual topology of the Internet. Moreover, in regard to the spreading of the sink trees, binary trees have been solely examined since more complex topologies are not considered to be of significant added value with

respect to the goal of the simulations. An example binary sink tree of depth 4 is illustrated in Figure 5.

For identifying the effectiveness of the NLRI information distribution to the nodes of a sink tree, it is essential to examine how the number of populated nodes can contribute to the reduction of the path length. With the term "populated nodes" we denote the BGRPP agents that are aware of the NLRI information of the destination domain (root of the tree) and therefore can identify the destination (sink tree) of the impending reservation request. These nodes are assigned the task of intercepting a reservation request before reaching the root of the tree and silently grafting it onto the sink tree. It would be ideal if every node of a sink tree was populated but obviously, the "number" of populated nodes performing this identification introduces a scalability problem. If this were true, then we would actually bring into scene again the problem that has been tackled by the BGP through the aggregation it performs on routes.

Given a particular binary sink tree, an "initial state" (number) of populated nodes needs to be created in relation to which, the mean path length of a potential reservation request is computed. This state is produced after performing a certain number of reservations towards the root of the tree. The nodes of the tree (not necessarily leaves) dedicated for initiating those reservations are randomly chosen. All the nodes that are traversed while the reservations make their way up to the root are transformed to populated nodes. In this way, an initial state of populated nodes can be produced whose topology significantly mitigates the potential degree of homogeneity introduced by the binary tree.

Our aim is to compute the impact of this state, i.e. of the number of populated nodes, on the average number of nodes that a reservation request has to traverse before striking a populated one. To that end, a certain amount of additional reservation requests to the ones that have produced the "initial state" needs to take place. As before, randomly selected nodes initiate those reservations, which do not modify the initial state of populated nodes, i.e. they do not produce further populated nodes, as is the case with the creation of the "initial state". Therefore, a mean value of the number of nodes traversed before striking populated one can be computed for a particular populated state.

The simulations were carried out for a variety of binary sink trees and a variety of "initial states" (number of populated nodes) for each tree. Nonetheless, we have chosen to demonstrate how a particular sink tree of depth 10 behaves to the augmentation of the populated nodes since it is considered as the most representative for an inter-domain reservation (4 transit AS domains). It can be seen from Figure 2 and Figure 3 how the number of populated nodes, which comprises a relatively low percentage (4%-20%) of the total number of nodes (2047), affects the average hop count of a PROBE (GRAFT) message. Having in mind that for a non-populated tree, where quiet grafting is not activated, the number of nodes traversed is equal to its depth, i.e. 10, it can be deduced that the percentage corresponding to the reduction of the actual hop count attains the values of around 47% to 77% for remarkably small percentages of populated nodes.

However, it can be seen from Figure 4, that in the case of sink trees of small depth (1 to 3 transit domains), the percentage reduction of the path length does not attain the same values. In fact, a comparison between 5 sink trees with depth values of 4, 6, 8, 10 and 12 is presented in terms of the path reduction for the same percentage of populated nodes. It is evident that the sink trees of small depth do not exhibit the same

effectiveness on the reduction of the path length for a certain percentage of populated nodes. For example, a sink tree of depth 6 with 20% populated nodes will attain a 63% reduction of the path length whereas a sink tree of depth 12, for the same percentage of populated nodes, will achieve an 80% reduction of the path length.

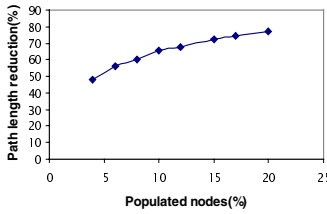


Fig. 2. Path length reduction vs. populated nodes

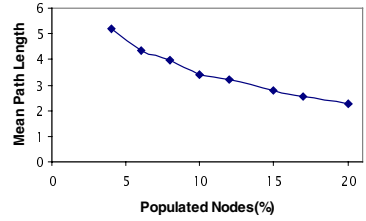


Fig. 3. Mean path length vs. populated nodes

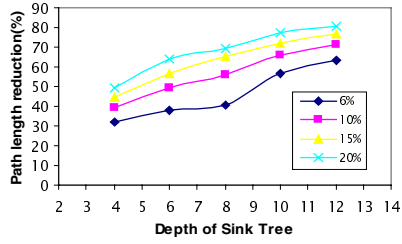


Fig. 4. Path length reduction for different sink trees

4.2 Performance Evaluation of Resource Cushion Mechanism

In order to evaluate the proposed quiet grafting mechanism, the resource cushion algorithm technique and the ability of early sink tree identification are used in conjunction. It is assumed that every BGRPP agent of the sink tree can identify the destination domain, and therefore can execute the following step; the BGRPP agent will check, whether it has enough resources to accommodate a new reservation request. If there are sufficient resources, the PROBE message will be terminated and a GRAFT message towards the corresponding source will be generated.

For that purpose, a simulation scenario is specified, which is based on the realization of a sink tree reflecting a real network that is consisted of a number of inter-connected DiffServ domains. In fact, a simple binary and complete tree is proposed, since the properties of symmetry simplify the interpretation of the results. Each node of the sink tree will be a BGRPP agent capable of performing quiet grafting.

In order to have a relatively large number of nodes, while keeping at the same time complexity at a low level, the depth of the tree for this simulation scenario was chosen to be equal to 4. That results in a total number of N = 31 nodes. This scenario could represent a real network topology, consisting of a number of source domains, the destination domain and the transit domains in order to form the binary tree as

depicted in Figure 5. We have also made the assumption that traffic is injected into the network from the leaves of the tree, which correspond to the source domains. Therefore, reservation requests are only generated at the edges of the network topology.

Concerning the traffic model used for the simulation scenario, a traffic generator that generates reservation requests with exponential distributed inter-arrival times and exponential distributed holding times is attached to nodes belonging to source domains. As regards the size of each reservation request, for the sake of simplicity, a homogeneous scenario is assumed where all the reservations have the same fixed size. It is assumed, without loss of generality that each request asks for one unit of bandwidth, since an infinitive bandwidth capacity is presumed for every node of the tree. Two different scenarios are considered with different traffic loads, in order to examine the effect of load on the performance of the quiet grafting mechanism. Therefore, two traffic conditions are examined with 20 and 100 flows average accordingly.

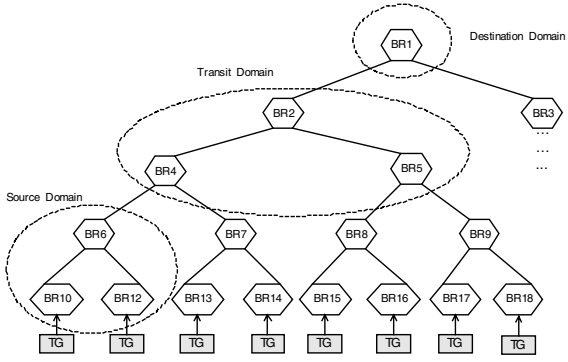


Fig. 5. The reference sink tree

The simulation scenario implies that in an initially "reservation free" tree, traffic flows will be generated at the leaves of the tree contemporarily, following the traffic pattern previously described. During the initial phase, each reservation request (PROBE message) has to travel all the way up to the root of the tree in order that resources are reserved. After the initial phase, sequential requests can be potentially accommodated from the already reserved resources, due to the resource cushion algorithm given the fact that the sink tree has been successfully early identified.

Two performance measures were used to rate quiet grafting mechanism performance: average signaling overhead and average resource utilization for the whole sink tree.

The average signaling overhead S for a sink tree with a number of n nodes, is defined as:

$$S = \frac{\sum_{j=1}^{n} r_{out}(j)}{\sum_{j=1}^{n} r_{in}(j)} * 100\% \quad (1)$$

where rin(j) is the number of received requests of node j and rout(j) is the number of forwarded requests of node j. The signaling overhead indicates the percentage of received resource reservation requests (signaling messages) forwarded by the nodes of the sink tree.
The average utilization ρ for the whole tree is defined as:

$$\rho = \frac{\sum_{j=1}^{n} R_{reserved}}{\sum_{j=1}^{n} R_{assigned}} * 100\ \% \quad (2)$$

In other words, it comprises the average utilization of the sink tree, which is defined as the bandwidth used for the active reservations (Rreserved) divided by the total amount of bandwidth assigned to the created sink tree (Rassigned).
The objective of the simulations is to present the effectiveness of the introduced resource cushion mechanism on reducing the number of messages processed by each node, limiting in this way the number of signaling messages, and on improving the accomplished performance.
Based on the realized resource cushion mechanism, there are two parameters, which need to be appropriately tuned for achieving the desired network performance: the RBS and the RP. The guidelines for setting these parameters compromises a trade-off between achieved resource utilization and signaling overhead. The results concerning the overall utilization and signaling overhead of the sink tree in relation to RP and RBS are shown in the following figures (all simulation scenarios have lasted 10 hours).
Figure 6 and Figure 7 present the effect of the release period RP on the performance of the resource cushion algorithm. As expected, the overall signaling overhead percentage decreases, respectively to the increase of the release period. By increasing the RP parameter (which represents the time during which the free resources should exceed the RBS), resources are released less frequent. Longer retained resources may accommodate future requests, limiting in this way the number of requests forwarded to the next nodes. On the other hand, the increase of the RP has a negative impact on the average utilization of the sink tree, since the resource status is checked less frequently, resulting in longer intervals between subsequently releases of resources. Notice that under heavy load conditions the signaling overhead is reduced to 2,2171%.
In Figure 8 and Figure 9, we can observe the impact of RBS parameter on the performance of the quiet grafting mechanism. As it can be seen, the effect of the RBS is not similar to the one of RP. We can observe that there is a maximum of request forwarding activity between small and large values of RBS. Moreover, the randomly changing but stationary load provokes a re-allocation of the released bandwidth, which results in request forwarding.
Notice that there is an optimum RBS that follows changing demand best. We have to stress here that lower RBSs do not decrease allocated bandwidth fast enough, while larger RBSs cannot follow small temporary demand changes. With a very small RBS, resources are not released fast enough to follow the changing bandwidth demand. Moreover, more requests are forwarded with higher RBS values, because releases follow demand closer. Nevertheless, large RBS values do not allow following small

demand changes. This decreases the number of forwarded requests to lower levels again beyond a certain RBS value.
This is justifiable if we consider that while the RBS is increased (but still gets small values), a greater amount of resources is released concluding in a higher level of utilization. Moreover, since the release resources procedure is more frequently performed, a smaller amount of resources are retained for accommodating future requests resulting in a higher signaling overhead. Nevertheless, as the RBS rises up to really great values, the possibility of accumulating that amount of free resources declines. This significantly impacts the utilization level, particularly under low load conditions. Accordingly, it is anticipated that the signaling overhead will be reduced.
Concluding, it is obvious from the presented results that the algorithm's performance is much more sensitive to the RP than to the RBS parameter. Moreover, the algorithm's performance is significantly improved under heavy load conditions.

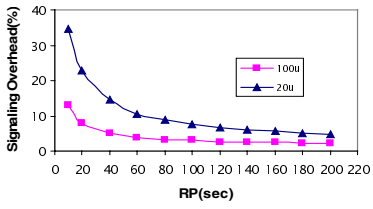


Fig. 6. Signaling Overhead vs. RP

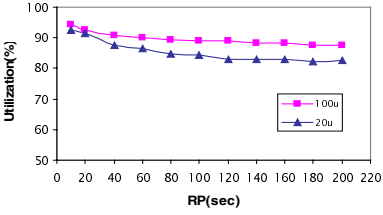


Fig. 7. Utilization vs. RP

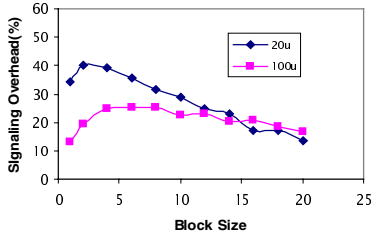


Fig. 8. Signaling Overhead vs. RBS

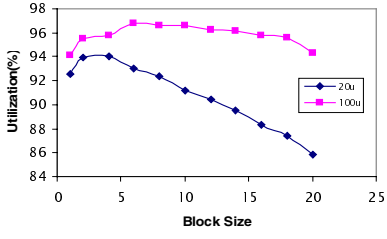


Fig. 9. Utilization vs. RBS

5 Conclusions

In this paper, a scalable extension to the BGRP approach, namely BGRP Plus, was presented. Based on the BGRP concepts, a set of mechanisms was proposed and analyzed followed by simulations supporting the efficiency and the performance of BGRP Plus. More specifically, a proposal for the identification of the sink tree was investigated and a resource cushion algorithm was described and evaluated in order to demonstrate the great improvement that is achieved in terms of scalability.
Summarizing, the gain from the use of quiet grafting mechanism over a scenario without this mechanism is approximately 80% concerning the signaling overhead.

There is clearly a trade off situation arising between signaling overhead and resource utilization. The cost of having high resource utilization is the high signaling overhead whereas an effort to limit the signaling overhead results in a degradation of the utilization. However, the resource cushion algorithm offers two parameters that can be used as tuning knobs to adapt the behavior of the quiet grafting mechanism appropriately.

Acknowledgment. This work was performed in the framework of IST Project AQUILA [11] (Adaptive Resource Control of QoS Using an IP-based Layered Architecture - IST-1999-10077) funded in part by the EU. The authors wish to express their gratitude to the other members of the consortium for valuable discussions.

References

1. Braden, R., Clark, D., Shenker, S.: Integrated services in the Internet architecture: an overview. RFC 1633, IETF, June 1994
2. Black, D., Blake, S., Carlson, M., Davies, E., Wang, Z., Weiss, W.: An Architecture for Differentiated Services. RFC 2475, IETF, December 1998
3. Nichols, K., Jacobson, V., Zhang, L.: A Two-bit Differentiated Services Architecture for the Internet. RFC 2638, IETF, July 1999
4. QBone Bandwidth Broker Architecture, Work in Progress, URL: http://qbone.internet2.edu/bb/bboutline2.html
5. Next Steps in Signaling (NSIS WG), URL: http://www.ietf.org/html.charters/nsis-charter.html
6. Salsano, S., Genova, V., Ricciato, F., Winter, M., Koch, B.F., Dimopoulou, L., Nikolouzou, E., Sampatakos, P., Venieris, I.S., Eichler, G.: Inter-domain QoS Signaling: the BGRP Plus Architecture. Internet Draft, May 2002
7. Pan, P., Hahne, E., Schulzrinne, H.: BGRP: Sink-Tree Based Aggregation for Inter-Domain Reservations. Journal of Communications and Networks, Vol. 2, No. 2, June 2000, pp. 157–167
8. Rekhter, Y. and Li, T.: A border gateway protocol 4 (BGP-4). RFC 1771, IETF, March 1995
9. Traina, P.: BGP-4 Protocol Analysis. RFC 1774, IETF, March 1995
10. NLANR: Nlanr as path lengths. Web site at http://moat.nlanr.net/ASPL/
11. Aquila project: URL: http://www.ist-aquila.org

Measurement-Based Admission Control in the AQUILA Network and Improvements by Passive Measurements

Marek Dąbrowski[1] and Felix Strohmeier[2]

[1] Warsaw University of Technology, Institute of Telecommunications,
Nowowiejska 15/19, 00-665 Warsaw, Poland
mdabrow5@tele.pw.edu.pl
[2] Salzburg Research Forschungsgesellschaft mbH, Advanced Network Center,
Jakob-Haringer-Straße 5/III, 5020 Salzburg, Austria
fstrohmeier@salzburgresearch.at

Abstract. This paper discusses the application of measurement-based admission control methods for efficient resource utilization in the premium variable bit rate service of the AQUILA QoS IP network. The design of the architecture for measuring aggregate mean bit rate to support admission control is presented together with the results of testbed experiments with the prototype implementation. Additionally, the paper discusses possible improvements of the current approach by application of an admission control algorithm taking into account the bit rate variance of the traffic aggregates. A method for measuring the bit rate variance of traffic aggregates produced by MPEG video sources is proposed and verified by simulation and experiments.[1]

1 Introduction

For the deployment of QoS in data networks, a major necessity is the restriction of user traffic to the networks' limitations. "Admission control" (AC) in general has to be used to ensure that upcoming traffic fits into the currently free resources (i.e. bandwidth) of the QoS network without degradation of the service quality of existing flows. Different kinds of admission control algorithms have been developed, summarized and evaluated e.g. in [5,9]. The input parameters for the algorithms are: (1) the current situation in the network, (2) the requested QoS for the upcoming flow (flow declaration) and (3) the overall network capacity. While (2) and (3) are usually known (1) is not. It has to be retrieved either by accumulation of the declarations of all previous flows (declaration-based admission control, DBAC) or by measurement (measurement-based admission control, MBAC). Depending on the expected traffic to the network, either one or the other AC algorithm will be best suited.

[1] This work was performed in the AQUILA project, partly funded by the EU (Adaptive Resource Control for QoS Using an IP-based Layered Architecture, ref. IST-1999-10077 [13]).

W. Burakowski et al. (Eds.): Art-QoS 2003, LNCS 2698, pp. 189–202, 2003.

The IST project AQUILA implemented a QoS architecture for IP networks [3] based on the DiffServ concept [1] and introduces four premium traffic classes for different kinds of applications in addition to the best effort traffic class. With these traffic classes, the currently available network services of the AQUILA QoS network are: Premium Constant Bit Rate (PCBR) for low bandwidth constant UDP traffic like voice calls, Premium Variable Bit Rate (PVBR) for variable UDP traffic like video streaming, Premium Multimedia (PMM) for long-living TCP streams like FTP or TCP-based video streaming and Premium Mission Critical (PMC) for short-living TCP streams like transactions and online games. The implementation supports the configuration of separate AC algorithms for each traffic class. The MBAC algorithms measure the mean bit rate of the traffic class aggregate in constant intervals [2,11].

After presenting the current AQUILA MBAC approach for the PVBR network service in Section 2, this paper proposes an improvement of the MBAC approach by measuring not only the mean bit rate but also the variance of the traffic aggregate. The knowledge of the variance allows the usage of more sophisticated MBAC algorithms like in [7] for a better link utilization. The paper discusses this algorithm and its gains in Section 3. In Section 4 the method for measuring the bit rate variance by passive link monitoring is proposed and verified by simulations and measurements with traffic traces.

2 MBAC for AQUILA PVBR Service

The Premium Variable Bit Rate (PVBR) service is designed for handling streaming VBR traffic with target QoS objectives defined as low packet losses and low packet delay. A candidate application for using this service is real-time video, like video-conference or live streaming video.

For the purpose of AC in this network service, a so-called rate envelope multiplexing (REM) scheme [4,5] with a quasi-bufferless link model (i.e. with small buffer for absorbing simultaneous arrivals of packets) is recommended. Let us assume a fluid-flow model, i.e. source i submits its traffic to the network with the instantaneous bit rate $\hat{X}_i(t)$, where $\hat{X}_i(t), i = 1, \ldots, N$, is a set of independent random variables. Assuming that the considered stochastic process has stationarity property, it is sufficient to omit the time index t and deal with the stationary distributions of X_i.

Packet losses occur when the total instantaneous bit rate of active sources exceeds the link capacity. For calculating the loss probability, typically the theory of Chernoff bounds for the sum of random variables is used [4,10]. Parameters of random variables X_i, which represent the characteristics of user traffic, are obtained either from user declarations or, alternatively, from measurements of real traffic in the network.

The DBAC methods assume, that during the QoS reservation setup phase users submit to the network declarations referring to the peak, h, and sustained, r, (as an upper bound for the mean, m) bit rates. On the basis of these values, the required link capacity, expressed in the form of effective bandwidth is

calculated, e.g. using Lindberger's formula [4]. Such method was tested at the beginning phase of AQUILA project [6]. Unfortunately, it appeared that it is rather difficult task for a user to precisely specify a priori a proper value of sustained bit rate. Remark, that this value is policed and incorrect declarations could cause undesirable packet dropping. In addition, even if it is possible to make correct declarations in the case of stored video, it is not possible to be done in the case of e.g. live video. As a consequence, a user will have rather tendency to over-declare the sustained bit rate, leading to ineffective bandwidth utilization. The above motivates introducing MBAC instead of DBAC. By applying effective MBAC algorithms, one can expect the following profits:

- Simplification of the traffic declarations; usually it is difficult for the user to specify accurate parameters other than the peak bit rate;
- Usage of knowledge about the volume of submitted traffic to the network. This should result in better network utilization (finally, leading to more accepted flows), since in some cases the carried traffic could vary from declaration;
- Observation of the stochastic nature of the user traffic more accurately than assuming traffic description by deterministic parameters, like done when using DBAC.

2.1 Effective Bandwidth with Known Mean Bit Rate

The MBAC method selected for the PVBR service is the algorithm based on Hoeffding bound [9,7]. It is relatively easy to implement since it requires only the measurements of aggregate mean bit rate. A new flow with declared peak bit rate h_0 is admitted to the system of capacity C with N running flows, each with a declared peak bit rate $h_i (i = 1, \ldots, N)$, and measured aggregate mean bit rate M, only if:

$$h_0 + M + \sqrt{\frac{\gamma}{2}\left(\sum_{i=1}^{N} h_i^2 + h_0^2\right)} \leq C \tag{1}$$

where $\gamma = -\log(P_{loss})$ and P_{loss} is the target packet loss probability, e.g. 10^{-4}.

Note that the considered AC algorithm does not require that the users declare the value of sustained bit rate, r. Instead, it uses the measured mean bit rate M of the traffic that users submit to the network. Therefore, the AC method allows the admission of more flows than the DBAC approach, when the traffic submitted by the users is considerably smaller than the declared one.

2.2 Implementation and Verification of Hoeffding Bound MBAC

In the AQUILA network, the AC agent (ACA) [3] is the component responsible for performing admission control on the network edge and polls the router in predefined constant time intervals L and reads the statistics related to the number of bits transmitted on the given link. The bit rate sample in interval j is then

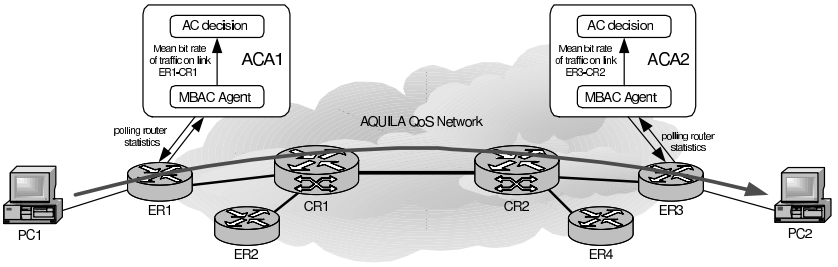


Fig. 1. AQUILA MBAC architecture and trial network

calculated as $S_j = \frac{D_j}{L}$, where D_j denotes the number of bits transmitted within j-th interval, and L is the length of the interval. A moving window algorithm is used for estimation of the stationary mean bit rate, i.e. it is calculated as arithmetic mean from bit rate samples S_j,$(j = 1, \ldots, W)$, collected in W latest measurement intervals.

The measurement mechanisms were implemented and verified in the laboratory testbed (see Figure 1), installed in Polish Telecom R&D in Warsaw. Artificial traffic (on the flow- and packet-level), was generated between PC1 and PC2. Flow inter-arrival time and flow duration were exponential distributed with mean equal to 10s and 120s, respectively. Within each flow, packet traffic was generated according to an MPEG4 trace from [12], with peak bit rate $h = 0.94Mbps$ and mean bit rate $m = 0.131Mbps$.

The mean bit rate M, measured on the ingress link ER1-CR1 by MBAC component in the ACA corresponding to router ER1, was monitored for the period of the test. The length of the measurement interval was $L = 2s$, and the measurement window size was $W = 10$ values. Figure 2 depicts the time-plot of M measured by ACA (dashed line), comparing to the real value of the aggregate mean bit rate, calculated as the real mean bit rate of one flow ($m = 0.131Mbps$), multiplied by the number of flows running at particular time instant.

One can observe, that the measured value closely follows the real mean bit rate. The experiment confirms that the measurement of the mean bit rate by

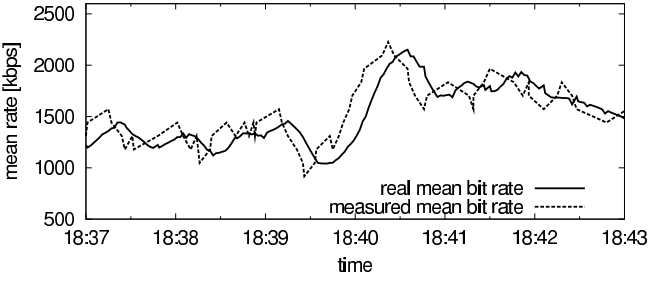


Fig. 2. Measured mean bit rate of aggregate traffic on the link ER3-CR2

router polling and moving window based estimation gives satisfactory results and can be effectively applied for the purpose of supporting admission control.

3 Improving MBAC by Measuring the Variance

The efficiency of AC can be further improved by taking into account not only the mean load, but also the variability of traffic submitted to the network. Below we recall the method of calculating effective bandwidth with known bit rate variance of aggregate traffic stream, originally presented in [7,8], and discuss its application for QoS IP networks.

3.1 Effective Bandwidth with Known Bit Rate Variance

The MBAC algorithm presented in [7,8] allows an improved estimation of the required bandwidth, by taking into account the measured bit rate variance of the traffic aggregate, additionally to the declared parameters of particular flows: peak bit rates h_i and sustained bit rates r_i. Notice, that these are the same parameters like users have to declare with DBAC. However, in DBAC, the real variability of traffic sources is unknown. Therefore, it is assumed that the bit rate variance of particular flow is bounded by a "worst-case" variance of ON-OFF source with peak and mean bit rates equal to h_i and r_i, respectively. On the contrary, the variance-based MBAC algorithm takes into account real stochastic characteristics of user traffic and thus should allow improvement of resource utilization.

A flow with declared peak bit rate h_0 and sustained bit rate r_0, is admitted to the system of capacity C, with N running flows, each with declared peak bit rate h_i and sustained bit rate $r_i (i = 1, \ldots, N)$, and with measured bit rate variance of aggregate stream equal to V, only if:

$$e_0 + \sum_{i=1}^{N} e_i - \delta_{K'+1} \left(\sum_{i=1}^{K'+1} r_i(h_i + r_i) - V \right) - \sum_{i=1}^{K'+2} \delta_i r_i (h_i - r_i) \leq C \qquad (2)$$

where coefficient δ_i is calculated from the declared parameters as:

$$\delta_i = \frac{(h_i - e_i)e_i}{h_i^2 (h_i - r_i) \log \frac{h_i - r_i}{h_i - e_i}} \left[\frac{h_i(e_i - r_i)}{r_i(h_i - e_i)} - \log \frac{e_i(h_i - r_i)}{r_i(h_i - e_i)} \right] \qquad (3)$$

Index K' is determined by numbering the sources such that $\delta_1 \leq \delta_2 \leq \ldots \leq \delta_N$ and solving the equation:

$$\sum_{i=1}^{K'} r_i(h_i - r_i) \leq V \leq \sum_{i=1}^{K'+1} r_i(h_i - r_i) \qquad (4)$$

Value of e_i represents the declaration-based effective bandwidth of a source, calculated as a unique solution of the equation:

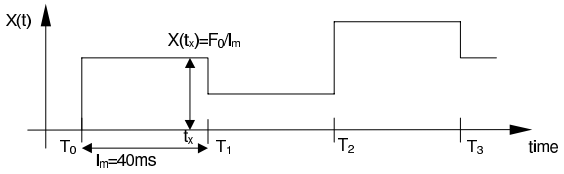


Fig. 3. Fluid-flow model of traffic produced by MPEG video source. Changes of continuous-time bit rate function occur in multiples of frame period, equal to 40ms

$$\frac{1}{h_i}\log\frac{e_i(h_i-r_i)}{r_i(h_i-e_i)}-\frac{1}{e_i}\log\frac{h_i-r_i}{h_i-e_i}=\frac{\gamma}{C} \tag{5}$$

Notice, that e_0 is the effective bandwidth of a new flow, which requests admission to the network.

3.2 Fluid-Flow Bit Rate Variance of MPEG Video Traffic

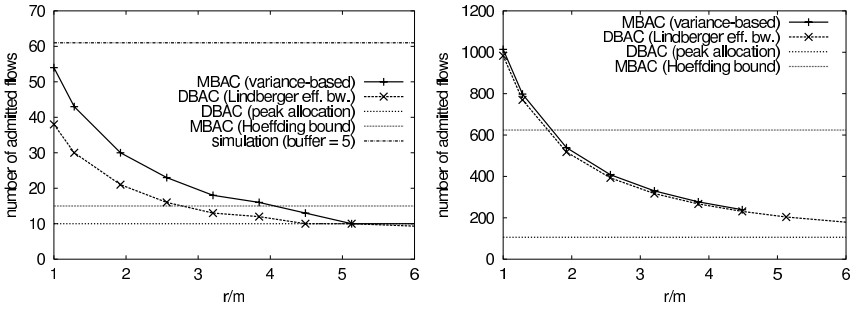
The MBAC algorithm recalled in Section 3.1 assumes that the bit rate variance of fluid-flow model of aggregate traffic stream is known. Below, we discuss the definition of such variance in the case of traffic produced by MPEG4 video source. A VBR video encoder generates frames of different sizes in constant frame intervals $l_m = 40ms$ (which corresponds to standard frame rate of 25 frames/s). Traces of MPEG4 sources are publicly available, e.g. from [12].

Trace file contains a sequence of sizes of video frames $F_m, m = 0, \ldots, M$, produced during the encoding process of a particular movie. Frames are emitted at the discrete time instants $T_m, m = 0, \ldots, M$. Following [12], we assume that the discrete frame trace can be converted into a fluid-flow by transmitting the frame of size F_m with constant rate $\frac{F_m}{l_m}$ within its frame period:

$$X(t)=\frac{F_m}{l_m} \qquad \text{for} \quad T_m \le t \le T_{m+1}, m=0,\ldots,M \tag{6}$$

We omit here the index i, representing the numbering of sources. Figure 3 illustrates the fluid-flow model of traffic produced by MPEG4 video source. Notice, that the function $X(t)$ corresponds to the realization of the stochastic process $\hat{X}(t)$ introduced in Section 2 as the theoretical model for the instantaneous bit rate.

Assuming a constant value of the frame interval, the variance of bit rate $X(t)$ can be easily calculated from the sequence of frame sizes F_m, obtained from the trace file. For example, the bit rate variance of the Star Wars IV movie encoded with medium quality level [12] is equal to $0.008Mbps^2$. The peak bit rate of this source is $h = 0.94Mbps$, while the mean bit rate is $m = 0.078Mbps$. Thus, the "worst-case" variance, being the variance of ON-OFF source, is equal to $m(h-m) = 0.067Mbps^2$, i.e. much larger than the bit rate variance of the fluid-flow model.



(a) 10Mbps (b) 100Mbps

Fig. 4. Comparison of MBAC and DBAC on different link capacities

3.3 Effectiveness of the Variance-Based MBAC

The effectiveness of variance-based MBAC method, in comparison to Hoeffding bound MBAC, Lindberger's effective bandwidth DBAC [4] and simple peak bit rate allocation scheme, was studied in detail. A homogenous set of Star Wars IV movie sources, encoded with medium quality was considered. Note, that the number of admissible flows in the case of variance-based MBAC, similarly as in the case of DBAC method, depends on the declared value of r. Let us define the over-declaration factor r/m, which represents the difference between user's declaration and actually submitted traffic. Figure 4 shows the effectiveness of investigated AC methods as a function of over-declaration factor, on the links with capacity 10 Mbps and 100 Mbps.

One can observe, that the profit from applying the variance-based MBAC in the considered traffic scenario, especially on the link with capacity 10 Mbps, can be significant. For example, assuming that users precisely characterize their traffic (over-declaration factor is equal to 1), the DBAC method admits 38 flows, while the variance-based MBAC admits 54 flows. The results of simulation experiments (carried out on the 10 Mbps link with buffer size of 5 packets, properly dimensioned to absorb simultaneous packet arrivals [4]) indicate, that 61 flows could be admitted without violating the 10^{-4} loss probability target.

Notice that in case of both, 10 Mbps and 100 Mbps links, the Hoeffding bound MBAC algorithm allows higher resource utilization than other methods if users significantly over-declare their traffic.

In Table 1 we present analysis of effectiveness of variance-based MBAC, comparing to Lindberger's effective bandwidth DBAC, in the homogenous scenarios with different video sources (representing movies and sport events). Proper declaration is assumed, i.e. $r/m = 1$.

One can observe, that in most cases the real fluid-flow variance is significantly smaller than the "worst-case" variance. Therefore, the usage of variance-based

Table 1. Profit of variance-based MBAC with different video traces

Quality	Title	Bit Rate [Mbps]		Variance [Mbps^2]		Gain MBAC [%]	
		peak	mean	worst-case	fluid-flow	10 Mbps	100 Mbps
High	Jurrasic Park I	3.3	0.77	1.95	0.201	50%	16%
	Silence	4.4	0.58	2.216	0.215	0%	28%
	Star Wars IV	1.9	0.28	0.454	0.034	100%	10%
	Formula I	2.9	0.84	1.73	0.12	0%	14%
	Office-Cam	2.0	0.4	0.64	0.19	25%	3%
Medium	Jurrasic Park I	1.7	0.27	0.386	0.048	67%	7%
	Silence	2.4	0.18	0.4	0.045	125%	13%
	Star Wars IV	0.94	0.08	0.069	0.008	42%	3%
	Formula I	1.4	0.29	0.322	0.037	38%	5%
	Office-Cam	1.0	0.11	0.098	0.065	0%	-9%

MBAC is profitable. It should be noted, that the approach has its limitations when the real variance is close to the worst-case variance (see e.g. the Office-Cam source), which anyway is not the case in most of the video sources.

4 Measuring the Bit Rate Variance of Traffic on the Link

The difficulty with measuring the bit rate variance is caused by the discrete nature of packet traffic in the IP network. The real traffic is not a fluid-flow, but a sequence of packets of certain sizes. On the other hand, the AC formulas presented in Section 3.1 assume the known variance of fluid-flow model of traffic, so the goal of the measurement procedure should be to approximate the parameters of the fluid-flow model from data obtained by monitoring the packet traffic in the network.

4.1 A Method for Measuring Bit Rate Variance

The proposed procedure for measuring bit rate variance of traffic carried on the link includes three steps:

- Passive monitoring of the link, which allows identifying the packet transmission events and counting the number of transmitted bits;
- Converting the data obtained by monitoring into the approximation of fluid-flow model of traffic;
- Estimation of variance of instantaneous bit rate of the fluid-flow approximation of traffic.

By using special hardware for passive link monitoring (see below in Section 4.3), it is possible to gather the packet sizes and times of packet transmission events. Then, we obtain the fluid-flow approximation by dividing the time into constant intervals and averaging the traffic rate within those intervals (see

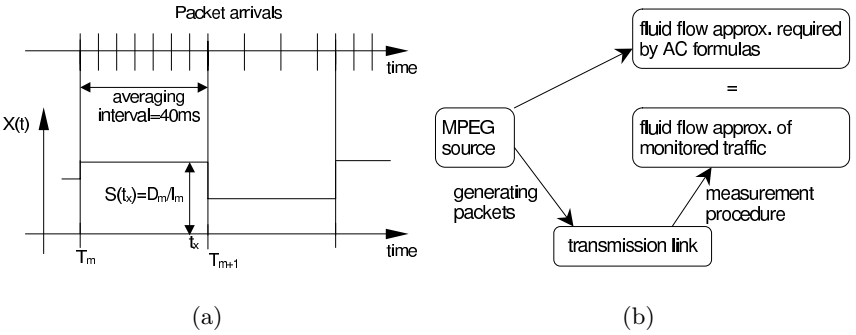


Fig. 5. Conversion of data obtained by passive link monitoring into the approximation of a fluid-flow model

Figure 5(a)). The rate in interval m is thus calculated as $S_m = \frac{D_m}{L}$, where D_m denotes the number of bits transmitted within m-th interval, and L is the length of the averaging interval. Remark, that in practical situations, it is possible that the link transmits only a part of the packet in one measurement interval, and the rest in the consecutive interval. This fact should be taken into account in the implementation of the averaging procedure.

The fluid-flow approximation of traffic carried on the link is thus:

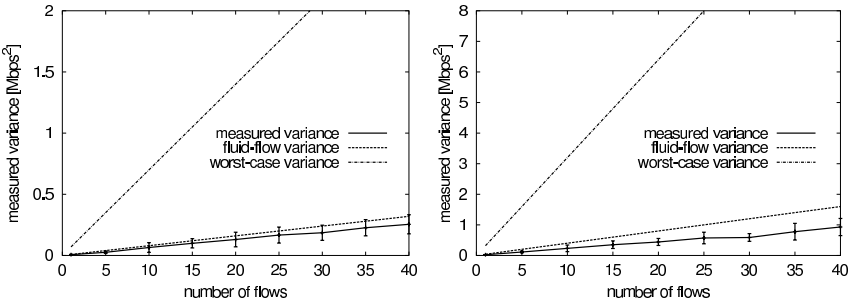
$$S(t) = \frac{D_m}{L} \qquad \text{for} \quad T_m \leq t \leq T_{m+1}, m = 0, \ldots, M \tag{7}$$

The value of $S(t)$ represents the bit rate of aggregate traffic. Note, that assuming independence of sources, the variance of $S(t)$ should be equal to the sum of variances of running flows.

In the case when the considered traffic is produced by MPEG video sources, the obtained fluid-flow approximation should possibly closely resemble the theoretical fluid-flow model of MPEG video source, presented in Section 3.2. Therefore, the proper choice for the length of averaging interval L seems to be the value of 40ms, equal to the length of the frame period of MPEG encoder, l_m. Figure 5(b) illustrates the concept of conversion of traffic monitored on the link into its fluid-flow approximation.

It should be noted, that the presented measurement procedure gives accurate results, when the averaging intervals are synchronized with the frame periods of the source. Unfortunately, this will be practically impossible, since the traffic observed on the link constitutes of a number of single flows, produced by video sources transmitting frames independently of each other.

The last step in the measurement procedure is the estimation of variance of the instantaneous bit rate of fluid-flow approximation of traffic. The required theoretical variance corresponds to the stationary case, when a certain number of flows are running in the system without arrivals or departures. We estimate



(a) Star Wars IV (b) Formula I

Fig. 6. Measurements of variance of traffic produced from traces; window size equal to flow duration

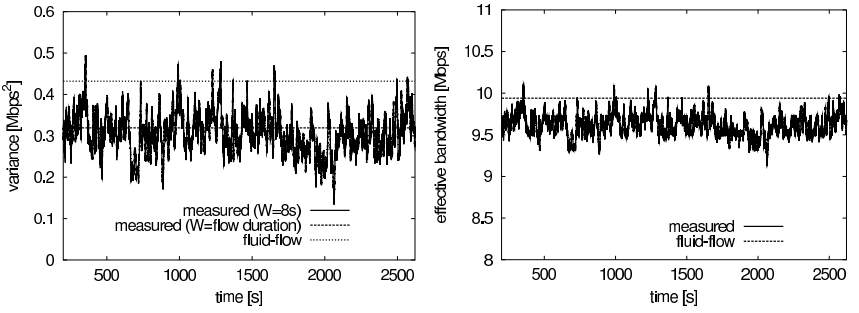
the stationary variance by taking into account the traffic rates collected within the moving window of certain length W. The desired variance estimate V is thus calculated as a sample variance of S_j, where $j = 1, \ldots, W$ denotes the bit rate within the j-th previous averaging interval. Notice, that assuming that the number of current interval is 0, the interval number 1 is the latest finished interval.

The fact, that the measurement procedure assumes that the monitored traffic is produced only by the MPEG video sources, constitutes certain limitation for applying the method. Anyway, in the case of AQUILA QoS IP network, the variance-based MBAC is considered in the PVBR network service, which is especially designed for carrying real-time video traffic, most probably being produced by standard MPEG encoders.

4.2 Verification of the Measurement Procedure by Simulations

The measurement method described in Section 4.1 was verified by simulation. In the first set of experiments, the length of the measurement window was equal to the duration of the flow. Therefore, the measured variance should be equal to the stationary bit rate variance, assuming a fixed number of flows in progress. Packet traffic was generated to the system according to the trace file. Two trace files were used: "Star Wars IV" and "Formula I", both encoded with medium quality. The parameters of the sources were given in Table 1. Each flow starts from a random frame within the trace file. The frame periods of particular sources are not synchronized, i.e. sources do not start sending frames exactly at the same time. The duration of each simulation is 1 hour and is equal to the duration of the trace.

Figure 6 depicts the measured bit rate variance in the series of simulation runs with different number of running flows. For comparison the "worst-case"



(a) Variance (b) Effective Bandwidth

Fig. 7. Time plot of variance and derived effective bandwidth with measurement window size $W = 8s$ (200 values)

variance is also shown, which is the sum of variances of ON-OFF sources with equivalent values of h and m (as assumed in the DBAC method). The third line depicts the sum of variances of the fluid-flow source model, calculated from the trace. This value should be considered as a reference, because this is the variance assumed by the model used for developing the theoretical AC formulas.

The 95% confidence intervals for results obtained in 20 simulation runs are presented. Remark, that the differences in consecutive simulation runs are caused by the randomness in the synchronization between averaging intervals and frame periods of the sources. The obtained results confirm, that the measured bit rate variance closely approximates the theoretical variance of the fluid-flow model. Although the measured value slightly underestimates the variance of the fluid-flow, the obtained results are satisfactory and prove that the proposed method works according to expectations.

Another set of simulation experiments was conducted, with the number of running flows fixed to 54. Now, the measurement window was set to $W = 200$, which corresponds to the window duration of 8 seconds. The time plot of variance estimation, recorded in consecutive intervals is depicted in Figure 7(a). One can observe, that the estimate of variance fluctuates around the value, which corresponds to the stationary variance. This is caused by the fact that the characteristics of video sources change during the movie duration, i.e. taking into account only a fraction of the flow in the measurement procedure causes some information loss.

Figure 7(b) plots the total effective bandwidth of running flows, calculated with the formulas from Section 3.1, taking into account the measured variance estimate. Note, that this is exactly the value, which would be calculated by the AC procedure for the purpose of deciding on the admission or rejection of a new flow, arriving at a particular time instant. As the number of running flows is

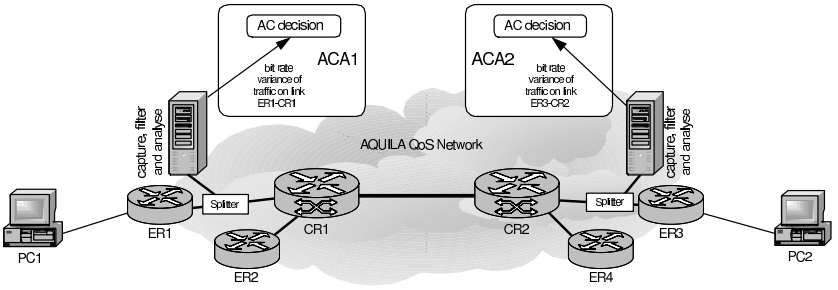


Fig. 8. Improved AQUILA MBAC architecture and trial network

constant throughout the simulation, the real value of total effective bandwidth also does not change. Observed variability of estimated effective bandwidth results from the fluctuations in time of variance estimation. Anyway, the effective bandwidth calculated by taking into account the measured variance is close to the theoretical value.

4.3 Feasibility Study and Implementation Issues

As described in Section 2 the current MBAC approach in AQUILA uses statistics from the routers for measuring the mean bit rate. To gather the input data for admission control, the router is polled in constant time intervals via the command line interface of the router, retrieving the number of transmitted bytes for the targeted traffic class. This means communication overhead, additional load on the router's control processor and the delays because of connection establishment and release to the router. Experiments in the trials of the AQUILA project showed, that for each router polling 4kBytes of data are transmitted between the router and the ACA. Therefore the time interval between two measurement requests must be in the order of seconds.

The approach described in Section 3 needs also the bit rate variance as input parameter. The measurement of bit rate variances demands the instantaneous bit rate to be measured in very small time intervals (e.g. every 40ms) and therefore cannot be realized by router polling. The proposed solution is passive link monitoring, which requires a packet capture card to be attached to the output link of the router. Depending on the transfer medium and on the link speed, different packet capture hardware is available on the market.

Figure 8 shows the concept of the new approach with using passive measurement equipment in contrast to the implementation with router polling as depicted in Figure 1. Depending on the system design, the data gathering, filtering and analysis can either be separated in a single machine or co-located with the ACA.

For the scenario described in this paper a DAG3.5 card [14] was attached to the monitored link (ATM/OC3) via an optical splitter, which supports capturing

rates up to $622Mbps$. For this kind of analysis only the IP headers (including the packet length information) were from interest, i.e. only the payload of the first ATM cell (48 bytes) from each IP packet was needed. For the analysis discussed in this paper the captured data has been filtered and analyzed off-line after the measurements. For the validation of this approach the measurement results of the capture process have been compared to the simulation results. The results in Table 2 for one and five flows shows a sufficient accuracy of the measurement with a relative error less than 10%.

Table 2. Measurement results vs. simulation results

No. of flows	simulation mean variance	variance calculated from DAG trace
1	0.00625	0.0068
5	0.03575	0.035

For the MBAC support, the data can be made available on-line e.g. by filtering the requested traffic class directly on the ARM processor mounted on the DAG board and on-line calculation of the bit rate variance in the hosting PC.

5 Conclusion

The paper discussed the application of MBAC methods within the PVBR service of the AQUILA QoS IP network. While the current implementation does only measure the mean bit rate in constant, large intervals ($\geq 2s$), an improvement by measuring the bit rate variance is proposed, i.e. the mean bit rate has to be measured in much smaller intervals (e.g. $40ms$). The comparison of the two methods showed a gain in the number of admissible flows when using the variance-based MBAC algorithms. The AQUILA architecture can be enhanced by the proposed improvements, if special equipment for passive link monitoring and additional software for on-line data analysis is integrated.

References

1. Blake, S. et al.: An Architecture for Differentiated Services. RFC 2475, December 1998
2. F. Ricciato et al.: Specification of traffic handling for the second trial. AQUILA deliverable D1302, December 2001
3. M. Winter et al.: Final system specification. AQUILA deliverable D1203, April 2002
4. Roberts, J., Mocci, U., Virtamo, J. (ed.): Final Report COST 242, Broadband network teletraffic: Performance evaluation and design of broadband multiservice networks. Lectures Notes in Computer Science 1155, Springer, 1996

5. Tran-Gia, P., Vicari, N. (ed.): Final Report COST 257, Impact of New Services on the Architecture and Performance of Broadband Networks. compuTEAM, Wuerzburg, 2000
6. Bak, A., Burakowski, W., Ricciato, F., Salsano, S., Tarasiuk, H.: A Framework for Providing Differentiated QoS Guarantees in IP-based Network. Computer Communications, March 2003
7. Brichet, F., Simonian, A.; Conservative Gaussian models applied to Measurement-based Admission Control. IWQoS'98, USA, May 1998
8. Brichet, F., Simonian, A.: Measurement-based CAC for video applications using SBR service. proceedings of the PMCCN conference, IFIP, Japan, November 1997
9. Floyd, S.: Comments on Measurement-based Admissions Control for Controlled-Load Services. Technical Report, Lawrence Berkeley Laboratory, July 1996
10. Gibbens, R.J., Kelly, F.P.: Measurement-based connection admission control. 15th International Teletraffic Congress, June 1997
11. Brandauer, C., Burakowski, W., Dąbrowski, M., Koch, B., Tarasiuk, H.: AC algorithms in AQUILA QoS IP network. To appear in European Transactions in Telecommunications
12. Fitzek, F.H.P., Reisslein, M.: MPEG-4 and H.263 video traces for network performance evaluation. Technical report TKN-00-06, Technical University Berlin, Dept. of Electrical Eng., Germany, October 2000
13. The AQUILA project web page. `http://www.ist-aquila.org`, February 2003
14. Endace Measurement Systems. `http://www.endace.com`, February 2003

An Implementation of a Service Class Providing Assured TCP Rates within the AQUILA Framework *

Christof Brandauer and Peter Dorfinger

Salzburg Research, Jakob-Haringer-Str. 5/III,
A-5020 Salzburg, Austria
{christof.brandauer,peter.dorfinger}@salzburgresearch.at

Abstract. This paper investigates an attempt to establish a QoS class that supports long-lived, bulk-data TCP flows that require a minimum rate from the network. The approach is based on a model for TCP flows subject to token bucket marking at the network edge and preferential dropping in the core network. The service class adds admission control functionality and a model for multi-RED queue management to the token bucket marker. The difficulty of parameterizing the mechanisms is discussed and analyzed in an explorative simulation study. A set of configuration parameters that enables a successful operation of the service class is identified and the achievable service provisioning is shown.

1 Introduction

The European IST project AQUILA (Adaptive Resource Control for QoS Using an IP-based Layered Architecture) [1] implements an IP-based quality of service (QoS) architecture on the basis of the Differentiated Services [2,3] philosophy. In the AQUILA approach, the network operator provides a set of *Network Services* to its costumers. A network service defines i) requirements concerning the admissible traffic and ii) the QoS properties provided for the admitted traffic.

One particular network service studied in AQUILA is called Premium Multimedia (PMM). It is intended for greedy TCP-based applications that require a minimum sending rate. Example applications are premium FTP data transfers or TCP-based streaming media applications. The network operator implements a network service by means of admission control, traffic conditioning, queue management, and scheduling.

In order to be able to perform the testbed / real-user trials we strive for an approach that could be implemented with the router equipment available in the project. In this paper we investigate the feasibility of a PMM implementation on the basis of *token bucket marking* and *preferential dropping*.

The foundation of the approach is an analytical model [4] for the TCP sending rate when a TCP flow is subject to a token bucket marker and preferential queue

* This work was partly funded by the European IST project AQUILA under contract IST-1999-10077

W. Burakowski et al. (Eds.): Art-QoS 2003, LNCS 2698, pp. 203–217, 2003.

management. The model shows that there are conditions where the sending rate of the TCP flow can be regulated by the configuration of the token bucket parameters. The goal of the PMM implementation is to permanently enforce these conditions by means of admission control and queue management.

If an application wants to utilize the PMM service it sends a reservation request to the AC entity. The request contains the requested rate R. The AC entity decides whether or not the request can be accepted. If the request is accepted, the requester's ingress edge device is reconfigured which involves the setup of a classifier and the token bucket marker. In any case, the AC decision is signalled back to the requester using the signaling facilities provided by AQUILA framework.

The paper is structured as follows: after briefly summarizing related work in the next section we review in more detail the results of [4] as relevant for this work in section 3. Subsequently, appropriate admission control (section 4) and queue management (section 5) entities are derived. The difficulty of finding appropriate parameter sets is discussed in section 6. The feasibility of the approach is investigated in a large simulation study reported in sections 7 and 8.

2 Related Work

There exist several works [5,6,7,8] that enforce TCP rate control by relying on an "invasive" mechanism where TCP header fields are modified. Packeteer [9] seems to have originally come up with this concept. Their TCP rate controller [10] modifies the receiver window and acknowledgement number and additionally modulates the rate of acknowledgements.

Several "non-invasive" mechanism based on packet marking algorithms have been investigated in the context of Differentiated Services. Many of these algorithms use a static marking profile [11,12,13,14,15,16,17]. Adaptive marking algorithms are proposed in [18,19,20,21].

Unlike these works the AQUILA architecture explicitly acts upon the assumption of an admission control entity limiting access to the offered service classes. The TCP rate controller (TRC) for long-lived TCP flows proposed in [22] essentially requires an admission control entity. It operates as a traffic conditioner at the edge of a domain. Given a requested rate and an estimation of the domain's delay it computes a target packet drop probability and a target delay on the the basis of a TCP model [23]. By enforcing the drop rate and introducing artificial delays the TCP flows are trimmed to the requested rate. The TRC is shown to be unbiased to the requested rate and RTT.

3 Token Bucket Marker

The authors of [4] model the impact of token bucket marking on greedy TCP flows. On the basis of a model for TCP sending behavior [24] they develop an

analytical model for determining the sending rate of a TCP flow when edge-routers use token bucket marking (with statically configured token bucket parameters) and core routers employ active queue management with preferential packet dropping. Using this model (we denote it as *TBM model* in the following) it is shown that there exist conditions where it is not possible to influence the service achieved by a TCP flow through a marking profile. For a different set of conditions it is, however, feasible to achieve a requested sending rate. In that case the sending rate A of a greedy TCP flow is given by eq. 1.

$$A = \begin{cases} R - \frac{3}{2Rp_2T^2} & R \leq \frac{3W}{2T} \\ \frac{4}{3}(R - \frac{3}{2T\sqrt{2}}\sqrt{Z + \frac{1}{p_2}} & R > \frac{3W}{2T} \end{cases} , where\ W = \sqrt{2(Z + 1/p_2)} + 2\sqrt{2Z} \tag{1}$$

Table 1 describes the parameters involved. The necessary condition is a so-called *under-subscribed* scenario which is defined as $p_1 = 0$ and $p_2 > 0$.

Table 1. Token bucket parameters

Parameter	Meaning	Unit
A	token bucket rate	packet/s
Z	token bucket size	packet
R	requested rate	packet/s
p_1	packet drop probability for in-profile packets	-
p_2	packet drop probability for out-profile packets	-
T	round-trip-time (RTT)	s

The results of [4] make the TBM model a promising candidate for a PMM implementation. The model, which is analytically derived from an accurate TCP model, provides closed-loop formulae for the computation of the appropriate marking profile required to achieve a target sending rate. In a simulation study [4] the TBM model is shown to be very accurate over a wide range of values for p_2, T, and R. Given the accuracy of the model and the possibility to practically implement the token bucket marking approach using today's routers we construct the PMM class on the basis of the TBM model.

The goal is to operate the service class in the region where the achieved TCP rate can be regulated through the configuration of the token bucket parameters. We seek to establish the required under-subscribed scenario by combining the token bucket marking with adequate admission control and queue management.

4 Admission Control

The goal of any admission control (AC) functionality is to limit the amount of traffic admitted to a particular service class such that the QoS objectives are

reached for all admitted flows. At the same time, the service class utilization should be maximized.

For PMM a declaration based approach is employed: the requested rate R is part of the reservation request sent to the AC entity. Using the TBM model, the AC entity computes the token bucket rate A according to eq. 1. Now, if $A > R$, the flow requires at least an available bandwidth of A in order to obtain R. If $A \leq R$, an amount of R resources must be available.

On this basis a simple AC rule can be formulated. The resources required by a single flow are expressed by the greater value of the token rate A and the requested rate R. The inequality in eq. 2 ensures that the bandwidth required by the aggregate traffic submitted to the PMM class is smaller than ρ times the reserved capacity C, where ρ is a (tunable) over-provisioning factor; it denotes the fraction of reserved capacity that is at most allocated to resource requests.

$$\sum_{i=1}^{N} \max(A_i, R_i) \leq \rho C \tag{2}$$

The number of flows in the PMM class, including the new one if being admitted, is denoted by N. Note that eq. 2 implicitly assumes that the aggregate PMM traffic is able to fully utilize the reserved capacity C. It is discussed in section 5 why this is a valid assumption. In the AQUILA framework the bandwidth is allocated to the different network services by means of WFQ-based scheduler at each router output port.

It follows from the TBM model that in an under-subscribed scenario each TCP flow achieves a minimum sending rate R_0 when the token bucket marks all packets as out-profile ($A = Z = 0$).

$$R_0 = \frac{1}{T}\sqrt{\frac{3}{2p_2}} \tag{3}$$

A reservation request for a rate $R < R_0$ can be either principally denied (because a rate as small as R cannot be achieved) or it has to be handled in the following way:

- the token bucket parameters are set to: $A = Z = 0$.
- for that particular request, R is replaced by R_0 in the computation of the sum in eq. 2 to take into account that the flow will in fact consume R_0 bandwidth.

5 Queue Management

In order to be combineable with the TBM model, the queue management mechanism must be able to enforce an under-subscribed scenario, i.e. $p_1 = 0$ and $p_2 > 0$. First of all, this requires the ability to distinguish between in-profile and out-profile packets, respectively. We employ a two-color extension [25] of RED [26] queue management. There is one parameter set for in-profile packets

$\{minth_i, maxth_i, maxp_i\}$ and one set for out-profile packets $\{minth_o, maxth_o, maxp_o\}$. A single average queue size is calculated over all arriving packets and depending on the color of the packet the corresponding set of parameters for that color is used. This approach is generally referred to as Weighted RED (WRED).

In order to optimally support the PMM class, the following queue size behavior should be enforced:

- the average queue size converges within the control range for out-profile packets, i.e., between $minth_o$ and $maxth_o$.
- the amplitude of oscillation of the average queue size is bounded and significantly smaller than the difference between $minth_o$ and $maxth_o$.
- the instantaneous queue size is (mostly) greater than zero and smaller than the buffer size.

Such a behavior is clearly beneficial for the PMM class: besides establishing the required under-subscribed scenario, the available link capacity can be fully utilized. First, this provides for optimal resource usage. Second, the predictability of bandwidth utilization is an important input for the AC algorithm. In fact, the AC rule has to take into account the amount of bandwidth the aggregate traffic stream is able to consume – not the capacity reserved for that traffic class.

Moreover, due to the enforced queue size behavior the controlled TCP flows experience a rather constant drop probability for out-profile packets and should be able to handle these drops without resorting to timeouts. Consequently, the sending behavior of the flows is rather smooth.

It must be noted that a careful selection of WRED parameters is required to achieve the above described performance. If the queue management parameters are chosen rather incidentally the average queue size will generally not exhibit such a behavior. See [27] and [28] for a discussion of these effects.

5.1 Implementation

We develop a quantitative model (subsequently called *WRED model*) for setting the parameters of WRED queue management. The WRED model is an extension of the quantitative RED model [28] which can be accessed via the Web under [29]. This RED model calculates the RED parameters $minth$, $maxth$, max_p, w_q, and the buffer size as a function of the scenario parameters bottleneck bandwidth, RTT, and number of TCP flows. The RED model has been developed by assembling an accurate analytical model of TCP sending behavior [24], an analytical model for setting w_q [30] and an empirical model providing the required difference between $minth$ and $maxth$. It is shown in [28] that under the load of long-lived TCP flows, RED's average queue size converges between $minth$ and $maxth$ and the amplitude of average queue size oscillation is about one third of the difference between $minth$ and $maxth$.

The idea of the WRED model is to achieve the same convergence behavior in a two-color environment but without having to drop in-profile packets. This would establish the under-subscribed condition as required by the TBM model.

With (W)RED queue management the long term average queue size is mostly dependent on the maximum drop probability max_p. This parameter determines the aggressiveness of dropping packets when incipient congestion is detected.

If in-profile packets are excluded from the dropping process the WRED parameter $maxp_o$ must be higher than the RED parameter max_p in order to achieve the same overall drop probability. Therefore, to adapt max_p correctly, knowledge of the expected ratio of out-profile packets is required. We define the number of out-profile packets divided by the total number of packets that arrive at the queue as the out-share. The out-share is estimated by a parameter called U. It is influenced by several factors as discussed in section 6.

Due to the existence of an AC framework, convergence of the average queue size within the control range for out-profile packets is feasible. This eliminates the need to drop in-profile packets for the sake of congestion avoidance / control. Thus, in order to exclude in-profile packets from the dropping process, we recommend to set the WRED parameters $minth_i$ and $maxth_i$ to the total buffer size and $maxp_i$ to 1. Note that with this approach there is practically no difference between WRED and RIO [31].

Due to space limitations we must omit here the derivation of the equations for the WRED model. Please refer to [32] for the details. A Web interface to the model can be accessed under [33].

6 Parameterization

The WRED model requires an estimate of the out-share as an input parameter. As discussed below, the out-share is influenced by many factors and may oscillate substantially. For the WRED model to be practically applicable it is thus crucial to achieve the desired convergence behavior even if the real out-share differs significantly from the estimated value U. We investigate in [32] the model's sensitivity on a correct estimate of the out-share. In this study, U is set to 0.5 in order to minimize the deviation from the real out-share. In the simulations where the out-share is significantly smaller than estimated (10% instead of 50%) the average queue size converges to a value higher than $(minth_o + maxth_o)/2$ but remains within the control range for out-profile packets. If the real out-share is higher (90% instead of 50%) the dropping is too aggressive and the average converges to a value lower than $(minth_o+maxth_o)/2$, but again remains between $minth_o$ and $maxth_o$. In any case, the convergence behavior is as desired and the performance of the WRED model is shown to not critically depend on U. The details can be found in [32].

Another critical input parameter needed for the WRED model is an estimate N of the number of flows. This dependency cannot be avoided as the RED behavior is intrinsically influenced by the number of flows. In order to compute a WRED parameter set it is therefore required to estimate the expected number of flows. As it will change over time there is no correct value for N. Clearly, there is a lower bound (zero or more flows) as well as an upper bound for the number of flows that can be active at any time in the system. The upper bound is

determined by the finite reserved capacity and the smallest acceptable requested rate.

Additionally, the TBM model requires the expected drop probability for out-profile packets p_2 as in input parameter. The value of p_2 must thus be estimated a priori through a constant value. Clearly, p_2 can fluctuate heavily as it depends on the level of congestion (number of flows) and the out-share. The out-share itself is influenced by:

- the size of the requested rates:
  it is a general property that TCP flows with smaller window sizes exhibit more "aggressiveness" than flows with larger windows. As a consequence, flows with lower bandwidth requests produce more out-profile packets than flows with higher bandwidth requests. This general effect is reinforced if the token bucket rate A is computed according to the TBM model (eq. 1), as in the TBM model $(A - R)$ is strictly monotonic increasing with increasing R.
- the portion of reserved capacity allocated to accepted requests:
  The PMM class is utilized by greedy TCP flows which always fully utilize the available capacity - independently of the requested rate. Clearly, the token bucket parameters A and Z limit the amount of packets that are marked as in-profile but the amount of out-profile packets is only limited by the portion of unallocated capacity.

Unfortunately it is not possible to make a worst-case estimation of p_2. In the one case, if p_2 is estimated too low, the sending rates of the TCP flows are over-estimated and the resulting token rates are too small. For those requests where $A > R$, the $\max(A, R)$ is smaller than the amount of bandwidth that would in fact be needed by that flow. Thus, in general, too many flows would be admitted.

In the other case, if p_2 is estimated too high, the resulting sending rates of the TCP flows are under-estimated and the computed token rates are too high. This again leads in general to a situation where too many flows are admitted because the real TCP rates are higher than the ones used for the admission control algorithm.

The TBM model requires an estimate T of the RTT. It could be estimated as the propagation plus transmission delays plus the average queueing delay at the WRED bottleneck. It is however difficult to know in advance the number of bottleneck routers in a flow' s path. Additionally, different flows generally have different destinations with different RTTs.

In lack of an analytical model that captures the behavior of the PMM implementation we try to discover parameter configuration dependencies by executing a large amount of packet level simulations. The goal is to discriminate between configurations which enable a successful PMM operation (if possible at all) from those configurations where the probability for a sending rate smaller than R is much larger than zero. Moreover, we investigate the influence of deviations in the parameter estimation on the usability of the various traffic control components.

7 Simulation Study

We use the well-known packet-level network simulator `ns-2` from [34] and extend it by admission control functionality. The AC entity decides on the basis of eq. 1, 2, 3 whether the request can be accepted.

The simulated topology is shown in figure 1. Applications are distributed over hosts S1–Sn and send to hosts R1–Rm, where n and m are chosen such that drops occur only at the output interface of B1. The AC entity controls access to the bottleneck link between routers B1 and B2. The bottleneck link has a capacity of 10 Mbps and a propagation delay of 20 ms. We simulate only PMM traffic and thus the full 10 Mbps are reserved for PMM. All other links have a capacity of 100 Mbps. The propagation delay of the links between B2 and hosts R1–Rm is either set as 70 ms for all links in the RTT_{equal} scenarios or set as 40 ms (B1–R1), 70 ms (B2–R2), and 100 ms (B2–R3) in the RTT_{var} scenarios. This $\{40, 70, 100\}$ cycle is repeated from R4–Rm. The RTT_{var} setup enables the study of competing flows under different RTTs.

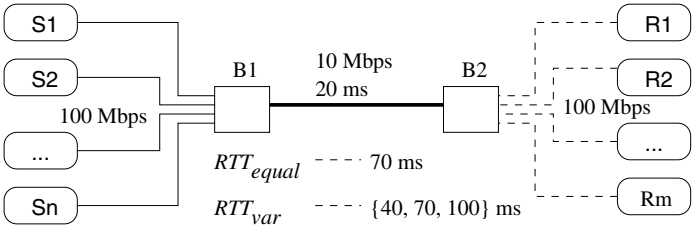


Fig. 1. Simulation topology

We use the `ns-2` built-in FTP application to simulate greedy, TCP based bulk-data transfers. The application lifetime is uniformly distributed between 50 and 150 seconds. While this is arguably not a choice matching real-world observations, we are initially not interested in very long transfers. The goal of the study is to first get an understanding what parameter configuration is useless / useful for the operation of TCL 3. Once this large parameter space can be reduced, more detailed studies can be performed and we argue that a configuration that optimizes transfers of FTP flows lasting for 150 seconds will also be beneficial for longer flow durations.

We have seen in initial simulations that the higher the ratio of the highest and smallest requested rate, the more difficult it is to provide the requested sending rates. This is a consequence of the weak estimation of certain parameters like p_2 or T. In fact, flows with a smaller requested rate tend to "steal" bandwidth from flows with a higher rate. To study this effect, we define a requested rate factor rF as $rF := \frac{R_{max}}{R_{min}}$ and include rF into the list of parameters that are varied between simulation runs.

We put an additional restriction on the rates that can be requested by allowing only a finite set of rates within the range $[R_{min} \dots R_{max}]$. The difference between R_{i+1} and R_i must be equal to the requestable rate distance rD. Like rF, rD is also varied over the simulation scenarios. In total, the requestable rates are the ones shown in table 2.

Table 2. Requestable rates

rF = 3	
$rD = 100$:	$R \in \{200, 300, 400, 500, 600\}$
$rD = 200$:	$R \in \{200, 400, 600\}$
$rD = 300$:	$R \in \{200, 500\}$
rF = 5	
$rD = 100$:	$R \in \{200, 300, 400, 500, 600, 700, 800, 900, 1000\}$
$rD = 200$:	$R \in \{200, 400, 600, 800, 1000\}$
$rD = 300$:	$R \in \{200, 500, 800, 1100\}$
rF = 8	
$rD = 200$:	$R \in \{200, 400, 600, 800, 1000, 1200, 1400, 1600\}$
$rD = 300$:	$R \in \{200, 500, 800, 1100, 1400\}$
$rD = 400$:	$R \in \{200, 600, 1000, 1400\}$

Resource reservation requests are generated according to an exponentially distributed inter-arrival time with a mean on 2 seconds. This results in a high-load scenario where at almost any time all resources are allocated to flows and the probability that a new request has to be denied is high. Although such a request blocking probability may be unrealistically high in an operative network, it is a worst case scenario where flows cannot consume unallocated bandwidth and thus more easily reach the required sending rate.

We introduce the term *traffic template* which represents exactly one possible combination of requested rate R and RTT T. When a new rresourcerequest is generated, one such traffic template is randomly (uniform) chosen and a request for the template's requested rate R is sent to the AC entity.

Besides choosing T according to the RTT_{equal} and RTT_{var} scenarios, respectively, we also vary the error in RTT estimation, called T_{dev}, as {0%, 25%, 50%, 75%, 100%}. This enables us to study the influence of a wrong RTT estimation. A T_{dev} of 0% means that T is set as a value that closely matches the RTT of the simulation scenario (propagation delays plus transmission delays plus average WRED queueing delay). If T_{dev} is larger than zero, the real RTT is around $T * (1 + T_{dev})$, i.e. T underestimates the real RTT. Underestimation is the more difficult case as this overestimates the TCP sending rate.

Concerning the WRED model, we investigate two approaches for setting the number of flows N. In one case, called N_{high}, we set $N = \frac{\rho C}{R_{min}}$. This estimation assumes that all requests are for the minimum rate R_{min} and thus N_{high} is generally an overestimation. In the other case, called N_{low}, we set $N = \frac{\rho C}{R_{avg}}$, where $R_{avg} = \frac{R_{min}+R_{max}}{2}$. Due to an unfairness on the request level – smaller

requests have a higher probability of being accepted – N_{low} is thus generally an underestimation.

Finally, we use values of $\{0.7, 0.8, 0.9\}$ for the ρ parameter of the AC formula (eq. 2).

Some parameters have been fixed for all simulations. The packet size is set to 1500 bytes. The WRED model is configured with $U = 0.5$. For the TBM model, the p_2 value of 0.1 is configured. This choice is the result of a prior study not shown here.

Table 3. Summary: variation of input parameters

ρ of admission control	$\in \{0.7, 0.8, 0.9\}$
number of flows for the WRED model	$\in \{N_{low}, N_{high}\}$
link delays in topology	$\in \{RTT_{equal}, RTT_{var}\}$
error in RTT estimation T_{dev}	$\in \{0\%, 25\%, 50\%, 75\%, 100\%\}$
requested rate factor rF	$\in \{3, 5, 8\}$
requested rate distance rD	$\in \{100, 200, 300[, 400]\}$

The input parameter space is summarized in table 3. In order to exhaustively explore this input space we simulate all 540 possible combinations of input parameters. Each simulation is run for 50000 simulated seconds. As the request blocking probability increases with larger requested rates, this long simulation time is needed to get enough results for traffic templates which request a rate of R_{max}.

8 Simulation Results

The primary QoS goal of the PMM class is to achieve a sending rate that is at least as high as the requested rate. We therefore define a *success* value s_t for each traffic template t in the following way:

$$s_t := \frac{card\{flow \in t : rate \geq R\}}{card\{flow \in t\}} \tag{4}$$

A whole simulation run is characterized by one *success* parameter S, where $S = \min(s_t)$. Interestingly, the resulting success values vary between 0% (i.e. there is no traffic template where all flows reach at least R) and 100% (i.e. all flows in all templates reach at least R).

To evaluate the optimality of the different parameter configurations we represent each simulation with a 7-dimensional vector: one dimension for the success S of the simulation and 6 dimensions due to the input parameters (see table 3).

In order to gain insight on how to ideally configure the PMM service we select those simulation vectors with a success $S \geq 0.99$. Such high success values can only be obtained if the input parameters are chosen among the ones shown in table 4.

Table 4. Input parameters resulting in high success S

ρ of admission control	$\in \{0.7\}$
number of flows for the WRED model	$\in \{N_{high}\}$
link delays in topology	$\in \{RTT_{equal}, RTT_{var}\}$
error in RTT estimation T_{dev}	$\in \{0\%, 25\%, 50\%, 75\%, 100\%\}$
requested rate factor rF	$\in \{3\}$
requested rate distance rD	$\in \{100, 200, 300\}$

We subsequently discuss the results and thereby look in detail at the influence of each parameter involved. The following paragraphs provide guidelines on how to optimally configure the PMM service class.

The choice of $\rho = 0.7$ provides enough safety margin (unallocated capacity) to compensate for the imperfections of the PMM traffic control combination. Among these imperfections are the impossibility to perfectly estimate changing parameters by a single static value and deviations between analytical models and the traffic under control.

Concerning the WRED model, better results can be achieved if the number of flows is set as N_{high}. The convergence behavior of the average queue size is indeed according to the objective of the WRED model. This justifies the correctness and usability of the WRED model even in an environment where the input parameters are not exactly known.

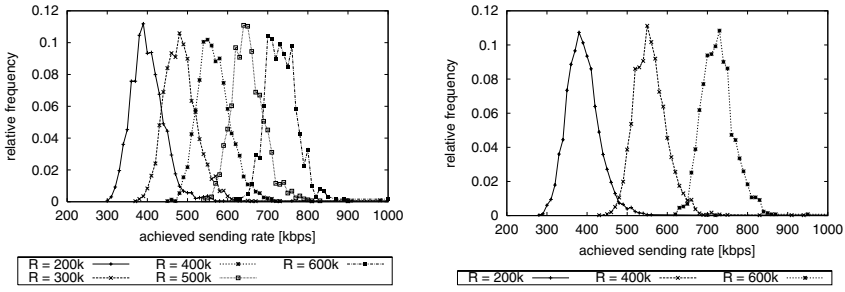
Another positive outcome of the simulation study is the result that under the restricted choice of ρ, rF, and rD as shown in table 4, differing RTTs (RTT_{var}) as well as wrongly estimated RTTs ($T_{dev} > 0$) do not present a major problem. This is a convenient property as finding a good estimate for the RTT is a difficult task.

Providing requestable rates over a broad range seems an impossible objective with the design as investigated in this study. In fact, good results can only be achieved if rF is not larger than 3. The distance rD between requestable rates has no impact on the success S.

For the further analysis the focus is on simulation scenarios where success $S \geq 0.99$. We fix the input parameters to $\rho = 0.7$, N_{high}, $rF = 3$, $T_{dev} = 50\%$ and take a detailed look at the distribution of sending rates for the RTT_{equal} / RTT_{var} scenarios and different values of rD. The sending rates are classified into bins of 10 kbps. Figure 2 shows the relative frequency of sending rates for each requestable rate in the RTT_{equal}, $rD = 100$ scenario. The requestable rates are listed in table 2.

As can be seen in figure 2, each flow achieves at least the requested sending rate, i.e. the success S of both simulations is 100%.

As far as the service differentiation within the PMM class is concerned, the scenario with $rD = 100$ (subfigure 2(a)) shows a suboptimal behavior. The user is offered a high number of requestable rates. However, the resulting service curves are significantly overlapping and the service offerings are thus not clearly distinguishable.



(a) $rD = 100$ (b) $rD = 200$

Fig. 2. Relative frequency of achieved sending rates in the RTT_{equal} scenario

In the $rD = 200$ case (subfigure 2(b)) the service curves are hardly overlapping. This provides for a clear distinction of the service delivered for different requests. The service distinction is even more pronounced in the $rD = 300$ scenario where only two rates (200 kbps, 500 kbps) are requestable.

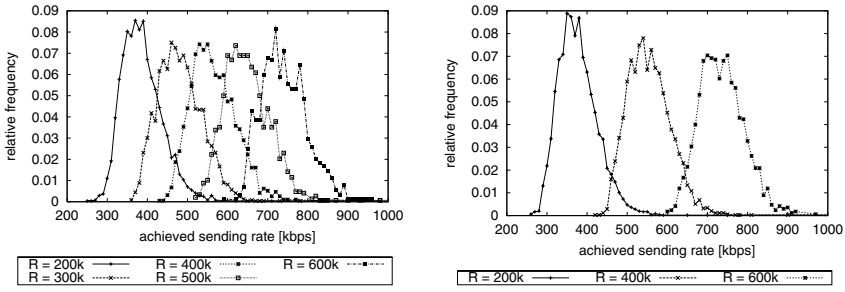
Looking at the shape of the service curves it makes sense to offer only a discrete set of requestable rates instead of a continuous range between $[R_{min} \ldots R_{max}]$. By offering a finite set of rates the operator can tune the service differentiation within the PMM class. This approves the usefulness of the discrete rates approach. A choice of $rD = 200$ results in a reasonable trade-off between the number of requestable rates and a clear service distinction.

In the RTT_{var} scenario, there are flows with different RTTs within each requested rate. Compared to the RTT_{equal} scenario, the resulting sending rates fluctuate more and the curves in figure 3 are thus slightly broader and lower. For $rD = 200$ the service differentiation within the PMM class is still well pronounced.

9 Conclusion

In this paper we report on an attempt to establish a QoS class for long-lived, bulk-data TCP flows that require a minimum rate from the network. The approach is based on a model for TCP flows subject to token bucket marking at the network edge-device and preferential dropping in the core network. This model is combined with an admission control functionality and a model for the parameterization of multi-RED queue management. The goal of the additional components is to enforce conditions under which the sending rate of the flows can be regulated through the token bucket marking profile.

The difficulty of finding a proper parameter set for the various input parameters of the service class is discussed. In a large simulation study a broad



(a) $rD = 100$ (b) $rD = 200$

Fig. 3. Relative frequency of achieved sending rates in the RTT_{var} scenario

spectrum of the input parameter space is explored in order to identify inter-parameter dependencies and discriminate between useless / useful service class configurations.

From these results guidelines on how to configure the PMM class are derived. While the models derived for admission control and queue management can be generally applied in the context of long-lived TCP flows and token bucket marking, the guidelines only apply to the PMM implementation studied in this paper.

Although ideal traffic handling is not feasible with a *static* marking profile the simulation results encourage the practicability of a real-world implementation. Despite simulations were run at a very high service class utilization (and thus an unrealistically high service request blocking probability) a set of configuration parameters that enables a successful operation could be identified. The QoS objectives can be more easily reached under a lower service utilization where more unallocated resources are available.

Initial testbed measurements have been performed but they were heavily influenced by the maximum queue size that could be configured in the router when the full AQUILA scheduling approach is employed (a combination of Priority Queueing and WFQ). This restriction did not allow the use of the WRED model as discussed in section 5 and consequently led to throughput degradations. We plan to repeat the measurements with the PMM class only as this allows the use of a FIFO scheduler where our equipment does not have the mentioned buffer size restrictions.

References

1. Adaptive Resource Control for QoS Using an IP-based Layered Architecture (AQUILA), IST-1999-10077, `http://www.ist-aquila.org/`.
2. Nichols, K., Blake, S., Baker, F., Black, D.: RFC 2474: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers (1998)

3. Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., Weiss, W.: RFC 2475: An Architecture for Differentiated Services (1998)
4. Sahu, S., Nain, P., Towsley, D., Diot, C., Firoiu, V.: On Achievable Service Differentiation with Tocken Bucket Marking for TCP. In: Proc. of the ACM SIGMETRICS'2000 Int. Conf. on Measurement Modeling of Computer Systems, Santa Clara, CA, USA. (2000)
5. Narvaez, P., Siu, K.Y.: An Acknowledgment Bucket Scheme for Regulating TCP Flow over ATM. In: Proceedings of IEEE Globecom. (1997)
6. Koike, A.: TCP flow control with ACR information (1997) ATM Forum/97-09989.
7. Kalampoukas, L., Varma, A., Ramakrishnan, K.K.: Explicit Window Adaptation: A Method to Enhance TCP Performance. In: Proceedings of INFOCOM '98. (1998)
8. Satyavolu, R., Duvedi, K., Kalyanaraman, S.: Explicit rate control of TCP applications (1998) ATM Forum Doc. Number 98-0152R1.
9. (Packeteer) `http://www.packeteer.com/` (Feb. 2003).
10. Karandikar, S., Kalyanaraman, S., Bagal, P., Packer, B.: TCP rate control. ACM Computer Communications Review (2000)
11. Elizondo-Armengol, A.: TCP-Friendly Policy Functions: Capped Leaky Buckets. In: Seventeenth International Teletraffic Congress (ITC17). (2001)
12. Heinanen, J., Guerin, R.: RFC 2698: A Two Rate Three Color Marker (1999)
13. Fang, W., Seddigh, N., Nandy, B.: RFC 2859: A Time Sliding Window Three Colour Marker (TSWTCM) (2000)
14. Lin, W., Zheng, R., Hou, J.C.: How to Make Assured Service More Assured. In: ICNP. (1999) 182+
15. Seddigh, N., Nandy, B., Pieda, P.: Bandwidth Assurance Issues for TCP Flows in a Differentiated Services Network (1999)
16. Makkar, R., Lambadaris, I., Salim, J., Seddigh, N., Nandy, B.: Empirical Study of Buffer Management Scheme for Diffserv Assured Forwarding PHB. In: Proceedings Ninth International Conferance on Computer Communications and Networks, Las Vegas, Nevada. (2000)
17. Azeem, F., Rao, A., Kalyanaraman, S.: A TCP-friendly traffic marker for IP differentiated services (2000)
18. Feng, W., Kandlur, D., Saha, D., Shin, K.: Adaptive Packet Marking for Maintaining End-to-End Throughput in a Differentiated Services Internet. In: IEEE/ACM Transactions on Networking. Volume 7. (1999) 685–697
19. Nandy, B., Seddigh, N., Pieda, P., Ethridge, J.: Intelligent Traffic Conditioners for Assured Forwarding Based Differentiated Services Networks. In: IFIP High Performance Networking (HPN 2000). (2000)
20. El-Gendy, M.A., Shin, K.G.: Equation-Based Packet Marking for Assured Forwarding Services. Infocom (2002)
21. Chait, Y., Hollot, C., Misra, V., Towsley, D., Zhang, H.: Providing throughput differentiation for TCP flows using adaptive two color marking and multi-level AQM. In: Proceedings of Infocom 2002. (2002)
22. Dorfinger, P., Brandauer, C., Hofmann, U.: A rate controller for long-lived TCP flows. In: Joint International Workshop on Interactive Distributed Multimedia Systems and Protocols for Multimedia Systems (IDMS/PROMS). (2002) 154–165
23. Padhye, J., Firoiu, V., Towsley, D., Kurose, J.: Modeling TCP Throughput: A Simple Model and its Empirical Validation. In: ACM SIGCOMM'98. (1998)
24. Padhye, J., Firoiu, V., Towsley, D., Krusoe, J.: Modeling TCP throughput: A simple model and its empirical validation. Proceedings of the ACM SIGCOMM '98 conference on Applications, technologies, architectures, and protocols for computer communication (1998) 303–314

25. Seddigh, N., Nandy, B., Pieda, P., Salim, J.H., Chapman, A.: An Experimental Study of Assured Services in a Diffserv IP QoS Network (1998)
26. Floyd, S., Jacobson, V.: Random Early Detection Gateways for Congestion Avoidance. IEEE/ACM Transactions on Networking **1** (1993) 397–413
27. Floyd, S.: The RED Web Page (1997) `http://www.aciri.org/floyd/red.html`.
28. Ziegler, T., Brandauer, C., Fdida, S.: A quantitive Model for Parameter Setting of RED with TCP traffic. In: Proceedings of the Ninth International Workshop on Quality of Service (IWQoS), 2001, Karlsruhe, Germany. (2001)
29. Brandauer, C.: Web interface to the RED model developed in [28] (2000) `http://www.salzburgresearch.at/~cbrand/REDmodel`.
30. Firoiu, V., Borden, M.: A study of active queue management for congestion control. In: Proceedings of IEEE Infocom, Tel Aviv 2000 IEEE Computer and Communications Societies Conference on Computer Communications (INFOCOM-00). (2000) 1435–1444
31. Clark, D., Fang, W.: Explicit allocation of best effort packet delivery service. IEEE/ACM Transactions on Networking **6** (1998) 362–373
32. Salsano, S., et al: Traffic handling studies (2003)
33. Brandauer, C.: Web interface to the WRED model (2001) `http://www.salzburgresearch.at/~cbrand/WREDmodel`.
34. Network Simulator ns-2, see `http://www.isi.edu/nsnam/ns/`.

Evaluation of the AQUILA Architecture: Trial Results for Signalling Performance, Network Services and User Acceptance*

Marek Dąbrowski[1], Gerald Eichler[2], Monika Fudała[1], Dietmar Katzengruber[3], Tero Kilkanen[4], Natalia Miettinen[4], Halina Tarasiuk[1], and Michael Titze[3]

[1] Warsaw University of Technology, Institute of Telecommunications, PL-00-665 Warsaw
{mdabrow5,mkrol,halina}@tele.pw.edu.pl
[2] T-Systems, Technologiezentrum, D-64307 Darmstadt
Gerald.Eichler@t-systems.com
[3] Telekom Austria AG, A-1030 Vienna
{Dietmar.Katzengruber,Michael.Titze}@telekom.at
[4] ELISA Communications, FIN-00381 Helsinki
{tero.kilkanen,natalia.miettinen}@elisa.fi

Abstract. A set of five practically manageable Network Services is proposed for the IP QoS AQUILA architecture and implemented in a prototype network. QoS capabilities of routers in conjunction with Admission Control (AC) and Resource Pool (RP) allow for an efficient handling of traffic requiring different packet transfer characteristics. The paper presents the trial results illustrating the effectiveness of the AQUILA network for providing a well-differentiated Network Services set. Performed trials cover both, technical parameters such as per Network Service behaviour, signalling performance and user acceptance referring to application behaviour using the entire AQUILA architecture.

1 Introduction

Both, business and private end-users are looking for reliable and provable network applications with focus on cheap and easy accessible service offers. From the point of a network operator or Internet service provider, Quality of Service (QoS) is a business opportunity. An important prerequisite for QoS offers towards the customer is a technique for precise specification of Network Services and their support at network level using Traffic Classes (TC).

Within the AQUILA project [2], a modular Resource Control Layer (RCL) is defined to cover both, intra- and inter-domain issues. A Resource Pool (RP) approach accompanied with appropriate Admission Control (AC) mechanisms guarantees scalability. Multiple trials were carried out to evaluate the architecture and its parts.

The paper is organised as follows. In chapter 2, AQUILA architecture and Network Services are described. Traffic handling mechanisms at packet and flow levels are presented in chapter 3. The obtained trial results are discussed in chapter 4. The results correspond to Resource Control performance and evaluation of AQUILA

* This work is partially funded by the European Union under contract number IST-1999-10077 "AQUILA".

W. Burakowski et al. (Eds.): Art-QoS 2003, LNCS 2698, pp. 218–233, 2003.

Network Services. Chapter 5 contains a summary of the paper and main trial achievements.

2 AQUILA Architecture and Network Services

This chapter provides an overview of QoS IP architecture developed by AQUILA project with special focus on offered Network Services.

2.1 Architecture for Intra- and Inter-domain

AQUILA architecture covers intra- and inter-domain parts as depicted on Fig. 1. Intra-domain part relies on the Resource Control Layer (RCL) that acts as distributed bandwidth broker, controlling and providing the resources of the underlying DiffServ network. RCL contains three main components, that are:

- the *Resource Control Agent (RCA)* which is responsible for the control and management of overall resources of the domain;
- the *Admission Control Agent (ACA)* manages the local resources of one edge or border router. An ACA communicates with other ACAs for allocating the resources.
- the *End-user Application Toolkit (EAT)* is a kind of middleware between end-user application and the network. EAT requests appropriate network resources for setting the connection.

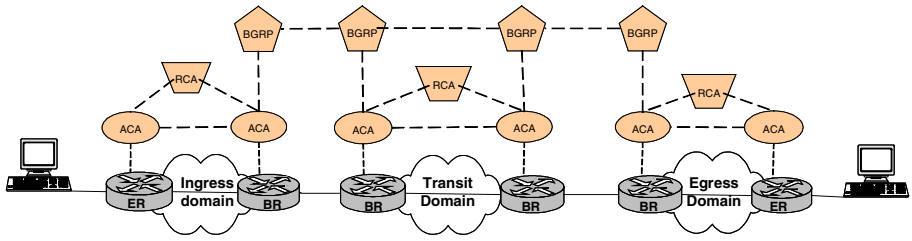


Fig. 1. AQUILA intra- and inter-domain Resource Control Layer architectures with associated message flows. H : Host, ER : Edge Router, BR : Border Router, BGRP : Border Gateway Reservation Protocol

The objective of ACA is to control the volume of traffic injected into the network and in this way to avoid network congestion. This approach is necessary for providing QoS guarantees in the network. Additionally, overall resources of the domain are represented in the form of resource pools which is suitable for effective management by RCA [2]. The resource pools is a mechanism for achieving dynamic resource allocation in a domain.

Since, the initial distribution of resources is based on assumed provisioning rules, those resources may be again more effectively redistributed based on observations of network traffic load. The exemplary interactions between the RPs entities of the tree are depicted on Fig. 2.

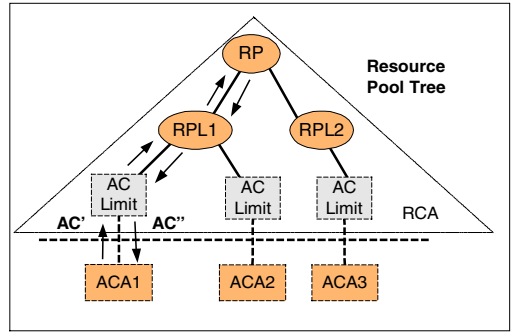


Fig. 2. Interactions between ACA and RCA

When a new resource request can not be admitted because the current assigned AC limit is fully allocated for running connections, the ACA will request (AC') additional resources from the RCA. If the RP does not have enough resources to accommodate the request, then additional resources are requested from the RP of the level above. Each RP runs the same algorithm, which is executed whenever resources should be redistributed. The actual amount of assigned resources (AC'') from a RP is based on its spare resources. The request for additional resources may be propagated up to the root of the tree, as depicted on Fig. 2. Moreover, in case resources are not used by an ACA, they are released to the upper level RP.

In inter-domain architecture, the enhanced BGRP (Border Gateway Reservation Protocol) protocol [12, 2] for making reservations across borders is applied. Fig. 1 depicts a network consists of source, transit and destination domains. For such a scenario, the associated RCL components for intra-domain resource management, BGRP agents for inter-domain resource control as well as the interactions between the components are shown.

2.2 Network Services

The AQUILA project [2] defined four manageable premium transport options (beside best effort) for IP traffic, as listed in Table 1. They are named Network Services. The Network Service characteristics could be defined by the network operator. The idea of these services is to provide a few specific offerings to the customer, which are easy to understand, suitable for specific groups of applications, and maintainable in large networks [4].

Fig. 3 describes the relations between the different entities and the role played by the Network Services. The operator of a DiffServ aware network needs a formalism in order to express technically what can be provided to its customers. The AQUILA consortium defined a generic Service Level Specification (SLS), capturing all the possible service offerings that can be provided over a DiffServ network.

Table 1. Network Services as defined within the AQUILA framework

Service	Goals/Focus
PCBR: Premium Constant Bit Rate	designed to serve constant bit rate traffic e.g. voice trunks and virtual leased lines
PVBR: Premium Variable Bit Rate	designed to provide effective transfer of streaming variable bit rate traffic e.g. video-conference and video
PMM: Premium Multi-Media	designed to support TCP (or TCP like) applications of greedy type e.g. ftp or adaptive non-real time streaming video, that require some minimum bandwidth
PMC: Premium Mission Critical	designed to support TCP (or TCP like) applications of non-greedy type e.g. online games or home-banking
STD: Standard	designed to carry best effort traffic

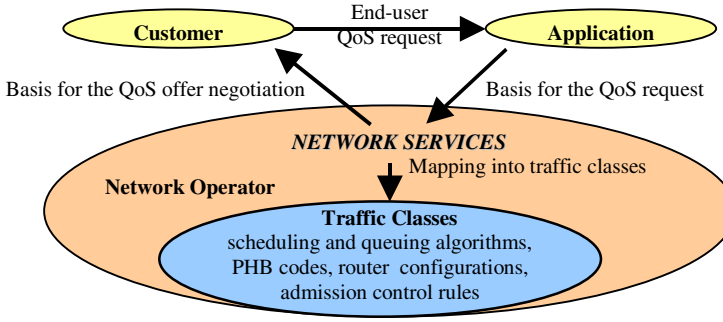


Fig. 3. Network Services as intermediate between application's/user's request and Traffic Classes

3 Traffic Handling Mechanisms at Packet and Flow Levels

3.1 Mechanisms at Packet Level

To achieve QoS prerequisites at network level, the key issues are appropriate router configurations beside appropriate network design (network topology or link dimensioning) and traffic engineering (routing, network load assessment) [1]. The proposed solution aims to produce reliable differentiated network QoS, known as Traffic Classes (TC).

At the router ingress classification, marking/colouring and policing/profiling are performed. Interface selection and queue selection lead to a realisation of the handling defined by the TC, where queue dimensioning and drop policies influence the packet handling. AQUILA uses five TCs, mirroring a face to face mapping of the five Network Services. While the PCBR, PVBR and STD TCs are tail dropped, PMM and

PMC make use of Weighted Random Early Detection (WRED). Scheduling among multiple queues influences the traffic merging. The TC for PCBR is prioritised, Priority Queuing (PQ) against the other four, which are handled with Weighted Fair Queuing (WFQ) with appropriate weights. Fig. 4 illustrates the packet flow through a routing entity. The decision made for the handling of selected packets is indicated by and stored in the precedence bits of the Type of Service (ToS) byte of the IP header.

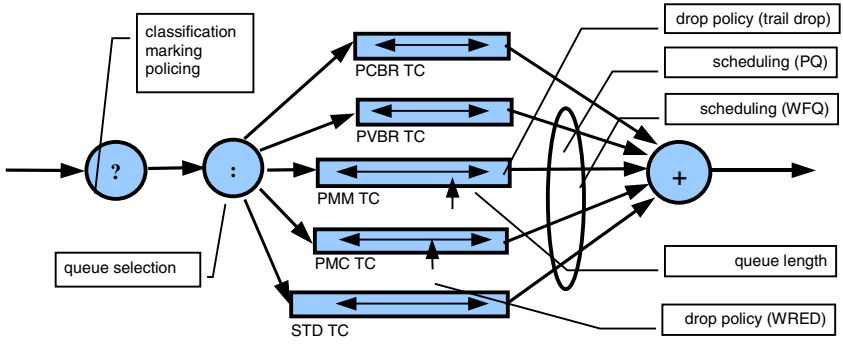


Fig. 4. Parameter setting points within a QoS sensitive router entity. After checking various selectors (?) the queue selection decision (:) is made. Packets of parallel queues have to be merged (+) before sending them onto the next link

3.2 Admission Control

Admission Control (AC) operates on the flow level and prevents the network against congestion by limiting the volume of submitted traffic. The decision of new flow acceptance/rejection is based on the flow traffic declaration and the current network load. New flow is only accepted when the QoS requirements for both the considered flow and the flows in progress are met. In AQUILA, adequate AC rules govern the volume of the traffic submitted to each TC. Different AC are implemented since the QoS objectives as well as the handled traffic profiles are different for each TC. For achieving maximum link utilisation profit, the bandwidth of the link is not strictly partitioned between TCs but could be dynamically allocated to each TC according to the demands (leading to full available link). For this purpose, additional rules (named joint AC) are implemented and situated on the top of AC associated to particular TCs [7].

3.2.1 Admission Control for PCBR and PVBR Traffic Classes

The PCBR and PVBR TCs are designed to carry streaming type of traffic. The QoS objectives provided by these TCs are defined in terms of low packet loss ratio as well as low packet delay and jitter. For achieving this, the REM (Rate Envelope Multiplexing) scheme is assumed with appropriate AC algorithms. A short buffer (for a few packets) is applied for absorbing packets arriving to the router output port from different sources at the same time.

The PCBR TC handles constant bit rate flows, so the users characterise the submitted traffic by the parameters of single token bucket mechanism: peak bit rate (PR) and bucket size. The DBAC (Declaration Based Admission Control) method is applied, i.e. decision on admitting or rejecting the new flow is taken only on the basis of the submitted traffic descriptors. There is assumed, that so called negligible jitter property [8] is valid, so the aggregation of a large number of CBR flows is modelled as a Poisson stream. Therefore, the maximum allowed utilisation in PCBR TC is calculated from the analysis of M/D/1/B system, taking into account the available buffer size and the target loss ratio. Details of the algorithm can be found in [1, 7].
The traffic submitted to the PVBR TC is of variable bit rate type. In the DBAC approach, additionally to the PR a user declares the value of sustained bit rate (SR), which usually is greater than the mean bit rate. Anyway, providing a priori the proper value for the SR parameter is rather difficult. Therefore, the MBAC (Measurement Based Admission Control) approach is also investigated in AQUILA. The applied Hoeffding bound algorithm (see details in [3, 11]) takes into account the measured mean bit rate of aggregate traffic in the PVBR TC, instead of user declarations of SR.

3.2.2 Admission Control for PMM and PMC Traffic Classes

The PMM TC was designed to provide throughput guarantees for TCP connections of greedy type. The guaranteed throughput per TCP connection should not be below the requested rate value. For the PMM TC two alternative AC algorithms are implemented. Each of them operates per TCP flow and is of declaration based type. They assume that a user, before establishing TCP connection, submits its request to the network. The traffic contract specifies the target requested bit rate (RR). Furthermore, on the basis of the RR and information about round trip time (RTT) of the TCP connection, the user declarations are mapped into the form of single token bucket parameters, rate (SR) and bucket size (BSS), constituting input parameters for the AC decision. The first of the two proposed AC is based on the token bucket marking (TBM); the second one enables an ideal TCP behaviour by setting an appropriate value for the advertised window size. The details of the algorithms are presented in [7, 9] and [3, 4].

The PMC TC is designed for handling non-greedy TCP traffic with no packet losses as QoS objective. The potential applications for using PMC are e.g. transaction oriented applications and www applications. In this case, a flow is characterised by parameters of dual token bucket mechanism, similarly as for PVBR TC. The proposed AC algorithm assumes that a demand is expressed in the form of the effective bandwidth value calculated for the RSM (Rate Sharing Multiplexing) scheme. The details of the AC rules for this TC are described in [1] and [7].

4 Trial Results

In this chapter the results from the trials are presented and discussed. They correspond to the RCL performance and evaluation of Network Services, including trials with real users. The trials were performed in the three trial sides Helsinki, Vienna and Warsaw.

4.1 Resource Control Performance

The aim of Resource Control performance measurements was to evaluate the reservation set-up and release times as well as the volume of signalling load in the inter-domain scenario. These parameters determine the scalability factor of AQUILA architecture.

4.1.1 Reservation Times

The reservation times were measured assuming network connections through three neighbouring domains (like in Fig. 1). No information about reservation transactions was recorded to the system log file to minimise additional delays. Twenty PCBR reservations were set-up and released. After one reservation was released, another reservation was immediately set-up. The timestamps were measured with a reservation generator which counted the time between sending the request to EAT and receiving the acknowledgement of the established reservation. The obtained results are presented in Table 2.

Table 2. Results of reservation processing times

Reservation	Set-up Time [s]		Release Time [s]	
	Average	Deviation	Average	Deviation
Initial	25.8	14.1	0.849	0.22
Subsequent	1.452	0.1	0.506	0.03

One can observe the significant difference between initial and subsequent reservation set-up. This difference was caused by the initial telnet connection to the router, requesting resources from RCA and first time initialisation of reservation related Java classes.

Initial reservations can be considered as a particular case and subsequent reservations as the standard case. Test results show that the average times for subsequent reservations were 1.45 seconds for reservation set-up and 0.506 seconds for reservation release. Additionally, the effect of requested network service on reservation set-up and release times was measured. The measurement was performed for both declaration and measurement based admission control schemes. The results show that either AC schemes or network services have no significant impact on reservation set-up and release times.

It was also measured if the number of the ongoing reservations has an impact to reservation set-up time. It was observed that the reservation set-up time did not change when the number of active reservations was increased.

4.1.2 Signalling Traffic

The amount of signalling traffic for reservation set-up was measured between RCL components. One reservation without existing sink-tree was made. In the intra-domain signalling the total amount of traffic was 64 kbytes for initial reservation and 50 kbytes for subsequent reservation. The largest component in both cases was router configuration (47% and 60 % respectively). Router contribution overhead is significant because of router telnet implementation inefficiencies.

For inter-domain reservations, additional signalling between BGRP agent and RCL components is necessary. When consecutive reservation joins the sink-tree, there is no additional signalling between BGRP agents.

4.2 Evaluation of Network Services

This section presents the measurement results illustrating the effectiveness of AQUILA Network Services, as introduced in chapter 2. The trials were mainly oriented on trial validation whether the assumed QoS objectives for particular Network Service were met under the allowed worst case traffic network conditions. All tests were carried out in intra-domain scenarios.

4.2.1 PCBR Service

The capabilities of PCBR service were measured assuming the test-bed network configuration as depicted on Fig. 5 with single bottleneck on the ingress link, 10 Mbps link connecting er1 and cr1. The packets of the foreground traffic submitted to PCBR service (emitted from PC1, with PC4 as the destination) were generated as Poisson stream, modelling a large number of CBR flows. The measured parameters were the PCBR packet loss ratio (P_{loss}) and packet delay characteristics. Furthermore, for getting more realistic traffic conditions in the network, traffic of lower priority services was also emitted (generated from PC2, with PC5 as the destination).

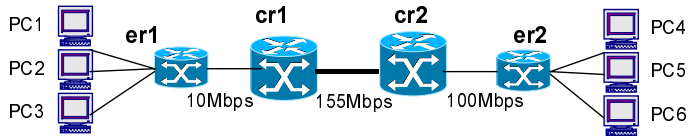


Fig. 5. Network configuration for testing PCBR and PVBR services. er : edge router, cr : core router

In the tests, the volume of bandwidth allocated for PCBR service, B_1, on the ingress link was changed from 1 to 10 Mbps. The mean bit rate of PCBR stream was equal to $B_1*\rho$, where $\rho = 0.58$ that corresponds to the buffer size of 5 packets and target $P_{loss} = 10^{-4}$ [1]. The traffic submitted as a background load to the STD service was also of Poisson type. The mean rate of this traffic was tuned to give the total offered load to the system equal to 120 % of the link capacity. In this way, the overload conditions on the link er1-cr1 were modelled.
The measured parameter values are shown in Table 3. One can observe that capabilities of PCBR service for all considered traffic scenarios are satisfied, even in the overload conditions on the bottleneck. The P_{loss} is below the target value 10^{-4} and the packet transfer characteristics are acceptable.

Table 3. PCBR service: the measured packet loss ratio (P_{loss}), one-way delay and delay variation (IPDV) vs. the bandwidth available for PCBR service

B_1 [Mbps]	PCBR traffic load [Mbps]	Lower priority background traffic load [Mbps]	P_{loss} of PCBR stream	Delay [ms]			IPDV [ms]	
				min	max	avg	avg	max
1	0.58	11.42	0	0.60	19.76	4.70	0.70	17.74
5	2.90	9.10	0	0.60	22.87	3.89	1.08	18.41
7	4.06	7.94	0	0.59	22.23	3.71	1.10	19.96
9	5.22	6.78	$4.5*10^{-5}$	0.59	24.59	3.6	1.09	22.26
10	5.80	6.20	$9.0*10^{-5}$	0.59	19.32	3.57	1.09	14.96

4.2.2 PVBR Service

Trial validation of PVBR service with MBAC AC was carried out similarly as for PCBR service, assuming topology from Fig. 5. As previously, the experiments were focused on measurements of P_{loss} and packet delay characteristics. The submitted PVBR traffic (generated from PC1, with PC4 as the destination) was modelled by a superposition of the maximum number of PVBR flows, N_{PVBR}, allowed by the AC. The QoS requirements for PVBR traffic are represented by packet loss ratio (P_{loss}) assumed at 10^{-4} level [7].

Table 4. PVBR service: the measured packet loss ratio (P_{loss}), one-way delay and delay variation (IPDV), vs. volume of PCBR and PVBR traffic

B_1 [Mbps]	PCBR traffic load [Mbps]	B_2 [Mbps]	N_{PVBR}	PVBR traffic load [Mbps]	P_{loss} of PVBR stream	Delay [ms]			IPDV [ms]	
						min	max	avg	avg	max
0	0	8.945	23	3.45	$1.26*10^{-4}$	2.84	17.50	3.95	0.49	12.99
0	0	4.238	7	1.05	$1.30*10^{-4}$	2.95	17.37	4.01	0.52	13.04
4	2.32	5.243	10	1.50	$1.36*10^{-4}$	2.87	23.34	4.16	0.70	19.13
4	2.32	2.658	3	0.45	$1.00*10^{-4}$	2.23	14.88	4.15	0.63	11.44
7	4.06	2.658	3	0.45	$1.18*10^{-4}$	2.43	21.70	4.41	1.00	17.86

The flows were of ON-OFF type with the following parameters: peak bit rate 0.5 Mbps and mean bit rate 0.15 Mbps. The volume of bandwidth B_2 allocated for PVBR service was changing from 2.658 to 8.945 Mbps. The remaining bandwidth was allocated for PCBR service (B_1), according to the joint AC rules [7]. The background traffic submitted to PCBR service (generated from PC2, with PC5 as the destination) was of Poisson type with mean bit rate equal to $B_1*\rho$, $\rho=0.58$. In addition, constant bit rate traffic was submitted to STD service (generated from PC3, with PC6 as destination). The rate of this traffic was tuned to produce permanent congestion conditions on the bottleneck 10 Mbps link.

The measured parameters associated to PVBR traffic service are collected in Table 4. They say that the measured P_{loss} are close to assumed target value (10^{-4}) as well as the packet delays are acceptable. This allows us to conclude that the impact of higher priority PCBR traffic on PVBR service is effectively regulated by the applied AC rules.

4.2.3 PMM Service

The PMM service was designed to guarantee the TCP throughput, which should not be below the requested rate. Since two alternative AC methods for PMM has been implemented, two groups of trials were performed: (1) for AC based on TBM and (2) for AC based on advertised window setting. The measured parameter was the TCP throughput. The obtained results were compared with the declared requested rate values.

The assumed test-bed topology for PMM is depicted on Fig. 6. This topology consists of 2 CISCO edge routers connected by 2 Mbps link (bottleneck link). The PC stations 1/2/3/4 are connected to the er1 router while PC 5/6 and PC 8 to the er2. The PC stations from 1 to 4 play role of TCP senders while the PC stations from 5 to 8 are the TCP receivers. In this configuration the maximum number of running TCP connections was 4.

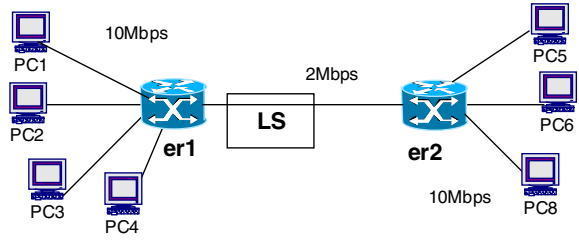


Fig. 6. Trial topology for PMM and PMC services. er: edge router, LS: Link Simulator

In the investigated network scenario, an additional transmission delay on the bottleneck link by the Link Simulator (LS) was also introduced. In this way, the effectiveness of the PMM service for more realistic RTT values was verified. For all tests, the minimum round trip time RTT^{min} was 108 ms, including additional one-way transmission delay equal to 50 ms on 2 Mbps link. The MTU was 1500 bytes and TCP MSS (Maximum Segment Size) was set to 1448 bytes.

The foreground traffic was produced by the number of consecutive TCP connections generated by single TCP greedy source. Particular TCP flow started after the previous one was finished (the consecutive TCP flows started each 8 minutes). A volume of data generated by TCP source corresponding to single flow was fixed to 10 Mbytes. The background traffic was generated by 3 parallel running greedy TCP connections. The number of these connections (4 connections), including foreground connection, was admitted according to the joint AC rules (in this case no additional flow could be admitted by ACA). The bandwidth B_3 allocated for PMM was equal to 2 Mbps. In addition, for AC algorithm based on TBM, parameter ρ_{PMM} was equal to 0.75 and T=202 ms, according to the recommendation from [9].

For both proposed AC algorithms two test cases were performed: (1) assuming homogenous TCP connections with the same requested rate values, and (2) assuming heterogeneous TCP connections differing in the requested rates.

For homogenous TCP connections test, two investigated AC approaches met the expectations and they guaranteed that the measured TCP throughput was above the requested rate. For the AC based on TBM the difference between the measured TCP throughput and the requested rate was hard to predict and depended on the number of running TCP connections. One could observe that in some cases this difference was significant. For the AC based on advertised window setting the measured TCP throughput, according to the expectations, was between the requested rate and the sustained rate, but rather closed to the requested rate. The reason that the measured TCP throughput was greater than the requested rate was mainly due to the error resulting from the assumed analytical approximation of average RTT, see [9]. The obtained results for the discussed case can be found in [6].

Table 5. Throughput characteristics for AC based on TBM. The parameters are: RR : Requested Rate, SR/BSS : token bucket parameters, Throughput : measured TCP throughput with a confidence interval of 95 %

TCP connections	#1/#2	#3/#4
SR (kbps)	40	392
BSS (bytes)	60000	60000
RR (kbps)	250	500
Throughput (kbps)	385 ± 110	473 ± 16

Table 6. Throughput characteristics for AC based on advertised window setting. Parameters description like Table 5 and W^{adv} : advertised window size

TCP connections	#1/#2	#3/#4
SR (kbps)	328	672
W^{adv}(bytes)	4274	8688
BSS (bytes)	4283	8463
RR (kbps)	232	521.7
Throughput (kbps)	275 ± 2	567.6 ± 2.5

For heterogeneous TCP connections test the AC algorithm based on TBM did not meet the expectations (see Table 5). In some cases the measured TCP throughput was below the requested rate. In addition, one can observe that by using this algorithm the TCP connections shared available bandwidth rather to the fair share than according to the requested rates. This was observed especially for more than 3 admitted connections.

For heterogeneous TCP connections test, the AC algorithm based on advertised window setting met the expectations (see Table 6). Similarly to the homogenous case, again the measured TCP throughput was between the requested rate and the sustained rate, but rather closed to the requested rate.

Summarising, the results referring to the TCP throughput say that for the case with homogenous connections both considered AC approaches work properly, for different number of admitted connections. However, this conclusion can not be extended to the case with heterogeneous TCP connections, where only the AC based on advertised window setting meets requirements. The main reason that the AC based on TBM failed in this case was that the assumed maximum buffer size (25 packets) was shorter than required from theoretical studies, see [9]. This was caused by the limitation of the routers used in trial (maximum buffer size for PQWFQ scheduler was only 64 packets for all network services).

4.2.4 PMC Service

The PMC service was designed to guarantee very low packet losses and low delay for non-greedy traffic usually controlled by TCP protocol. The trial was performed assuming that PMC service was separated from other network services. During the trial the packet loss ratio was measured. By assuring low packet loss ratio one can expect the low transaction delay by avoiding packet retransmission.

The test-bed topology is depicted on Fig. 6. In this case, no additional transmission delay by the Link Simulator (LS) was introduced. The PMC foreground and background traffic was sent between terminals PC2-PC6 and PC1-PC5, respectively. Since PMC requires relatively large room, the almost whole router output port buffer space was dedicated for PMC services (60 packets). Moreover, the buffer management mechanism WRED was applied with parameters fixed according to [7].

Two test cases were taken into account: (1) homogenous case, when all submitted TCP connections have the same characteristics and (2) heterogeneous case, when TCP connections have different characteristic.

In the test the packet loss rate was measured after 100 measurement cycles. Each measurement cycle begun with simultaneous starting up of a given number of TCP flows and ended after completing all transfers. During single TCP connection a predefined amount of data was transferred corresponding to a typical size of www pages. The number of simultaneous running connections was determined by defined AC algorithm for PMC service [7]. The test was performed under the minimum possible RTT value with negligible propagation delay. This condition constitutes the worst case for the PMC traffic.

The obtained results for heterogeneous case are shown in Table 7. In the presented case, two different types of flows were simultaneously submitted into the system. The number of admitted flows (4 flows) was determined by available bandwidth B_4 equals 2 Mbps. More test cases for PMC service can be found in [6].

Table 7. PMC service: packet loss ratio (P_{loss}) characteristics for heterogeneous case. PR, SR, BSS: dual token bucket parameters

TCP connections	#1/#2	#3/#4
Amount of transferred data per flow [bytes]	36200	73848
PR [Mbps]	10	10
BSS [bytes]	15000	30000
SR [kbps]	340	170
P_{loss}	0	0

Taking into account the obtained results for both homogenous and heterogeneous cases one can conclude that PMC service is able to guarantee low packet losses (in the performed tests no losses were observed). Moreover the AC algorithm designed for PMC service properly determines the maximum number of admitted flows.

4.3 Real User Trial for PCBR Service

In this section, the results of the real user listening-opinion trial with VoIP application as well as trial with using the NetQual software are presented.

4.3.1 Listening-Opinion Trial

The listening-opinion trial aims at assessing QoS perceived by real users and expressed by their subjective opinion. This assessment was done by measurements of logatom (non-sense words) articulation, which gives statistical information about voice transfer quality. In other words, the probability of successful speech transfer on the basis of the perceived phonetic speech elements was calculated.

The calculated parameters were:

$$W_{n,k} = \frac{P_{n,k}}{T_k} * 100\,[\%] \quad \text{(a)} \qquad W_L = \frac{1}{N * K} \sum_{n=1}^{N} \sum_{k=1}^{K} W_{n,k}\ [\%] \quad \text{(b)} \qquad (1)$$

where: $W_{n,k}$: logatom articulation measured during listening logatoms from k-th test list by n-th listener, $P_{n,k}$: the number of correctly received logatoms from k-th test list by n-th listener, T_k : the number of read logatoms from k-th test list, W_L : average logatom articulation, N: the listener number, K : the number of read test list;

$$s = [\frac{1}{N * K - 1} \sum_{n=1}^{N} \sum_{k=1}^{K} (W_{n,k} - W_L)^2]^{1/2} \qquad (2)$$

where S is the mean square deviation, which is used for calculating of logatom articulation dispersion.

The real user test-bed topology in Warsaw is depicted on Fig. 5. Foreground VoIP connection was established between PC1 and PC4, while background traffic was generated between PC2 – PC5 and PC3 – PC6.

The trial was repeated under different traffic conditions. In the scenario #1 (reference scenario) only single VoIP connection (tested connection) was established in the network. For the scenario #2, both tested VoIP connection as well as background traffic were handled by AQUILA Network Services (including STD). In this case foreground traffic (VoIP flow) was generated into PCBR, while background traffic to both PCBR (Poisson stream with mean rate 5.136Mbps) and to STD (Poisson stream with mean rate 6.8 Mbps) services. As a consequence the total offered traffic to the access link (between er1 and cr1 routers) was equal to 1.2 times link capacity and produced overload condition. Finally, in the scenario #3, comparing to the scenario #2, tested VoIP traffic was served by STD.

Trial procedure was the following. Five listeners and speaker, who tested VoIP application, were situated in acoustic separate rooms. The speaker was reading the prepared logatom lists, while listeners wrote down the received logatoms. The voice

quality was estimated on the basis of the probability of correctly received logatoms. Before starting the experiment, the listeners passed the training with the speaker, by listening to the selected logatom lists. Then for the scenario #1, #2 and #3 listeners were listening to three logatom lists (100 logatoms each). Furthermore, for each scenario the logatom articulation ($W_{n,k}$) was calculated according to the formula (3a). Finally, average logatom articulation (W_L) and mean square deviation (S) were counted according to the formulas (3b) and (4). In addition, on the basis of W_L, the MOS (Mean Opinion Score) index was evaluated, in approximate way, according to the conversion rate given by the Polish standard, see Table 8.

Table 8. Average logatom articulation (W_L) and mean square deviation (S) calculated under different traffic conditions

Trial scenarios	Average logatom articulation (W_L)	Mean square deviation (S)	MOS
Scenario #1: reference	74.1 %	7.1 %	4.0
Scenario #2: VoIP using PCBR	71.9 %	9.8 %	3.8
Scenario #3: VoIP using STD	46.1 %	9.6 %	1.9

On the basis of the obtained results one can conclude as follows. Measured W_L for both scenarios #1 and #2 was similar and on acceptable level for IP network (for a telephone network, with 64 kbps voice channel – MOS is 4.4 , with 16 kbps voice channel – MOS is 4.2). Results obtained in the scenario #3 were much worse comparing to the scenario #2 and evaluated quality was on unacceptable level.

In the Vienna test-bed similar intra-domain trials with German logatoms were performed, which also approved the presented results.

4.3.2 Trial with NetQual

In order to compare the real user trial measurements, which resulted by the perceived speech quality of the users further tests were performed using NetQual [10]. NetQual system enables the execution of sample wave files, which are recommended by the ETSI and used for MOS verification tests.

Therefore, a sample wave files was injected into the network on one side and recorded on the other side. Then the reference sample (reference, indicated by the white line) file and the recorded (coded, represented by the black line) file were compared and analysed by NetQual to get statistical values. Fig. 7 shows a scenario with a 100 % loaded network and a sample wave file injected in STD whereas Fig. 8 represents the same loaded network using PCBR for the sample wave file. The x-axes shows the time in seconds whereas the y-axes illustrates the signal level in dB. In Fig. 8 the similarity of the signals hardly exists, which indicates that the reference signal was highly distorted during the transmission. As a consequence, every voice conversation is impossible. On the other hand, in Fig. 8 both signals are nearly similar, which represents good quality and a MOS value of about 3.

Concluding, the obtained results in both trial sides confirm the expectations that VoIP needs a prioritised service in IP networks. The PCBR service in AQUILA supports VoIP in a very good way even in extremely congested traffic conditions.

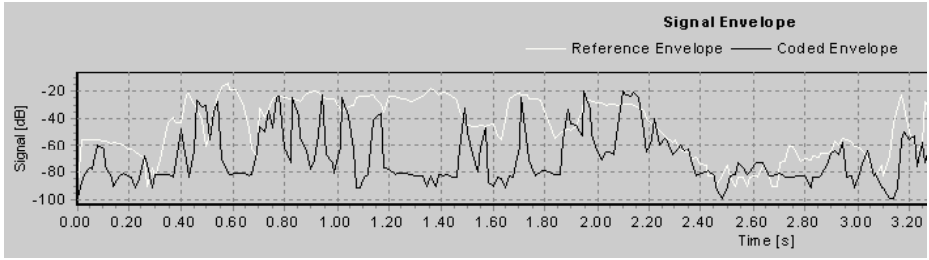


Fig. 7. Signal envelopes for STD service and 100 % background traffic

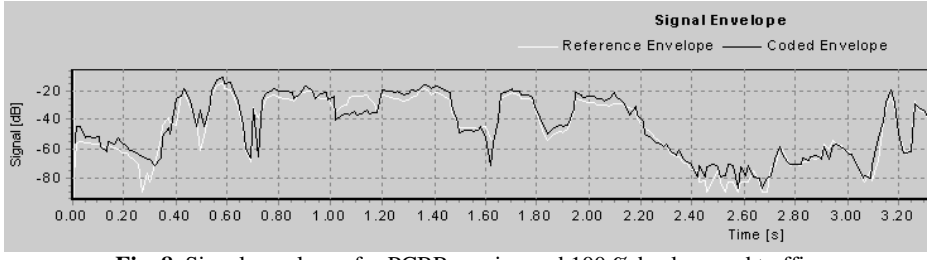


Fig. 8. Signal envelopes for PCBR service and 100 % background traffic

5 Summary

A pilot implementation of the AQUILA solution has been evaluated during project trials. The main objectives of the trials were evaluation of the proposed Network Services for intra- and inter-domain scenarios, real user trials for different applications (voice, video and audio streaming, interactive games) as well as the trials of Resource Control performance. In this paper only selected part of the trials is presented.

The focus was set on Resource Control performance for inter-domain network topology and Network Services evaluation for intra-domain (single domain) including real user trials for the PCBR service. Main achievements of the presented trials are the following:

1. The trial confirmed that the assumed QoS objectives for PCBR, PVBR and PMC are met. It means that Admission Control mechanisms for these Network Services fulfil their role according to the expectations. *Four premium Network Services cover a wide range of applications.*
2. The results for PMM service were presented for two alternative AC algorithms. Only one of them based on advertised window setting met the QoS objectives for homogenous and heterogeneous case. The second one based on TBM worked properly for homogenous case only. *There is a strong need for AC to produce QoS on a DiffServ aware network.*
3. The RCL performance tests show that set-up and release times for subsequent reservations are acceptable. It was also observed that AC scheme, traffic class and number of ongoing reservations have no significant impact on reservation processing times. *QoS add-ons should be manageable, scalable and well performing.*

4. The presented results from real user trials for PCBR service show that QoS perceived by real users expressed by their subjective opinion is on acceptable level taking into account obtained logatom articulation and MOS index values. *Beside all technical support user acceptance is the main focus which has to be supported by understandable operator offers.*

Detailed description of all trials performed during second phase of the AQUILA project one can find in [6].

References

1. Bak, A., Burakowski, W., Ricciato, F., Salsano, S., Tarasiuk, H.: A Framework for Providing Differentiated QoS Guarantees in IP-based Network. Computer Communications 26 (2003) 327–337
2. Engel, T. et al.: AQUILA Adaptive Resource Control for QoS Using an IP-based Layered Architecture. IEEE Communications Magazine, Vol. 41 No. 1, January 2003
3. Brandauer, C., Burakowski, W., Dąbrowski, M., Koch, B., Tarasiuk, H.: AC algorithms in Aquila QoS IP network. 2nd Polish-German Teletraffic Symposium PGTS 2002, Gdańsk, Poland, September 2002. (Accepted for publication at the European Transactions on Telecommunications, 2003)
4. Eichler, G., Thomas, A., Widera, R.: Proven IP Network Services: From End-User to Router and vice versa. Innovative Internet Computing Systems, IICS 2002, Springer LNCS Vol. 2346, 2002
5. Brandauer, C., Dorfinger, P.: An implementation of a service class providing assured TCP rates within the AQUILA framework. Workshop on Architectures for Quality of Service in the Internet jointly held with the Final AQUILA Seminar Art-QoS 2003, Warsaw, Poland, 2003.Will be published in LNCS series
6. Kopertowski, Z. et al.: Second Trial Report. AQUILA deliverable D3202, Warsaw, February 2003
7. Ricciato, F., Salsano, S. (ed.): Specification of traffic handling for the second trial. AQUILA deliverable D1302, Rome, September 2001
8. Roberts, J., Mocci, U., Virtamo, J. (ed.): Broadband network teletraffic: Performance evaluation and design of broadband multiservice networks. Final Report COST 242, Lectures Notes in Computer Science 1155, Springer, 1996
9. Salsano, S. (ed.): Traffic handling studies. AQUILA deliverable D1303, Rome, November 2002
10. SwissQual AG – NetQual, URL: http://www.swissqual.com/html/netqualpage.htm
11. Tran-Gia, P., Vicari, N. (ed.): Impact of New Services on the Architecture and Performance of Broadband Networks. Final Report COST 257, compuTEAM, Wuerzburg, 2000
12. Winter, M. (ed.): Final System Specification. AQUILA deliverable D1203, Munich, April 2002

CSPF Routed and Traffic-Driven Construction of LSP Hierarchies

Michael Menth, Andreas Reifert, and Jens Milbrandt

Department of Distributed Systems, Institute of Computer Science,
University of Würzburg, Am Hubland, 97074 Würzburg, Germany
{menth,reifert,milbrandt}@informatik.uni-wuerzburg.de

Abstract. The objective of this work is the analysis of reservation aggregation and the description of a network architecture for scalable Quality of Service support. This architecture applies Differentiated Services for packet forwarding, admission control is done on a per flow basis at the access, and resource allocation is based on reservations. The reservations of individual flows are aggregated recursively to achieve scalability in the core. Multiprotocol Label Switching is applied to reflect these aggregates in Label Switched Paths (LSPs). The construction of the LSP hierarchy is traffic-driven and based on explicit routes that are determined by Constraint Shortest Path First routing. We describe the architecture of that system and suggest several mechanisms to operate it also in networking scenarios with heavy signaling load.

Keywords: QoS, resource allocation, admission control, CSPF, MPLS, LSP hierarchy

1 Introduction

One of the challenges in future data communication networks is the provisioning of toll quality data transport for real-time applications, i.e. Quality of Service (QoS) for the traffic in terms of loss and delay bounds for transported packets must be met.

For this purpose, the Internet Engineering Task Force (IETF) proposed the Integrated Services (IntServ) approach [1,2] for IP networks. This, however does not scale in large networks with respect to reservation handling, packet classification and scheduling in presence of a large number of flow reservations in a router. In contrast to that, the Differentiated Services (DiffServ) [3] allows only for a few relatively differentiated transport service classes which makes this paradigm highly scalable. However, it does not provide reservations and absolute QoS guarantees. Multiprotocol Label Switching (MPLS) [4] is a promising technology that offers various means for Traffic Engineering (TE). When aggregated flows are tunneled in a Label Switched Path (LSP), their reservations can be subsumed to a single one along that path and their packets are unified under a common MPLS label for classification and scheduling. This achieves scalability in the Label Switching Routers (LSRs) [5]. Furthermore, MPLS allows for the

W. Burakowski et al. (Eds.): Art-QoS 2003, LNCS 2698, pp. 234–251, 2003.

integration of Constraint Shortest Path First (CSPF) routing leading to a better utilization of network resources [6].

In this work, we investigate a network architecture that combines principles from IntServ, DiffServ, and MPLS. It offers good real-time QoS support and it is scalable in large networks. We suggest protocol actions to build LSP hierarchies in a traffic-driven and distributed manner. We propose mechanisms that protect the network from signaling overload not only in extreme networking scenarios [7].

This paper is structured as follows. Section 2 gives a short introduction to important aspects of IntServ, DiffServ, MPLS, and CSPF. In Section 3, we describe our network architecture that is based on these concepts and explain the required protocol actions. We introduce further enhancements to the basic structure to achieve signaling scalabiltiy even in extreme networking scenarios. Section 4 illustrates the performance of this approach with respect to resource utilization and signaling stability. Section 5 concludes this work and gives an outlook on further activities.

2 Concepts for QoS Support in IP Networks

The architecture under study comprises many known concepts of today's methods for QoS support in IP networks. Therefore, we briefly describe their most important aspects in this section. The IETF has suggested two main alternatives to enhance IP networks with real-time capabilities. These are the IntServ and the DiffServ approach. Recently, MPLS has been defined to facilitate the TE process, e.g. data tunneling and route pinning. The latter feature may be used in combination with constraint based routing.

2.1 Integrated Services

IntServ is characterized by the separate handling of each individual end-to-end (e2e) micro flow. The Resource Reservation Protocol (RSVP) [8] is used to establish an e2e flow path and reservation states with knowledge about this flow in all routers along the path of this flow. These states contain among other information a flow specification that comprises the *Tspec* and *Rspec* parameters to indicate the expected data rate and the desired QoS for the reservation. They are used to manage the capacity on every outgoing interface and to enforce policing on a per flow basis. In particular, Admission Control (AC) uses these data to decide whether an additional flow can be admitted. A separate queue and a scheduling state are maintained for each flow to meet the required QoS objectives. This, however, is clearly a difficult task for routers as soon as the number of flows is in the order of a few ten thousands which can be easily reached in backbone networks. Hence, IntServ does not scale in large networks and can not be applied. Therefore, reservation aggregation [9] has been suggested to overcome this drawback.

2.2 Differentiated Services

The DiffServ approach allows only for a few traffic classes. The Differentiated Services Code Point (DSCP) in the IP header is used to mark the different Per Hop Behaviors (PHBs) that tell the routers to treat the corresponding IP packet with low or high priority in the forwarding process.

No per flow information is stored and, as a consequence, this architecture scales well for large networks because the forwarding process operates on aggregated traffic and not on single micro flows. Policers and shapers at the network edges try to control the traffic volume entering the network. But simple traffic conditioning impairs the transport QoS of all flows with the same DSCPs in the same way since the approach lacks AC. It can not support high QoS for some flows at the expense of the rejection of others.

A so-called bandwidth broker solves that problem by introducing AC on a per flow basis at the network edges [10]. The packet classification and scheduling inside the network is still done according to the DSCPs. The bandwidth broker needs to know all flows and their routes in the network to avoid congestion on the links. Hence, AC is done in an almost central manner and faces similar scalability issues like IntServ [11]. Distributed and hierarchically structured bandwidth brokers try to mitigate that effect [12,13,14]. The remaining key feature of DiffServ is that the packet classification and scheduling relies only on the DSCP in the packet headers and keeps the forwarding engine simple.

2.3 Multiprotocol Label Switching

MPLS is a mechanism to allow packet switching instead of routing over any network layer protocol [4]. The ingress LSR of a LSP equips an IP packet with a label of 4 bytes and sends it to the next LSR. The LSRs classify a packet according to its incoming interface and to its label. Based on this information and the Incoming Label Map (ILM), label swapping is performed and the packet is forwarded to the particular outgoing interface. The egress LSR only removes the label from the IP packet header. In practice, modern routers are capable to process both IP and MPLS packets. Hence, the label swapping process requires entries for every LSP in the Management Information Base (MIB) of the LSRs, so there is again a state per session like in IntServ.

There are two major alternative protocols for establishing a LSP. RSVP with Tunneling Extensions (RSVP-TE) is a modification of RSVP [15] and is able to distribute the labels. The Constraint Based Label Distribution Protocol (CR-LDP) [16] has been designed particularly for that goal, though the IETF seems now to go along with RSVP-TE. A LSP may be established and associated with bandwidth reservations, e.g. using the primitives of RSVP. Thus, the LSP represents then a virtual link that borrows its resources from the links connecting its LSRs. The more general Label Distribution Protocol (LDP) is not able to make reservations [17].

The label distribution and the label switching paradigm allows for explicit route pinning which facilitates fast rerouting and load balancing. Furthermore,

packets from different flows can be tunneled through a LSP. The label makes the aggregation visible in the LSRs and eases the mapping of a packet to a specific aggregate. It also bypasses control messages that are related to the individual flows at the LSRs.

MPLS implements the connection concept. Therefore, it is often viewed as modified version of the Asynchronous Transfer Mode (ATM) with variable cell size though there is a profound difference: ATM enables a two-fold aggregation with its virtual connection and virtual path concept while MPLS allows for many-fold aggregation using multiple label stacking, i.e. a LSP may be transported over other LSPs. This feature helps to build scalable network structures, so-called LSP hierarchies [18,19,20,21].

2.4 Constraint Shortest Path First Routing

Open Shortest Path First (OSPF) is the common routing algorithm in Autonomous System (AS) networks constituting the Internet. The routing in the Internet has two major drawbacks. All packets with the same destination are routed along the same path. This can lead to overloaded and poorly utilized links at the same time. Packets that require real-time transportation need a path where enough resources are available to avoid extensive waiting times and packet loss in the router internal queues. But the IP routing mechanism is unaware of the free link capacities. In contrast, Constraint Shortest Path First Routing (CSPF) takes the free resources of the links into account and finds the shortest path through a network whose links offer a desired QoS - if there is any such path available. This route may differ from the shortest path that is taken by OSPF. LSPs may be used to provide tunnels along these routes to bypass conventional IP routing. The blocking probability for data flows that require stringent QoS can be reduced this way when default paths are highly loaded.

3 Reservation Aggregation

The hierarchical partition of networks into access and core implicates that the number of flows increases towards the core. This is a problem for per flow reservations because the large number of flows in the core is not manageable in the routers. Flow aggregation towards the core achieves scalability for flow classification and resource reservation. We briefly characterize what we understand in general by aggregation and deaggregation, describe the tunnel and funnel concept, and explain how an appropriate signaling can be achieved with existing protocols. Finally, we also present another kind of aggregation that is implemented in a distributed bandwidth broker.

3.1 Aggregation in General

Aggregation. Several flows with the same or similar requirements are summarized along a common subpath of their routes to a single flow from the viewpoint of a

router. This reduces the flow information quantity which is stored in the routers along the common subpath. Aggregation can be applied recursively.

Deaggregation. When a flow aggregation terminates, the individual flows become visible again in the deaggregating router. The attributes of the original flows (e.g. final destination) are restored. The aggregation method influences how much of the original information can be recovered.

Aggregation Aspects. In DiffServ, packets with the same demand for QoS are marked with the same DSCP. This aggregates many streams with respect to queuing and scheduling. Routing in the Internet is also performed on traffic aggregates: only a subnet mask is decisive for the routing decision. All the information needed for packet scheduling and routing is recorded in the packet header.

The flowspecs (e.g. peak rate) of reservations are needed for the resource management of the links. Their information is not related to a single packet but to the whole flow and it is stored in the routers. Hence, when reservations are aggregated, the flow related information in the routers must also be aggregated. If it is needed again after deaggregation, it must be preserved. Therefore, reservation aggregation is more difficult than aggregation for scheduling and routing.

There are basically two ways how reservations can be aggregated. We explain these fundamental concepts using MPLS terminology. Aggregation using MPLS means that two different flows are equipped with the same label and are forwarded in the same manner. In principle, there are two alternatives to accomplish this: tunnels and funnels.

3.2 Tunnel Aggregation

We talk about tunneling flows if the uppermost label in their packet headers remains in place and a new common label is put onto the label stack. This new label corresponds to a new connection in the aggregated context (cf. Figure 1), i.e. a new aggregating LSP is set up. The individual reservations of the contained flows are (naively spoken) summed up to compute the size of the aggregate reservation. The flows are transported over the LSP and when the egress router of that LSP removes the uppermost label, the original flows are restored. In particular, their connection context and reservation information is present in the deaggregating router.

The LSP acts as a logical link and the intermediate LSRs do not see the individual reservations because their control messages are bypassed as MPLS packets at the interior LSRs of the LSP. Hence, tunnels reduce the state information in the intermediate routers, they allow for reservation aggregation and deaggregation because the original flows are recovered. MPLS tunnels are applied in [22] whereas other tunnels are applied for reservation aggregation in [9] and [13].

We sketch out how tunnels in MPLS are set up using RSVP-TE. The ingress LSR issues a PATH message with the resource demand that is forwarded hop by

hop to the egress LSR. This pass is used to install a path state in every participating router to indicate the previous hop, to store flow related information, and to make label requests to downstream next hops. In addition, the PATH message contains information about the available capacity on the already traversed route so that the demand can be adapted if there is a shortage of resources. The egress LSR triggers a RESV message back to the ingress LSR that distributes the MPLS labels upstream and establishes the reservation for the LSP by setting up a reservation state. As RSVP-TE is only an extension of RSVP, the path and reservation states are soft, i.e. they are refreshed by periodically sent PATH and RESV messages.

The tunnel concept scales poorly in some network topologies. When full connectivity in a star-shaped network is to be realized, the center node has to handle all possible tunnels which amounts to exactly $N \cdot (N-1)$ LSPs. Thus, the number of aggregates scales quadratically with the network size. Although mostly not all LSPs are stored in the MIB of a single router, this is still not a good scaling behavior.

3.3 Funnel Aggregation

We say that LSP flows are merged into a new aggregate if the uppermost label in their packet headers is substituted by a new common label. Figure 1 visualizes the resulting sink tree towards a common destination and motivates the name funnel for this kind of aggregation. In contrast to tunnel aggregation, no new connection is created to carry the aggregated context but the aggregate information of the new aggregate is associated with the merged flow in the downstream LSP context. The information about the individual flows is lost and can not be recovered at the end of the sink tree.

To achieve full connectivity in a network with N nodes, every node needs to hold $N-1$ LSPs since every other router can then be reached by equipping the packets with the corresponding label for the destination machine. This means that the number of paths scales linearly with the network size.

LSP Multipoint-to-Point Trees. In MPLS it is possible to construct multipoint-to-point forwarding trees that have a common sink as destination of the transported traffic. So far, the LDP has been developed for label distribution in MPLS but no reservations can be set up with LDP. CR-LDP is able to set up reservations but it explicitly declares the construction of multipoint-to-point LSP for further study. In RSVP-TE, there are different filter styles. The fixed filter (FF) style only allows for point-to-point LSPs. With the wildcard filter (WF) reservation style, a single shared reservation is used for all senders to a session. The total reservation on a link remains the same regardless of the number of senders. This reduces the amount of reservation information at the egress on the one side but on the other side, the size of the aggregated reservation can not be adapted to the number of sending sources. Hence, this is not a scheme for reservation aggregation. The shared explicit (SE) style allows a receiver to explicitly specify the senders to be included in a reservation. There is a single reservation on a

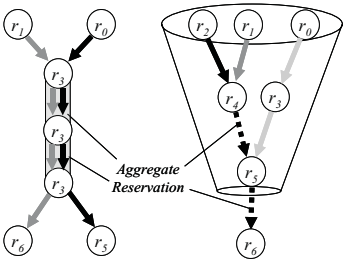


Fig. 1. Tunnel aggregation in contrast to funnel aggregation

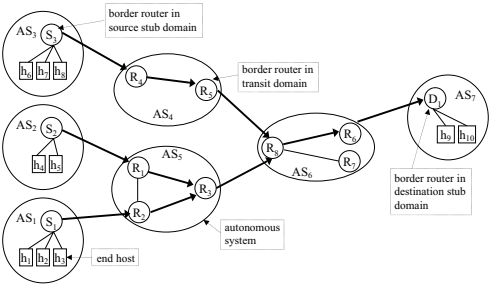


Fig. 2. Signaling in BGRP

link for all the listed senders and this option may be used to support sink tree reservations provided that the Explicit Route Objects (EROs) of the different sessions are the same. But the bandwidth computation of the merged reservations is not appropriate: "When SE-style reservations are merged, the resulting filterspec is the union of the original filterspecs, and the resulting flowspec is the largest flowspec [8]. For sink tree reservation aggregation we rather need the sum and not the maximum of the individual reservation sizes. Hence, there is no label distribution protocol that supports sink tree reservations for the purpose of reservation aggregation.

Signaling of Sink Tree Reservation Aggregation with BGRP. The Border Gateway Reservation Protocol (BGRP) [23] has been conceived for inter-domain use and to work in cooperation with the Border Gateway Protocol (BGP) for routing. It is used for reservations between border routers only. BGRP aggregates all inter-domain reservations with the same destination AS gateway into a single funnel reservation, no matter of their origin.

Figure 2 shows the setup of sink tree reservations in BGRP. A PROBE message is sent from a source border router S_1 to the destination border router D_1. It is processed at the intermediate border routers, collects them as well as information about available transit capacities in their AS. Unlike in RSVP, no "PATH" state is established, i.e. stateless probing is done. A sink tree is uniquely identified by the IP address of the destination AS and an ID for the destination border router D_1. The destination border router returns a GRAFT message together with the tuple (IP address of destination AS, ID of border router) along the reversed path collected in the PROBE message. The required reservation states are established or updated if they already exist.

So far, this is very similar to RSVP: one pass is used to collect path information in order to reserve an appropriate amount of resources on the way back. Hence, both protocols consist of a path information pass (PIP) and a resource reservation pass (RRP). There are also simpler protocols like Boomerang [24,25]

that combine the PIP and the RRP. They require only a single signaling pass but they need a second pass back to the sender to notify the successful establishment of the reservation.

The difference between BGRP and RSVP is that in BGRP the reservation request is expressed by a relative capacity offset because the issuing source can not know the resulting size of the aggregate reservation on the downstream links. Such an offset must be signaled exactly once and must arrive at the sink of the reservation tree. Therefore, BGRP requires reliable communication for signaling whereas RSVP messages are sent in unreliable datagrams. BGRP is also a soft state protocol but in contrast to RSVP, only neighboring routers exchange explicit REFRESH messages. They keep the reservation alive and interchange absolute reservation values.

3.4 Source Tree Flow Aggregation in Aquila

The project Aquila [12,26,27] implements a distributed and scalable bandwidth broker architecture which gains its scalability also from aggregation. The capacity of all links is controlled by a central Resource Control Agent (RCA) which distributes shares of link capacities to so-called Admission Control Agents (ACA). An ACA is a bandwidth broker associated with a single edge router and handles only local admission requests. The resource assignment from the RCA to the ACAs can be viewed as a reservation for aggregated flows starting from a common ingress border router to all egress border routers. In contrast to BGRP, these flows form a source tree instead of a sink tree. The reservations for all flows are basically known for all links in the network but the scalability comes from the fact that only the edge router knows about them. Therefore, the reservations do not need to implement RSVP states. Note that this kind of aggregation can not be applied to MPLS or RSVP-like concepts that rely on reservation states in the network. The Aquila architecture is flexible since the RCA and ACA are able to negotiate bandwidth and to make their interaction more scalable. The RCA may be implemented as a hierarchically structured and distributed entity.

4 Distributed and Traffic-Driven Setup of a LSP Hierarchy with Integration of CSPF

In this section, we describe how CSPF routed e2e reservation may be established within a LSP hierarchy using a traffic-driven LSP setup. We envision a hierarchically structured multi-service network that is capable for real-time transport. QoS is realized for flows in the network by e2e reservations, i.e. AC is performed for all links in the network (cf. Figure 3). For scalability reasons we apply reservation aggregation using MPLS technology, thus introducing additional virtual links for which AC also applies. Databases at the ingress routers store the available capacity of all links and LSPs in the network. This information is then used to compute for each reservation request an appropriate path through the network, i.e. the shortest possible path providing enough capacity to carry the

flow. These constraint-based routes increase the success probability of AC when the network is highly loaded. The IETF discusses a similar concept with a centralized solution: A Path Computation Server (PCS) computes routes based on a link database tracking the network-wide resource utilization [28,29].

In the following, we suggest mechanisms to implement a low degree of overreservation for reservation aggregates to reduce their signaling frequency. We propose that a protocol signals the available capacity of physical and virtual links to the link databases at the ingress routers. Based on this information, Explicit Routes (ER) are computed and enhanced by LSP hierarchy information. We explain modifications to RSVP(-TE) that are necessary for hierarchical LSP setup. Finally, we delimit this architecture against existing approaches.

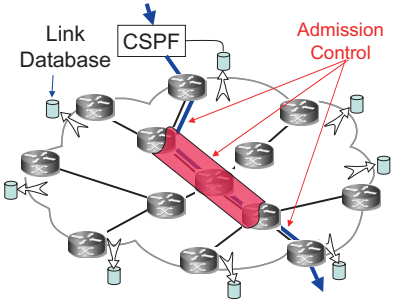


Fig. 3. Link databases allow CSPF computation and AC is performed in the network

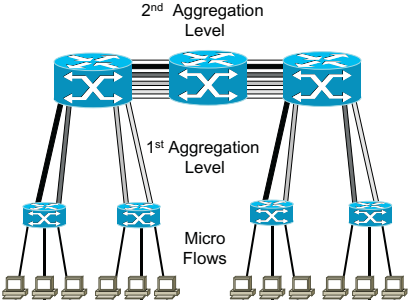


Fig. 4. Hierarchical tunnel aggregation between access routers

4.1 Resource Management and Hierarchical Aggregation

As outlined before, reservations are a possible means for resource management. If parts of the resources of a physical link have been dedicated to a LSP, they can not be reused until they are released. Reservation aggregation reduces the number of reservation states in intermediate routers. Tunnels and funnels may be used and they may also be applied recursively. In the following, we suggest the static structure of some basic aggregation options using MPLS technology because it also supports scalability for packet classification.

No Reservation Aggregation. If no reservation aggregation is performed, individual RSVP reservations initiate states along their paths through the network. This is the standard IntServ architecture.

Simple Tunnel Aggregation between Access Routers Using LSPs. LSPs may be used between access routers to tunnel e2e reservations through the core network.

The number of LSPs in the core is limited by the square of the number of access routers and is independent of the traffic load. If alternative paths are possible, this number can be exceeded because new LSP tunnels are required for additional alternative routes. As outlined before, LSP tunnels can be established by already existing signaling protocols.

Hierarchical Tunnel Aggregation between Access Routers. LSPs may be used between routers of the same depth, i.e. where paths start and end with a link between two different depths (cf. Figure 4). The depth of the LSP is the depth of its ingress and egress router. If possible, the LSP of depth d is carried over another LSP of depth $d-1$.

Simple Aggregation between Access Routers Using a Sink Tree LSP. LSPs may be used between access routers to tunnel e2e reservations through the core network. The LSPs themselves are arranged as a sink trees. The number of LSPs in the core is limited by the number of access routers and is independent of the traffic load. If alternative paths are possible, this number can be exceeded because for an additional alternative path a new LSP funnel may be required. As outlined before, LSP sink trees with reservations can not be established by already existing label distribution protocols. But enhancements to MPLS signaling protocols similar to BGRP would solve that problem.

4.2 Description of the PIP and RRP

Explicitly routed e2e reservations as well as LSPs are set up using a PIP and a RRP. They are also used to increase or decrease the reserved rate of reservations, therefore, we explain them briefly beforehand. The RSVP(-TE) sender triggers a PATH message that contains an ERO and follows the ER towards the destination. The PATH message carries a desired minimum and maximum capacity value, collects the available bandwidth on the path, and adjusts the flowspec ($Tspec$). The destination router returns an RESV message with an appropriate $Tspec$ parameter and the required resources are reserved on the way back, i.e. the used links dedicate $Tspec$ to the new reservation. When the RESV message is back at the sender, it is informed about the new $Tspec$ for future PATH messages that refresh the reservation states periodically. If the reservation failed because of lack of bandwidth, this is marked at the router to prevent another such reservation setup or the increase of reserved bandwidth during the next *NoResourceInterval* for that reservation. This failure should be propagated to the lower hierarchy levels until the e2e reservation is notified by an error message. Since the error message returns to the ingress border router, this failure information should be added to the local link database for the next *NoResourceInterval* in order to prevent the same requests in the near future.

4.3 Reduction of Signaling Frequency by Overreservation

Reservation aggregation increases the scalability in real-time networks by reducing the number of states in the router MIBs. If an aggregate reservation corresponds exactly to the sum of sizes of the aggregated reservations, it is updated whenever an RSVP-signaled flow starts, changes in rate, or ends. This change is also propagated to higher level LSPs if their aggregate reservations are also tight. Therefore, the amount of signaling is rather increased than reduced due to LSP capacity updates. Fortunately, it is possible to trade signaling frequency for bandwidth efficiency. When the bandwidth of a LSP is updated, its reservation is set to a larger value than the required sum of aggregated reservations in order to serve future requests from the residual bandwidth.

We define several attributes for physical links and LSPs. $Tspec$ is the amount of bandwidth that is assigned to a LSP by its reservation from the links along its path to itself. The "$Tspec$" of a physical link corresponds to its physical bandwidth which does not change. Only the link itself can dispose of that capacity. When a flow passes the AC of a link, some of the link capacity is dedicated to the reservation of the flow. The *AllocatedBandwidth* is the sum of all capacities of a (virtual) link that are already assigned to reservations. The *FreeBandwidth* of a link is the difference between its $Tspec$ and its *AllocatedBandwidth*. This is the capacity that may be used for serving new requests. For conservative AC without overbooking, $FreeBandwidth \geq 0$ is always true. The *AllocatedBandwidth* may be larger than the sum of individual e2e reservations (*UsedBandwidth*) that are indeed carried over the link. *UsedBandwidth* is a value that is in general not available in the router MIBs since it is intentionally concealed by aggregation and overreservation.

According to Figure 5, the LSP acquires more bandwidth from the links supporting it (if possible) when its *AllocatedBandwidth* does not suffice to serve a new request. With RSVP-TE this may be done using the SE style [15] and a new PIP. When the *AllocatedBandwidth* falls below a predefined *UpdateThreshold*, some of the $Tspec$ has to be redistributed to the links supporting the LSP. This is done by sending a RESV message with reduced capacity requirements. During signaling a decrease of LSP resources, the LSP is in an inconsistent state and should store arriving signaling packets in its Sleep Queue (SQ) which will be explained later. The overreservation dynamics have been studied in [7,30].

4.4 Signaling Available Bandwidth for LSPs

The *AvailableBandwidth* of a physical link is its $FreeBandwidth$ and the *AvailableBandwidth* of a LSP at a certain LSR is the minimum of the *AvailableBandwidth* of all links supporting this LSP from this LSR to the egress LSR. Hello messages [15] provide a means for rapid link and node failure detection and are exchanged every 5 ms. We propose to use either these HELLO messages or extra messages to exchange additional information about *AvailableBandwidth* of LSPs. This signaling pass is illustrated in Figure 6. The *AvailableBandwidth* of a LSP at a certain LSR is computed as the minimum of the *AvailableBandwidth* at the

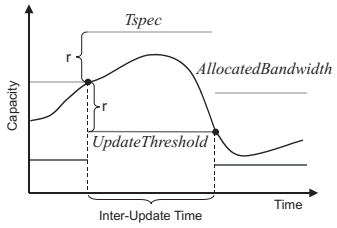


Fig. 5. Overreservation decreases the mean inter-update time of a LSP reservation

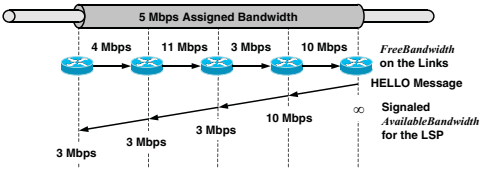


Fig. 6. The signaling of available bandwidth for a LSP using Hello messages

next hop LSR and the available bandwidth on the link from the LSR to the next hop LSR. With this enhancement, the LSPs have an approximated knowledge about the bandwidth that they can potentially allocate. Hence, the value of the *HelloInterval* influences the system accuracy.

4.5 Monitoring the *AvailableBandwidth* in the Link Databases at the Border Routers for CSPF

To enable CSPF computation at the border routers, the link database needs the available resources of the links and LSPs in the network. For the provision of the required information we suggest that every router in the network multicasts the *AvailableBandwidth* of all links that are under its control towards all border routers. The information quantity rises with the network size and, being more critical, the update frequency rises with the number of successfully admitted flows. Therefore, the simple approach, i.e.e where changed resource utilization is signaled to all local databases, can not scale for large network structures from the signaling point of view. Therefore, it is important that the update frequency is limited, i.e. not all changes are signaled to the databases. Therefore, the router should send the *AvailableBandwidth* information only once in a *DatabaseUpdateInterval*. Hence, the CSPF algorithm operates on obsolete information. As a consequence, the *DatabaseUpdateInterval* should be small but it must be sufficiently large to keep the amount of signaling traffic in the network low. A simple protocol like RTP should be used to perform that task.

4.6 Computation of an ER

When a new flow requests a reservation through a network, AC is performed for every link of the path. This process must succeed at all AC points, otherwise the desired reservation can not be set up. Based on its link database, the ingress border router computes, using CSPF, a constraint-based route that has enough capacity and that can serve as an ER for the reservation setup. The ER may

deviate from normal IP routing, therefore, it is convenient that LSPs are used for route pinning. The output of the CSPF algorithm is a constraint-based route consisting of physical and virtual links. This constraint-based route is resolved into a Hierarchical Explicit Route (HER) that contains the ER together with some information about the intended LSP hierarchy.

4.7 Hierarchical Explicit Routes

An ER is a predefined path and a HER is an ER that contains information about the intended LSP hierarchy concerning the ER. From the properties of a LSP hierarchy [5] one can conclude that the flow-specific LSP hierarchy can be marked by parentheses. The HER $(l_0, (l_1, (l_2, l_3), l_4), l_5)$ (cf. Figure 7) denotes that the flow is first transported over a simple physical link l_0, over a LSP along the links l_1, l_2, l_3, and l_4, and, finally, over physical link l_5. The used LSP consists of the physical link l_1, another LSP consisting of physical links l_2 and l_3, and the physical link l_4. This notation does not determine whether LSPs are tunnels or funnels. This may be configured in the routers if there is a suitable signaling protocol.

Hence, the ingress border router that determines the ER must also add some hierarchy information to tell the e2e reservation where a LSP should be used. The appropriate aggregation hierarchy depends on the used technology and is still a matter of research. The LSPs do not need to exist yet, they may also be constructed on demand which requires some changes to the signaling protocol RSVP(-TE). The advantage of this approach is that not all possibly required direct and detour LSPs need to be established in advance but only when needed. This reduces again the number of simultaneous states in the routers while keeping the system flexible. Based on the HER, the e2e reservation is set up.

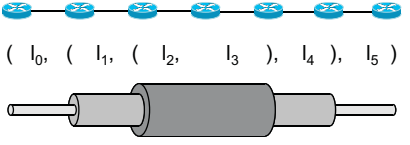


Fig. 7. The LSP hierarchy for the hierarchical explicit route $(l_0, (l_1, (l_2, l_3), l_4), l_5)$

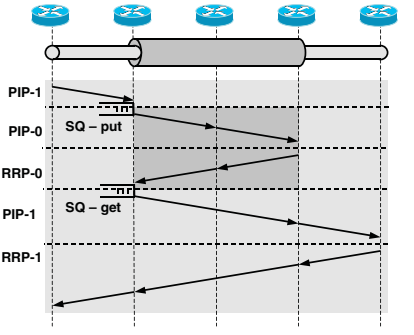


Fig. 8. The PIP is suspended for the construction of a higher level LSP

4.8 Reservation and LSP Setup

E2e reservations as well as LSPs are set up by initiating a PIP with an ERO and a flowspec as described before. Upon receipt of the PATH message, the receiver triggers the RRP. When the used links dedicate $Tspec$ to the new LSP reservation, they increase in turn their $AllocatedBandwidth$. If all required logical links (i.e. LSPs) in the ERO do already exist, the e2e reservation setup does not differ from normal RSVP-TE signaling.

The setup procedure becomes more complex when LSPs in the ERO do not exist yet. The PIP proceeds until a router recognizes that a desired LSP is not yet established. The router suspends the PIP of the ongoing e2e reservation or LSP setup because the path state can not be set up for the missing LSP. The construction of the missing LSP is triggered and the first action is the setup of a SQ that stores all signaling packets as long as the LSP is in an inconsistent state, i.e. until it is established. The PIP of the missing LSP is triggered with an adapted ERO by the router. This may further cascade if other LSPs are missing. When a PIP arrives at its destination, the RRP starts and reserves the resources on every link. If a logical link has not enough free resources anymore, the RESV message may be suspended and put into the SQ of the LSP and the capacity of the link is increased if possible. Note that this may cascade recursively. When the result returns, the RRP is resumed. If the increase succeeded, the RRP continues up to the initiator of the PIP and the e2e reservation or LSP is set up (cf. Figure 8). Otherwise, if the PIP or RRP have failed, an error is returned to the initiator of the PIP and the setup is rolled back. In both cases, all items in the SQ are processed and appropriate actions like forwarding of PATH messages or returning of error messages are performed.

4.9 Reservation Teardown

When a reservation is torn down, it gives its capacity back to its supporting logical links which decrease their $AllocatedBandwidth$. If an $AllocatedBandwidth$ falls below an $UpdateThreshold$, a decrease of the reserved bandwidth is triggered for resource efficiency. This action may cascade to higher levels in the hierarchy. In contrast to capacity increases, this is done asynchronously concerning the reservation teardown process. When a supporting link has finally no other reservations or LSPs to support and its SQ is also empty, it may wait for another $TeardownDelay$ time until it triggers its own teardown. This should be done to leave as few LSPs alive as possible. There are two possibilities how the LSP can realize that it is not needed anymore. If PATH states are used, there must be no PATH state for the LSP waiting for the establishment of a RESV state. If stateless PIPs are used (like PROBEs in BGRP), they should leave a timestamp at the ingress LSR. After a sufficiently long time one can assume that no corresponding RRP message will return and the empty LSP can be torn down.

4.10 Transport Resilience

In this work we have left out the resilience aspect. RSVP(-TE) has the capability of rerouting when network resources fail. This is not intended in the described system. The carried traffic has real-time requirements and, therefore, it is crucial that there are enough resources available on the rerouted path. Otherwise, AC fails and so does the rerouting approach. MPLS technology offers fast rerouting mechanisms for failure scenarios using precomputed backup LSPs with some shared backup capacity. Backup capacities may also rely on other resources where traffic may be preempted if necessary. This is favorable especially in muli-service transport networks. This is also subject for further research.

4.11 Differences to IntServ

At first sight, the presented network architecture is pretty close to IntServ because we have only considered the reservation process so far. But our approach is more scalable due to reservation aggregation. The aggregated reservations are supported by LSPs. Packets can thereby be classified in a scalable manner using only a few MPLS labels. Tracking of individual flows for single reservations is not anymore required in the core. Packets are policed and shaped only on an aggregate basis.

Another important difference to IntServ is that traffic handling works on an aggregate basis in the routers. As in DiffServ, packets are scheduled for transmission according to the DSCP in their IP packet header which can also be inherited by a LSP. Only traffic with the same DSCP value may be aggregated in a LSP, i.e. this attribute can be stored with the forwarding tables and the DSCP for labeled packets can be inferred. This allows for simple service differentiation while AC is maintained.

The traffic with less stringent QoS requirements may consume the bandwidth left over by the real-time flows. This is realized by appropriate scheduling mechanisms in the routers [31,32] that are not considered in this work.

4.12 Differences to Conventional DiffServ Bandwidth Brokers

The above proposed architecture has two different types of databases.

The routers store the traversing flows or flow aggregates in their MIBs. This information is required for local AC and resource management purposes. Every individual flow in the network is known at least in the ingress and egress border router. But inside the network they are concealed by aggregate tunnels. All these data are timely accurate because they must prevent overload on links such that real-time guarantees can be granted. Note that a hierarchical network structure is important for the scaling of this approach.

In contrast to DiffServ without bandwidth brokers, our architecture is able to support real-time guarantees. We explicitly state that our approach is different from a central DiffServ bandwidth broker because of the better scaling properties. A bandwidth broker in DiffServ stores all flows in the network in a more

or less central database. The amount of that information is proportional to the overall number of flows in the network which does not work in large networks for scalability reasons. In addition, flow policing can be done in the network based on LSP classification and not only according to DSCP aggregates.

The link databases have a global view on the *AvailableBandwidth* of all links and LSPs in the network and provide this information for CSPF computation at the ingress routers. Hence, the size of the link databases is proportional to the number of links in the network that are also advertised by routing protocols. The information quantity is independent of the number of transported flows. If these data are obsolete, the derived EROs might lead to increased reservation blocking probability inside the network. Although obsolete data might decrease the resource efficiency, they can not corrupt the QoS of already admitted flows. The link databases together with CSPF can reduce the flow blocking probability and increase the network utilization for real-time traffic. This does not exist in DiffServ prototypes either, because route pinning is hard to be done with conventional IP routing.

5 Conclusion

In this paper, we gave a short overview of current mechanisms to support QoS in data communication networks. We have presented the aggregation technique as a means to achieve reservation state scalability in the routers. Aggregation for reservations can be implemented using tunnels or funnels. Tunnel reservations can be realized using LSPs while funnel reservations are only signaled by BGRP in an inter-networking context so far. We explained the basic signaling procedures for both approaches. Our main contribution is the description of a DiffServ-based network architecture that relies on e2e reservations. Individual reservations are aggregated by bandwidth-adaptive LSPs in the core which leads to reservation state scalability. We suggested to use a low degree of overreservation for the aggregate reservations coupled with a hysteresis to achieve signaling scalability. We suggested operations for an automated, dynamic, and incremental setup of such a LSP hierarchy. These operations reuse existing label distribution and resource reservation protocols as much as possible.

The available resources on the network links are signaled regularly to the resource utilization databases at the ingress border routers which use that information to compute constraint-based routes using the CSPF algorithm. MPLS helps to establish these explicit routes. The integration of CSPF reduces the blocking probability for new requests in a highly loaded network.

Our solution scales well if the network structure is hierarchical. The LSP hierarchy setup is automated, therefore, it is still a low-cost solution since no human interaction is required. Apart from AC, TE is performed by the integration of CSPF and leads to an optimized resource utilization. We believe that this architecture may be one step towards scalable next generation QoS networks.

For future work, there are still some open issues concerning the presented network architecture. Those may be evaluated by simulations or analytical in-

vestigations. Parameters like the *DatabaseUpdateInterval*, *NoResourceInterval*, and others need to be set in an appropriate way. MPLS may also be used for fast link and node failure recovery. For resource efficiency, thorough planning of backup LSPs using shared resources or preemption of other reservations is a crucial issue.

References

1. Braden, B., Clark, D., Shenker, S.: RFC1633: Integrated Services in the Internet Architecture: an Overview. http://www.ietf.org/rfc/rfc1633.txt (1994)
2. Wroclawski, J.: RFC2210: The use of RSVP with IETF integrated services. ftp://ftp.isi.edu/in-notes/rfc2210.txt (1997)
3. Blake, S., Black, D.L., Carlson, M.A., Davies, E., Wang, Z., Weiss, W.: RFC2475: An Architecture for Differentiated Services. ftp://ftp.isi.edu/in-notes/rfc2475.txt (1998)
4. Rosen, E.C., Viswanathan, A., Callon, R.: RFC3031: Multiprotocol Label Switching Architecture. http://www.ietf.org/rfc/rfc3031.txt (2001)
5. Menth, M., Hauck, N.: A Graph-Theoretical Concept for LSP Hierarchies. Technical Report, No. 287, University of Würzburg, Institute of Computer Science (2001)
6. Ashwood-Smith, P., Jamoussi, B., Fedyk, D., Skalecki, D.: Improving Topology Data Base Accuracy with Label Switched Path Feedback in Constraint Based Label. http://www.ietf.org/internet-drafts/draft-ietf-mpls-te-feed-05.txt (2002)
7. Menth, M.: A Scalable Protocol Architecture for End-to-End Signaling and Resource Reservation in IP Networks. In: 17^{th} International Teletraffic Congress, Salvador de Bahia, Brazil (2001) 211–222
8. Braden, B., Zhang, L., Berson, S., Herzog, S., Jamin, S.: RFC2205: Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification. ftp://ftp.isi.edu/in-notes/rfc2205.txt (1997)
9. Baker, F., Iturralde, C., Le Faucheur, F., Davie, B.: RFC3175: Aggregation of RSVP for IPv4 and IPv6 Reservations. http://www.ietf.org/rfc/rfc3175.txt (2001)
10. Terzis, A., Wang, J., Ogawa, J., Zhang, L.: A Two-Tier Resource Management Model for the Internet. In: Global Internet Symposium'99. (1999)
11. Günther, M., Braun, T.: Evaluation of Bandwidth Broker Signaling. In: International Conference on Network Protocols ICNP'99. (1999) 145–152
12. Politis, G.A., Sampatakos, P., Venieris, I.: Design of a Multi-Layer Bandwidth Broker Architecture. In: Interworking, Bergen, Norway (2000)
13. Teitelbaum, B., Hares, S., Dunn, L., Narayan, V., Neilson, R., Reichmeyer, F.: Internet2 QBone: Building a Testbed for Differentiated Services. IEEE Network Magazine (1999)
14. Zhang, Z.L.Z., Duan, Z., Hou, Y.T.: On Scalable Design of Bandwidth Brokers. IEICE Transaction on Communications **E84-B** (2001) 2011–2025
15. Awduche, D.O., Berger, L., Gan, D.H., Li, T., Srinivasan, V., Swallow, G.: RFC3209: RSVP-TE: Extensions to RSVP for LSP Tunnels. http://www.ietf.org/rfc/rfc3209.txt (2001)
16. Jamoussi, B., et al.: RFC3212: Constraint-Based LSP Setup using LDP. http://www.ietf.org/rfc/rfc3212.txt (2002)
17. Andersson, L., Doolan, P., Feldman, N., Fredette, A., Thomas, B.: LDP Specification. http://www.ietf.org/rfc/rfc3036.txt (2001)

18. Menth, M., Hauck, N.: A Graph-Theoretical Notation for the Construction of LSP Hierarchies. In: 15^{th} ITC Specialist Seminar, Würzburg, Germany (2002)
19. Kompella, K., Rekhter, Y.: LSP Hierarchy with Generalized MPLS TE. http://www.ietf.org/internet-drafts/draft-ietf-mpls-lsp-hierarchy-08.txt (2002)
20. Hummel, H., Grimminger, J.: Hierarchical LSP. http://www.ietf.org/internet-drafts/draft-hummel-mpls-hierarchical-lsp-01.txt (2002)
21. Hummel, H., Hoffmann, B.: O(n**2) Investigations. http://www.ietf.org/internet-drafts/draft-hummel-mpls-n-square-investigations-00.txt (2002)
22. Li, T., Rekhter, Y.: RFC2430: A Provider Architecture for Differentiated Services and Traffic Engineering (PASTE). ftp://ftp.isi.edu/in-notes/rfc2430.txt (1998)
23. Pan, P., Schulzrinne, H.: BGRP: A Tree-Based Aggregation Protocol for Interdomain Reservations. Journal of Communications and Networks **2** (2000) 157–167
24. Fehér, G., Németh, K., Maliosz, M., Czslényi, I., Bergkvist, J., Ahlard, D., Engborg, T.: Boomerang - A Simple Protocol for Resource Reservation in IP Networks. In: "IEEE Workshop on QoS Support for Real-Time Internet Applications ", Vancouver, Canada (1999)
25. Menth, M., Martin, R.: Performance Evaluation of the Extensions for Control Message Retransmissions in RSVP. In: 7^{th} International Workshop on Protocols For High-Speed Networks (PfHSN 2002), Berlin, Germany (2002)
26. Engel, T., Nikolouzou, E., Ricciato, F., Sampatakos, P.: Analysis of Adaptive Resource Distribution Algorithsm in the Framework of a Dynamic DiffServ IP Network. In: 8^{th} International Conference on Advances in Communications and Control (ComCon8), Crete, Greece (2001)
27. Koch, B.F.: A QoS Architecture with Adaptive Resource Control: The AQUILA Approach. In: 8^{th} International Conference on Advances in Communications and Control (ComCon8), Crete, Greece (2001)
28. Lee, C.Y., Ganti, S. Hass, B., Naidu, V.: Path Request and Path Reply Message. http://www.ietf.org/internet-drafts/draft-lee-mpls-path-request-04.txt (2002)
29. Vasseur, J.P., Iturralde, C., Zhang, R., Vinet, X., Matsushima, S., Atlas, A.: RSVP Path Computation Request and Reply Messages. http://www.ietf.org/internet-drafts/draft-vasseur-mpls-computation-rsvp-03.txt (2002)
30. Fu, H., Knightly, E.: Aggregation and Scalable QoS: A Performance Study. In: Proceedings of IWQoS 2001, Karlsruhe, Germany (2001)
31. Menth, M., Schmid, M., Heiß, H., Reim, T.: MEDF - A Simple Scheduling Algorithm for Two Real-Time Transport Service Classes with Application in the UTRAN. In: IEEE INFOCOM'03, San Francisco, USA (2003)
32. Bonald, T., Roberts, J.W.: Performance of Bandwidth Sharing Mechanisms for Service Differentiation in the Internet. In: 13^{th} International Teletraffic Congress Specialist Seminar, Monterey, USA (2000)

Load Balancing by MPLS in Differentiated Services Networks

Riikka Susitaival, Jorma Virtamo, and Samuli Aalto

Networking Laboratory, Helsinki University of Technology
P.O.Box 3000, FIN-02015 HUT, Finland
{riikka.susitaival, jorma.virtamo, samuli.aalto}@hut.fi

Abstract. Multi Protocol Label Switching (MPLS) assigns a short label to each packet and packets are forwarded according to these labels. The capability of MPLS of explicit routing as well as of splitting of the traffic on several paths allows load balancing. The paper first concentrates on two previously known approximations of the minimum-delay routing. Using these load balancing algorithms from the literature as a starting point, the main goal of this paper is to develop optimization algorithms that differentiate classes in terms of mean delay using of both routing and WFQ-scheduling. Both optimal and approximative algorithms are developed for the joint optimization of the WFQ-weights and routing. As a result it is found that the use of the approximations simplifies the optimization problem but still provides results that are near to optimal.

Keywords: MPLS, load balancing, differentiated services, scheduling

1 Introduction

In the conventional IP routing, forwarding decisions are made independently in each router, based on the packet's header and precalculated routing tables. MPLS (Multi Protocol Label Switching) is a flexible technology that enables new services in IP networks and makes routing more effective [1,2]. It combines two different approaches, datagram and virtual circuit, as a compact technology. MPLS is based on short fixed length labels, which are assigned to each packet at the ingress node of the MPLS cloud. These labels are used to make forwarding decisions at each node. This simplifies and improves forwarding. The architecture of MPLS is defined in [3].

One of the most significant applications of MPLS is Traffic Engineering. Traffic Engineering (TE) concerns performance optimization of operational networks [4]. Traffic Engineering using MPLS provides mechanisms to route traffic that have equal starting point and destination along several paths. The most important benefit of traffic splitting is the ability to balance load.

MPLS and its Traffic Engineering capabilities could provide technical support to the implementation of Quality of Service (QoS). The differentiation of services can be obtained by an alternative flow allocation that has the same

W. Burakowski et al. (Eds.): Art-QoS 2003, LNCS 2698, pp. 252–264, 2003.

principles as the load balancing methods. In order to make differentiation more effective, scheduling mechanisms, like WFQ-scheduling, can be utilized in the same context.

Our goal is to adjust routing and scheduling parameters that optimize differentiation of experienced service of different classes in terms of their mean delays. We use load balancing methods as a starting point in the further development. The routing and scheduling methods to be introduced are divided into two types. The first type tries to optimize only flow allocation so that differentiation is achieved. The second type of methods makes use of WFQ-weights. In each node, each service class has a WFQ-weight and the bandwidth is shared according to these weights using approximation of parallel independent queues. More details of these methods can be found in [5].

The rest of this paper is organized as follows: In the second section we concentrate on the three previously known load balancing algorithms. In section 3 we introduce flow allocation methods that differentiate traffic classes by routing only. We present the flow allocation model that makes use of WFQ-scheduling in section 4. We develop both optimal and approximative algorithms. In section 5 we present numerical results of all algorithms. Finally, section 6 makes some conclusions.

2 Load Balancing Algorithms

Load balancing methods make an attempt to balance load in the network and therefore achieve better performance in terms of delay. The basic optimization problem minimizes the mean delay in the network. The use of the link delays of M/M/1-queues leads to a non-linear optimization problem (NLP). Many exact algorithms have been introduced to this optimization, the most famous one being Gallager's algorithm from year 1977 [6].

2.1 Minimum-Delay Routing

First we formulate the load balancing as an optimization problem, which minimizes the mean delay of the network. Consider a network consisting of N nodes. A pair of nodes (m, n) can be connected by a directed link (m, n) with bandwidth equal to $b_{(m,n)}$. The number of links is denoted by L and the topology is denoted by T, which is a set of node-pairs. Let $A \in \mathbb{R}^{N \times L}$ be the matrix for which $A(i, j) = -1$ if link j directs to node i, $A(i, j) = 1$ if link j leaves from node i and $A(i, j) = 0$ otherwise.

The traffic demands are given by the matrix $[d_{(i,j)}]$, where i is the ingress and j is the egress node. $R_{(i,j)} \in \mathbb{R}^{N \times 1}$ is a vector for each ingress-egress pair (i, j) such that $R_{(i,j),k} = d_{(i,j)}$, if k is the ingress node, $R_{(i,j),k} = -d_{(i,j)}$, if k is the egress node, and $R_{(i,j),k} = 0$ otherwise. Demands and capacities are assumed to be constant (or average traffic rates). Let $x_{(i,j),(m,n)}$ be the allocated traffic of ingress-egress pair (i, j) on link (m, n). Then the total traffic on the link (m, n) is

$$X_{(m,n)} = \sum_{(i,j)} x_{(i,j),(m,n)}. \tag{1}$$

The formulation of the optimization problem that minimizes the mean delay in the whole network is as follows:

$$\begin{aligned} &\text{Minimize } E[D] = \tfrac{1}{\Lambda} \textstyle\sum_{(m,n)} \frac{X_{(m,n)}}{b_{(m,n)} - X_{(m,n)}}, \\ &\text{subject to the constraints} \\ &X_{(m,n)} < b_{(m,n)}, && \text{for each } (m,n), \\ &Ax_{(i,j)} = R_{(i,j)}, && \text{for each } (i,j), \end{aligned} \tag{2}$$

where Λ is the total offered traffic of the network. The last equation in (2) states that, at every node n, incoming traffic of each ingress-egress pair must be equal to outgoing traffic.

2.2 Flow Allocation Using Two-Step Algorithm

The algorithm that calculates an optimal flow allocation in terms of mean delay may be approximated by using the approach presented in [7]. The approximative algorithm solves first the paths to be used by LP-optimization and, after that, allocates the traffic to these paths using NLP-optimization.

The pair-based flow formulation that minimize the maximum link load is as follows:

$$\begin{aligned} &\text{Minimize } [-\epsilon Z + \sum_{(m,n)} w_{(m,n)} \sum_{(i,j)} x_{(i,j),(m,n)}] \\ &\text{subject to the constraints} \\ &x_{(i,j),(m,n)} \geq 0; \quad Z \geq 0, \\ &\textstyle\sum_{(i,j)} x_{(i,j),(m,n)} + \sqrt{b_{(m,n)}} Z \leq b_{(m,n)}, && \text{for each } (m,n) \text{ with } b_{(m,n)} > 0, \\ &Ax_{(i,j)} = R_{(i,j)}, && \text{for each } (i,j), \end{aligned} \tag{3}$$

where $w_{(m,n)}$ is a cost weight of link (m,n) and Z is a free parameter that describes the minimum value of the proportional unused capacity.

When the LP-problem (3) is solved and variables $x_{(i,j),(m,n)}$ found, we have to find paths for each ingress-egress pair. The algorithm to define paths to ingress-egress pair (i,j) is as follows: We have the original topology T, which consists of the set of directed links. Because one ingress-egress pair uses only part of the whole topology, we define a new topology $T'_{(i,j)}$, which consists of the links for which $x_{(i,j),(m,n)}$ differs from zero. Let $L'_{(i,j)}$ be the number of links in $T'_{(i,j)}$. Topology $T'_{(i,j)}$ is loop-free, because if there were loops, the original allocation would not be optimal. We search all possible paths to ingress-egress pair (i,j) by

a breadth-first-search algorithm. The set of these paths is denoted by $P_{(i,j)}$. Let $K_{(i,j)}$ be the number of paths and $Q_{(i,j)} \in \mathbb{R}^{L'_{(i,j)} \times K_{(i,j)}}$ be the matrix, where

$$Q_{(i,j),(l,k)} = \begin{cases} 1, & \text{if path } k \text{ uses link } l \text{ of topology } T'_{(i,j)}, \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

for each ingress-egress pair (i, j). Let $y_{(i,j),k}$ be the traffic allocation in path k. Unknown $y_{(i,j),k}$'s can be solved from matrix equation

$$Q_{(i,j)} y_{(i,j)} = x'_{(i,j)}, \quad \text{for each } (i,j), \tag{5}$$

where $x'_{(i,j)} \in \mathbb{R}^{L'_{(i,j)} \times 1}$ is the flow vector for each ingress-egress pair (i, j). Finally, if element k of the solution vector $y_{(i,j)}$ differs from zero, ingress-egress pair (i, j) uses path k, else not. So reducing the unused paths from path set $P_{(i,j)}$ we get actual path set $P'_{(i,j)}$.

The objective is to find an optimal flow allocation to these paths. Let $P'_{(i,j),k}$ be the k:th path of node-pair (i, j), $K'_{(i,j)}$ be the number of paths in $P'_{(i,j)}$, and $\phi_{(i,j),k}$ be the fraction of $d_{(i,j)}$ allocated to path $P'_{(i,j),k}$. The structure of path k is defined by $\delta_{(m,n),(i,j),k}$ as follows:

$$\delta_{(m,n),(i,j),k} = \begin{cases} 1, & \text{if path } P'_{(i,j),k} \text{ uses link } (m,n), \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Note that the traffic allocation $y_{(i,j),k}$ above is a special case of traffic allocation $d_{(i,j)}\phi_{(i,j),k}$. Our objective is to minimize the mean delay of the total network. So the optimization problem is as follows:

Minimize $\frac{1}{\Lambda} \sum_{(m,n)} \frac{\sum_{(i,j),k} \delta_{(m,n),(i,j),k} d_{(i,j)} \phi_{(i,j),k}}{b_{(m,n)} - \sum_{(i,j),k} \delta_{(m,n),(i,j),k} d_{(i,j)} \phi_{(i,j),k}}$,

subject to the conditions

$$\begin{aligned} &\textstyle\sum_{(i,j),k} \delta_{(m,n),(i,j),k} d_{(i,j)} \phi_{(i,j),k} < b_{(m,n)}, && \text{for each } (m,n), \\ &\phi_{(i,j),k} \geq 0, && \text{for each } (i,j),k, \\ &\textstyle\sum_{k=1}^{K'_{(i,j)}} \phi_{(i,j),k} = 1, && \text{for each } (i,j). \end{aligned} \tag{7}$$

2.3 Heuristic Approach

The heuristics presented in [8] to allocate traffic into the network using a particular level of granularity is very simple. The traffic granularity refers to the level of traffic aggregation [8]. The finer the level of the granularity the finer the traffic partitioning to different paths. In this algorithm, depending on the level of granularity, traffic demands from ingress to egress node are divided into streams (e.g. using some hashing function). The streams are sorted in descending order in terms of their traffic demand. After that each stream is routed sequentially one at a time to the shortest path defined by Dijkstra's algorithm using the mean delay of an $M/M/1$-queue as a link cost.

3 Methods to Achieve Differentiation by Routing

In this section we present such flow allocation methods that try to differentiate traffic classes by routing only. The traffic classes with a higher priority are routed along the paths that are not congested, for example. We differentiate classes by minimizing the sum of weighted mean delays, by fixing the ratio of mean delays, and using a heuristic approach.

3.1 Optimization Using Cost Weights

We consider the situation where the total traffic is divided into traffic classes, into the gold, silver and bronze classes, for example. The gold class has the highest priority and the bronze class the lowest priority. Each traffic class l has its own traffic matrix $[d_{l,(i,j)}]$, where i is the ingress node and j is the egress node. $R_{l,(i,j)} \in \mathbb{R}^{N\times 1}$ is an array for each class l and ingress-egress pair (i,j), where $R_{l,(i,j),k} = d_{l,(i,j)}$, if k is the ingress node, $R_{l,(i,j),k} = -d_{l,(i,j)}$, if k is the egress node and $R_{l,(i,j),k} = 0$ otherwise. The total traffic offered by class l is denoted by Λ_l. Let $x_{l,(i,j),(m,n)}$ be the allocated traffic of ingress-egress pair (i,j) of class l on link (m,n). So the total traffic of class l on link (m,n) is

$$X_{l,(m,n)} = \sum_{(i,j)} x_{l,(i,j),(m,n)}, \text{ for each } l,(m,n). \tag{8}$$

Let w_l be the cost weight associated to traffic class l. Additional notation used is the same as in section 2.2.

The purpose is to divide traffic into paths so that classes with higher priority achieve smaller mean delay than other classes. The optimization problem, where we minimize the sum of the weighted mean delays of the classes, is as follows:

$$\begin{aligned}
&\text{Minimize } \sum_l w_l E[D_l] = \sum_{(m,n)} \frac{\sum_l \frac{w_l X_{l,(m,n)}}{\Lambda_l}}{b_{(m,n)} - \sum_l X_{l,(m,n)}},\\
&\text{subject to the constraints}\\
&\sum_l X_{l,(m,n)} < b_{(m,n)}, && \text{for each } (m,n),\\
&Ax_{l,(i,j)} = R_{l,(i,j)}, && \text{for each } l,(i,j),
\end{aligned} \tag{9}$$

where traffic allocations $x_{l,(i,j),(m,n)}$ are decision variables.

When the cost weights of different classes differ sufficiently, the optimization function tries to minimize the mean delays of gold class at the expense of lower priority classes. As a result, the routing obtained by the optimization function above differs from the load balanced routing, because in the load balancing the delays of the links are balanced to be almost equal and the differentiation could occur only if the paths are of different length.

3.2 Optimization with a Fixed Mean Delay Ratio

Now we fix the ratio of the mean delays of various classes to some value in order to differentiate classes. For example, in the case of two classes (gold and silver),

the ratio of the mean delays between the silver and the gold class could be fixed to parameter q:

$$\frac{E[D_{l_2}]}{E[D_{l_1}]} = \frac{\frac{1}{\Lambda_{l_2}} \sum_{(m,n)} \frac{X_{l_2,(m,n)}}{b_{(m,n)} - \sum_l X_{l,(m,n)}}}{\frac{1}{\Lambda_{l_1}} \sum_{(m,n)} \frac{X_{l_1,(m,n)}}{b_{(m,n)} - \sum_l X_{l,(m,n)}}} = q. \tag{10}$$

After that the optimization can be done by minimizing the mean delay of either class.

Compared to the optimization in the previous section, this approach does not include cost weights and the actual ratio of the mean delays is known before the optimization. In order to make the optimization procedure easier it is useful to constraint the ratio of the mean delays to some small interval rather than to the exact value.

3.3 Heuristics

There exists a demand to provide also simple routing methods that can be implemented without heavy optimization. The approach that routes traffic to the network near optimally but still obtains the differentiation in terms of mean delay tries to adapt the heuristic approach presented in section 2.3.

The heuristic approach in the case of two classes (gold and silver) is as follows: The gold class is routed using heuristics introduced in 2.3. Then the allocated traffic of the gold class is multiplied by $1 + \Delta$. The silver class is then routed using heuristics in 2.3.

The idea of the heuristics is that the links used by the gold class look more congested than they actually are. So the routing scheme tries to balance load by routing the silver class by some other way. That is, the artificial congestion forces the silver class to avoid links used by the gold class and therefore the gold class should achieve more bandwidth. The choice of the parameter Δ depends on how much there is traffic offered compared to the capacity of the network.

4 Methods to Achieve Differentiation in Mean Delay Using Routing and WFQ-Scheduling

The possibilities to provide differentiated services using routing only are limited. To achieve certain ratio of mean delays may lead up to disadvantageous routing because the low priority class is routed along long and congested paths in order to artificially obtain the desired ratio of the mean delay.

Weighted Fair Queueing (WFQ) as a packet scheduling mechanism divides bandwidth among the parallel queues. Each queue achieves a guaranteed bandwidth, which depends on the WFQ-weight of that queue and the link capacity. We try to find optimal routing that differentiates the quality of service by including the WFQ-weighs to the optimization function as free parameters.

Because WFQ-scheduling is a work-conserving discipline, the bandwidth that is guaranteed for a class in our model can be viewed as the lower bound. Actually,

the bandwidth available to a class can be much greater as the other classes may not always use the bandwidth reserved for them.

The bandwidth of each link (m, n) is shared according to WFQ-weights. We approximate the behavior of WFQ-scheduling as follows: Let $\gamma_{l,(m,n)}$ be the WFQ-weight that determines the proportion of total bandwidth that is given to class l. The bandwidth $b_{l,(m,n)}$ of class l on link (m, n) is thus

$$b_{l,(m,n)} = \gamma_{l,(m,n)} b_{(m,n)}, \text{ for each } l, (m, n). \tag{11}$$

The sum of the WFQ-weights of the classes on each link must equal to one. As a result, we have changed over from the WFQ-scheduling system to the system of parallel independent queues with link capacities described above.

4.1 Optimization Using Cost Weights

In this section we obtain the differentiation between classes by using the cost weights as in (9). The gold class gets the greatest cost weight and so on. The joint optimization of flow allocation and WFQ-weights where the sum of weighted mean delays is minimized is as follows:

$$\begin{aligned}
&\text{Minimize } \textstyle\sum_l w_l E[D_l] = \sum_l \frac{w_l}{\Lambda_l} \sum_{(m,n)} \frac{X_{l,(m,n)}}{\gamma_{l,(m,n)} b_{(m,n)} - X_{l,(m,n)}}, \\
&\text{subject to the constraints} \\
&X_{l,(m,n)} < \gamma_{l,(m,n)} b_{(m,n)}, && \text{for each } l, (m, n), \\
&Ax_{l,(i,j)} = R_{l,(i,j)}, && \text{for each } l, (i, j), \\
&0 < \gamma_{l,(m,n)} < 1, && \text{for each } l, (m, n), \\
&\textstyle\sum_l \gamma_{l,(m,n)} = 1, && \text{for each } (m, n),
\end{aligned} \tag{12}$$

where traffic allocations $x_{l,(i,j),(m,n)}$ and WFQ-weights $\gamma_{l,(m,n)}$ are decision variables. This straightforward optimization is referred to "Str".

The optimization problem presented in (12) is quite heavy and time-consuming. We introduce near optimal algorithms that make the size of the problem smaller and the calculation easier. The first two algorithms first allocate the traffic into the network and after that optimize WFQ-weights. The structure of both algorithms is as follows:

1. Allocate the traffic into the network without WFQ-weights so that the weighted sum of mean delays is minimized. The formulation of the optimization algorithm is presented in section 3.1.
2. Fix the traffic allocation obtained in the first step.
3. Determine WFQ-weights using optimization problem (12) applied to the fixed link flows. Now the number of free variables equals the number of WFQ-weights, the number of links in the network multiplied by the number of classes minus one.

The cost weights of the optimization function are selected twice, in steps 1 and 3. In the first two-step algorithm (referred to as "2StepV1") we select the cost weights in the first step to be

$$w_l = \frac{\Lambda_l}{\sum_k \Lambda_k}. \tag{13}$$

Now the flow allocation is optimal and differences in mean delays do not appear in the first step. This algorithm makes only use of WFQ-scheduling when trying to differentiate classes. In the second two-step algorithm (referred to as "StepV2") the cost weights in the first and third steps are equal. So this algorithm utilizes both routing and WFQ-scheduling when differentiating classes and can therefore be closer to the optimal.

The third approximative algorithm makes use of the two-step algorithm presented in section 2.2 and is referred to as "QoS-LP-NLP". The paths are first calculated using the linear optimization that minimizes the maximum link load. Then the traffic is allocated to these paths and WFQ-weights are determined using the non-linear optimization (12) applied to the fixed paths.

4.2 Fixing the Link Delay Ratio

In the routing with WFQ-weights, if the ratio of total mean delays is fixed like in the algorithm presented in section 3.2, the optimization problem is demanding. One possibility is to fix the mean delay ratios at the link level. If the lengths of paths of different classes do not differ significantly, this approach should result approximately in the same mean delay ratio in the whole network.

We consider the case with two classes, gold and silver. We fix the ratio of link delays to parameter q and solve WFQ-weights $\gamma_{l_1,(m,n)}$'s as a function of the traffic of both classes, that is

$$\gamma_{l_1,(m,n)}(X_{l_1,(m,n)}, X_{l_2,(m,n)}) = \frac{qb_{(m,n)} + X_{l_1,(m,n)} - qX_{l_2,(m,n)}}{b_{(m,n)}(q+1)}, \text{ for each } (m,n). \tag{14}$$

The optimization problem of the flow allocation with the fixed ratio of mean delays is almost similar to optimization problem (12). The difference is that the WFQ-weights are not free parameters in the approach with fixed link delays. If we want to differentiate the classes by fixing the link delays only, the cost weights of the optimization function are equal to one (referred to as "FLD"). We can also optimize the sum of the weighted mean delays (referred to as "FLDW"). The problem is now how to determine the cost weights in relation to the ratio of link delays.

5 Numerical Results

The formulations of the optimization problems were written using a General Algebraic Modelling System (GAMS), which is a high-level modelling language

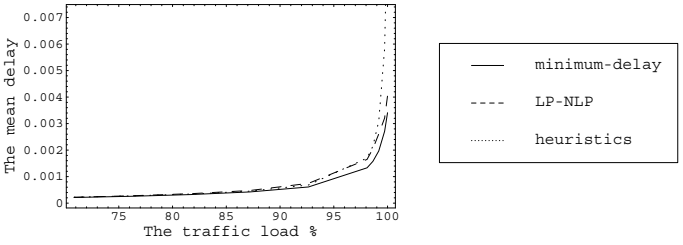


Fig. 1. The mean delay as a function of the traffic load

for mathematical programming problems [9]. We have used solver module Minos 5 in our optimizations.

The algorithms are tested in a known test-network, which consists of 10 nodes, 58 links and 72 ingress-egress pairs. The link capacities and the traffic demands are available at web-page http://brookfield.ans.net/omp/random-test-cases.html. In order to make optimizations simpler, we study only the case of two classes, gold and silver. The traffic matrices of both classes are equal, the half of the original demands, so that the comparison between the traffic classes is easier.

5.1 Load Balancing Routing

The mean delay and the relative deviation of the mean delay from the optimal as a function of the traffic load of the three routing methods of section 2 are presented in Figure 1 and 2. With the heuristic approach, we use the granularity level 32, except in the cases of heavy load (the traffic load is over 95%) when the used granularity level is 128. We can see that the mean delays of different methods do not differ significantly. Only when the traffic load is near to the total capacity of the network is the performance of the minimum-delay routing notable.

We find that the maximum number of paths used per each ingress-egress pair is only 3 in both minimum-delay routing and LP-NLP routing in the case of heavy traffic load (98%). Only 20% of the pairs in LP-NLP routing and 30% of the pairs in minimum-delay routing use more than one path. The computation time for minimum-delay routing is about 6.6 seconds, whereas it is for LP-NLP routing ten times smaller, about 0.6 seconds.

5.2 Methods to Achieve Differentiation Using Routing Only

The three approaches in section 3 make an attempt to differentiate the classes by routing only. The first optimization problem minimizes the sum of weighted mean delays. As a result, the ratio of the mean delay of the silver class to the mean delay of the gold class and the growth of the total mean delay as a function of the cost weight of the gold class is presented in Figure 3.

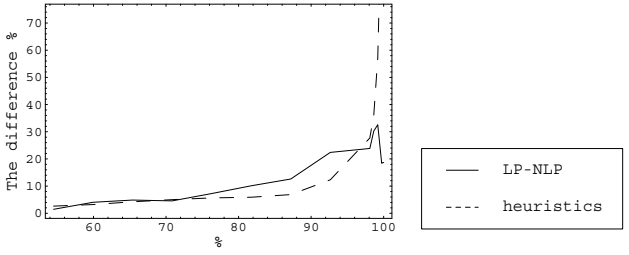


Fig. 2. The relative deviation of the mean delay from the optimal as a function of the traffic load

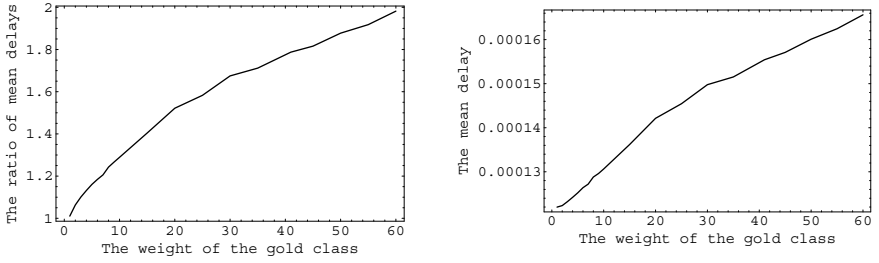


Fig. 3. The ratio of mean delays and the total mean delay as a function of the weight of the gold class

We take the total mean delay of the network as a function of the ratio of mean delay as a performance indicator. The increase in mean delay describes the cost of achieving a certain level of differentiation. All three methods are compared in Figure 4. The figure shows that the first and second optimizations generate the same result. However, the benefit of optimization function (10) is that the ratio of mean delays is known a priori, while the cost weights of optimization function (9) must be determined.

5.3 Methods to Achieve Differentiation by Routing and WFQ-Scheduling

First we have implemented the optimization method that minimizes the weighted sum of mean delays straightforwardly and the optimization methods that divide the problem into two steps (introduced in section 4.1).

A near optimal routing can also be achieved using an iterative approach (referred to as "2StepIt"). The flow allocation and the WFQ-weights are optimized alternatingly. The number of iterations is ten, which means that ten flow allocations and ten WFQ-weight determinations are done in the optimization.

The ratio of the mean delays and the total mean delay of all five algorithms as a function of the weight of gold class are presented in Figure 5 and 6. Note

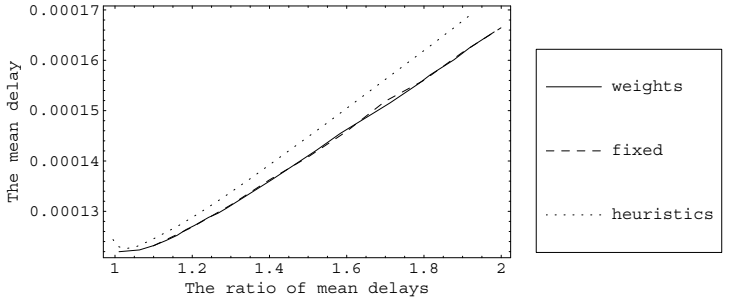


Fig. 4. The total mean delay as a function of the ratio of mean delays

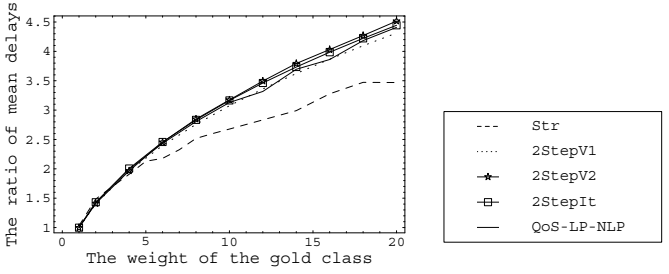


Fig. 5. The ratio of mean delays as a function of the weight of the gold class

that also the total mean delay is calculated using the approximation of parallel queues. Thus the figures in this subsection are not comparable with the figures in subsection 5.2. The irregularities in the curve of the straightforward routing are perhaps a consequence of numerical errors in the optimization procedure.

We have also implemented optimizations that fix the ratio of link delays to some parameter q (problems "FLD" and "FLDW"). In the latter problem we have implemented only one case, where the cost weight is three times greater than link delay ratio q. In Figure 7 we present the relation between the ratio of link delays and the ratio of mean delays of different classes. In the problem, where we fix only q ("FLD"), the ratio of mean delays seems to be smaller than the ratio of link delays. The explanation is that the routing algorithm tries to balance traffic load by routing classes that achieve more bandwidth through long routes.

Finally, we compare all the methods. The performance metric is the same as in section 5.2, the total mean delay of the network as a function of the ratio of mean delays. The results are presented in Figure 8. The straightforward optimization seems to have the smallest mean delay. The difference to other algorithms is significant when the ratio of mean delays is small. When the ratio is greater, the performance of the two-step algorithm that utilizes both routing

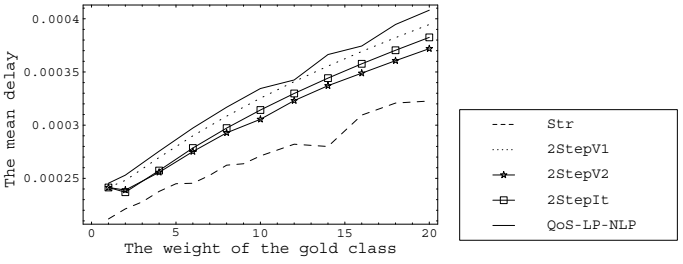


Fig. 6. The total mean delay as a function of the weight of the gold class

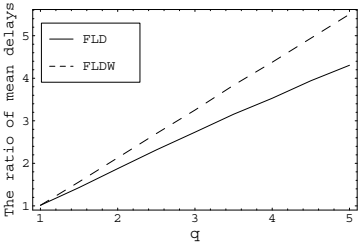


Fig. 7. The ratio of mean delays as a function of q

and WFQ-weights ("2StepV2") is near to optimal. As a conclusion, the differentiation methods that make use of both routing and WFQ-scheduling ("Str", "2StepV2", "2StepIt", "FLDW") perform better than the differentiation methods that make use of only WFQ-scheduling ("2StepV1", "QoS-LP-NLP" and "FLD").

6 Conclusions

We have used load balancing algorithms as a starting point when developing methods that try to differentiate traffic classes in terms of the mean delay. In the first model differentiation is achieved by using routing only. The mean delay using the algorithm that uses cost weights and the algorithm that fixes the ratio of mean delays are equal. The advantage of the latter algorithm is that the cost weights need not be known in advance. The mean delay using the heuristic approach is a little greater than using the other two algorithms.

We have also presented a model where the bandwidth of each link is shared among the traffic classes according to the WFQ-weights. The optimization problem is to minimize the weighted mean delay. In addition, near-optimal heuristic algorithms have been introduced. We notice that the use of the algorithm that makes use of both routing and WFQ-scheduling gives the best result.

The bandwidth guaranteed by WFQ-scheduling to each class is the theoretical minimum. As a result, the actual ratio of mean delays may differ from the

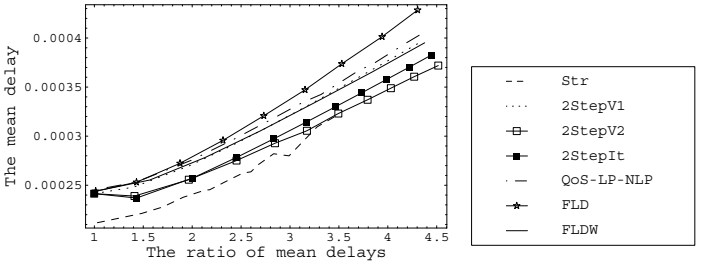


Fig. 8. The total mean delay as a function of the ratio of mean delays

result obtained by the optimizations. It would be interesting to know whether the actual ratio of mean delays is greater or smaller. A simulation study of WFQ-scheduling may provide some answers. That would also help the comparison between differentiation with routing only and differentiation with scheduling and routing. Adaptive algorithms used in the situation where the traffic matrices of the classes are unknown would be useful.

References

1. A. Viswanathan, N. Feldman, Z. Wang and R. Callon, Evolution of Multi-Protocol Label Swithing, IEEE Communication Magazine, Vol 36, Issue 5, pp. 165–173, May 1998.
2. G. Armitage, MPLS: The Magic Behind the Myths, IEEE Communications Magazine, Vol 38, Issue 1, pp. 124–131, January 2000.
3. E. Rosen, A. Viswanathan and R. Callon, Multiprotocol Label Switching Architecture, IETF RFC 3031, January 2001.
4. D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell and J. McManus, Requirements for Traffic Engineering over MPLS, IETF RFC 2702, September 1999.
5. R. Susitaival, Load Balancing by MPLS in Differentiated Services Networks, Master's Thesis, Helsinki University of Technology, 2002.
6. R.G. Gallager, A Minimum Delay Routing Algorithm Using Distributed Computation, IEEE Transactions on Communication, Vol COM-25, Nr 1, pp. 73–85, January 1977.
7. T. Ott, T. Bogovic, T. Carpenter, K. R. Krishnan and D. Shallcross, Algorithms for Flow Allocation for Multi Protocol Label Switching, Telcordia Technical Memorandum TM-26027, 2001.
8. A. Sridharan, S. Bhattacharyya, R. Guérin, J. Jetcheva and N. Taft, On The Impact of Aggregation on The Performance of Traffic Aware Routing, Technical report, University of Pennsylvania, July 2000.
9. http://www.gams.com.

An Integrated Scheduling for Multiple Loss Priority Traffic in E-PON OLT Switches

Myoung Hun Kim and Hong Shik Park

Information and Communications University
58-4, Hwaam-Dong, Yuseong-gu, Daejeon, 305-732, Korea
{fiaa,hspark}@icu.ac.kr

Abstract. In this paper, we deal with the problem of scheduling packets in an input queued switch when both unicast and multicast traffic are present over broadcasting networks such as Ethernet-Passive Optical Network (E-PON). We propose a Multicast Bypass (MULBY) architecture to perform integration efficiently and to support QoS class traffic. The architecture helps existing unicast scheduling algorithms to support integrated traffic. The MULBY is devised for easy implementation eliminating the need of traditional integrated scheduling problems such as complex fanout splitting and unicast integration problem [1]. Numerical analysis and simulation study is performed to show that the proposed MULBY architecture improves average latency of unicast traffic significantly even in the high ratio of multicast traffic when round-robin matching (RRM) or *i*SLIP is used as a scheduling algorithm.

1 Introduction

Switching systems are prevailed almost all over the current networks. Emerging optical networks also use the switching systems wherever they are used in access networks. The E-PON [2] is viewed as a technology for full-service access networks and especially it is designed to include multimedia broadcasting services. The problem of scheduling unicast cells in an input-queued switch has received much attention. It is known that the maximum weight matching (MWM) algorithm delivers up to 100% throughput in a single switch [3], [4], [5]. However, it is too complex to implement since it requires $O(N^3)$ iterations in the worst case. Several good switch scheduling algorithms have been proposed; *i*SLIP [6], *i*LQF [7], RPA [8], and MUCS [9]. With centralized implementations the run-time of these algorithms is $O(N^2)$ or more. However, the performance of these algorithms is poor compared to MWM under non-uniform input traffic: they induce very large delays and their throughput can be less than 100%. The problem of scheduling multicast cells in an input-queued switch has received interests. A lot of work done has been centered around the basic scheme of putting a copy network before the fabric of a point-to-point switch. [10] gives an excellent performance analysis of a multicast switch with this copy network. It utilizes a

W. Burakowski et al. (Eds.): Art-QoS 2003, LNCS 2698, pp. 265–275, 2003.

partial multicast discipline, infinite buffering and a random scheduling (RND) scheme in the case of uniform copy traffic.

Unicast and multicast traffic are expected to co-exist in varying proportions, and hence scheduling algorithms that serve both types of traffic efficiently are required. Options include multicast algorithms that treat unicast as a special case of a multicast packet with a fanout of unity, or unicast scheduling algorithms that effectively convert a multicast packet to multiple unicast packets using a copy network. And there are traditional integrated scheduling problems such as complex fanout splitting and unicast integration problem [1]. We are of the opinion that none of them will perform as efficiently as an algorithm that takes care of both types of traffic while making its scheduling decision, especially since unicast and multicast traffic occurs in varying proportions of the total traffic.

One of the typical problems with the implementation of a switch over the E-PON is the difficulty of QoS guarantee. Many applications use multiple classes of traffic with different priority levels. The basic arbitration algorithms such as round-robin, *i*SLIP, and so on can be extended to include requests at multiple priority levels at the expense of performance and complexity. The performance and complexity get worse if unicast and multicast traffic are integrated. These features lead to infeasible complexity level when it comes to implementation in the end.

The MULBY architecture adopts multiple loss priority queue scheme that is simple to implement and is not to degrade the performance. The architecture controls loss performance by using multiple thresholds. Our primary goal is to investigate a simple switch architecture and efficient scheduling algorithm for integrated traffic in an E-PON system. And different QoS classes are also considered with multiple loss priority queue.

2 Multicast Bypass (MULBY) Architecture for Integrated Traffic Switching

It is our purpose in this section to present a buffer management and a scheduling strategy in the input queued switch for integrated unicast and multicast traffic. Figure.1. shows the proposed Multicast Bypass switch architecture. We assume the switch operates on fixed-size cells. Each input has N virtual output queues (VOQs), one for each output. We place extra number of Y multicast output ports and these ports only transmit multicast traffic. As a result each input has $N+Y$ VOQs. Here Y VOQs are called multicast VOQs.

2.1 Modeling of Multiple Loss Priority Queue to Support QoS

In E-PON, optical line terminal (OLT) switch may support a number of packet streams having different priority levels. It is expected that different priority levels are associated with different packet streams and they will be multiplexed. Now we are

focusing on the input multicast VOQ that is implemented with multiple loss priority queue characteristics.

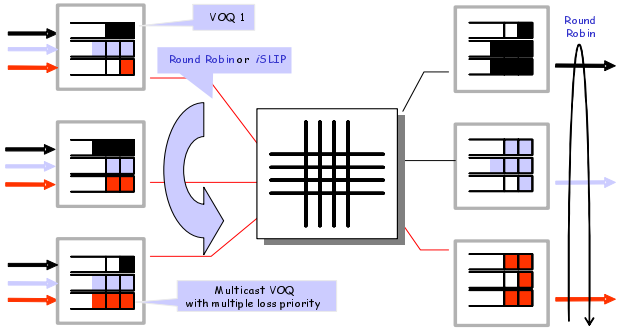


Fig. 1. Multicast Bypass switch architecture

Loss priority is implemented by partitioning the buffer into as many levels of occupancy as there are priority classes; cells of a particular priority class are either admitted or discarded depending upon the prevailing level of buffer occupancy. As shown in Figure.2, the high priority queue is fed with delay sensitive traffic which consists of J packet loss priority classes corresponding to the threshold $B_1, B_2, \ldots, B_j, \ldots, B_J$, where j is the priority number and $j=1$ is the highest packet loss priority. When the buffer content is in the buffer occupancy level L_j, i.e. $B_j < X_t < B_{j+1}$, the only cells admitted to the buffer belong to classes $\{1, 2, \ldots, J-j\}$. Here X_t is buffer occupancy at time t. Thus at level L_0 packets of all classes are admitted. And, at the boundary B_{j+1} $(0 \le j \le J-1)$ packets of class $(J-j)$ are admitted only if their presence does not cause the buffer occupancy to exceed B_{j+1}. That is, at the upper boundary of each level, cells of the class with the lowest priority, among those admitted at the level, may be lost and the loss rate is determined by the residual capacity of the output rate.

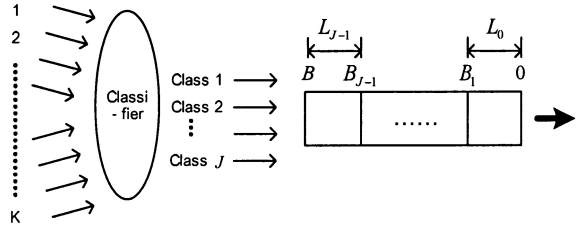


Fig. 2. Multiple loss priority queue for a multicast VOQ

A packet stream generated by each user source is modeled as an ON/OFF process, in that the transition rate from ON state to OFF state is α and the rate of transition rate from OFF state to ON state is β. When K of such ON/OFF packet streams are multiplexed, the result can be represented by an $(K+1)$ state Markov modulated process [11], where the state i represents the number of ON state packet streams

2.2 Scheduling Algorithms in the MULBY Architecture for Integrated Traffic

Properties of the E-PON are such that it cannot be considered either shared medium or a point-to-point network; rather, it is a combination of both. Because the Ethernet is broadcasting by nature, in the downstream direction (from network to user), it fits perfectly with the E-PON architecture: packets are broadcast by the OLT and extracted by their destination optical network unit (ONU) based on the media access control (MAC) address. We notice that broadcasting and multicasting service, an important target application in E-PON, have larger volume of downstream traffic than upstream traffic. And E-PON has a broadcasting downstream channel and serves non-cooperative users. The key concept of the proposed MULBY switch architecture is coming from the broadcasting downstream channel. This feature can reduce the amount of burden from switching. In other word, there is no need to perform complex integrated unicast and multicast traffic scheduling in a switch fabric when the MULBY architecture is applied.

Let consider an $N \times (N+Y)$ switch, and see Figure.3. We aim to schedule both unicast and multicast packets. Every input port has virtual output queues (VOQs) for the output ports and additionally one VOQ for multicast traffic entering from the input port. Here Y is the number of multicast bypass queue/port that is accepting multicast traffic only. This means that other unicast output queues/ports do not have to process any multicast traffic.

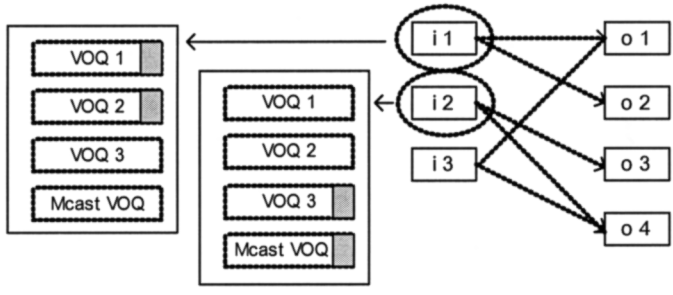


Fig. 3. Request step of MULBY switch

Here we consider two scheduling algorithms such as round-robin matching and *i*SLIP. The MULBY_RR scheduling algorithm uses rotating priority called round robin matching (RRM) to schedule each active input and output in turn. The main charac-

teristic of the MULBY_RR is its simplicity: it is readily implemented in hardware and can operate at high speed. The MULBY_RR scheduling algorithm is a variation of simple basic round-robin matching algorithm. For the process of multicast traffic in RRM with crossbar fabric point-to-point matching is required. When a multicast packet enters in an input port the multicast packet will be copied into multiple destination VOQs in the input port if the point-to-point matching and crossbar switch is used. Then the copied multicast packets in the VOQs will be switched like a unicast packet. However this mechanism has a drawback; if the ratio of multicast traffic increase latency of all packets will increase because of copying multicast packets into multiple VOQs. This affects the performance of switching of unicast traffic seriously. To cope with the degradation of integrated traffic switching performance the MULBY_RR algorithm eliminates the process of copying multicast packets. Instead MULBY_RR places the multicast packets into a dedicated multicast VOQ in each input port. Then the multicast VOQ is dealt with same as other unicast VOQ competing for the possibly less number (Y) of multicast bypass output queue. For example, when the multicast VOQ in input port 1 and the multicast VOQ in input port 2 have waiting packets they need arbitration if the number of multicast bypass queues in output is one ($Y=1$). The MULBY_RR scheduling algorithm consists of three steps (e.g. request, grant, and accept) to resolve the contention. Let assume $N=3$ and $Y=1$.

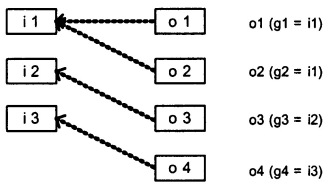


Fig. 4. Grant step

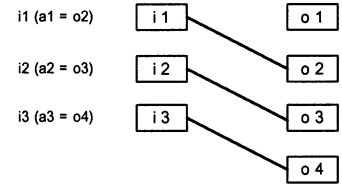


Fig. 5. Accept step

Request step: See Figure.3. Each input sends a request to every output for which it has a queued packet.

Grant step: See Figure.4. If an output receives any requests, it chooses the one that appears next in a fixed, round-robin schedule starting from a previously granted input (pointer). For example, when an input 3 (i3) and an input 1 (i1) are requesting simultaneously for an output 1 (o1), the input 1 gets a grant (g1 = i1) if a previously granted input was input 4.

Accept step: See Figure.5. If an input receives multiple grants, it accepts the one that appears next in a fixed, round-robin schedule starting from a previously accepted output (pointer). For example, when an input 1 (i1) receives two grants from an output 1 (o1) and an output 2 (o2) simultaneously, the input 1 accepts only one grant (a1 = o2) if a previously accepted output was output 1.

However the RRM has a feature of low throughput. The reason for the poor performance of the RRM lies in the rules for updating the granted pointers at the output arbiters that is called 'synchronization of output arbiters.' The MULBY_RR also has

the same problem. The *i*SLIP algorithm is viewed as a good alternative of RRM by reducing the synchronization of the output arbiters. *i*SLIP is identical to RRM except for a condition placed on updating the grant pointers. The *Grant* step of RRM is changed to:

Grant step: If an output receives any request, it chooses the one that appears next in a fixed, round-robin schedule starting from the previously granted pointer. The output notifies each input whether or not its request was granted. The pointer is incremented to one location beyond the granted input if and only if the grant is accepted in following *Accept* step.

We can expect the performance improvement using the *i*SLIP algorithm in MULBY architecture, that is called MULBY_ *i*SLIP. The MULBY_ *i*SLIP is identical to the *i*SLIP in terms of the improved *Grant* step function. Performance evaluation through simulation shows the MULBY_ *i*SLIP performs better than the MULBY_RR.

3 Performance Analysis

The E-PON is able to multiplex several types of traffic onto the uplink and downlink. Considering that the input streams of a queue consist of i packet streams which are in ON state $(s=1)$ at time t. Each stream generates traffic of an arbitrary number J of multiple priority classes. And each source when in state $s\ (s=0,1)$ generates traffic of class $j\ (j=1,2,...,J)$ at rate r_s^j. Hence each of the K sources is specified by $(\mathbf{G};r^1,...,r^J)$ where $r^j=[r_0^j,\ r_1^j]$. We define $\pi_i^j(t,x)\ (0\le i\le K, 1\le j\le J)$ as Cumulative Distribution Function for the priority j cell in the queue at time t, where i streams are in the ON state. In other word it means the probability that the queuing buffer occupancy is less than or equal to $x\ (B_j\le x\le B_{j+1})$ while i streams are in the ON state. $\pi_i^j(t,x)$ can be obtained by utilizing $\pi_i^j(t+\Delta t,x)$. Then we have

$$\pi_i^j(t+\Delta t,x)=\Pr(i-1,i,\Delta t)+\Pr(i+1,i,\Delta t)+\Pr(i,i,\Delta t), \tag{1}$$

here $\Pr(i-1,i,\Delta t)$ is the probability of transition from the state $i-1$ to i at time Δt. The second and the third term are defined by the same token. Each term are defined by following formula

$$\begin{aligned}\Pr(i-1,i,\Delta t)&=[K-(i-1)]\alpha\Delta t\pi_{i-1}^j(t,x)\\ \Pr(i+1,i,\Delta t)&=(i+1)\beta\Delta t\pi_{i+1}^j(t,x)\\ \Pr(i+1,i,\Delta t)&=[1-\{(K-i)\alpha+i\beta\}\Delta t]\pi_i^j(t,x-(\gamma_i^j-c)\Delta t).\end{aligned} \tag{2}$$

Here γ_i^j is the sum of traffic rates of all classes admitted to the buffer at the condition of time t, $N_{on}=i$, and $X_t\in L_j$. Here N_{on} is the number of streams that is in ON state. In other words, by virtue of the implementation of loss priority γ_i^j is the sum of the rates at which cells of classes $1,2,...,J-j$ are generated when $N_{on}=i$. That is, $\gamma_i^j=\sum_{p=1}^{J-j}\lambda_i^p$ $(j=0,1,...,J-1)$, where

λ_i^p = the rate of generation of class p traffic, given $N_{on} = i$

$$= i\, r_1^p + (K-i)\, r_0^p . \tag{3}$$

On substituting for λ_i^p, we obtain

$$\gamma_i^j - c = i\omega^j - c^j , \tag{4}$$

where $\omega^j = \sum_{p=1}^{J-j} (r_1^p - r_0^p)$, and $c^j = c - K\sum_{p=1}^{J-j} r_0^p$. Here c is output rate.

In (1) let Δt go to zero, it represents the following differential function:

$$(\gamma_i^{J-j} - c)\frac{d\pi_i^j(x)}{dx} = [K-(i-1)]\alpha\pi_{i-1}^j + (i+1)\beta\pi_{i+1}^j(x) - [(K-i)\alpha + i\beta]\pi_i^j(x)$$

$$(1 \le j \le J,\ 0 \le i \le K, \pi_{-1}^j(x) = 0, \pi_{K+1}^j(x) = 0) . \tag{5}$$

Define $\boldsymbol{\pi}(x) = \boldsymbol{\pi}^j(x) \equiv [\pi_0^j(x), \pi_1^j(x), ..., \pi_K^j(x)]^T$, then it can be expressed in the following compact matrix form:

$$\mathbf{D}^j \frac{d\boldsymbol{\pi}(x)}{dx} = \mathbf{M}\boldsymbol{\pi}(x) \quad (x \in L_j), \tag{6}$$

where $\mathbf{D}^j = diag\{-c^j,\ \omega^j - c^j, 2\omega^j - c^j, ..., K\omega^j - c^j\}$ and $\mathbf{M} = (K+1)\times(K+1)$ tri-diagonal matrix. Note that the linear growth property of the diagonal elements of $\mathbf{D}^j$ is the drift matrix for the j^{th} buffer occupancy level. Assuming $\gamma_i^j - c$ is not equal to zero for any i, ($0 \le i \le K$). The piece-wise linear form of (6) gives the following structure to the solution: for $j = 0,1,...,J-1$,

$$\boldsymbol{\pi}(x) = \boldsymbol{\pi}^j(x) = \sum_{i=0}^{K} a_i^j \exp(\mathbf{z}_i^j x)\, \mathbf{V}_i^j \quad (x \in L_j) . \tag{7}$$

Here $(\mathbf{z}_i^j, \mathbf{V}_i^j)$ are solutions to J separate sets of eigenvalue problems: for ($j = 0,1,...,J-1$)

$$\mathbf{D}^j \mathbf{z}_i^j \mathbf{V}_i^j = \mathbf{M}\mathbf{V}_i^j, (i = 0,1,...,K). \tag{8}$$

The specific structure of $(\mathbf{D}^j, \mathbf{M})$ has allowed all thee eigenvalues and eigenvectors to be obtained in closed form [12]. That is, vector $\mathbf{z}^j = [z_0^j,\ z_1^j, ..., z_K^j]$ is eigenvalues of $(\mathbf{D}^j)^{-1}\mathbf{M}$, and $\mathbf{V}_i^j$ is eigenvector of $(\mathbf{D}^j)^{-1}\mathbf{M}$. It remains to obtain the coefficients $\{a_i^j\}$ from boundary conditions.

For each buffer occupancy level we separate the aggregate-source states which give a downward drift to the buffer content from those which give an upward drift:

$$E_D^j = \{i \mid \gamma_i^j < c\},\ E_U^j = \{i \mid \gamma_i^j > c\} \quad (j = 0,1,...,J-1) \tag{9}$$

Then the boundary conditions in (9) can be obtained as below

$$\begin{aligned} &\pi_i^0(0) = 0 && (i \in E_U^0) \\ &\pi_i^j(B_{j+1}) = \pi_i^{j+1}(B_{j+1}) && (i \in E_D^j \cup E_U^{j+1},\ j = 0,1,...,J-2), \\ &\pi_i^{J-1}(B) = P_i && (i \in E_D^{J-1}) \end{aligned} \tag{10}$$

where $P_i = {}_K C_i \cdot P_{on}^i \cdot (1-P_{on})^{K-i}$ is the probability that i sources are in ON-state ($s = 1$) and $P_{on} = \alpha/(\alpha+\beta)$ is the probability that a source is in the ON-state. Substitution of the expressions for $\{\pi_i^j(x)\}$ given in (7) yields a system of linear equations in the

number $J(K+1)$ coefficients $\{a_i^j\}$, which have to be solved numerically. One can show that the number of unknown coefficients $\{a_i^j\}$ precisely equals the number of equations, i.e. there are as many eigenvalues as there are feasible states. Hence, the equations suffice to determine the unknown coefficients $\{a_i^j\}$. This completes the description of the procedure for calculating $\boldsymbol{\pi}(x)$. Now the steady state distributions satisfying the boundary conditions of (10) can be used to calculate the throughput for the traffic with different priority classes. The throughput of cells of class $(J-j)$:

$$T^{J-j} = \sum_{i=0}^{K} \lambda_i^{J-j} \pi_i^{j+1}(B_{j+1}) - \sum_{i \in E_U^j \cap E_D^{j+1}} \{(\gamma_i^j - c)(\pi_i^{j+1}(B_{j+1}) - \pi_i^j(B_{j+1}))\}, \tag{11}$$

where $\pi_i^{j+1}(B_{j+1}) - \pi_i^j(B_{j+1}) = \Pr(N_{on}, X = B_{j+1})$ and $0 \le j \le J-2$.

The throughput of class 1, the class of highest priority, is

$$T^1 = \sum_{i=0}^{K} \gamma_i^{J-1} \pi_i^{J-1}(B) + c \cdot \{1 - \sum_{i=0}^{K} \pi_i^{J-1}(B)\}, \tag{12}$$

where $1 - \sum_{i=0}^{K} \pi_i^{J-1}(B) = \Pr(\textit{buffer full})$.

The loss probability for the class is obtained from its throughput. The loss probability for the class: $L^j = 1 - T^j / A^j$. Here $A^j = \sum_{i=0}^{K} \lambda_i^j P_i$ and $1 \le j \le J$.

4 Numerical Results

4.1 Numerical Results for the Multiple Loss Priority Queue

In this section we investigate numerical results obtained from fluid models for the performance analysis of statistical multiplexing of one multicast VOQ with loss priority traffic class. Each individual source is modeled as a Markov modulated fluid source consisting of the ON state and the OFF state. In the ON-OFF process, the ON state is uniformly distributed and the transition between the ON and the OFF states is controlled by a continuous-time Markov chain which determines the rate of fluid generating.

The input traffic of class j (j=1,2) consists of homogeneous independent ON-OFF sources. Figure.6. shows the cell loss probability as a function of the buffer size and the fixed portion of class 1 when offered load is 95% and B_1 is 90%. As expected, the cell loss probability decreases as the buffer size increases and as fixed portion of class 1 decrease. Figure.7. shows the cell loss probability for different threshold values, respectively, where the traffic offered load of the Class 1 is fixed at 30% of the given output rate. This figure shows clearly how the threshold B_1, as it is varied from 50% to 90% of total buffer size, controls the throughput of an associated traffic class. That is,

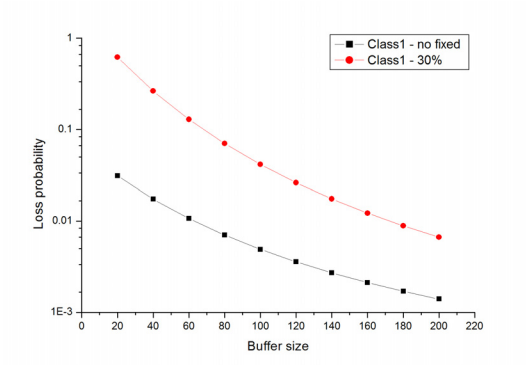


Fig. 6. Loss probability vs. buffer size

the effect of the different threshold values on the performance of cell loss probability is significant for traffic classes.

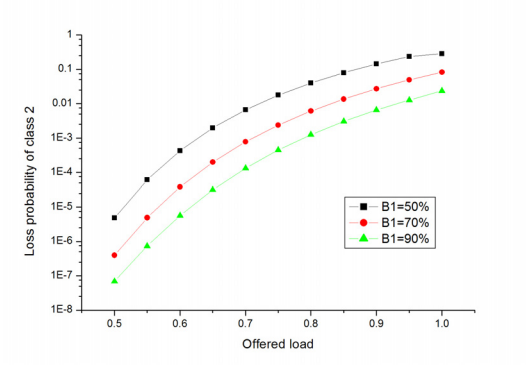


Fig. 7. The effect of threshold on loss probability

4.2 Performance Evaluation of RRM and *i*SLIP Scheduling in MULBY Switch

We evaluate performances of the proposed MULBY architecture using RRM and *i*SLIP scheduling algorithm with the SIM [13]. Current simulation is focusing on average latency that fixed packets have experienced to be served by a switch. We focus on performance of unicast traffic latency variation according to multicast ratio. We compare the MULBY_RR with the round-robin (RR) having a feature of the copy architecture instead of the MULBY architecture. And we compare the MULBY_*i*SLIP with *i*SLIP having a feature of the copy architecture instead of the MULBY architecture.

We setup 8×8 crossbar switching as a basic architecture and multicast bypass output queue number Y=1. In other word, 8 input multicast VOQs competing for one multicast bypass output port when there are waiting packets in the head of all input multicast VOQs. Uniform independent and identically distributed Bernoulli arrivals are used. The percentage of multicast traffic is from 5% to 50%.

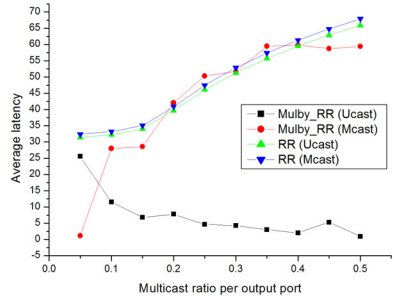


Fig. 8. MULBY_RR : Average latency vs. multicast ratio

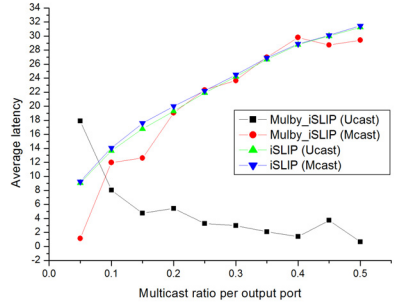


Fig. 9. MULBY_*i*SLIP: Average latency vs. multicast ratio

See Figure.8, and Figure.9. The average latencies of unicast traffic with the MULBY decrease even though the ratio of multicast traffic increases. Meanwhile the average latency of unicast traffic with no MULBY increases according to average latency of multicast traffic when the ratio of multicast traffic increases. These results of the MULBY architecture are coming from no copying mechanism. And we can notice that the MULBY_*i*SLIP performs better than the MULBY_RR due to the improved synchronization scheme of output arbiters as we already expected. If Y increases better performance will be expected.

5 Conclusions

We have proposed a simple switching architecture, called multicast bypass (MULBY) for integrated traffic in E-PON system. A major feature of the MULBY architecture is the elimination of complex scheduling algorithm while supporting integrated unicast and multicast traffic switching. Performance evaluation using well-known RRM and *i*SLIP scheduling algorithm shows that the proposed MULBY architecture improves an average latency of unicast traffic significantly even in the high ratio of multicast traffic. However this architecture is only applicable to broadcasting featured networks

such as E-PON. To support QoS in the MULBY switch we use the multiple loss priority queue for a multicast VOQ while avoiding complex approaches of prioritized scheduling such as extended *i*SLIP, *i*LQF, RPA, and MUCS. We have evaluated cell loss performance due to buffer overflow at input multicast VOQ. The numerical results clearly demonstrate the influence of the threshold and buffer size in the control of loss probability. The MULBY architecture satisfies the simplicity and efficiency, which are the most important criteria in designing of a switch, using the key concept of broadcasting downstream channel in E-PON.

Acknowledgement. This work was supported by the Korea Science Engineering Foundation and OIRC.

References

1. Andrews, M., Khanna, S. and Kumaran, K.: Integrated scheduling of unicast and multicasttraffic in an input queued switch. IEEE Infocom, 1999
2. Kramer, G., Pesavento, G.:Ethernet Passive Optical Network : Building a Next Generation Optical Access Network. IEEE Communications Magazine, pp. 66–73, February 2002
3. McKeown, N.W., Anantharam, V. and Walrand, J.: Achieving 100% throughput in an input-queued switch. Proceedings of IEEE INFOCOM '96, pages 296–302, San Francisco, CA, March 1996
4. Tassiulas L., Ephermides A.: Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. IEEE Trans. On Automatic Control, vol. 37, Dec. 1992, pp. 1936–1948
5. Dai J., Prabhakar B.: The throughput of data switches with and without speedup. IEEE INFOCOM 2000, vol. 2, Tel Aviv, Mar. 2000, pp. 556–564
6. McKeown N.: iSLIP: a scheduling algorithm for input-queued switches. IEEE Trans. on Networking, vol. 7, n. 2, Apr. 1999, pp. 188–201
7. Mekeown N.: Scheduling algorithms for input-queued cell swtches. Ph.D. Thesis, Unv. of California at Berkely, 1995
8. Ajmone Marsan M., Bianco A., Leonardi E., Milia L.: RPA: a flexible scheduling algorithm for input buffered switches. IEEE Trans. on Communications, vol. 47, n. 12, Dec. 1999, pp. 1921–33
9. Duan H., Lockwood J.W., Kang S.M., Will J.D.: A high performance OC12/OC48 queue design prototype for input buffered ATM switches. IEEE INFOCOM'97, vol. 1, Kobe, 1997, pp. 20–28
10. Hayes J., Breault, R. and Mehmet-Ali, M.: Performance Analysis of a Multicast Switch. IEEE Trans. Comm., vol 39, no. 4, pp 581-587, Apr 1991
11. Schwartz, M.: Broadband Integrated Networks. Prentice-Hall, Englewoodcliffs, NJ, 1996
12. Anick, D., Mitra D. and Sondhi, M.M.: Stochastic theory of a data-handling system with multiple sources. Bell System Tech. J., 61, 1982, 1871–1894
13. McKeown, N.W., Prabhakar, B. and Anantharam, V.: SIM: A Fixed Length Packet Simulator. http://klamath.stanford.edu/tools/SIM/, 1999

Differentiation and Interaction of Traffic: A Flow Level Study

Eeva Nyberg and Samuli Aalto

Networking Laboratory
Helsinki University of Technology
P.O.Box 3000, FIN-02015 HUT, Finland
{eeva.nyberg,samuli.aalto}@hut.fi

Abstract. We study what kind of differentiation can be achieved using DiffServ without admission control and using a relative services approach, i.e. the rate of the flow should be in proportion to the contracted rate. We model analytically differentiation mechanisms from such proposals as AF and Simple Integrated Media Access (SIMA) with emphasis on modelling the interaction between various differentiation mechanisms and traffic flows of type TCP and UDP. We first review main results of an earlier and more detailed packet level model and then introduce more abstract flow level models. As a result we show how the flow level models are powerful in explaining how marking thresholds and mechanisms determine the differentiation in a network. The flow level models are able to produce results similar to the packet level models, and can be used as an abstraction of a DiffServ network and give system level guidelines for both network and differentiation design.

Keywords: DiffServ, proportional differentiation, TCP, flow level

1 Introduction

New service concepts have been designed to overcome the problems of converged networks: delivering real time applications in the best-effort Internet. One of these concepts is Differentiated Services (DiffServ). DiffServ is termed a scalable architecture, as inside the network quality treatment is only offered to aggregate classes not to each individual traffic flow. The scalability is achieved at the price of losing strict quantitative guarantees and bounds on the level of service.

For the purpose of our evaluation, we divide differentiation into two categories: *assured services* and *relative services*. In assured services the flow should receive a rate at least equal to the contracted rate, while in relative services the rate of the flow should be in proportion to the contracted rate.

We then study what kind of differentiation can be achieved using DiffServ without admission control. Specifically, previous work on DiffServ modelling and especially on the Assured Forwarding (AF) service proposal [4] have shown that assured services cannot be achieved without admission control, [13], [15], [7], [14], [3] and [12]. We challenge this claim by stating that a more valid service

W. Burakowski et al. (Eds.): Art-QoS 2003, LNCS 2698, pp. 276–290, 2003.

requirement in DiffServ is relative services, where the received rate does not have to be equal to the contracted rate, but where the link bandwidth should be divided in proportion to the contracted rates.

In [13] and [15] the analytical relationship between the contracted rate and the transmission rate of TCP flows was studied. The works include a model for TCP and a model for the priority marker, under the assumption of deterministic loss rates for different priorities, without an explicit buffer model for how the loss probabilities depend on the TCP rates. In [7] a model for assured service that includes a buffer model with different discarding levels for different priorities is studied, but the paper does not include a TCP model nor a marking model for the traffic. A similar model is given in [14] and [8], with some considerations on modelling TCP flows. All the papers, however, still lack combining both a TCP model and a buffer model.

In a previous paper [10] and in [9] we gave a detailed packet level model and a closed loop modelling of the dependency of the flow sending rate on the metering, marking, dropping and scheduling mechanisms. Furthermore, we studied the interaction between various differentiation mechanisms and traffic flows of type TCP and UDP, only done previously in simulation studies by [3] and [12].

In the present paper we continue the analytical modelling of differentiation mechanisms from such proposals as AF and Simple Integrated Media Access (SIMA) [6] with emphasis on modelling the interaction between various differentiation mechanisms and traffic flows of type TCP and non-TCP, e.g. UDP. We first review the main results of the packet level model and then introduce flow level models to compare to our packet level models of [10]. The flow level model is a more abstract model than the detailed packet level model. On the flow level the emphasis is on modelling the interaction between traffic flows assuming some general models of the differentiation mechanisms, while the packet level models were based on detailed models of the mechanisms themselves and the response of traffic flows to the feedback signals of the network.

The paper is organized as follows. In section 2 we discuss the marking, forwarding and discarding mechanisms employed for differentiating traffic. We then review the results for the packet level model, in section 3, and in section 4 present the flow level model. Section 5 compares the models through numerical results and section 6 concludes the paper.

2 Mechanisms for Differentiating Traffic

Throughout the paper we study how delay and priority aggregates can be used to achieve relative differentiation of flows and to protect traffic with different service requirements, e.g. delay and drops, from each other. We identify the main QoS mechanisms used to achieve differentiation in AF and SIMA and formulate a generic DiffServ architecture. We thus do not explicitly model existing DiffServ proposals, but study what mechanisms are needed at different parts of the network to achieve differentiation.

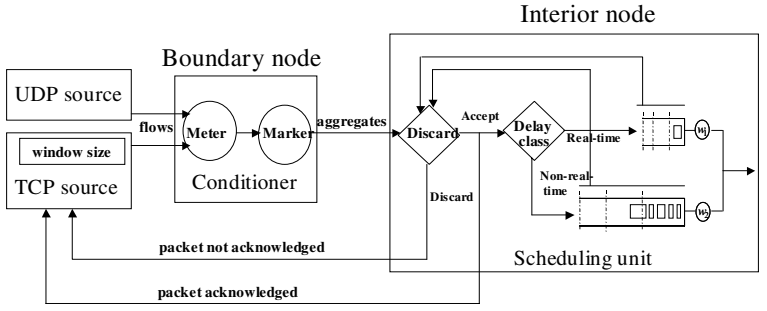


Fig. 1. Components of a DiffServ network including feedback to TCP sources

If the service model is of type assured services, then as long as the flow sends at a rate less than the contracted rate, the flow is marked in profile and thus only two or three priority levels are needed. If the service model is of type relative services, then the priority can be determined based on the ratio of the sending rate to the contracted rate, thus the more priority levels, the more flexible the marking is in terms of covering a wider range of sending rates. The main difference between the service concepts is the division of excess capacity and the division of capacity in overload situations.

We divide the DiffServ mechanisms into two categories: classification and conditioning of flows at the boundary node and forwarding of flow aggregates through packet level mechanisms inside the DiffServ node. Figure 1 summarizes the components, each discussed separately below.

2.1 Reference Model

Consider a DiffServ network with a single bottleneck link, with capacity $C = 1$, loaded by a fixed number of flows. We divide flows into two delay classes, real-time streaming traffic and non-real-time elastic traffic, $d = 1, 2$, respectively. Within each delay class, we have I priority levels, $i = 1, \ldots, I$. Level I refers to the highest priority, i.e. flows at that level encounter the smallest packet loss probability. Each flow is given a weight ϕ that reflects the value of the flow, i.e. the contracted rate of the flow, and flows are grouped according to this weight. There are L^d different groups of flows of delay class d, each group l with a characteristic packet sending rate $\nu(l)$ and weight $\phi(l)$. Let $\mathcal{L}^d$ denote the set of such flow groups. Finally, let $n(l)$ denote the number of flows in any group l. For TCP flows $\nu(l)$ depends on the network state, while for UDP flows $\nu(l)$ is fixed and does not change even if the network congestion level changes.

2.2 Conditioning Flows at the Boundary Nodes

At the conditioner, the packets of a flow are marked and aggregated to priority levels. We adopt the proposal in [6], where the priority level $pr(l)$ of the flow depends on $\nu(l)$ and $\phi(l)$ as follows:

$$pr(l) = \min[i = 1, 2, \ldots, I : \nu(l) \geq t(l, i)]. \quad (1)$$

Corresponding threshold rates would then be

$$\begin{aligned} t(l, 0) &= \infty, \\ t(l, i) &= \phi(l)a(i), \ i = 1, ..., I - 1, \\ t(l, I) &= 0, \end{aligned} \quad (2)$$

where, based on [6], $a(i) = 2^{I/2-i-0.5}$. Note that $a(i-1)/a(i) = 2$ for all i. Note also that we adopt here the relative services approach by defining that a flow sending at its contracted rate receives the middle priority and not the highest priority.

We presented in [10] two **metering and marking alternatives** to mark flows to priority levels: *Token bucket* (TB) and *Exponential weighted moving average* (EWMA). Independent of the marking scheme we say that a flow has priority $i = pr(l)$ as given by equation (1).

TB. The token or leaky bucket principle, referred to, e.g. in the AF specification, is a popular metering principle. For three priority levels the metering and marking may be performed with two token buckets for each group l, with rates $t(l, 1) > t(l, 2)$ and capacities c. Traffic of the flow passes the two buckets sequentially, starting from the bucket $[t(l, 1), c]$, and is marked to highest priority if there are enough tokens in all the buckets, to middle priority if only in the first bucket there were enough tokens and to lowest priority if the traffic is out-of profile already in the first bucket. We illustrated in [11], using simulations, that TB can be modelled as :

– *Per packet marking:* Only those packets of a flow that exceed the marking threshold are marked to the lower priority level. The ratio of packets that have priority i is

$$\frac{\min[\nu(l), t(l, i-1)] - \min[\nu(l), t(l, i)]}{\nu(l)}. \quad (3)$$

Cascaded token buckets split the flow into sub-streams $i = pr(l), pr(l)+1, \ldots, I$, where $pr(l)$ is given by equation (1). All the packets of sub-stream i have the same mark i corresponding to the ratio given in equation (3).

EWMA. The measurement results of previous time instants are taken into account, but exponentially dampened according to a time parameter α and the time interval that has elapsed since the measurement was done. The marking is then performed based on predefined thresholds on the resulting measured arrival rate. The measured bit rate $mbr(k, j)$ of a flow k at the moment of transmission of the j:th packet is given in [6], it can be derived from the traditional discrete EWMA updated at fixed intervals of length $1/\delta$ time slots, with time scale α/δ. The j:th packet of flow $k \in l$ has priority i, if

$$t(l, i) \leq mbr(k, j) < t(l, i - 1).$$

We have illustrated in [11] that EWMA can be modelled as :

– *Per flow marking:* Once the measured rate of a flow exceeds a marking threshold, all packets of the flow are marked to the same priority level.

All the packets of the flow then have the same mark $pr(l)$ given by equation (1).

2.3 Forwarding and Discarding Packets inside the DiffServ Nodes

Bandwidth between delay aggregates must be divided in terms of delay requirements. Bandwidth between priority aggregates, on the other hand, must be divided in terms of packet loss probabilities and result in a division according to flow weights assuming that the relative services approach is adopted. Furthermore, this must be done across delay classes. Low latency classes should not starve bandwidth from the other classes and the elastic delay aggregate with high priority should not be discarded before the low latency aggregate with lower priority.

Discarding. We have a system with two delay classes, serviced by two separate buffers, where the buffer sizes are chosen according to the delay requirements of the delay aggregates. Both buffers have I discarding thresholds, one for each priority class. Consider two different **discarding mechanisms**:

– *Independent discarding:* Each buffer acts locally as a separate buffer, discarding appropriate priority levels according to its buffer content.
– *Dependent discarding:* The content of both buffers determines which priority level is discarded, in both buffers.

For example, when one buffer is heavily loaded and the other is almost empty, *independent discarding* would discard traffic belonging to different priority levels in different buffers, while *dependent discarding* would discard traffic of high priority also from the buffer that is almost empty.

On the packet level, discarding can be modelled using a buffer model. Let m^d denote the number of packets in the buffer of delay class d. The *independent discarding* is implemented by giving, separately for each delay class d, thresholds $K^d(i)$, where $K^d(I) = K^d$, the size of the buffer, and $K^d(0) = 0$. The minimum priority level accepted is then $pr_a^d = f_d(\frac{m^d}{K^d})$. The *dependent discarding*, proposed in [6], is implemented by giving a two-dimensional monotonic function that determines the minimum priority level accepted when in state (m^1, m^2), $pr_a = f(\frac{m^1}{K^1}, \frac{m^2}{K^2})$, e.g.

$$pr_a = a + b \cdot \Big(\frac{m^1}{K^1} + \frac{m^2}{K^2}\Big), \tag{4}$$

$$pr_a = a + b \cdot \sqrt{\Big(\frac{m^1}{K^1}\Big)^2 + \Big(\frac{m^2}{K^2}\Big)^2}. \tag{5}$$

We use equation (5) as a basis for the discarding function in the packet level model. Note that equation (4) is similar to the discarding in a system with only one buffer shared by the two delay classes.

Scheduling. The traffic that is not discarded is placed in either one of the two buffers. We restrict our analysis of scheduling mechanisms to the different weights possible in the Weighted Fair Queuing (WFQ) scheduling principle. The capacity of the link is divided according to predetermined weights w^1 and w^2, with $w^1 + w^2 = 1$, unless one of the buffers is empty, as then the other buffer has use of total link capacity.

3 Packet Level Model for TCP Flows

In [10] we presented a packet level model, where it was assumed that UDP traffic is sent at a constant rate $\nu(l)$, classified to appropriate priorities, resulting in aggregate arrival intensities $\lambda(i)$. The TCP flows on the other hand respond to the congestion in the network, e.g. loss probability feedback signal, by adjusting their sending rate. For elastic traffic we solve the resulting equilibrium intensity using the fixed point method. We briefly present the equations in parameterized form needed to solve the equilibrium rate $\nu(l, x_l)$, for elastic flows $l \in \mathcal{L}^{\text{TCP}}$. Defining $\mathbf{x} = \{x_l, l \in \mathcal{L}\}$, where x_l is an auxiliary variable used for parameterization. For a detailed discussion see [9].

The TCP flows are characterized by their round-trip time. Let $RTT(l)$ denote the round-trip time of flows in group $l \in \mathcal{L}^{\text{TCP}}$. Let $q(l, \mathbf{x})$ denote the packet loss probability of flow group l and $p^d(i, \mathbf{x})$ the packet loss probability for delay class d at priority level i. Assuming that the dynamics of the buffer is faster than that of TCP we can use the steady state TCP throughput expression of, e.g. Kelly [5]. Then the equilibrium throughput $\nu(\mathbf{x}) = \{\nu(l, x_l), l \in \mathcal{L}\}$ amounts to solving the fixed point equation

$$\nu(l, x_l) = \frac{1}{RTT(l)} \sqrt{2 \frac{1 - q(l, \mathbf{x})}{q(l, \mathbf{x})}},\ l \in \mathcal{L}^{\text{TCP}}. \tag{6}$$

where, using the *per flow* marking mechanism,

$$\begin{aligned} q(l, \mathbf{x}) = {} & p^d(\lfloor pr(l, x_l) \rfloor, \mathbf{x})(\lfloor pr(l, x_l) \rfloor + 1 - pr(l, x_l)) \\ & + p^d(\lfloor pr(l, x_l) \rfloor + 1, \mathbf{x})(pr(l, x_l) - \lfloor pr(l, x_l) \rfloor), l \in \mathcal{L}^d, \end{aligned} \tag{7}$$

and using the *per packet* marking scheme,

$$q(l, \mathbf{x}) = \sum_{j=1}^{I} p^d(j, \mathbf{x}) \frac{\min\left[\nu(l, x_l), t(l, j-1)\right] - \min\left[\nu(l, x_l), t(l, j)\right]}{\nu(l, x_l)},\ l \in \mathcal{L}^d. \tag{8}$$

The loss probabilities $p^d(i, \mathbf{x})$ are solved for the one buffer case assuming an $M/M/1/K$ model with state dependent arrival intensities. For the two buffer case they can only be solved numerically assuming a model of two dependent $M/M/1/K$ queues with state dependent arrival intensities, see [9].

The priority level is $pr(l, x_l) = I$ in the range $0 \leq x_l \leq 2$, and from there on the parameterized form is

$$pr(l, x_l) = \begin{cases} -x_l + I + 1 + i, & 2i \leq x_l \leq 2i + 1,\ i = 1, \ldots, I - 1 \\ I - i, & 2i + 1 \leq x_l \leq 2i + 2,\ i = 1, \ldots, I - 2. \end{cases}$$

For $x_l \geq 2I - 1$, we have $pr(l, x_l) = 1$.

Finally, the vector form of the arrival intensity of priority level i is

$$\lambda^d(\mathbf{x}) = \{\lambda^d(i, \mathbf{x}) \mid i = 1, \ldots I\},$$

where the arrival intensity of priority level i is with *per flow* marking

$$\lambda^d(i, \mathbf{x}) = \sum_{l \in \mathcal{L}^d : |pr(l,x_l) - i| < 1} n(l)\nu(l, x_l)(1 - \mid pr(l, x_l) - i \mid). \tag{9}$$

and with *per packet* marking

$$\lambda^d(i, \mathbf{x}) = \sum_{l \in \mathcal{L}^d : pr(l,x_l) \leq i} n(l)(\min[\nu(l, x_l), t(l, i-1)] - \min[\nu(l, x_l), t(l, i)]). \tag{10}$$

4 Flow Level Model

In the previous section we gave a detailed packet level model including scheduling and discarding functions. Let us now study the resulting bandwidth division β assuming that the packet handling in a buffer can be approximated on the flow level as a Processor Sharing mechanism, which divides capacity equally among all flows, $\beta = 1/n$. Then we are able to study the conditioning mechanisms by modelling how the introduction of priority levels i and flow groups l affect the actual bit rate of a flow, $\beta(l, i)$.

The flows with the highest mark I have, in our flow level model, a strict priority over all the other flows. Among these high priority flows, the bandwidth is divided as fairly as possible, i.e. each receives an equal share unless this exceeds the corresponding threshold rate.

4.1 Differentiation Mechanisms and Resulting Bandwidth Shares

One buffer. Assume two TCP flow groups, $\phi(1) > \phi(2)$ and one buffer. The flows or substreams of flows in the same priority level share bandwidth equally up to their threshold rate. Because the threshold rate for group 1 is higher than for group 2, group 1 flows share also among themselves the extra bandwidth left over by group 2 flows in the same priority level. If *per flow* marking is used, then the general rule to determine the bandwidth shares $\beta(l, i)$ for all the $n(l, i)$ flows in group l and at level i is, cf. [1],

$$\begin{cases} \beta(1, i) = \min\{\max\{\dfrac{C(i)}{n(i)}, \dfrac{C(i) - n(2, i)t(2, i-1)}{n(1, i)}\}, t(1, i-1)\}, \\ \beta(2, i) = \min\{\dfrac{C(i)}{n(i)}, t(2, i-1)\}, \end{cases} \tag{11}$$

where $C(I) = C = 1$ and

$$C(i) = \max\{C(i+1) - n(1, i+1)t(1, i) - n(2, i+1)t(2, i), 0\}$$

refers to the remaining capacity for the flows with mark i, $n(i) = n(1, i) + n(2, i)$.

If *per packet* marking is used, we consider substreams of flows instead of entire flows, as a flow has substreams in different priority levels. The general rule to determine the bandwidth shares $\beta(l,i)$ is , cf. [1],

$$\begin{cases} \beta(1,i) = \min\{\beta(1,i+1) + \max\{\dfrac{C(i)}{s(i)}, \dfrac{C(i) - s(2,i)\delta(2,i-1)}{s(1,i)}\}, t(1,i-1)\}, \\ \beta(2,i) = \min\{\beta(2,i+1) + \dfrac{C(i)}{s(i)}, t(2,i-1)\}, \end{cases} \tag{12}$$

where $\beta(l,I+1) = 0$, $C(I) = C = 1$ and

$$C(i) = \max\{C(i+1) - s(1,i+1)\delta(1,i) - s(2,i+1)\delta(2,i), 0\}$$

refers to the remaining capacity for the substreams with mark i, $\delta(l,i) = t(l,i) - t(l,i+1)$, $s(l,i) = n(l,1) + \ldots + n(l,i)$ and $s(i) = s(1,i) + s(2,i)$.

Two buffers with dependent discarding. Assume now that, instead of two TCP flow groups sharing the same buffer, we have one group in each delay class, so that group 1 consists of UDP flows, sending at a fixed rate of $\nu(l)$ and group 2 consists of TCP flows. Assume further that packet discarding in the buffers is dependent. Now the UDP flows in group 1 divide the remaining capacity among themselves, but never receive more than their boundary rate. The TCP flows in group 2 then divide among themselves the capacity that is left over by the flows in higher priority levels and by the UDP flows or substreams of the same priority level. Because we assume dependent discarding, UDP flows of lower priority levels than the TCP flows do not affect the bandwidth share of the TCP flows. The equations are then for *per flow* marking,

$$\begin{cases} \beta(1,i) = \min\{\dfrac{C(i)}{n(1,i)}, t(1,i-1), \nu(1)\}, \\ \beta(2,i) = \min\{\max\{\dfrac{C(i) - n(1,i)t(1,i-1)}{n(2,i)}, 0\}, t(2,i-1)\}, \end{cases} \tag{13}$$

where

$$C(i) = \max\{C(i+1) - n(1,i+1)t(1,i) - n(2,i+1)t(2,i), 0\}$$

as before.

If *per packet* marking is used, we again consider the substreams of flows. The equations are then of the form

$$\begin{cases} \beta(1,i) = \min\{\beta(1,i+1) + \dfrac{C(i)}{s(1,i)}, t(1,i-1), \nu(1)\}, \\ \beta(2,i) = \min\{\beta(2,i+1) + \dfrac{C(i) - s(1,i)\delta(1,i-1)}{s(2,i)}, t(2,i-1)\}, \end{cases} \tag{14}$$

where

$$C(i) = \max\{C(i+1) - s(1,i+1)\delta(1,i) - s(2,i+1)\delta(2,i), 0\}$$

as before. Note that now, $\delta(1,i) = \min\{\nu(1), t(1,i)\} - \min\{\nu(1), t(1,i+1)\}$ and $\delta(2,i) = t(2,i) - t(2,i+1)$.

4.2 Interaction of Priority Levels and Bandwidth Shares

To model the interaction between TCP and the marking mechanisms we need to determine the priority levels $pr(l)$ as a function of the number of flows $n(l)$ in each group l. These priority levels, in turn, determine uniquely the *network state* $\mathbf{n} = (n(l,i);\, l = 1,2;\, i = 1,2,\ldots,I)$, namely

$$n(l,i) = \begin{cases} n(l), & \text{if } i = pr(l), \\ 0, & \text{if } i \neq pr(l). \end{cases}$$

As soon as the network state is known, the bandwidth shares can be calculated from equations (11) - (14), depending on the marking and buffer models used.

Consider a non-elastic CBR source using UDP that does not react to packet losses but tries to maximize the sending rate, then the bandwidth share does not necessarily equal the sending rate. The result will be that the packets have a low priority and high sending rate, but the bandwidth share of the flow may become very small. True TCP, instead, seems to behave smartly tending to optimize, in the first place, the bandwidth share of the flow and not the sending rate. It follows that the priority level of the UDP flows remains constant, whereas TCP sources experiment whether it is more useful to limit the sending rate and have a high priority or increase the sending rate and have a lower priority. More precisely, we set up a game between two flow groups with the following rules:

1. For UDP flows the priority levels are constant and determined from the sending rates. For TCP flows the initial priority levels are the highest ones, $pr(l) = I$.
2. The groups make decisions alternately.
3. Flows belonging to the UDP group do not adjust their sending rate, while at level $pr(l)$, TCP group $l \in \mathcal{L}^{\mathrm{TCP}}$ decides to raise the level by one if $pr(l) < I$ and the resulting bandwidth share $\beta'(l, pr(l)+1)$ is higher than the original one $\beta(l, pr(l))$. If the level is not raised, group l decides to lower the level by one if $pr(l) > 1$ and the resulting bandwidth share $\beta'(l, pr(l)-1)$ is higher than the original one. All the bandwidth shares are calculated from equations (11) - (14).
4. The game ends whenever it is beneficial to all TCP groups to keep the current priority levels. The final bandwidth share $\theta(l)$ for a flow in group l will be $\beta(l,i)$ corresponding to the final level $pr(l) = i$.

In principle, it may happen that the game does not end, but remains in a loop consisting of a number of states. However, our numerical experiments suggest that such a unique final state is always achieved.

5 Numerical Results

With the models presented above, we can now study, first on the packet level, how the sending rate of TCP flows affect the dropping probabilities of the network and vice versa, with the effect of UDP flows in the network taken into account.

We then compare the packet level results, to the flow level model. For the sake of comparison, we will also present flow level results for the case of two TCP flow groups.

In all cases, we evaluate the resulting differentiation under the different mechanism options based on the relationship between the contracted rate, i.e. the weight, of the flow and the share of bandwidth achieved by the flow belonging to a specific aggregate. Our main purpose is to study what mechanisms give elastic traffic, which adjusts its sending rate according to the state of the network, incentives to maximize primarily its bandwidth share and not its sending rate.

5.1 Packet Level

Assume that the flows in group $l = 1$ are UDP flows that have a constant sending rate $\nu(1)$. Then the fixed point equation is only solved for the TCP flows in group $l = 2$. We study the ratio of achieved throughputs $\frac{\theta(1)}{\theta(2)} = \frac{\nu(1)(1-q(1))}{\nu(2)(1-q(2))}$ under three constant sending rates $\nu(1)$, chosen so that under *per flow* marking, the flows are in the highest (solid line), middle (gray line), and lowest (dashed line) priority $\nu(1) = 0.039$, $\nu(1) = 0.079$, and $\nu(1) = 0.16$, respectively. Our basic scenario for three priority levels is the following: $I = 3$, $\mu = C = 1$, $RTT = 1000/\mu$ and $(\phi(1), \phi(2)) = (0.08, 0.04)$, i.e. ratio of weights $k = 2$. Each set of pictures depicted in figures 2 (one buffer) and 3 (two buffers) show the ratio between throughputs of flows as a function of the total number of flows, under the condition $n(1)/n(2) = 1/2$.

One buffer with TCP and UDP traffic. The case of one delay class and one buffer, with size $K = 39$ is depicted in figure 2. When the UDP traffic has the highest priority, there is no visible difference between the marking schemes and as long as $\nu(1) \leq 0.039$, there is no way of limiting its sending rate. For *per flow* marking we notice that the UDP flows do not always receive more bandwidth than the TCP flows. In fact, only when the fixed priority of the UDP flows is optimal in relation to the load of the network, the UDP flows receive more bandwidth than the TCP flows.

Two buffers with TCP traffic and UDP traffic. For the two buffer model we can compare the result with the one delay class case, to see if our conjecture on the difference of marking schemes holds, and how the weighted capacity and discarding affects these results. These results for the two buffer case were presented in [10], and due to lack of space, we only replicate the comparison of discarding methods and marking alternatives for priority queuing ($w^1 = 1, w^2 = 0$) in figure 3. We notice that figure 2 depicting the same scenario for one delay class roughly corresponds to two delay classes with dependent discarding, i.e. in times of low load it is optimal to have low priority and in times of high load to have high priority. The TCP flow is able to adjust its priority to differ from the priority of the aggressive flows and therefore only when the constant priority coincides with the load level of the network, does the aggressive flow get more than the TCP flow.

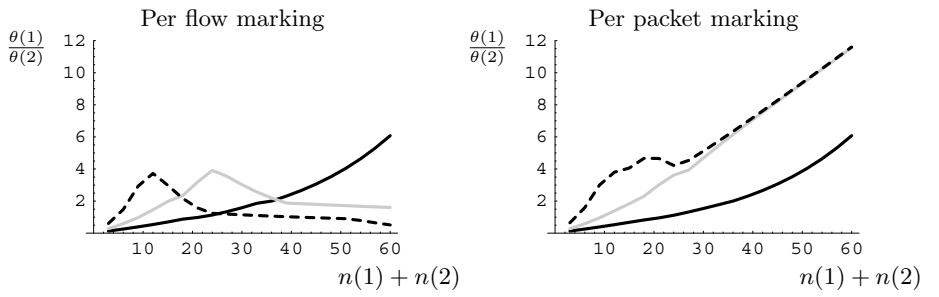


Fig. 2. Packet level model for one buffer with 66% TCP and 33% UDP traffic. UDP flows are in the highest (solid line), middle (gray line), and lowest (dashed line) priority with $\nu(1) = 0.039$, $\nu(1) = 0.079$, and $\nu(1) = 0.16$, respectively.

When priority queuing is used to schedule traffic, only with *per flow* marking and dependent discarding TCP flows can be protected from bandwidth exhaustion by the UDP flows. It was shown in [10] that when independent discarding is used, the results are the same for one priority level and three priority levels, then bandwidth shares can only be controlled by the scheduler weights. However, there is no clear dependency between the ratio of weights of the scheduler (w^1/w^2) and the ratio of weights of the flow $(\phi(1)/\phi(2))$.

5.2 Flow Level

TCP and UDP traffic. The flow level model seems to lead to similar results as the more detailed packet level model in the case of dependent discarding. Note that though in the packet level models it is necessary to assume that packets arrive to the buffer with an arrival intensity of a Poisson process, in the flow level models we only assume that the link divides capacity equally among all flows, with no restrictive assumptions of the packet arrival process.

The packet level and flow level models differ quantitatively, but qualitatively the flow level model also shows that using both *per flow* marking and dependent discarding, gives a powerful incentive for the flows to be TCP friendly [2], whereas the combination of *per packet* marking and dependent discarding, encourages the flows to behave selfishly. Figure 4 depicts this behavior.

As the number of flows increases, i.e. the congestion of the network increases the bandwidth share for UDP flows is independent of their sending rate or priority level when *per packet* marking together with dependent discarding is used. With *per flow* marking and dependent discarding we again deduce that the elastic TCP flows gain from adjusting their rates to the network conditions as long as there are enough priority levels so that UDP traffic is not in the highest priority level.

When there are few users on the link, it is optimal for the streaming traffic to send at lowest priority $\nu(1) = 0.16$. As the number of users increases it would be optimal for the streaming traffic to drop its sending rate to, e.g. $\nu(1) = 0.079$,

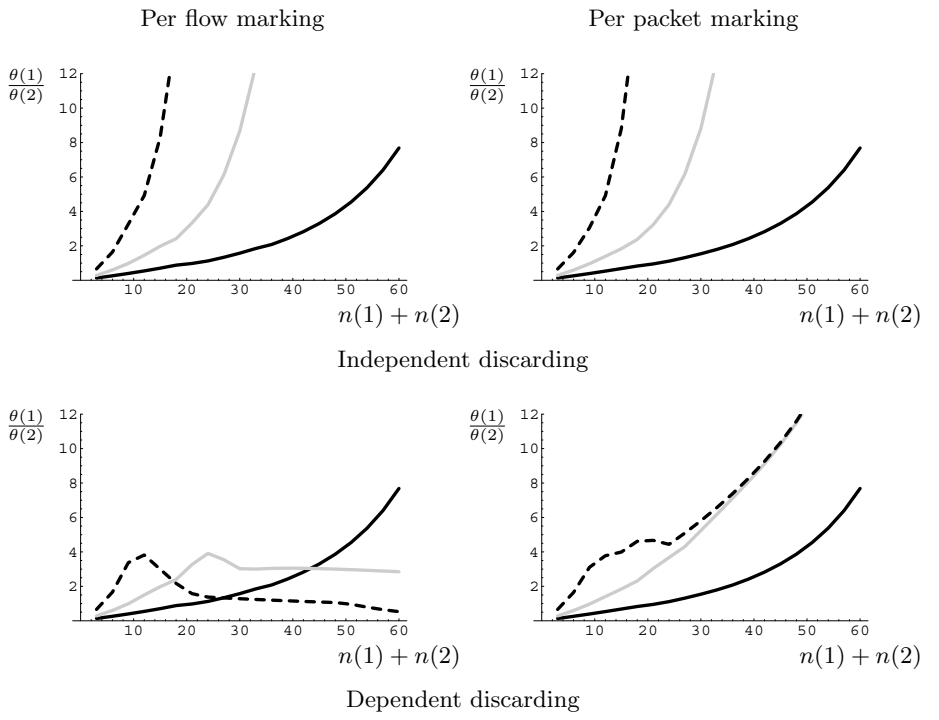


Fig. 3. Packet level model showing the effect of marking and discarding for two buffer priority queueing 66% of the flows are TCP and 33% UDP. UDP flows are in the highest (solid line), middle (gray line), and lowest (dashed line) priority with $\nu(1) = 0.039$, $\nu(1) = 0.079$, and $\nu(1) = 0.16$, respectively.

and as the number of users further increases it would be optimal to send at the highest priority level, with intensity $\nu(1) = 0.039$. Under all other mechanisms, it is always optimal to send as much as possible, even if all or some of the packets are then marked to the lowest priority level.

TCP traffic only. As a final example, let us study, using flow level models, the bandwidth division in a link with only TCP traffic. Figure 5 shows the throughput ratios, when group 1 traffic is also TCP traffic.

In [9], we showed that also for the case of all groups having TCP traffic, the packet level and flow level models give similar results. In a graph, where the number of group 1 flows are on the x-axis and the number of flows of group 2 are on the y-axis, we can divide the plot of throughput ratios into 3–5 different areas depicted in figure 6. The areas show how the TCP flows adjust their sending rate as the number of flows in the system change. They at the same time adjust their priority level to attain a better throughput. With per flow marking it is advantageous for flows in group 1 to move up in priority before flows in group 2. The flow model equations (11)–(14) thus give expressions that explain how, when and to what rate the TCP flows adjust their sending rate.

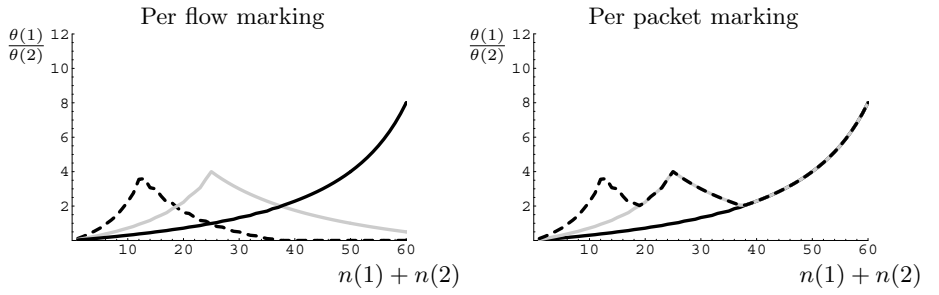


Fig. 4. Flow level model for two buffers with 66% TCP and 33% UDP traffic. UDP flows are in the highest (solid line), middle (gray line), and lowest (dashed line) priority with $\nu(1) = 0.039$, $\nu(1) = 0.079$, and $\nu(1) = 0.16$, respectively.

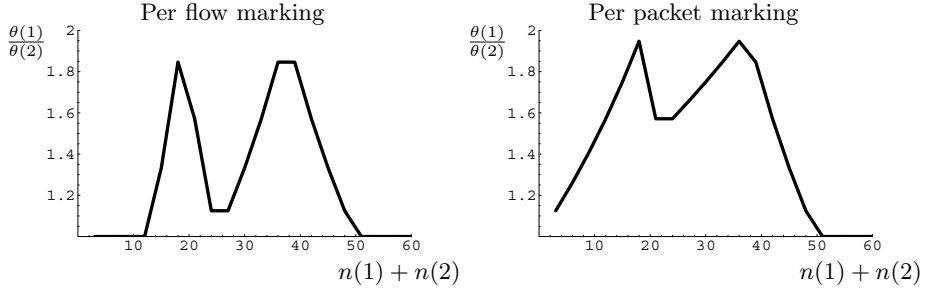


Fig. 5. Flow level model for one buffer and two TCP flow groups, 66% are group 2 and 33% group 1 traffic.

6 Conclusions

Numerical results on both the packet level and flow level show that dependent discarding controls the throughput of non-responsive flows better than independent discarding. This is due to the fact that under the fixed threshold mechanism when the TCP buffer is congested, packets in the UDP buffer are also discarded to alleviate the congestion. Only when the UDP flows are sending at a rate low enough to attain highest priority are they able to use more than their fair share of the bandwidth. By having enough priority levels, i.e. more than three, this effect is also diminished. The use of *per flow* marking and dependent thresholds thus gives a powerful incentive for flows to adjust the sending rate according to the state of the network and thus be TCP friendly.

The paper also showed how the flow level models are powerful in explaining how marking thresholds and mechanisms determine the differentiation in a network. The flow level models are able to produce results similar to more detailed packet level models, and can be used as an abstraction of a DiffServ network and give system level guidelines for both network and differentiation design. Furthermore, being a simpler model, the flow level model can be used for

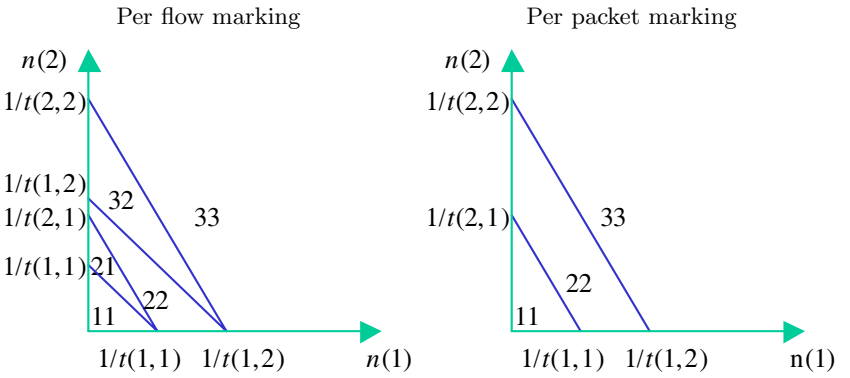


Fig. 6. Resulting priority levels $(pr(1), pr(2))$ as given by the flow level model for two TCP flow groups as a function of the number of flows in the two groups, $n(1)$ and $n(2)$.

more general network topologies, this we leave for an interesting further research topic. Another further research direction is to introduce statistical behavior to the number of flows present in the network.

Acknowledgements. Eeva Nyberg's research is supported by the Academy of Finland and in part by grants from the Nokia and TES foundations.

References

1. S. Aalto and E. Nyberg. Flow level models of diffserv packet level mechanisms. In *Proceedings of the Sixteenth Nordic Teletraffic Seminar, NTS 16*.
2. S. Floyd and K. Fall. Promoting the use of end-to-end congestion control in the Internet. *IEEE/ACM Transactions on Networking*, 7(4):458–472, August 1999.
3. M. Goyal, A. Durresi, R. Jain, and C. Liu. Performance analysis of Assured Forwarding. IETF Draft October 1999.
4. J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. *Assured Forwarding PHB Group*, June 1999. RFC 2597.
5. F. Kelly. Mathematical modelling of the Internet. In *Proc. of Fourth International Congress on Industrial and Applied Mathematics*, pages 105–116, 1999.
6. K. Kilkki. Simple Integrated Media Access.
Available at http://www-nrc.nokia.com/sima, 1997.
7. M. May, J. Bolot, C. Diot, and A. Jean Marie. Simple performance models for Differentiated Services schemes for the Internet. In *Proceedings of IEEE INFOCOM*, pages 1385–1394, March 1999.
8. L. V. Nguyen, T. Eyers, and J.F. Chicaro. Differentiated service performance. In *Procedings of Fifth IEEE Symposium on Computers and Communications ISCC 2000*, pages 328–333, 2000.
9. E. Nyberg. How to achieve fair differentiation: Relating flow level qos requirements to diffserv packet level mechanisms. Licentiate thesis, Helsinki University of Technology, Networking Laboratory, September 2002.

10. E. Nyberg and S. Aalto. How to achieve fair differentiation. In *Networking 2002*, Pisa, Italy.
11. E. Nyberg, S. Aalto, and R. Susitaival. A simulation study on the relation of diffserv packet level mechanisms and flow level QoS requirements. In *Intl. Seminar, Telecommunication Networks and Teletraffic Theory*, St. Petersburg, Russia, 2002.
12. P. Pieda, N. Seddigh, and B. Nandy. The dynamics of TCP and UDP interaction in IP-QoS Differentiated Service networks. In *3rd Canadian Conference on Broadband Research (CCBR)*, November 1999.
13. S. Sahu, P. Nain, D. Towsley, C. Diot, and V. Firoiu. On achievable service differentiation with token bucket marking for TCP. In *Proceedings ACM SIGMETRICS'00*, pages 23–33, June 2000.
14. S. Sahu, D. Towsley, and J. Kurose. A quantitative study of differentiated services for the Internet. In *Proc. IEEE Global Internet'99*, pages 1808–1817, 1999.
15. I. Yeom and A. L. N. Reddy. Modeling TCP behavior in a differentiated services network. *IEEE/ACM Transactions on Networking*, 9(1):31–46, 2001.

Application-Oriented Evaluation of Measurement Estimation

Adam Wierzbicki[1] and Lars Burgstahler[2]

[1] Polish-Japanese Institute of Information Technology
Chair of Computer Networks, Warsaw, Poland
adamw@icm.edu.pl
[2] Institute for Communication Networks and Computer Engineering
University of Stuttgart, Germany
burgstahler@ikr.uni-stuttgart.de

Abstract. The design and operation of telecommunication networks often requires measurement-based decisions. Examples of such decisions are QoS-based routing, where the measurement of the utilized link capacity influences the path selection, or cache location that uses measurements of average TCP throughput. However, the variability of measurements of network conditions makes it difficult to use them for such decisions. In QoS routing, decisions need to be stable in order to avoid packet reordering; in cache location, a small change in the measurements can lead to a completely different location decision. Therefore, estimation algorithms are used to smooth the measurements. The quality of decisions based on estimations depends on how close the estimation is to real conditions, and on how variable it is. There is a trade-off between these two objectives that makes it difficult to choose an appropriate estimation method. In this paper, an approach that uses multi-criteria analysis to evaluate the quality of an estimation algorithm is introduced. It is shown how different applications require different criteria and how the preferences of these criteria can be set by the algorithm designer. Some example estimation algorithms are evaluated on synthetic and real traffic traces using the proposed approach.

1 Introduction

Measurements of network conditions can be of great use in many areas of telecommunications. For some functionality, such as congestion control, such measurements are essential. On the other hand, it is not always possible to use the measurements without any further processing (or estimation). The TCP protocol relies on measurements, yet some of the most important improvements to the protocol were made when better methods of measurement estimation were utilized. TCP congestion control has a single objective: to adapt as fast as possible to the available bandwidth. In other applications, the use of measurements can have more than one objective.

An example application is QoS-based routing, e.g. in a DiffServ network where routers measure the available capacity on the links, possibly separately

W. Burakowski et al. (Eds.): Art-QoS 2003, LNCS 2698, pp. 291–304, 2003.

for each service class. With this information the router then can select e.g. the shortest path for high priority real time traffic and the widest path — which could be longer — for low priority bulk traffic (i.e. different routing algorithms would be used). An alternative usage of the information is load balancing for increased utilization of a network if different paths are available between a source-destination node pair.

However, load-based routing should not lead to frequent switching between different paths. Otherwise, packets would arrive out-of-order and provoke retransmission, decreasing thereby the performance of the network on the transport layer. Hence, the router needs to use an estimation that has two objectives: efficient path selection and routing stability.

Another example is the location of Internet caches and design of Content Delivery Networks (CDN). These locations can be planned in order to reduce the average delays of users. To do so, network management measures the average available TCP throughput on the paths from the clients to the potential cache locations and from there to the origin servers [1]. However, available TCP throughput is extremely variable, and the location decisions are very sensitive to small changes in the input data. Again, it is important to stabilize the measurements before they can be used to make decisions.

An estimation algorithm can simplify decision making by reducing the variability of the measurements. However, the quality of such a decision depends strongly on the quality of the estimation. There can be different requirements that estimation algorithms must meet, and they can sometimes be contradictory. Stability and accuracy are examples of such requirements. A stable estimation makes decisions very easy, but the estimates may be far too inaccurate to allow for a correct decision.

For applications that have different (and contradictory) requirements that the estimation algorithms must meet, we propose to use *multi-criteria analysis* to evaluate the quality of the algorithms. Multi-criteria analysis allows for an objective evaluation of different estimation algorithms that will take into account all diverse application requirements. As different applications (e.g., cache location, bandwidth measurement for routing) have different requirements, different criteria of algorithm performance (e.g., stability, accuracy, delay) can be selected. The criteria can be combined in a way that takes into account the preferences of the algorithm designer. The subject of this paper is the development of evaluation methods for the design of estimation algorithms that must satisfy diverse requirements, and the use of these to evaluate estimation algorithms on traces of network conditions (real and synthetic).

The remainder of this paper is organized as follows: Section 2 gives an overview on some estimation algorithms we have evaluated so far. Section 3 focuses on multi-criteria approaches for the evaluation of these algorithms. Section 5 describes the scenario we used for the evaluation, i.e. the traffic and the parameterization of the algorithms. Further, the selected criteria for two different applications are presented. In Sect. 6 we show and discuss the results and Sect. 7 gives a conclusion and a further outlook.

2 Estimation Algorithms

Although estimation algorithms change the shape of the measurement, the main purpose is to adapt the measurement to the needs of the application, i.e. calculating reasonable and usable values or to forecast values. The result of an estimation is one single new value y_t based on a series of measured values x_t and sometimes older estimates y_{t-i}. In contrast, smoothing only focuses on the form of a curve and not the meaning of the measurement. A smoothing algorithm modifies the data set to make it smooth and nearly continuous and to remove or to diminish outlying points. Estimation and smoothing usually add extra delay to the measurement.

2.1 Median Filter

The *median* is the value in the center of an ordered list. Usually the median is used as a *running median*, where a window of the size $L = 2N+1$ (N is the order of the median) is shifted continuously on the measurement. The values within the window are ordered and the $(N+1)^{st}$ value is the median. It is assigned to the time when the value in the middle of the window was measured. Since the median can only be determined, when the last value of the window is measured, a delay of N samples is always introduced.

A median filter has three important characteristics: First, for a given N, all peaks with a length $l_p \leq N$ are completely removed. Peaks with a larger length will not be altered, not even by multiple application of the same filter [2]. Secondly, median filters preserve discontinuities, provided the discontinuity exceeds some minimum duration. This minimum duration refers directly to the first characteristic, i.e., it has to last for at least $N+1$ samples. Thirdly, median filters follow polynomial curves rather tight. This is important when polynomial (e.g., linear) smoothing is applied before the estimation [3].

2.2 Exponential Moving Average

The general *exponential moving average (EMA)* of order N is described as follows:

$$y_t = \alpha_0 \cdot y_{t-N} + \alpha_1 \cdot y_{t-N+1} \cdots + \alpha_{N-1} \cdot x_t, \quad \text{with} \sum_{k=0}^{N} \alpha_k = 1 \tag{1}$$

However, in most cases the most basic form, a 1^{st}-order EMA, is used. The formula then is simplified to $y_t = (1-\alpha) \cdot y_{t-1} + \alpha \cdot x_t$. EMAs behave like a low pass, i.e., they filter high frequencies. In our case, a high frequency corresponds to a high variability within a short time. The responsiveness of the EMA is controlled by the weights $\alpha_k (0 \leq k < N)$. In the simple case of a 1^{st}-order EMA, a high α leads to a fast reaction to changing measurement, but the estimation follows the measurement too close. A low α leads to a more stable behavior but also to a larger delay and thus the accuracy of the estimation gets worse.

The main disadvantage of the basic EMA is its disability to adapt to a changed characteristic of the measured values. However, there are a number of dynamic EMAs where the weight is calculated dynamically depending on e.g., the time distance between the measured values or the change of the values themselves. An overview of dynamic EMAs can be found in [4].

2.3 Discrete Intervals with Hysteresis

Any estimation filter can be stabilized further by a simple estimation algorithm that frequently reduces the standard deviation. The algorithm divides the range of possible measurement results into *intervals* (not necessarily of the same size). In its simplest form, the algorithm replaces the estimate with the midpoint of the interval in which the median lies. Therefore, the width of the intervals determines the stability and error of the algorithm: by taking a wide enough interval (from min. to max. measurement), the estimate could be reduced to a straight line (a constant estimate). The desired width of the interval has to be determined by observation of the values of the estimate.

However, the described simple form of the algorithm is not sufficient. If the estimate lies close to the border of two intervals, even very small variations can change the interval and thus the value of the estimate. In such a case, the algorithm would increase the standard deviation. To avoid this behavior, *hysteresis* can be used. To change the interval, the estimate has to cross the border by a sufficient value. The value can be specified by a proportion of the width of the interval. For example, if the hysteresis parameter is equal to 0.3, and the estimate increases, the estimate has to exceed the lower border of an interval of width h by at least $0.3h$ in order to change the interval. A similar rule is applied when the estimate decreases.

Discrete intervals with hysteresis can be used for every estimation algorithm. For our evaluations, we used this combination for a median filter (*m9*) as well as for exponential moving averages with a weight $\alpha = 0.3$ and $\alpha = 0.7$ (*ema3DF* and *ema7DF*).

2.4 Smoothing

In our evaluation *hanning* was used as proposed for signal processing in [3]. An apodization function is used on a window of size L to calculate the smoothed value which is associated to the time in the middle of the window, thus the function adds a delay of N samples ($L = 2N + 1$). The apodization function for the hanning window is described by

$$\mathrm{Hn}(i) = \frac{1}{2}\left[1 - \cos\left(\frac{2i\pi}{L-1}\right)\right], \quad \text{with } 0 \leq i < L-1 \tag{2}$$

and therefore the smoothed value m_t for the time t results to

$$m_t = \sum_{k=0}^{L} \mathrm{Hn}(k) \cdot x_k \tag{3}$$

where x_k is the k^{th} measured value in the smoothing window.

2.5 Filter Chains

The concatenation of a smoothing and an evaluation algorithm can already be considered as a *filter chain*. Sometimes it can also be useful to chain several estimation algorithms. As described in [2], applying the same median filter multiple times on a measurement can help to remove variabilities completely up to a certain degree. Such a stable estimate is called a *root* to this filter. By chaining different filters, their different characteristics can be combined.

3 Evaluation Methods

The two main desirable characteristics of the estimations of measured data are: *stability* and a *good fit*. The reason for using estimations is to decrease the variability of the original measurement. On the other hand, the least variable estimate is a constant estimate, which is also unsatisfactory. Therefore, it is required that an estimate should fit the original measurement fairly well.

As was mentioned before, there exists a tradeoff between these two objectives. However, the two characteristics cannot be measured on a common scale. It is entirely unclear how much of the fit can be sacrificed to decrease the variability by some amount. To answer these questions, it would be useful to have a scale of comparison of the two characteristics. Yet, if one estimation has a better fit than another, but a worse variability, which of the two should be chosen, and consequently, which estimation algorithm is better?

These questions have been the subject of *multi-criteria decision analysis and support*. This area of operations research deals with the process of decision making and optimization in the case when there are many conflicting and incomparable objectives (called *criteria*). The methods of ranking outcomes (in our case, estimations) that have been developed by multi-criteria decision analysis could therefore be used for the evaluation of estimation algorithms that require stability and good fit.

Note, that the described evaluation methods are not to be used in a running system (e.g. a router). They evaluate the quality of an estimation algorithm off using traffic traces. Whether the estimation algorithm itself influences the performance of the system (e.g. in a router) or not (location of the CDN) is out of the scope of this paper. However, if the complexity of an estimation algorithm is important (e.g. for a router), it can be used as an additional characteristic. Yet we will not discuss this characteristic in the following.

3.1 Criteria for Evaluation of Estimation Algorithms

The first step of multi-criteria decision analysis is to identify the criteria that should be used to evaluate outcomes. Below, seven criteria for the evaluation of an estimate will be introduced. Each criterion will be expressed by a formula that uses the following notation: x_t is the measurement made at time t; y_t is the estimate available at time t (not using any measurement that has been made

later than t, which means, that the estimate is generally delayed); N is the number of measurements; $\overline{x}$, $\overline{y}$ are the mean values of the measurement and the estimate, and s_x, s_y are the standard deviations of the measurement and the estimate respectively.

Consider first the variability of an estimate. The simplest (and most intuitive) criterion of this characteristic is

1. the *standard deviation* of the estimate, s_y.

The quality of the fit of an estimate is harder to evaluate, since there is no single intuitive candidate for a measure of this characteristic. Several different criteria could be used:

2. the *mean absolute error* (MAE) of the estimate:

$$\mathrm{MAE} = \frac{1}{\mathrm{N}} \sum_{\mathrm{t}} |\mathrm{y_t} - \mathrm{x_t}|. \tag{4}$$

3. the R^2 *measure* of the estimate:

$$R^2 = 1 - \frac{\sum_t (y_t - x_t)^2}{\sum_t (x_t - \bar{x})^2} \quad (\text{and } 0 \text{ if } s_x = 0). \tag{5}$$

4. the *correlation coefficient* (CC) of the estimate and the measurement:

$$\mathrm{CC} = \frac{1}{\mathrm{N} \cdot \mathrm{s_x} \cdot \mathrm{s_y}} \sum_{\mathrm{t}} (\mathrm{y_t} - \overline{\mathrm{y}})(\mathrm{x_t} - \overline{\mathrm{x}}) \tag{6}$$

5. the Pearson correlation coefficient (PCC) of the estimate and the measurement

All of these criteria are related to the "global" fit of the estimate to the measurement. However, in practical applications, it is often more important how well, in the worst case, the estimate performs as a forecast of the measurement, than how well it fits on the average. To express this type of fit, a measure of forecast quality of the estimate could be used. Such measures take as their parameters the desired length of a forecast. Since we can update estimates (and forecasts) continuously by taking new measurements, the length of the desired forecast is not dictated by a measurement period, but should be determined by the application. As an example, let the desired length of the forecast be three observation periods, $h = 3$. Since the application requires that the forecast be good even in the worst case, and since the measures of forecast quality are minimized, the criterion should be the maximum of a forecast quality measure. Two such measures have been taken into consideration, both independent of the scale of the measurement and estimate:

6. the *maximum of the mean absolute percentage error of the forecast* (max. MAPEF), taken over any time period where the forecast can be compared to the measurement:

$$\mathrm{MAPEF_{max}} = \max_{\mathrm{t}} \frac{1}{(\mathrm{h}+1)} \sum_{\mathrm{i=t}}^{\mathrm{t+h}} \left| \frac{\mathrm{y_t} - \mathrm{x_t}}{\mathrm{x_t}} \right|. \tag{7}$$

7. the *maximum of the Theil inequality coefficient* (max. TIC) of the forecast, taken over any time period where the forecast can be compared to the measurement:

$$\mathrm{TIC}_{\max} = \max_t \frac{\sqrt{\sum_{i=t}^{t+h}(y_t - x_t)^2}}{\sqrt{\sum_{i=t}^{t+h} x_t^2} + \sqrt{\sum_{i=t}^{t+h} y_t^2}}. \tag{8}$$

Of these seven criteria, only one is used for the evaluation of estimate variability, while all others are different measures of estimate fit. To evaluate an estimation algorithm, not all criteria have to be used. Instead we can limit ourselves to two criteria, one of variability and the other one of fit.

3.2 Choosing Good Estimations Using Multi-criteria Methods

Given a set of estimation algorithms, a measurement, and a set of criteria Q_i for evaluation, it should be possible to select algorithms that produce good estimations. The concept of *Pareto-optimality* (in multi-criteria decision analysis; game theory uses the term *Nash equilibrium*) can be used for that purpose. For simplification, let all criteria be minimized. An estimation y' is Pareto-optimal if no other estimation y'' has values of all criteria smaller than the values of the same criteria for estimation y'. Several of the estimations for a given measurement can satisfy this condition. However, an estimate that is not Pareto-optimal need no longer be taken into consideration.

To choose one estimation from the Pareto-optimal estimations it is necessary to use an *objective function* (OF) that combines all criteria. Such an objective function must express the preferences of the algorithm designer, who should be able to specify what levels of the criteria he requires. To do so, the range of variability of any criterion should be known. This is also a requirement for scaling the criteria, so that they may be aggregated without bias.

To establish a common scale of comparison of all criteria, it is necessary to find "best" and "worst" values of any criterion. These values can be estimated in the following way: the "best" value of the standard deviation of an estimate is clearly 0 (achieved by a constant estimate). The "worst" value is the standard deviation of the original measurement. For the measures of fit, the situation differs. The "best" value is usually 0 (with the exception of R^2, hence, $1 - R^2$ will be used instead). The "worst" value can be 1 for some of the measures, such as the correlation coefficient or the Theil inequality coefficient, or a larger value. For the other fit criteria, the "worst" value can be obtained by considering a constant estimate equal to the mean or median of the measurement. Let the "best" and "worst" values of the criteria be denoted by Q_i^u and Q_i^n, respectively, where $i \in 1, \ldots, k$, and k is the number of all criteria (in our case, 6). Recall that for minimized criteria, $Q_i^u \leq Q_i^n$.

The algorithm designer can specify for each criterion a level that would satisfy him completely (an aspiration level, Q_i^a), and a level that should be achieved at least (a reservation level, Q_i^r). For minimized criteria, $Q_i^u \leq Q_i^a < Q_i^r \leq Q_i^n$ such an approach follows the multi-criteria methodology called the *reference-point*

method [5]. These levels could be specified in absolute terms or in terms of relative distance between the "best" and "worst" values. Once the levels are specified, each of the criteria can be scaled using a so-called *achievement function*:

$$\sigma_i = \begin{cases} 1 + \alpha \cdot \frac{Q_i^a - Q_i}{Q_i^a - Q_i^u}, & Q_i^u \leq Q_i < Q_i^a \\ \frac{Q_i^r - Q_i}{Q_i^r - Q_i^a}, & Q_i^a \leq Q_i < Q_i^r \\ \beta \cdot \frac{Q_i^r - Q_i}{Q_i^n - Q_i^r}, & Q_i^r \leq Q_i \leq Q_i^n \end{cases} \tag{9}$$

The achievement function is piecewise linear, and the coefficients α and β can be obtained from the reservation and aspiration levels. The slope of the achievement function in the interval $[Q_i^a, Q_i^r]$ is known (it is equal to $\frac{1}{|Q_i^a - Q_i^r|}$). β can be chosen to be twice that slope, and α to be half the slope.

Now, the objective function of the reference-point method can be expressed using the achievement function:

$$Q^{RP} = \min_i \sigma_i(Q_i, Q_i^a, Q_i^r) + \epsilon \sum_i \sigma_i(Q_i, Q_i^a, Q_i^r) \tag{10}$$

The parameter ϵ is a small, positive number, usually 0.01. If ϵ was too large, the reference-point method would become similar to a simple sum. The value of the objective function (10) can be negative (since the achievement functions can be negative). By changing the parameters Q_i^a and Q_i^r, any solution can be chosen as the optimal solution — in other words, the objective function is sufficiently flexible to express any preferences of the algorithm designer.

4 Preferences of Algorithm Designers

Two applications have been considered for the evaluation of estimation algorithms. The first is the use of estimations for the design of CDNs, which requires stable estimations of available throughput. The second is the use of estimations for QoS-based routing with the purpose of load-balancing. This application requires more accurate estimations than the first one.

The preferences of the algorithm designer for the first application value stability over accuracy. However, it is clearly unreasonable to use an algorithm that provides inaccurate estimates, if the accuracy can be improved without sacrificing stability. The first criterion is standard deviation: $Q_1 = s_y$. For the expression of accuracy, the algorithm designer chose $Q_2 = 1 - \mathrm{CC}$. The aspiration and reservation levels were chosen with respect to the relative distance between the "best" and "worst" values. Let the distance between the "best" and "worst" values for criterion i be $\delta_i = |Q_i^n - Q_i^u$. The aspiration and reservation levels for Q_1 were chosen as $Q_1^a = Q_1^u + 0.2\delta_1$, and $Q_1^r = Q_1^u + 0.95\delta_i$. The aspiration and reservation levels for Q_2 were chosen as $Q_2^a = Q_2^u + 0.5\delta_2$, and $Q_2^r = Q_2^u + 0.99\delta_2$.

The preferences of the algorithm designer for the second application value accuracy over stability. The first criterion is standard deviation: $Q_1 = s_y$. For the expression of accuracy, the algorithm designer chose $Q_2 = 1 - \mathrm{CC}$ and $Q_3 =$

$\text{MAPEF}_{\max}$. The aspiration and reservation level was also chosen with respect to "best" and "worst" values. For Q_1, they were chosen as $Q_1^a = Q_1^u + 0.8\delta_1$ and $Q_1^r = Q_1^u + 0.99\delta_1$. For Q_2, they were chosen as $Q_2^a = Q_2^u + 0.1\delta_2$ and $Q_2^r = Q_2^u + 0.25\delta_2$. For Q_3, they were chosen as $Q_3^a = Q_3^u + 0.2\delta_3$ and $Q_3^r = Q_3^u + 0.3\delta_3$.

The selection of appropriate aspiration and reservation levels by the algorithm designers was an iterative process. The algorithm designers started with some initial values, then explored the available solutions through the ranking produced by the objective function, and next could modify their values and iterate. During this process, the algorithm designers were able to learn about the expression of their preferences using aspiration and reservation levels.

5 Time Series for Evaluation

5.1 The Synthetic Traces

The synthetic traffic was generated with the help of the IKR simulation library (IKRSimLib). A variable number of traffic generators was used to generate self-similar traffic for different traces. The link's bandwidth was limited so that overload was possible. The burst length was Pareto-distributed ($\alpha = 1.6$, min. burst length 3750 byte). The resulting traffic was measured during 4000sec with a granularity of 1 sample per 5 seconds (i.e., there are 800 samples). The trace corresponds to what a router has to process to calculate link state information based on by-passing traffic. The traces will be referred to as *synthetic trace* synt−1 (6 generators), synt−2 (8 generators) and synt−3 (16 generators).

Table 1. Parameterization of the filter chains

Name	Type	Parameters		
		EMA	Hanning	Median
emaX	EMA	$\alpha = 0.X$		
emaXm3	EMA/Median	$\alpha = 0.X$		1^{st}-order
emaXDF	EMA, Disc. Intervals w. Hysteresis	$\alpha = 0.X$		
m9	Median			4^{th}-order
m9DF	Median, Disc. Intervals w. Hysteresis			4^{th}-order
hXmY	Hanning/Median		$w = X$	$(\frac{Y-1}{2})$-order
h3m3m3	Hanning/Median/Median		$w = 3$	1^{st}-order
m3h3m3	Median/Hanning/Median		$w = 3$	1^{st}-order
c	Mean	of whole measurement sample		

5.2 Traces of Measurements of Internet Conditions

The real traffic traces are measurements made by the NIMI infrastructure [6] (courtesy of Vern Paxson and Andrew Adams). The traces contained measurements of the available TCP throughput, which is measured using the treno tool [7]. The measurements were made daily during March and April, 2001. Due to

some irregularities in the measurement time, the traces do not contain measurements made at an exact point in the day; however, the measurements were chosen in such a way that all measurements were made within working hours (using local time at the starting point of the measurement). The traces will be referred to by the start-, and end-point of the measurement, e.g. *Grouse-Cern*.

5.3 Evaluated Estimation Algorithms

To evaluate the estimation algorithms described in Sect. 2, different combinations were used on all of the traces. Table 1 shows the parameterization of the different filter chains.

6 Results

The estimation algorithms were evaluated on all available traces using two sets of preferences for the two applications described above. Naturally, the outcomes of the two evaluations were different; this difference may be used to illustrate the tradeoffs introduced by certain algorithms. First, the results of the evaluation of estimation algorithms for the first application (CDN design) will be shown.

The first stage of multi-criteria analysis is the selection of Pareto-optimal solutions. This process can be demonstrated on the example of estimations obtained by the considered algorithms on one of the traces (for example, synt−3). Figure 1 shows all estimations on a plane with Q_1 plotted on the x axis and Q_2 on the y axis. Of the 14 estimations, only 5 are Pareto-optimal: c, $ema3DF$, $ema3$, $ema7DF$ and $ema7$.

Another way of evaluating the estimations is to rank them using the objective function. Such a ranking can be shown for our example on Fig. 2. On the figure, the standard deviation increases from right to left. It can be seen that a reduction of the standard deviation always increases the objective function, under the chosen preferences.

This method has the advantage that it chooses one Pareto-optimal solution (the best in the ranking) out of all. The objective function of the reference point method always chooses a Pareto-optimal solution as the best one; however, it is possible, that non-Pareto-optimal solutions rank higher than Pareto-optimal ones. On the figure, it can be seen that the objective function indeed selects the Pareto-optimal solution c as best, but the third solution in the ranking, $m9DF$, is not Pareto-optimal. Such a situation occurred infrequently on the evaluated traces (for 10 traces, it occurred in 3 cases), but it is necessary to remove such suboptimal solutions from the ranking. By such a procedure it is possible to obtain a ranking of Pareto-optimal solutions, which for our example is shown on Table 2.

From the ranking it can be seen that the constant estimate c, equal to the mean of all values in the measurement sample, was always chosen as the best estimation. This can be explained by the preferences of the algorithm designer,

who valued stability over accuracy. The constant estimate has a standard deviation of 0 (equal to q_1^U) and a correlation of 0 that results in $Q_2 = 1$ (halfway between q_2^U and Q_2^N). To beat it under the specified preferences, a more accurate estimation algorithm would have to be 100% accurate ($Q_2 = 0$) and have a standard deviation that is at least twice smaller than the standard deviation of the measurement, which is impossible. However, the constant estimate is not a practical estimation algorithm, since it would take very long to obtain the necessary data; also, it is not robust to trends or structural changes in the measurement. This would have been apparent if the algorithm designer would have chosen a different criterion of accuracy, for example MAE.

The ranking also shows that the second-best choice is usually an algorithm that was combined with the discrete intervals with hysteresis. The reason for this can be explained on Fig. 1. The result of combining the algorithms *ema*3 and *ema*7 is a reduction of the standard deviation at the expense of Q_2. This is not always the case (sometimes the standard deviation increases, if the resulting estimation consistently over-, or underestimates the measurement), but usually the discrete intervals with hysteresis allow to trade off accuracy for stability, which was preferred by the algorithm designer.

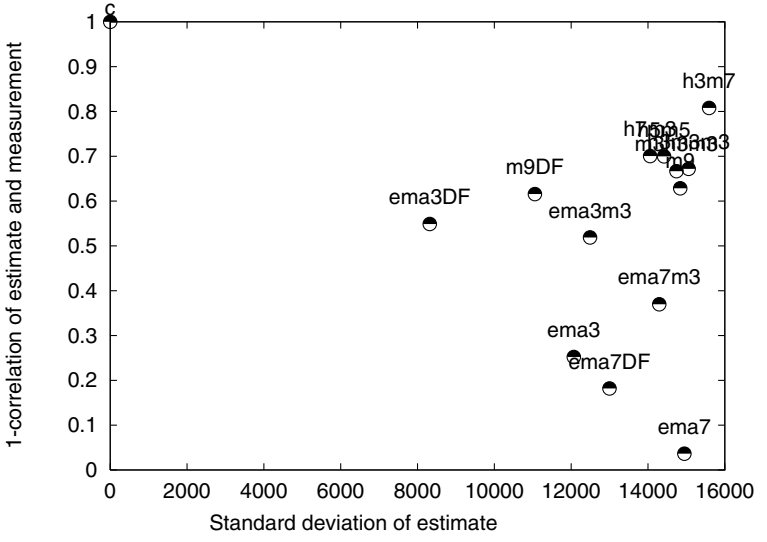


Fig. 1. Selection of Pareto optimal solutions using two criteria

The evaluation of estimation algorithms for the second application (routing) used one more criterion ($\text{MAPE}_{\max}$) and different aspiration and reservation values that represented preferences that valued accuracy more than stability. The ranking of results using the objective function was produced in the same way as for the first application.

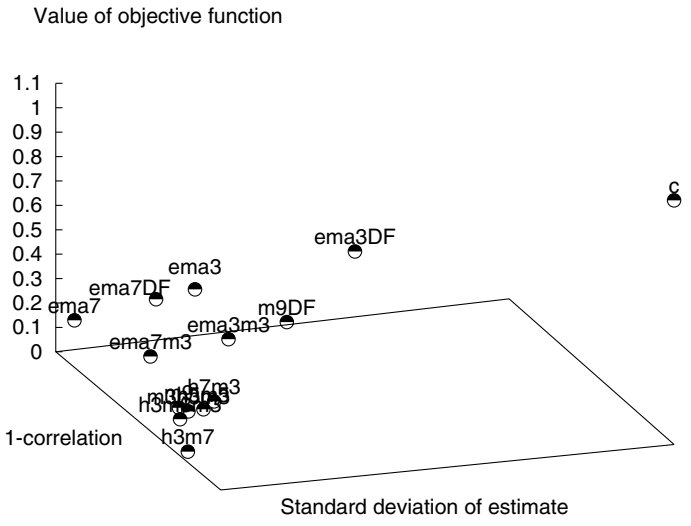


Fig. 2. Selection solutions using the objective function

The results shown in Table 2 are quite different than for the first application. In all cases, the *ema*7 algorithm scored first; for the first application, this algorithm produced too variable estimations, and never appeared in the first three positions of the ranking. For most other traces, the $ema7_{DF}$ algorithm is second, and *ema*3 is third. *ema*3 is on third place of the ranking for the first application for three traces (it was in the first five positions of the ranking for most traces). On the other hand, the constant estimate c appears only once in the first three positions of the ranking for the second application. Clearly, this algorithm is too insensitive for an application that requires accuracy.

Estimation using discrete intervals clearly decreases the performance of the estimations for the second application, which supports the conclusion that this algorithm trades off accuracy for stability.

7 Conclusion

In this paper, we first presented an evaluation approach for estimation algorithms that uses multi-criteria analysis. The approach was used on two examples with different requirements of estimation algorithms. It was shown how multi-criteria evaluation can be adapted to the preferences of an algorithm designer, and how the results of the analysis can be used to obtain insight into algorithm operation.

The question of the best algorithm for either of the two applications remains open, since we have not evaluated a sufficient number of algorithms, and it is possible to use more advanced estimation techniques. However, the chosen evaluation methodology should be useful for a further exploration of this topic.

Table 2. Overview of the three best estimation algorithms for each trace (OF-based)

	CDN				Routing				
Measure-ment	Estimation algorithm	s_y	1-CC	Value of OF	Estimation algorithm	s_y	1-CC	max MAPE	Value of OF
synt-1	c	0	1.00	1.00	$ema7$	19683	0.04	0.42	1.0000
	$h3m7$	12385	0.94	0.55	$ema7m3$	18047	0.38	1.21	-0.3000
	$ema3m3$	13999	0.53	0.46	c	0	1.00	1.04	-0.3200
synt-2	c	0	1.00	1.00	$ema7$	19982	0.04	0.33	1.0000
	$h3m7$	13798	0.82	0.49	$ema7DF$	18960	0.06	0.38	-0.0200
	$ema3DF$	14327	0.33	0.46	$ema7m3$	18218	0.38	0.74	-0.7100
synt-3	c	0	0.00	1.00	$ema7$	14948	0.04	0.11	1.0100
	$ema3DF$	8320	0.55	0.63	$ema7DF$	12998	0.18	0.17	0.0036
	$ema3$	12069	0.25	0.34	$ema7m3$	14295	0.37	0.35	0.0010
info to tahoe	c	0	1.00	1.00	$ema7$	475	0.03	4.41	0.0024
	$m9DF$	0	1.00	1.00	$ema7DF$	491	0.04	7.43	0.0023
	$ema3m3$	234	0.71	0.72	$ema3$	276	0.23	10.97	0.0018
grouse to cern	c	0	1.00	1.00	$ema7$	1980	0.03	1.21	0.0028
	$ema3DF$	1647	0.31	0.27	$ema7DF$	1859	0.06	1.39	0.0023
	$ema3$	1658	0.22	0.27	$ema3$	1658	0.22	4.81	0.0015
grouse to nimi1	c	0	1.00	1.00	$ema7$	1365	0.06	2.02	0.0023
	$m9DF$	0	1.00	1.00	$ema7DF$	1505	0.16	1.85	0.0018
	$ema3$	1174	0.47	0.32	$ema3$	1174	0.47	3.57	0.0003
grouse to tahoe	c	0	1.00	1.00	$ema7$	2000	0.04	0.49	0.0028
	$m9DF$	959	0.53	0.70	$ema7DF$	2138	0.11	0.65	0.0024
	$ema3DF$	1720	0.32	0.25	$ema3$	2026	0.22	0.97	0.0017
grouse to tracer	c	0	1.00	1.00	$ema7$	158	0.13	0.66	0.0027
	$m9DF$	0	1.00	1.00	$ema3$	203	0.74	1.49	-0.1500
	$ema3DF$	135	1.10	0.36	$ema7DF$	235	0.40	0.69	-0.1800
grouse to stanford	c	0	1.00	1.00	$ema7$	1231	0.05	1.02	0.0026
	$m9DF$	493	1.04	0.84	$ema7DF$	1187	0.10	1.26	0.0023
	$ema3DF$	806	0.49	0.57	$ema3$	840	0.34	1.87	0.0010
grouse to pendragon	c	0	1.00	1.00	$ema7$	1200	0.09	1.02	0.0023
	$m9DF$	0	1.00	1.00	$ema7DF$	788	0.11	1.22	0.0023
	$ema7DF$	788	0.11	0.56	$ema3$	1390	0.49	2.11	0.0002

Our approach allows also to preselect estimation algorithms that satisfy certain requirements.

The adaptation of the multi-criteria methodology for a particular application is made by the selection of appropriate aspiration and reservation levels. This process cannot be thought of as a single-step process, but should be regarded as an iterative procedure. The algorithm designer can learn how to express his preferences using aspiration and reservation levels, and he can modify his preferences over time, as he grows more acquainted with the tradeoffs and characteristics of the problem.

It is not always possible to specify the criteria for estimation algorithm evaluation using closed-form, analytical formulas, as in this paper. An example could be the algorithm that TCP uses for RTT estimation. It was designed with the single objective that the protocol should use only the available bandwidth. Introducing reduced delay as an additional objective could require another estimation algorithm. The criteria used to evaluate algorithm performance in this case would require a simulation of the TCP flow control algorithm that uses a particular estimation method. However, regardless of the method of criteria calculation, the evaluation scheme and the usage of the objective function remains the same as shown above.

References

1. Wierzbicki, A.: Models for internet cache location. In: 7th International Web Caching Workshop. (2002)
2. Gallagher, N.C., Wise, G.L.: A theoretical analysis of the properties of median filters. IEEE Transactions on Acoustics, Speech, and Signal Processing **ASSP-29** (1981) 1136–1141
3. Rabiner, L.R., Sambur, M.R., Schmidt, C.E.: Applications of a nonlinear smoothing algorithm to speech processing. IEEE Transactions on Acoustics, Speech, and Signal Processing **ASSP-23** (1975) 552–557
4. Burgstahler, L., Neubauer, M.: New modifications of the exponential moving average algorithm for bandwidth estimation. In: Proceedings of the 15^{th} ITC Specialist Seminar, Würzburg, Germany. (2002) 210–219
5. Wierzbicki, A.P., Makowski, M., Wessels, J., eds.: Model-Based Decision Support Methodology with Environmental Applications. Kluwer Academic Publishers (2000)
6. National Laboratory for Applied Network Research (NLANR): National internet measurement infrastructure. http://www.ncne.nlanr.net/nimi/ (2002)
7. Pittsburgh Supercomputing Center, Carnegie Mellon University: About the psc treno server. http://www.psc.edu/networking/treno_info.html (2000)

Author Index

Aalto, Samuli 252, 276
Aguiar, Rui L. 18
Akar, Nail 100
Arabas, Piotr 76

Beaujean, Christophe 18
Bolla, Raffaele 49
Brandauer, Christof 203
Burgstahler, Lars 291

Choi, JinSeek 64

Dąbrowski, Marek 189, 218
Davoli, Franco 49
Dimopoulou, Lila 177
Do, Tien Van 88
Dorfinger, Peter 203
Drobnik, Oswald 29

Eichler, Gerald 218

Fudała, Monika 218

Garcia, Carlos 18
Gozdecki, Janusz 18

Hussmann, Heinrich 154

Kaczmarek, Sylwester 127
Kálmán, Barnabás 88
Kamola, Mariusz 76
Kang, Minho 64
Katzengruber, Dietmar 218
Kemmel, Falk 165
Kilkanen, Tero 218
Kim, Myoung Hun 265
Kim, Sungchang 64
Király, Csaba 88
Koch, Bert F. 154
Kooij, Robert E. 115
Krieger, Udo 29

Liebsch, Marco 18

Malinowski, Krzysztof 76
Maniatis, Sotiris 165
Marques, Victor 18
Matthes, Michael 29
Melin, Eric 18
Menth, Michael 234
Miettinen, Natalia 218
Milbrandt, Jens 234
Moreno, Jose Ignacio 18

Narloch, Marcin 127
Nikolouzou, Eugenia 177
Nyberg, Eeva 276

Østerbø, Olaf 115

Pacyna, Piotr 18
Pándi, Zsolt 88
Park, Hong Shik 265
Potts, Martin 1
Prokopp, Daniel 29

Reifert, Andreas 234
Repetto, Matteo 49
Ricciato, Fabio 139

Sahin, Cem 100
Salsano, Stefano 177
Sampatakos, Petros 177
Strohmeier, Felix 189
Susitaival, Riikka 252

Tarasiuk, Halina 218
Thomas, Anne 165
Titze, Michael 218
Tran, Hung Tuan 139
Tsetsekas, Charilaos 165

Venieris, Iakovos S. 177
Virtamo, Jorma 252

Wal, J.C. van der 115
Wierzbicki, Adam 291

Ziegler, Thomas 139