Helsinki University of Technology Department of Communications and Networking
Teknillinen korkeakoulu Tietoliikenne- ja tietoverkkotekniikan laitos
Espoo 2009 Report 5/2009

QUALITY OF SERVICE AND RESOURCE MANAGEMENT IN IP AND WIRELESS NETWORKS

Jani Lakkakorpi

Dissertation for the degree of Doctor of Science in Technology to be presented with due permission for public examination and debate in Auditorium S4 at Helsinki University of Technology / Aalto University (Espoo, Finland) on the 22nd of January, 2010, at 12 o'clock noon.

Helsinki University of Technology
Faculty of Electronics, Communications and Automation
Department of Communications and Networking

Teknillinen korkeakoulu
Elektroniikan, tietoliikenteen ja automaation tiedekunta
Tietoliikenne- ja tietoverkkotekniikan laitos

Distribution:
Helsinki University of Technology
Department of Communications and Networking
P.O. Box 3000
FI-02015 TKK
Tel. +358 9 470 25300
Fax +358 9 470 22474

ISBN 978-952-248-242-6
ISBN 978-952-248-243-3 (PDF)
ISSN 1797-478X
ISSN 1797-4798 (PDF)

Multiprint Oy
Espoo 2009

ABSTRACT OF DOCTORAL DISSERTATION	HELSINKI UNIVERSITY OF TECHNOLOGY P.O. BOX 1000, FI-02015 TKK http://www.tkk.fi

Author	Jani Lakkakorpi		
Name of the dissertation	Quality of Service and Resource Management in IP and Wireless Networks		
Manuscript submitted	17.8.2009	Manuscript revised	30.10.2009
Date of the defence	22.1.2010		
☐ Monograph		☒ Article dissertation (summary + original articles)	

Faculty	Faculty of Electronics, Communications and Automation
Department	Department of Communications and Networking
Field of research	Teletraffic theory
Opponents	Professor Timo Hämäläinen (University of Jyväskylä)
	Professor Enzo Mingozzi (University of Pisa, Italy)
Pre-examiners	Docent Yevgeni Koucheryavy (Tampere University of Technology)
	Professor Enzo Mingozzi (University of Pisa, Italy)
Supervisor	Professor Jorma Virtamo (Helsinki University of Technology)

Abstract

A common theme in the publications included in this thesis is the quality of service and resource management in IP and wireless networks. This thesis presents novel algorithms and implementations for admission control in IP and IEEE 802.16e networks, active queue management in EGPRS, WCDMA, and IEEE 802.16e networks, and scheduling in IEEE 802.16e networks. The performance of different algorithms and mechanisms is compared with the prior art through extensive ns-2 simulations.

We show that similar active queue management mechanisms, such as TTLRED, can be successfully used to reduce the downlink delay (and in some cases even improve the TCP goodput) in different bottlenecks of IP, EGPRS, WCDMA, and IEEE 802.16e access networks. Moreover, almost identical connection admission control algorithms can be applied both in IP access networks and at IEEE 802.16e base stations. In the former case, one just has to first gather the link load information from the IP routers.

We also note that DiffServ can be used to avoid costly overprovisioning of the backhaul in IEEE 802.16e networks. We present a simple mapping between IEEE 802.16e data delivery services and DiffServ traffic classes, and we propose that IEEE 802.16e base stations should take the backhaul traffic load into account in their admission control decisions.

Moreover, different IEEE 802.16e base station scheduling algorithms and uplink channel access mechanisms are studied. In the former study, we show that proportional fair scheduling offers superior spectral efficiency when compared to deficit round-robin, though in some cases at the cost of increased delay. Additionally, we introduce a variant of deficit round-robin (WDRR), where the *quantum* value depends on the modulation and coding scheme.

We also show that there are several ways to implement ertPS in an efficient manner, so that during the silence periods of a VoIP call no uplink slots are granted. The problem here, however, is how to implement the resumption after the silence period while introducing as little delay as possible.

Keywords	quality of service, resource management, active queue management, admission control, scheduling		
ISBN (printed)	978-952-248-242-6	ISSN (printed)	1797-478X
ISBN (pdf)	978-952-248-243-3	ISSN (pdf)	1797-4798
Language	English	Number of pages	87 p. + app. 68 p.
Publisher	Department of Communications and Networking, Helsinki University of Technology		

☒ The dissertation can be read at http://lib.tkk.fi/Diss/2010/isbn9789522482433/

VÄITÖSKIRJAN TIIVISTELMÄ	TEKNILLINEN KORKEAKOULU PL 1000, 02015 TKK http://www.tkk.fi

Tekijä	Jani Lakkakorpi		
Väitöskirjan nimi	Palvelun laatu ja resurssien hallinta IP- ja langattomissa verkoissa		
Käsikirjoituksen päivämäärä	17.8.2009	Korjatun käsikirjoituksen päivämäärä	30.10.2009
Väitöstilaisuuden ajankohta	22.1.2010		
☐ Monografia		☒ Yhdistelmäväitöskirja (yhteenveto + erillisartikkelit)	

Tiedekunta	Elektroniikan, tietoliikenteen ja automaation tiedekunta
Laitos	Tietoliikenne- ja tietoverkkotekniikan laitos
Tutkimusala	Teleliikenneteoria
Vastaväittäjät	Professori Timo Hämäläinen (Jyväskylän yliopisto)
	Professori Enzo Mingozzi (University of Pisa, Italy)
Esitarkastajat	Dosentti Yevgeni Koucheryavy (Tampereen teknillinen yliopisto)
	Professori Enzo Mingozzi (University of Pisa, Italy)
Työn valvoja	Professori Jorma Virtamo (Teknillinen korkeakoulu)

Tiivistelmä

Tämän väitöskirjan julkaisujen yhteinen nimittäjä on palvelun laatu (QoS) ja resurssien hallinta IP-verkoissa ja langattomissa verkoissa. Työssä esitetään uusia algoritmeja ja toteutuksia pääsynvalvontaan IP- ja IEEE 802.16e -verkoissa, aktiiviseen jononhallintaan EGPRS-, WCDMA- ja IEEE 802.16e -verkoissa sekä skedulointiin IEEE 802.16e -verkoissa. Eri algoritmien ja mekanismien suorituskykyä verrataan olemassa oleviin ratkaisuihin simulaatioiden (ns-2) avulla.

Työssä osoitetaan, että samankaltaisia aktiivisen jononhallinnan mekanismeja (kuten esim. TTLRED) voidaan hyvällä menestyksellä käyttää pienentämään viiveitä (ja joissain tapauksissa jopa parantamaan TCP:n läpäisyä) erilaisissa IP-, EGPRS-, WCDMA- ja IEEE 802.16e -pääsyverkkojen ”pullonkauloissa”. Tämän lisäksi lähes identtisiä pääsynvalvonta-algoritmeja voidaan käyttää sekä IP-pääsyverkoissa että IEEE 802.16e -tukiasemilla. Ensin mainitussa tapauksessa tarvitaan luonnollisesti mekanismi, jolla linkkikuormatieto kerätään IP-reitittimiltä.

Työssä ehdotetaan myös DiffServ–mekanismien soveltamista IEEE 802.16e -pääsyverkoissa, jotta backhaul-linkkien ylimitoittaminen voitaisiin välttää. Työssä esitetään, kuinka IEEE 802.16e -palvelunlaatuluokat voidaan sitoa DiffServ-palvelunlaatuluokkiin ja kuinka IEEE 802.16e -tukiasemien tulisi ottaa pääsyverkon IP-linkkien kuorma huomioon pääsynvalvontapäätöksissään.

Tämän lisäksi työssä tutkitaan IEEE 802.16e -tukiaseman skedulointialgoritmeja sekä ”uplink channel access” -mekanismeja. On todettu, että proportional fair -skedulointi käyttää radiokapasiteettia selvästi tehokkaammin kuin deficit round-robin -skedulointi. Joissakin tapauksissa proportional fair -algoritmi tosin johtaa suurempiin viiveisiin. Työssä ehdotetaan myös muunnelmaa deficit round-robin -algoritmista (WDRR), jossa *quantum*-parametri riippuu käytössä olevasta modulaatiosta.

Työssä myös esitetään eri tapoja toteuttaa ertPS-palvelu niin, että VoIP-puhelun hiljaisten jaksojen aikana tukiasema ei jaa yhteydelle uplink-kapasiteettia. Ongelmana kuitenkin on, kuinka tukiasema voi mahdollisemman nopeasti jatkaa kapasiteetin jakamista hiljaisen jakson jälkeen.

Asiasanat	palvelunlaatu, resurssien hallinta, aktiivinen jononhallinta, pääsynvalvonta, skedulointi		
ISBN (painettu)	978-952-248-242-6	ISSN (painettu)	1797-478X
ISBN (pdf)	978-952-248-243-3	ISSN (pdf)	1797-4798
Kieli	englanti	Sivumäärä	87 s. + liitteet 68 s.
Julkaisija	Tietoliikenne- ja tietoverkkotekniikan laitos, Teknillinen korkeakoulu		
☒ Luettavissa verkossa osoitteessa http://lib.tkk.fi/Diss/2010/isbn9789522482433/			

Preface

This thesis consists of eight publications that were all written when I was with Nokia Research Center and later with Nokia Devices R&D. The work was carried out in different research projects, where the focus was on quality of service in IP access networks, EGPRS, WCDMA, and IEEE 802.16e. The first publication included in this thesis was published in the year 2005. However, the work had started already in the year 2002.

First, I would like to thank all my co-authors (in alphabetical order): Olli Alanen, Renaud Cuny, Juha Karhula, Jani Moilanen, Jukka Salonen, Alexander Sayenko, and Ove Strandberg. I am very grateful for all your comments and suggestions; they have surely improved the quality of the publications.

Then I want to thank my former line managers at Nokia for giving me the opportunity to write academic publications and supporting my travel plans for different conferences all over the world, despite the tight travel budgets. In chronological order: Jussi Ruutu, Timo Knuutila, Andreas Heiner, Veli-Matti Teittinen, Janne Parantainen, Marko Mahkonen, Kari Pehkonen, Soni Koul, Shkumbin Hamiti, and Riku Pirhonen.

Moreover, I have received lots of valuable advice from my former Nokia colleagues Kalevi Kilkki, Sergey Balandin, Sandro Grech, Vilho Räisänen, Haitao Tang, and Carl Eklund – just to name a few.

Professor Jorma Virtamo has supervised my studies since 1999. Then I had the honor to be the first M.Sc. from Helsinki University of Technology majoring in teletraffic theory. I am very grateful for all the guidance I have received from him during the past ten years. It has been a privilege to work for someone with such a deep knowledge and wide recognition.

I also want to express my gratitude to my current line manager Professor Jörg Ott, who has given me lots of time and freedom to finish this thesis. Naturally, finalizing this thesis has stolen some time from my current research on delay-tolerant networking.

Last, but definitely not least, I want to thank my family and friends for their continuous support.

Espoo, July 3, 2009

Jani Lakkakorpi

Abbreviations

3GPP	3rd Generation Partnership Project
AF	Assured Forwarding
AM	Acknowledged Mode
AMC	Adaptive Modulation and Coding
AQM	Active Queue Management
ARED	Adaptive Random Early Detection
ARQ	Automatic Repeat request
B	Byte
BB	Bandwidth Broker
BE	Best Effort
BG	Background
BLER	Block Error Rate
BPSK	Binary Phase Shift Keying
BS	Base Station
CAC	Connection Admission Control
CID	Channel Identifier
CONSER	Collaborative Simulation for Education and Research
CPU	Central Processing Unit
CQI	Channel Quality Indicator
DARPA	Defense Advanced Research Projects Agency
DCH	Dedicated Channel
DiffServ	Differentiated Services
DL	Downlink
DRR	Deficit Round-Robin
DSCP	Differentiated Services Code Point
DTCH	Dedicated Traffic Channel
E2E	End-to-End
ECN	Explicit Congestion Notification
EDGE	Enhanced Data Rates for GSM Evolution
EF	Expedited Forwarding
EGPRS	Enhanced General Packet Radio Service
FP	Frame Protocol

FTP	File Transfer Protocol
GGSN	Gateway GPRS Support Node
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
GTP	GPRS Tunneling Protocol
H-ARQ	Hybrid ARQ
HSDPA	High-Speed Downlink Packet Access
HS-DSCH	High-Speed Downlink Shared Channel
HTTP	Hypertext Transfer Protocol
IA	Interactive
IntServ	Integrated Services
IP	Internet Protocol
IPTV	Internet Protocol Television
MAAC	Measurement-Aided Admission Control
MAC	Medium Access Control
MBAC	Measurement-Based Admission Control
MS	Mobile Station
MTU	Maximum Transfer Unit
NRT	Non-Real Time
nrtPS	Non-Real Time Polling Service
NSF	National Science Foundation
OFDM	Orthogonal Frequency Division Multiplex
OFDMa	Orthogonal Frequency Division Multiple Access
OSPF	Open Shortest Path First
PBAC	Parameter-Based Admission Control
PDCP	Packet Data Convergence Protocol
PDPC	Packet Discard Prevention Counter
PDU	Protocol Data Unit
PFC	Packet Flow Control
PHB	Per Hop Behavior
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
RAB	Radio Access Bearer

RAN	Radio Access Network
RED	Random Early Detection
RFC	Request For Comments
RLC	Radio Link Control
RNC	Radio Network Controller
RRM	Radio Resource Management
RT	Real Time
RTG	Receive-Transmit Transition Gap
rtPS	Real-Time Polling Service
RTT	Round-Trip Time
SAMAN	Simulation Augmented by Measurement and Analysis for Networks
SDU	Service Data Unit
SGSN	Serving GPRS Support Node
SLA	Service Level Agreement
SNMP	Simple Network Management Protocol
SNR	Signal-to-Noise Ratio
SS	Subscriber Station
TCP	Transmission Control Protocol
TDD	Time Division Duplex
TDMA	Time Division Multiple Access
THP	Traffic Handling Priority
TM	Transparent Mode
TTG	Transmit-Receive Transition Gap
TTI	Transmission Time Interval
TTLRED	Time-to-Live Random Early Detection
UDP	User Datagram Protocol
UE	User Equipment
UGS	Unsolicited Grant Service
UL	Uplink
UM	Unacknowledged Mode
UMTS	Universal Mobile Telecommunications System
VoIP	Voice over Internet Protocol
WCDMA	Wideband Code Division Multiple Access
WiMAX	Worldwide Interoperability for Wireless Access

Contents

PREFACE **7**

ABBREVIATIONS **8**

CONTENTS **11**

LIST OF PUBLICATIONS **13**

1 INTRODUCTION **14**

1.1 QoS AND RESOURCE MANAGEMENT IN IP AND WIRELESS NETWORKS 14
1.2 METHODOLOGY 16
1.3 OUTLINE OF THE THESIS 17

2 IMPLEMENTING QUALITY OF SERVICE IN IP NETWORKS **18**

2.1 SCHEDULING 18
2.2 ACTIVE QUEUE MANAGEMENT 19
2.3 DIFFERENTIATED SERVICES 20

3 ADAPTIVE CONNECTION ADMISSION CONTROL FOR DIFFERENTIATED SERVICES ACCESS NETWORKS **22**

3.1 RELATED WORK 22
3.2 MODIFIED BANDWIDTH BROKER FRAMEWORK 24
3.3 ADAPTIVE AF WEIGHT TUNING 29
3.4 ADAPTIVE EF AND RT RESERVATION LIMIT TUNING 30
3.5 PERFORMANCE EVALUATION 30
3.6 SUMMARY 34

4 ACTIVE QUEUE MANAGEMENT IN WCDMA NETWORKS **35**

4.1 WCDMA NETWORK AND RNC 35
4.2 ACTIVE QUEUE MANAGEMENT 38
4.3 RELATED WORK 39
4.4 TTLRED 39
4.5 PERFORMANCE EVALUATION 40
4.6 SUMMARY 43

5 ACTIVE QUEUE MANAGEMENT IN EGPRS NETWORKS **44**

5.1 FLOW CONTROL IN EGPRS 44
5.2 ACTIVE QUEUE MANAGEMENT SCHEMES 46
5.3 RANDOM DROPPING OR MARKING BASED ON PACKET LIFETIME 47
5.4 SIMULATION MODEL 48
5.5 PERFORMANCE EVALUATION 50
5.6 SUMMARY 50

6 QUALITY OF SERVICE AND RESOURCE MANAGEMENT IN IEEE 802.16E **52**

6.1 IEEE 802.16E 52
6.2 RELATED WORK 54
6.3 ACTIVE QUEUE MANAGEMENT 54
6.4 BS SCHEDULING ALGORITHMS 55
6.5 BACKHAUL QOS 57
6.6 CONNECTION ADMISSION CONTROL 58
6.7 UPLINK CHANNEL ACCESS AND ERTPS RESUMPTION MECHANISMS 60

6.8 IEEE 802.16E SIMULATION MODEL 61
6.9 PERFORMANCE EVALUATION 65
6.10 SUMMARY 74

7 AUTHOR'S CONTRIBUTION 79

8 CONCLUSIONS 80

REFERENCES 82

List of Publications

P1 J. Lakkakorpi, O. Strandberg, and J. Salonen, "Adaptive Connection Admission Control for Differentiated Services Access Networks," IEEE Journal on Selected Areas in Communications, vol. 23, no. 10, pp. 1963–1972, Oct. 2005.

P2 J. Lakkakorpi and R. Cuny, "Comparison of Different Active Queue Management Mechanisms for 3G Radio Network Controllers," in Proc. IEEE WCNC 2006, Las Vegas, Nevada, USA, Apr. 2006, pp. 80–85.

P3 R. Cuny and J. Lakkakorpi, "Active Queue Management in EGPRS," in Proc. IEEE VTC2006-Spring, Melbourne, Australia, May 2006, pp. 373–377.

P4 J. Lakkakorpi, A. Sayenko, J. Karhula, O. Alanen, and J. Moilanen, "Active Queue Management for Reducing Downlink Delays in WiMAX," in Proc. IEEE VTC2007-Fall, Baltimore, Maryland, USA, Oct. 2007, pp. 326–330.

P5 J. Lakkakorpi, A. Sayenko, and J. Moilanen, "Comparison of Different Scheduling Algorithms for WiMAX Base Station: Deficit Round-Robin vs. Proportional Fair vs. Weighted Deficit Round-Robin," in Proc. IEEE WCNC 2008, Las Vegas, Nevada, USA, Mar. 2008, pp. 1991–1996.

P6 J. Lakkakorpi and A. Sayenko, "Backhaul as a Bottleneck in IEEE 802.16e Networks," in Proc. IEEE Globecom 2008, New Orleans, Louisiana, USA, Nov.–Dec. 2008, pp. 1–6.

P7 J. Lakkakorpi and A. Sayenko, "Measurement-Based Admission Control Methods for Real-Time Services in IEEE 802.16e," in Proc. Second International Conference on Communication Theory, Reliability, and Quality of Service (CTRQ), Colmar, France, Jul. 2009, pp. 37–41.

P8 J. Lakkakorpi and A. Sayenko, "Uplink VoIP Delays in IEEE 802.16e Using Different ertPS Resumption Mechanisms," in Proc. Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM), Sliema, Malta, Oct. 2009, pp. 157–162.

1 Introduction

Traditionally, capacity issues in fixed IP networks have been solved through overprovisioning as bandwidth has been relatively cheap. In wireless networks, however, "throwing bandwidth at the problem" has never been a viable option and it would be very expensive to overprovision radio links. Thus, different quality of service (QoS) mechanisms have been implemented and used in wireless networks more extensively than in fixed IP networks.

Even though emerging wireless technologies (such as IEEE 802.16, also known as WiMAX[1]) promise very high data rates, it is our firm belief that all network capacity will eventually be used as applications need more bandwidth than before. One example of bandwidth-hungry applications is peer-to-peer file sharing. During the last couple of years, peer-to-peer services have become one of the most important sources of Internet traffic. However, HTTP is still the dominant application [Mai09]. Moreover, IPTV (a digital television service delivered using IP) is becoming more and more popular [Won08].

1.1 QoS and Resource Management in IP and Wireless Networks

This thesis consists of eight publications, P1–P8. A common denominator in all these publications is QoS and resource management. Active queue management (AQM), for example, can be applied in a similar manner in the different bottlenecks of enhanced general packet radio service (EGPRS), wideband code division multiple access (WCDMA), and IEEE 802.16e (also known as mobile WiMAX) networks in order to keep the (per-connection) queues short and thus delays also short. The reason for the queue buildup is TCP flow control: when TCP packets are sent in bigger bursts, they have to wait for service in the bottleneck buffer. This can be changed by randomly dropping or marking packets so that the TCP sender will react and send packets in a smoother manner, resulting in shorter bottleneck queues. We demonstrate this in Publications P2–P4. The AQM mechanisms presented include our own contribution, time-to-live based random early detection (TTLRED), where the average queue size that is normally used in packet dropping decisions is replaced with the packet lifetime.

Moreover, it is possible to apply very similar measurement-based connection admission control (CAC) mechanisms in differentiated services (DiffServ) capable IP access networks and at an IEEE 802.16e base station (BS). In the former case, we use the bottleneck link load in our admission decisions, while in the latter case it is the number of free slots. Publication P1 presents our modified bandwidth broker framework and adaptive CAC algorithms (that tune the

[1] WiMAX stands for worldwide interoperability for wireless access.

scheduling weights and reservation limits) for DiffServ access networks, while Publication P7 presents two novel measurement-based CAC algorithms for IEEE 802.16e.

In Publication P5, different IEEE 802.16e BS scheduling algorithms are studied. In this study, we show that proportional fair (PF) scheduling offers superior spectral efficiency when compared to deficit round-robin (DRR), though in some cases at the cost of increased delay. Moreover, we introduce a variant of DRR (weighted DRR, WDRR) that takes the current channel conditions into account in a manner similar to PF.

In Publication P6, we note that backhaul could also become a bottleneck in IEEE 802.16e networks in the future, when radio capacity is increased. As an alternative to the costly overprovisioning of the backhaul, we propose that well-known IP QoS mechanisms such as DiffServ should be used in order to support real-time connections on the backhaul. We make a proposal for how IEEE 802.16e data delivery services should be mapped to DiffServ traffic classes. Moreover, it is proposed that the IEEE 802.16e BS should take backhaul load into account in its connection admission decisions.

In Publication P8, we show that there are different ways to implement an extended real-time polling service (ertPS) in IEEE 802.16e in such a manner that during the silence periods of a voice over IP (VoIP) call no uplink (UL) slots are granted. The research problem here is how to re-activate, i.e., resume the connection after the silence period with as little additional delay as possible. Granting one polling slot periodically during the silence periods of the VoIP connection is a simple solution. However, the polling slots eat too much capacity. The main alternatives to polling are contention, multicast polling, and the use of the fast feedback channel.

With basic contention, there is no separation between the VoIP connections that use contention only for sending a single packet after a silence period and connections that use contention as their primary means to obtain bandwidth. This means that the code division multiple access (CDMA) codes can collide and thus the VoIP resumption delays can easily grow too big. Multicast polling is otherwise similar to basic contention, except that now there is a dedicated contention region for a certain group of subscriber stations (SS). Thus, CDMA codes from VoIP users and other SSs cannot collide. However, according to the IEEE 802.16 standard [802.16], common request backoff parameters have to be used in basic contention and multicast polling. This is clearly a problem because of the strict delay requirements of VoIP.

The use of the fast feedback channel (i.e., the channel quality indicator channel, CQICH) for carrying the ertPS VoIP resumption codeword seems to be the most efficient way to re-activate ertPS VoIP connections.

1.2 Methodology

All the performance evaluations in this thesis are performed by simulation studies. The primary reason for this is that in most cases the systems are very complex and thus analytical modeling would be quite difficult. In all studies, the simulation tool was the Network Simulator, ns-2 [NS2].

The Network Simulator is a discrete event simulator intended for networking research. The history of ns-2 goes back to 1988, when the predecessor of ns-2 (simply called ns) was born as a variant of the REAL network simulator [Kes88]. Since then, ns has evolved substantially. Currently, the development of ns-2 is supported through the Defense Advanced Research Projects Agency (DARPA), with Simulation Augmented by Measurement and Analysis for Networks (SAMAN) and through the National Science Foundation (NSF) with Collaborative Simulation for Education and Research (CONSER), both in collaboration with other researchers. Since ns-2 has always been open source software, a big community of ns-2 users and co-developers exists.

The "building blocks" of the simulator are written in C++ and the simulation scripts that run the executable are written in the OTcl language. The scripts describe the network topology, protocols, workload, and control parameters. Simulation results (such as end-to-end IP packet delay, packet loss, and TCP goodput) can be parsed from default trace files. However, in most cases, it is more feasible to define one's own trace files and output only that information one is interested in.

The ns-2 simulator provides extensive support for different variants or transmission control protocol (TCP), (ad hoc) routing, and multicast protocols. If something does not exist in the "main trunk" (i.e., an official release), it can always be added. Because of its popularity, there is a good deal of third-party ns-2 code publicly available.

Nevertheless, sometimes it is more beneficial to write additional modules from the scratch than trying to understand code written by other people. The code in the "main trunk" can usually be trusted, since it is used by hundreds of researchers and most bugs have been fixed over the years (this is the case, e.g., with the TCP code). However, third-party code is often not very extensively tested.

The author has written several additional modules for ns-2, including many traffic sources, modified bandwidth broker framework, EGPRS and WCDMA modules. All these modules were used in order to obtain the results presented in this thesis. Moreover, there was a close co-operation between the University of Jyväskylä (JYU) and the author. JYU delivered the basic IEEE 802.16e module and the author added new features on top of the basic version. These

additional features include, e.g., support for active queue management, connection admission control, and different scheduling algorithms.

1.3 Outline of the Thesis

This thesis is organized as follows: Chapter 2 is an overview of well-known QoS mechanisms, Chapter 3 presents the modified bandwidth broker framework and our adaptive connection admission control algorithms for DiffServ-capable IP networks, and Chapter 4 presents our active queue management proposal (TTLRED) for radio network controllers in WCDMA networks, while Chapter 5 continues with the same theme and presents active queue management for 2G-SGSN in EGPRS networks.

Chapter 6 is about our IEEE 802.16e related proposals: it includes active queue management, BS scheduling with the focus on algorithms that take the channel conditions into account, a proposal of how to map IEEE 802.16e data delivery services to DiffServ traffic classes, two connection admission control algorithms, and uplink channel access related enhancements.

Finally, Chapter 7 summarizes the author's contribution to the articles included in this thesis and Chapter 8 presents the key findings of the thesis in a nutshell.

2 Implementing Quality of Service in IP Networks

One of the design principles in the Internet protocol (IP) has been that datagrams are delivered as best effort (BE), that is, all packets are treated equally in the routers and no guarantees about packet delivery are given. However, nowadays, the formerly more or less unused type-of-service (TOS) field in the IP packet header can be used for differentiating more important packets from less important ones. Additionally, if one wants to support different types of reservations[2] (e.g., "hard" versus "soft" reservations) for different types of connections, the routers need to be equipped with the proper mechanisms, such as priority queuing or deficit round-robin (DRR) [Shr96] for packet scheduling and random early detection (RED) [Flo93] for congestion management, in order to allow this differentiation.

2.1 Scheduling

Priority queuing is the most straightforward way to implement delay differentiation for multiple traffic classes, e.g., for voice and data traffic [Arm00]. In priority queuing, a queue with a higher priority is served as long as there are packets left in that queue, and only then can one start serving a queue with a lower priority. However, strict priority queuing can lead to starvation of the lower traffic class(es) and thus weighted scheduling (or a combination of strict priority and weights) is sometimes preferred.

Because of its simplicity and low computational complexity, deficit round-robin (DRR) [Shr96] is one of the most popular ways to implement weighted scheduling in IP routers. In DRR, the *deficitCounter* of each active queue[3] is increased by a *quantum* when the queue has its turn. If the size of the head-of-line packet in this queue is smaller than or equal to the *deficitCounter*, the packet is sent and the *deficitCounter* is decremented by the size of the packet. Sending packets is continued as long as the *deficitCounter* allows. If the *deficitCounter* is too small for the head-of-line packet, we move to the next queue. The deficit that is stored in the *deficitCounter* of the queue is then saved for the next round. If all the packets in the queue were served, the *deficitCounter* is reset to zero.

[2] A reservation is a set of connection admission control, packet scheduling, and dropping rules that have to co-operate in order to realize the desired behavior. There can be admission-controlled users and non-admission controlled users. The former group consists of those users that can actually benefit from reservations (e.g., VoIP or video streaming users), while the latter group, using, for example, file transfer protocol (FTP), would not necessarily benefit from reservations.

[3] An active queue means that there are packets in the buffer.

2.2 Active Queue Management

TCP global synchronization is a phenomenon that occurs when the bottleneck queue (in some IP router) gets full and several packets belonging to different TCP flows are dropped at the same time. In this kind of scenario, all the affected TCP flows will have to reduce their sending rate, since rate control (in most TCP variants) is largely based on packet drops.

Probably the best-known active queue management (AQM) method, random early detection (RED) [Flo93], aims at preventing TCP global synchronization by dropping random packets when the exponentially averaged queue size exceeds the minimum threshold (*minTh*). A packet drop every now and then prevents the congestion window at the TCP sender from growing too big and thus the packet flow from the sender becomes smoother, resulting in a shorter queue at the bottleneck router. The bigger the averaged queue size, the greater the probability of a packet drop. When the averaged queue size reaches the maximum threshold (*maxTh*), the dropping probability is at its peak level (*maxDP*). Naturally, queue size averaging weight (w_{AQS}) is also a configurable parameter. Fig. 1 illustrates packet dropping probability as a function of averaged queue size.

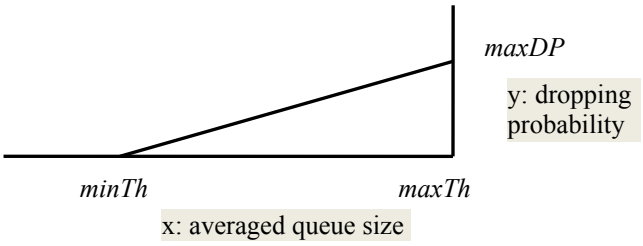

Fig. 1. RED: dropping probability as a function of averaged queue size.

Explicit congestion notification (ECN) [RFC3168] is a more advanced scheme. Otherwise, everything is done just like in RED but instead of the packet being dropped it is marked, if both the TCP connection and the router support ECN (which is not always the case). ECN allows a TCP receiver to inform the sender of congestion in the network by setting the ECN-Echo flag upon receiving an IP packet marked with congestion experienced (CE) bit(s). The TCP sender will then reduce its congestion window.

RED has been widely deployed in Internet routers. However, it has not been extensively used, mostly because of difficulties in selecting the optimal parameters. Tuning RED parameters has proven to be a difficult problem and it has been a popular topic in many publications. In [Mis00], the authors claim that there is a flaw in the RED queue averaging mechanism, which they believe is a cause of the tuning problems.

2.3 Differentiated Services

After the failure of integrated services (IntServ) [RFC1633] quality of service concept, differentiated services [RFC2475] has been the biggest effort for adding QoS to the Internet. This effort eventually resulted in a scalable architecture, where no state information is needed in the core routers but these can concentrate on packet forwarding using appropriate scheduling and dropping mechanisms. Packet scheduling and dropping decisions in the core routers are solely based on differentiated services code points (DSCP) that are set by the edge routers. In addition to packet marking, edge routers also take care of traffic classification and policing. Packet remarking, dropping, and shaping can be applied to non-conforming flows. The standardized per-hop behaviors (PHB) are expedited forwarding (EF) [RFC3246] and assured forwarding (AF) [RFC2597]. Naturally, BE is supported as well.

According to its original definition, EF is a "virtual leased line" treatment that can be used to build a low-loss, low-latency, low-jitter, assured bandwidth end-to-end service through DiffServ domains. EF is usually implemented with priority queuing, which requires the use of a rate limiter that protects other traffic. Naturally, strict edge policing has to be applied for this kind of traffic.

AF, on the other hand, can be used for different purposes. According to the official definition, the AF PHB group provides delivery of IP packets in four independently forwarded AF classes. Within each class, an IP packet can be assigned one of three different levels of drop precedence. Moreover, the reordering of packets belonging to the same microflow is not permitted if the packets belong to the same AF class. Weighted scheduling using, e.g., DRR is probably the only reasonable way to implement AF, since RFC 2597 states the following: "each AF class is in each DS node allocated a certain amount of forwarding resources (buffer space and bandwidth)." Differentiated packet dropping can be implemented by applying RED for each AF class and drop precedence level separately.

One practical way to use AF is to implement the so-called "Olympic service," which consists of three service classes: gold, silver, and bronze corresponding to AF classes 3, 2, and 1. Packets are assigned to these classes so that gold class packets experience less delay than silver class packets, which in turn experience less delay than bronze class packets. AF class selection can be based, for example, on application requirements. Moreover, packets within each AF class can be further separated by assigning them high, medium, or low drop precedence, corresponding to AF drop precedence levels 3, 2, or 1. Packets with a low drop precedence level are the most important ones. The drop precedence level of a packet can be assigned, for example, by using a leaky bucket traffic policer at the network edge.

A major problem with AF is the management of scheduling weights; it is very hard to build the aforementioned “Olympic service” unless the weights are configured in strict priority fashion. Otherwise, we would have to know the traffic mix accurately and adjust the scheduling weights accordingly. Of course, “Olympic service” is not the only service model that can be implemented with AF.

Probably the biggest handicap of DiffServ is that the management issues are not properly dealt with. It is crucial that the volume of EF traffic on a bottleneck link stays below the corresponding limit as otherwise the rate limiter will start dropping EF packets. Likewise, in the case of “Olympic service,” the traffic volumes of different AF classes should be linked to the corresponding AF weights in order to implement the delay differentiation goal. Admission control is something that could fill this gap. Actually, it is even mentioned in RFC 2597 that the AF PHB group can be used to implement a low-loss and low-latency service using an overprovisioned AF class – if the maximum arrival rate to that class is known a priori in each DiffServ node. It is also mentioned that the specification of the required admission control services is beyond the scope of the RFC.

3 Adaptive Connection Admission Control for Differentiated Services Access Networks

As more and more IP based applications with QoS requirements keep on emerging, it is becoming more and more evident that there is a need for connection admission control (CAC) in IP networks, too. For a TCP-based file transfer, there is nothing dramatic in a situation where, for instance, the bandwidth of the connection is suddenly halved; we just have to wait a little bit longer. For a VoIP conversation, however, this would not be acceptable because of the excessive packet delay and loss.

The last mile in many access networks consists of narrow-bandwidth links, e.g., leased lines. DiffServ can help to utilize these links in the most effective manner. DiffServ is managed through service level agreements (SLAs), whose enforcement should preferably include dynamic admission control [Bre00]. Otherwise, narrow-bandwidth access networks could become heavily congested or underutilized. It is possible to exercise dynamic admission control in IP networks, e.g., by probing the path with active measurements or using an agent called a bandwidth broker [RFC2638] to assist in deciding whether a connection is admitted to the network or not. In this thesis, we have studied the latter approach.

In Publication P1, we propose adaptive connection admission control mechanisms for DiffServ access networks. After first introducing the original bandwidth broker framework, we present our modified bandwidth broker framework (that knows the link loads) and then the algorithms for adaptively adjusting the AF weights and the reservation limits. Finally, simulations are used for validating the proposed ideas.

3.1 Related Work

There are many alternative ways to do admission control in IP networks – although none of them is widely used at this moment. These schemes can be first divided into parameter-based admission control (PBAC) and measurement-based admission control (MBAC). MBAC can be further divided into schemes that involve active measurements (i.e., sending probe packets) and schemes that do not involve active measurements.

In Publication P1, we have focused on the bandwidth broker approach. Bandwidth brokers have traditionally been designed to support PBAC only [Sch98]. However, we introduce modifications to the traditional bandwidth broker framework that will enable both the use of link measurements and reservation information in admission control decisions. The use of

adaptive packet scheduling weights and adaptive reservation limits in the admission control algorithm will be introduced, too.

Because of the fact that average bit rates can be substantially lower than the corresponding requested peak rates, the use of PBAC can leave the network underutilized. Link load measurements are needed for more efficient network utilization. EF and BE loads have already been suggested for the QBone architecture [Tei99]. In theory, it is possible that all admitted traffic sources start sending data at their peak rates at the same time. However, the probability for this is extremely low – especially if the number of traffic sources is very high. Moreover, it is possible to protect oneself against such an event by carefully combining MBAC and PBAC.

We present a flexible admission control mechanism for DiffServ access networks by extending and modifying the existing bandwidth broker framework proposed by Nichols *et al*. [RFC2638] and later implemented by Schelén [Sch98]. The extra information needed for measurement-based admission decisions – link loads – is retrieved from router statistics and periodically sent to the bandwidth broker agent of a routing domain. As a second enhancement, we allow connection admission control for multiple traffic classes, e.g., EF, AF1, and AF2. The motivation for doing CAC for selected AF traffic is that there are real-time applications with relaxed QoS requirements. These traffic sources (e.g., video or audio streaming) do not need the "virtual wire" treatment. Some statistical guarantees, however, have to be provided.

Wang *et al*. [Wan01] present an adaptive scheduling scheme to support premium service, i.e., EF in the DiffServ architecture. The scheme is designed for weighted packet scheduling, e.g., weighted round-robin (WRR). The core idea is to tune the scheduling weights of different traffic classes adaptively, according to the dynamics of the average queue sizes. The goal is naturally to achieve low loss, delay, and jitter for the premium service. The authors claim that neither strict admission control nor accurate traffic conditioning are needed. We disagree with this claim – in our opinion, admission control becomes necessary immediately when the connection arrival intensity is high enough.

Kawahara and Komatsu [Kaw02] introduce a scheme, called dynamically weighted queuing, for allocating bandwidth fairly in the DiffServ architecture. The proposed method estimates the number of active users in each class by simple traffic measurements. This estimate is then used in tuning the weights assigned to the queues of different classes. Shimonishi *et al*. [Shi02] propose a similar technique, where the sum of committed information rates (CIRs) of active flows in each class is estimated, and the link bandwidth is allocated according to the sum of CIRs. CIR-proportional allocation is combined with equal-share allocation in order to guarantee some resources for the best effort class connections with zero CIRs.

Neither of the aforementioned schemes, however, involves admission control. At least, these schemes do not couple admission control and adaptive scheduling weights.

3.2 Modified Bandwidth Broker Framework

The envisioned access network is assumed to be equipped with DiffServ routers; DiffServ is needed in realizing the bandwidth reservations. In addition to that, an improved bandwidth broker that knows all link loads and existing bandwidth reservations per DiffServ class is needed. Different admission control rules are applied for different types of connections and the same traffic class separation is performed in the routers as well. Packet scheduling has an essential role in protecting admission-controlled traffic from non-admission controlled traffic – as well as in protecting the admission-controlled traffic classes from each other. During congestion periods it is usually the non-admission controlled traffic that has to adapt, while sufficient QoS is guaranteed for the admission-controlled traffic classes. However, some resources are reserved for the non-admission controlled traffic, too. Naturally, the admission control function in the bandwidth broker is fully aware of the packet scheduling parameters.

In order to realize flexible bandwidth sharing, link capacity is divided in a hierarchical fashion so that there is a configurable total limit for all admission-controlled (non-BE) traffic, configurable limits for real-time (RT) and non-real time (NRT) traffic aggregates, and configurable limits for each admission-controlled traffic class. Moreover, for each stage, there is a reservation limit and a load limit – it is possible to combine MBAC and PBAC. By carefully constructing the admission control hierarchy and setting the corresponding limits, we can fulfill the QoS requirements of our traffic without compromising the goal of high bottleneck link utilization.

Since our admission control framework supports multiple admission-controlled classes, we need to pay special attention to scheduling issues. If there is a single admission-controlled class, say EF, we only have to make sure that the reserved EF bandwidth and the measured EF load are below their corresponding limits on each link. Packet scheduling will protect EF traffic from non-admission controlled traffic – but only if the relationship between the applied limits and the scheduling parameters is appropriate. However, if there are multiple admission-controlled classes, we have to take their scheduling weights into account in admission control or configure the weights in strict priority fashion, which would result in the aforementioned “Olympic service” model – and, thus, in delay differentiation between the AF classes. If the AF scheduling weights are not configured in strict priority fashion, we have to somehow figure out what the most suitable weights are. We propose a novel mechanism, where the AF scheduling weights partly depend on the volume of reservation requests per AF class.

The other adaptive feature that we propose relates to reservation limits – if they are set high enough, admission decisions will be made based on link measurements only. This approach, "pure MBAC," has its risks. If a sudden "burst" of connection arrivals is experienced, there is nothing that MBAC can do – it will take some time until the updated link loads arrive at bandwidth broker. The solution is to make the reservation limits adaptive – they would depend on the current link loads. This will provide us with "safety margins."

In our modified bandwidth broker framework, we have added a CAC agent in all routing domain nodes (see Fig. 2). One of these CAC agents will act as the bandwidth broker by storing the information on reservations and measured link loads within the routing domain. Just like in [Sch98], the bandwidth broker knows the routing topology by listening to OSPF [Moy98] routing messages. Link bandwidths within the routing domain are obtained through the simple network management protocol (SNMP) [RFC1157].

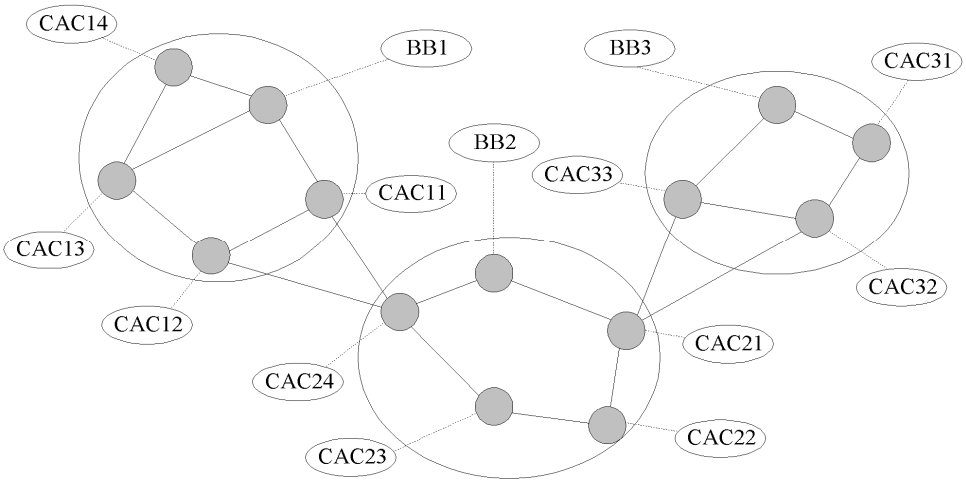

Fig. 2. Bandwidth brokers, CAC agents, and their routing domains.

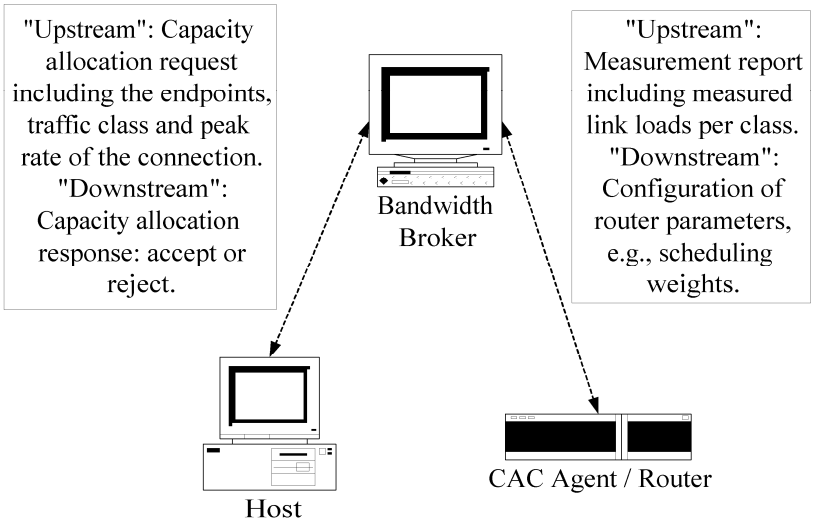

Fig. 3. Required signaling traffic in the proposed bandwidth broker framework. Periodic measurement reports provide the bandwidth broker with link loads and, thus, allow the use of MBAC.

In addition to reserved link capacities for different traffic classes, the admission decision is based on measured link loads on the path between the endpoints (see Fig. 3). If there is not enough unoccupied and unreserved bandwidth on the path, the connection is blocked. Note that the maximum reservable bandwidth on a link can exceed the link capacity. Thus, when the maximum reservable bandwidth is high enough, it is the unoccupied bandwidth only that matters. The relationship between the maximum reservable bandwidth and link bandwidth is configurable for each traffic class.

Available Bandwidth Calculation

In the proposed scheme, a CAC agent monitors and updates the loads of those links that are attached to its local router, while the bandwidth broker applies exponential averaging on the loads received from all CAC agents. The CAC agents send their current link loads periodically to the bandwidth broker. Exponential averaging weight (w), measurement period (p), and sampling period (s) should be carefully selected. The optimal values for w and p depend on traffic patterns and how fast we want to adapt to changes in link loads. During a single measurement period, the link loads are sampled p/s times, and at the end of each measurement period the maximum value is selected to represent the current load. Whenever a measurement report arrives at the bandwidth broker, the link database is updated by recalculating the applicable link loads and unoccupied link bandwidths for each traffic class

$$load_{class} = (1-w) * load_{class} + w * currentLoad_{class}, \tag{1}$$

$$unoccBw_{class} = bw * (loadLim_{class} - load_{class}). \tag{2}$$

Unreserved bandwidths are updated whenever a reservation is set up, modified, or torn down, while available bandwidths are calculated only when there is a resource request for a specific path

$$unresvBw_{class} = bw * (resvLim_{class} - resv_{class}), \tag{3}$$

$$avBw_{class,path} = \min_{link \in path} (\min(unoccBw_{class,link}, unresvBw_{class,link})), \tag{4}$$

where bw denotes the link bandwidth, $load_{class}$ the measured link load for a given class, and $resv_{class}$ the reserved link capacity for a given class.

For AF classes, the calculation of unoccupied bandwidth can be more complex. This is due to weighted scheduling between the AF queues. We can either configure the weights for all AF

queues in strict priority fashion and apply (2), or we can take the AF weights ($weight_{AFi}$) into account when calculating the unoccupied bandwidth values for each link

$$unoccBw_{AFi} = bw * \min(loadLim_{AFi} - load_{AFi}, 1 - load_{EF} - \frac{load_{AFi}}{weight_{AFi}}). \quad (5)$$

Flexible Connection Admission Control

If we pay too much attention to real-time application requirements, it may be impossible to use business [Kil02] or any other objectives in CAC decisions. In flexible CAC, real-time connections cannot claim all the bandwidth since link bandwidth between RT and NRT traffic is shared dynamically. Instead of a constant value, the load limit for RT traffic will be the minimum of total load limit less NRT traffic load and maximum RT load limit. Similarly, the load limit for NRT traffic will be the minimum of total load limit less RT traffic load and maximum NRT load limit. The total load limit is there in order to protect non-admission controlled traffic. We can also take the reserved link capacities into account in our admission decisions – reservation limits for RT and NRT traffic are calculated just like the load limits.

We can prioritize either PBAC or MBAC by tuning the maximum capacity that can be reserved for a given traffic class on a link ($resvLim_{class}$). If the reservation limit is low enough, it will be the PBAC that will rule. Fig. 4 illustrates the load/reservation limit hierarchy. Three limits can affect each admission decision: total limit, RT/NRT limit, and own class limit. However, each level in the hierarchy does not have to affect, i.e., we can, for example, set the NRT limit to equal the total limit. Note that a limit cannot exceed its parent class limit.

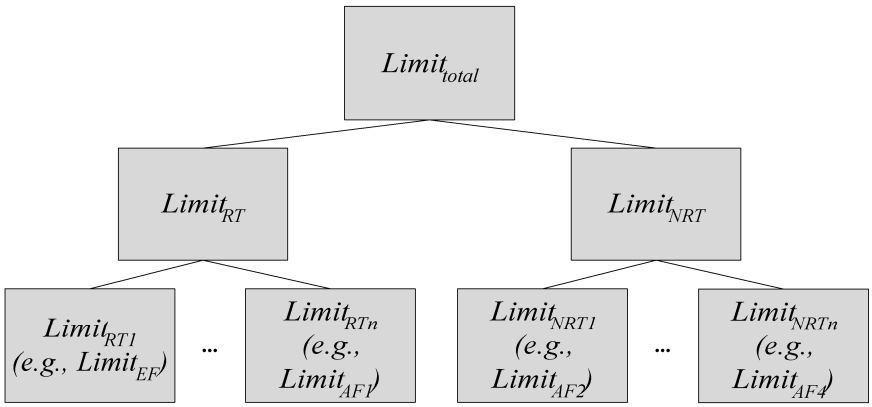

Fig. 4. Load/reservation limit hierarchy.

Probably the most practical way to apply flexible CAC is to configure all AF scheduling weights in strict priority fashion so that AF1 has the biggest weight – this results in delay differentiation between different AF classes. However, it is also possible to apply (5) for calculating the unoccupied bandwidths for AF classes. The latter method will most probably

result in lower admission ratios and resource utilization, but it may be useful when the goal of using AF is not delay differentiation but something else – like bandwidth sharing.

```
Bandwidth Broker:
for each admission request:
  classify connection (class = EF/AF1/AF2)
  admit = true
  if (class != AF2)
    calculate avBw_class, path and avBw_RT, path
    if ((avBw_class, path < requestedBw) OR (avBw_RT, path < requestedBw))
      admit = false
  else
    calculate avBw_NRT, path
    if (avBw_NRT, path < requestedBw)
      admit = false
  if (admit == true)
    for all links on the path:
      resv_class += requestedBw
      re-calculate unresvBw_class, unresvBw_RT, unresvBw_NRT

for each connection tear-down:
  classify connection (class = EF/AF1/AF2)
  for all links on the path:
    resv_class -= requestedBw
    re-calculate unresvBw_class, unresvBw_RT, unresvBw_NRT

for each measurement report arrival:
  update link database: re-calculate unoccBw:s

All CAC agents (including Bandwidth Broker):
timer expires:
  update link loads
  send update to Bandwidth Broker
  set timer to expire after p seconds
```

Fig. 5. Flexible CAC algorithm instance: admission decisions for RT (EF, AF1) and NRT (AF2) connections.

In flexible CAC, RT could denote, for example, the aggregate of EF and AF1 traffic classes. However, the scope of RT can be extended to cover more traffic classes. Similarly, NRT could include just AF2 traffic, but its scope can be extended to cover more traffic classes (see Fig. 4). Adjustable parameters are the following: $loadLim_{total}$, $loadLim_{RT\_MAX}$, $loadLim_{NRT\_MAX}$, $resvLim_{total}$, $resvLim_{RT\_MAX}$, $resvLim_{NRT\_MAX}$, and the load and reservation limits of individual traffic classes (e.g., EF, AF1, and AF2).

Fig. 5 illustrates how admission decisions are made in an example flexible CAC instance with three traffic classes. New connections request resources (peak rate from source to destination) from the bandwidth broker of their own routing domain. Other bandwidth brokers may have to be consulted as well if the destination is not in the same domain. If there are enough resources, the requested bandwidth for the admitted connection is added to reserved values for all links along the path. Otherwise, the connection is rejected. Policing is needed for all admitted flows to keep their peak bit rates below the agreed ones.

3.3 Adaptive AF Weight Tuning

Flexible CAC offers two operating modes for calculating the available bandwidth for AF classes: we can either have strict priority like AF weights and omit them in the calculation, or we can take the non-strict priority AF weights into account when calculating the available bandwidths. If we want to protect our non-admission controlled traffic (also, in shorter time scale), the latter mode is preferable.

In case we have two AF classes only, there is no need to tune the scheduling weights due to the fact that the other one, AF2, is the BE class. Thus, fixed weight allocations should be enough. With a more complex instance of flexible CAC, however, we might want to tune the AF1 and AF2 weights. If we give our "best effort" class, AF3, a fair share of forwarding resources, say 10%, it is no longer possible to have strict priority like weights (e.g., 90:9:1) for the three AF classes. Moreover, static normal AF weights could result in low bottleneck link utilization.

The AF weights are tuned individually for each link. The tuning process receives periodic input about the unoccupied AF bandwidths for every link within the bandwidth broker area. If certain thresholds are reached, new AF scheduling weights for the involved links and the CAC algorithm are calculated and taken into use.

The bandwidth broker monitors continuously the $unoccBw_{AFi}$ values. The minimum values from each link are stored into link database. After each periodical check, every T_W s, these values are reset. If certain thresholds were reached, new AF weights are applied for the involved links. If the $minUnoccBw_{AFi}/bw$ value is smaller than *lowThreshold* or larger than *highThreshold*, we shall update $weight_{AFi}$ for the link in question. After each measurement report arrival, it is checked whether $unoccBw_{AFi}$ is smaller than $minUnoccBw_{AFi}$. If that is the case, a new $weight_{AFi}$ is computed and stored,

$$weight_{AFi} = load_{AFi} / (1 - load_{EF} - unocc), \tag{6}$$

where *unocc* denotes the amount of unoccupied capacity that we would like to be always available. In general, *lowThreshold* should be less than *unocc*, which should be less than *highThreshold*. A negative value will immediately (after measurement report arrival) trigger AF weight tuning. Naturally, the final AF weights depend on the number of AF classes (*N*), excluding the "best effort" class

$$weight_{AFi} := weight_{AFi} / (\sum_{j=1}^{N} weight_{AFj}) * (1 - weight_{BE}). \tag{7}$$

3.4 Adaptive EF and RT Reservation Limit Tuning

One weakness of our admission control framework is that there is no protection against a sudden burst of connection arrivals. Of course, one could solve this problem with strict reservation limits. However, that would lead to low bottleneck link utilization. A better solution could be the use of adaptive reservation limits for real-time traffic. The goal of EF and RT reservation limit tuning is to achieve more stable link utilization, even in the presence of bursty connection arrivals and, thus, higher admission ratios.

The EF and RT reservation limits are tuned individually for each link. The tuning process receives periodic input about the EF and RT loads for every link within the bandwidth broker area. If certain thresholds are reached, new reservation limits are calculated and taken into use.

The bandwidth broker monitors periodically (every T_R seconds) the $load_{EF}$ and $load_{RT}$ values of each link. If $load_{class}$ does not fall into the desired interval, $resvLim_{class}$ for the link in question shall be updated. A parameter called *increment* denotes the amount of capacity by which we can increment or decrement the reservation limit. If the reservation limit is too low compared with the actual link usage, the reservation limit will be increased. Similarly, if the reservation limit is too high compared with the actual link usage, the reservation limit will be decreased. It should be noted that we are by no means disabling the measurement-based part of our admission control scheme – connections can be blocked because of exceeded load threshold already before it would be time to adjust the reservation limit. Fig. 6 illustrates how the proposed algorithm can be implemented.

```
Bandwidth Broker:
timer expires:
  for each link:
    if (load_EF < (loadLim_EF - increment))
      resvLim_EF = resv_EF + increment
    if (load_EF > (loadLim_EF + increment))
      resvLim_EF = resv_EF - increment
    if (load_RT < (loadLim_RT - increment))
      resvLim_RT = resv_RT + increment
    if (load_RT > (loadLim_RT + increment))
      resvLim_RT = resv_RT - increment
  set timer to expire after T_R seconds
```

Fig. 6. EF and RT reservation limit tuning algorithm.

3.5 Performance Evaluation

In order to find out packet loss rates with different traffic loads and algorithms as well as understand the dynamics of the proposed algorithms, we use a modified version of the ns-2 simulator [NS2]. Six simulations with different seed values are run in each simulated case in order to achieve small enough 95% confidence intervals. Simulation time is always 1200 seconds, of which the first 600 seconds are discarded as warming period. All cases are

simulated with eight connection arrival intensities. We use a flexible CAC instance with three classes: EF, AF1, and AF2 (EF and AF1 belong to RT superclass). Admission control parameters are listed in Table I, while the simulation topology is illustrated in Fig. 7.

Network Parameters

All routers implement the standard PHBs; EF is realized as a priority queue and AF with a DRR system consisting of three queues. The EF queue is equipped with a token bucket rate limiter having a rate of 0.8 link bandwidth and a bucket size 4500 bytes. Default, strict priority like, quanta for AF1, AF2, and AF3 queues are the following: 1800, 180, and 20 bytes. Queue sizes are given in kilobytes: 5 for EF, 15 for AF1, 20 for AF2, and 25 for AF3. AF packets with drop precedence level of three and two are dropped already when the corresponding AF queue filling level is 0.767 and 0.883, correspondingly.

TABLE I
ADMISSION CONTROL PARAMETERS

Parameters	SP like AF weights	Normal AF weights	Adaptive AF weights
$weight_{AF1}$	0.9	0.45	adaptive
$weight_{AF2}$	0.09	0.45	adaptive
$weight_{AF3/BE}$	0.01	0.1	0.1
T_W	N/A		10.0 s
lowThreshold	N/A		0.05
highThreshold	N/A		0.15
unocc	N/A		0.1
T_R	N/A or 10.0 s (w. EF/RT res. limit tuning)		
increment	N/A or 0.05 (w. EF/RT res. limit tuning)		
$resvLim_{EF}$	10.0 or adaptive (w. EF/RT res. limit tuning)		
$resvLim_{RT\ MAX}$	10.0 or adaptive (w. EF/RT res. limit tuning)		
$resvLim_{AF1}$	10.0		
$resvLim_{AF2}$	10.0		
$resvLim_{NRT\ MAX}$	10.0		
$resvLim_{total}$	10.0		
$loadLim_{EF}$	0.5		
$loadLim_{AF1}$	0.5		
$loadLim_{AF2}$	0.9		
$loadLim_{RT\ MAX}$	0.9		
$loadLim_{NRT\ MAX}$	0.9		
$loadLim_{total}$	0.9		
s	500 ms		
p	1.0 s		
w	0.5		

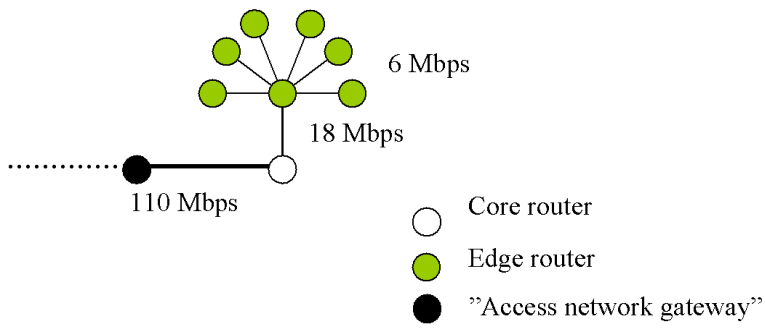

Fig. 7. Access network topology.

Traffic Characteristics

Connections are set up between the access network gateway and edge routers. Bursty connection arrivals are created with a two-state Markov chain, where the transition probabilities from normal state to burst state and vice versa are both 0.1. The decision on next state is triggered by a connection arrival, i.e., upon each connection arrival the state of the system may change with the stated probability. In the normal state, new connections arrive at each edge router with exponentially distributed interarrival times. However, in the burst state, the interarrival time is always zero. Holding times are exponentially distributed with a mean of 100 seconds for RT (EF and AF1) connections and 250 seconds for other connections. Our traffic mix consists of VoIP calls, videotelephony, video streaming [Mag88], web browsing [Mol00], and e-mail downloading [Bol99]. There are three different service levels within each AF class – their selection is based on subscription information, however, they do not have any effect on admission decisions. Signaling traffic between the bandwidth broker and the CAC agents is modeled in semi-realistic fashion. CAC agents do send real-router load reports to bandwidth broker but resource requests and replies are modeled in a statistical fashion. Bandwidth broker is located at the gateway that connects the access network to service provider's core network. Service mapping is done according to Table II. Simple token bucket policers are used to limit the sending rates of admitted sources.

TABLE II
TRAFFIC MIX AND SERVICE MAPPING

Service	Service level	PHB	Share of offered connections	Requested bandwidth (peak rate)
VoIP calls	N/A	EF	20.0%	36 kbps
Videotelephony	N/A	EF	20.0%	84 kbps
Video streaming	Gold	AF11	4.0%	250 kbps
	Silver	AF12	4.0%	250 kbps
	Bronze	AF13	4.0%	250 kbps
Guaranteed browsing	Gold	AF21	8.0%	250 kbps
	Silver	AF22	8.0%	250 kbps
	Bronze	AF23	8.0%	250 kbps
Normal browsing and e-mail downloading	Gold	AF31	8.0%	N/A
	Silver	AF32	8.0%	N/A
	Bronze	AF33	8.0%	N/A

Simulation Results

The loss-load graph of Fig. 8 illustrates the main results of our simulations. Bottleneck link utilization is maximized either with adaptive or strict priority like AF weights. Without adaptive reservation limits, AF1 packet loss can momentarily be prohibitive due to bursty connection arrivals. AF1 packet loss is minimized when reservation limit tuning is used together with strict priority like AF weights. With normal AF weights, AF packet loss is somewhat higher. Moreover, AF packet loss is decreased also in the case where AF weights are tuned in conjunction with the reservation limits. This is a nice result, since it proves (although informally) that the two tuning processes are not disturbing each other.

Fig. 9 illustrates how the admission-controlled load develops as a function of connection arrival intensity. The dynamic weights for AF1 and AF2, as well as reservation limits for EF and RT are illustrated Figs. 10 and 11, correspondingly. The purpose of these two graphs is just to illustrate how AF weights and reservation limits are tuned.

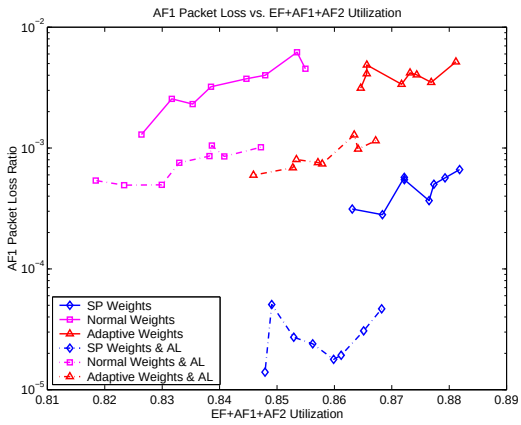

Fig. 8. AF1 loss versus EF+AF1+AF2 load.

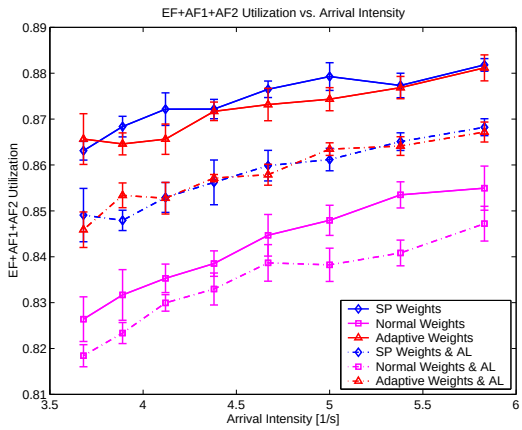

Fig. 9. EF + AF1 + AF2 load versus connection arrival intensity.

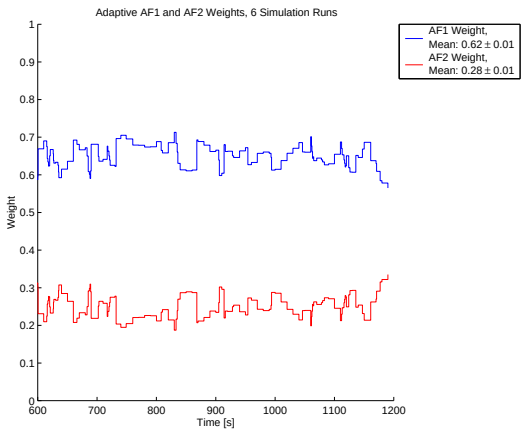

Fig. 10. Adaptive AF1 and AF2 weights.

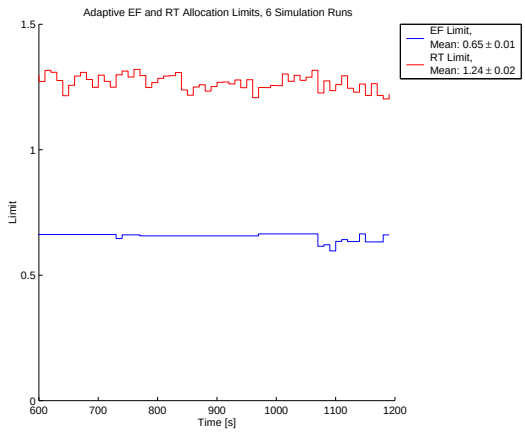

Fig. 11. Adaptive EF and RT allocation limits.

3.6 Summary

In this chapter, we introduced a modified bandwidth broker framework that utilizes link load information in admission decisions. On top of that, we also presented and evaluated algorithms for adaptively adjusting the AF weights and the reservation limits. Simulations were used for validating the proposed ideas.

There is a need for non-strict priority like AF weights – the main motivation arises from the desire to protect non-admission controlled traffic. Thus, AF weights should be taken into account in the admission control algorithm. Simulations show that static, non-strict priority like AF weights result in a lower bottleneck link utilization than adaptive AF weights. Naturally, the poor performance experienced with static AF weights means that the weights were inappropriate considering the traffic mix. However, the ideal scheduling weights cannot be known beforehand.

Adaptive reservation limits are an effective way to protect oneself against bursty connection arrivals and still maintain high bottleneck link utilization. The tuning of EF and RT reservation limits seems to lower the bottleneck utilization a little. That is the price one has to pay for the "safety margins."

The signaling overhead is very low, both in the basic framework and in the adaptive enhancements. Thus, there is no real trade-off between the bottleneck link utilization level and signaling traffic.

4 Active Queue Management in WCDMA Networks

Universal mobile telecommunications system (UMTS) is one of the third-generation (3G) mobile telecommunications technologies. The most common form of UMTS uses wideband code division multiple access (WCDMA) as the underlying air interface. UMTS and its use of WCDMA are standardized by the third-generation partnership project (3GPP).

The scope of this chapter is to study the end-to-end performance of TCP-based radio access bearers (RAB) in a WCDMA network, with or without AQM in the radio network controller (RNC). We shall take high speed downlink packet access (HSDPA) into account, since the effective air interface bandwidth that a user gets may, in that case, vary more dynamically than in the case of dedicated channels (DCH). However, it should be noted that HSDPA flows have dedicated buffers in the RNC, too. DCH flows usually have more constant data rates than HSDPA flows. Nevertheless, these (TCP-based) flows may experience long buffering delays in the RNC, as the downlink direction is more likely to be a bottleneck than the uplink direction. Thus, techniques that aim at keeping the buffer size small are useful in both DCH and HSDPA cases. In Publication P2, we present a number of existing AQM techniques adapted or tailored for the RNC, along with our own proposal, TTLRED, where the average queue size (that is normally used in packet dropping decisions) is replaced with the packet lifetime.

First, we describe how packets are buffered at the RNC. Then we present existing and novel solutions for reducing the buffering delays. Finally, simulations are used for comparing the performance of different AQM solutions. The simulation results indicate that our relatively simple TTLRED performs as well as the other schemes that have more configurable parameters.

4.1 WCDMA Network and RNC

RNC is responsible of RAB admission control and radio resource management, i.e., allocating bit rates to RABs – in the case of both dedicated and shared radio channels [TS23.107]. Gateway GPRS support node (GGSN) and serving GPRS support node (SGSN) are consulted in RNC's admission control decisions. Moreover, RED [Flo93] and different scheduling weights are applied for different traffic classes. However, we consider neither SGSN nor GGSN as a system bottleneck in these studies. Fig. 12 illustrates the WCDMA packet data user plane protocol stacks that are used between the different network elements.

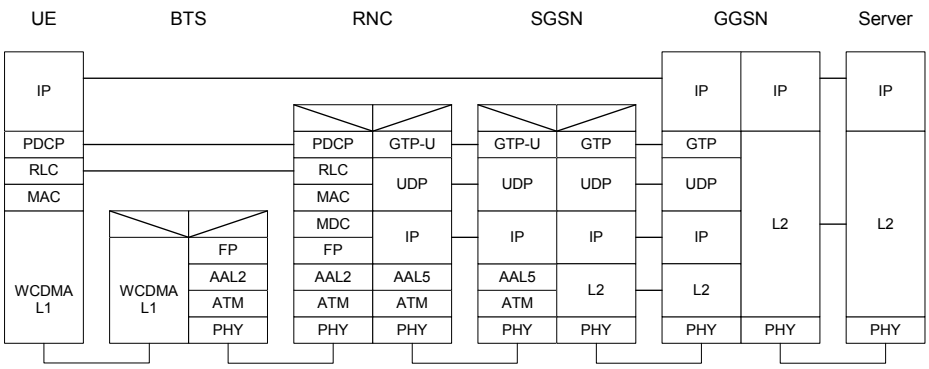

Fig. 12. WCDMA packet data user plane protocol stacks.

PDCP and RLC Buffers

In the RNC, there are dedicated packet data convergence protocol (PDCP) [TS25.323] and radio link control (RLC) [TS25.322] buffers for each RAB. The maximum per-RAB PDCP buffer length usually depends on the traffic class. For example, 10 kB might be used for the conversational class and 30 kB for all other traffic classes. Naturally, there has to be a limit for the total amount of buffer memory. However, we do not consider that as a bottleneck.

Unacknowledged mode (UM) and acknowledged mode (AM) RLC on dedicated traffic channel (DTCH) have fixed-size buffers. The default size for both transmission and receiving buffer is the maximum AM window size times the maximum protocol data unit (PDU) size. The same buffer size is used for UM RLC. For example, the following values might be used with DTCH: maximum window size of 768 PDUs and maximum PDU size of 42 B.

It should be noted that the same data is stored in both PDCP and RLC buffers. This enables the use of various features in PDCP buffering, e.g., GPRS tunnelling protocol (GTP) packet reordering. Moreover, features that target in enhancing the performance of TCP can be implemented with the help of PDCP buffering.

RLC/MAC Protocols

In transparent mode (TM) and UM RLC, higher layer packets are simply segmented and equipped with appropriate overhead before they are sent to the UE or the RNC. In AM RLC, however, RLC frames are not cleared from the retransmission buffer until they have been acknowledged. Moreover, the higher layer packet cannot be cleared from the PDCP buffer until its final RLC frame is cleared from the RLC buffer [TS25.322].

Acknowledgements can be polled in many different ways, e.g., by using counters and timers. When an RLC frame with the poll bit is received, the receiver answers either with a standalone acknowledgement or a piggybacked one. The standalone acknowledgements are put to the tail

of the RLC buffer while the piggybacked acknowledgement information is added to the first non-pending RLC frame in the RLC buffer.

When an acknowledgement is received, the retransmission buffer is cleared from those frames whose identifier is found from the acknowledgement. All other pending RLC frames, whose identifier is lower than the highest identifier in the acknowledgement, are considered lost and they need to be retransmitted. Naturally, there is a limit for the number of retransmissions.

HSDPA and its Flow Control

HSDPA [TS25.308] is a concept within WCDMA specifications whose main target is to increase user peak data rates and QoS, and to generally improve spectral efficiency for downlink asymmetrical and bursty packet data services.

When implemented, the HSDPA concept can co-exist on the same carrier as Release'99 WCDMA services. Furthermore a user can download packet data over HSDPA, while at the same time having a speech call. HSDPA offers theoretical peak data rates on the order of 10 Mbps and in practice more than 2 Mbps.

Compared to the Release'99 architecture, HSPDA introduces a short 2 ms transmission time interval (TTI), adaptive modulation and coding (AMC), multicode transmission, fast physical layer (L1) hybrid ARQ (H-ARQ), and it moves the packet scheduler from the RNC to the Node-B where it has easy access to air interface measurements. The latter facilitates advanced packet scheduling techniques, meaning that the user data rate can be adjusted to match the instantaneous radio channel conditions [Wig03].

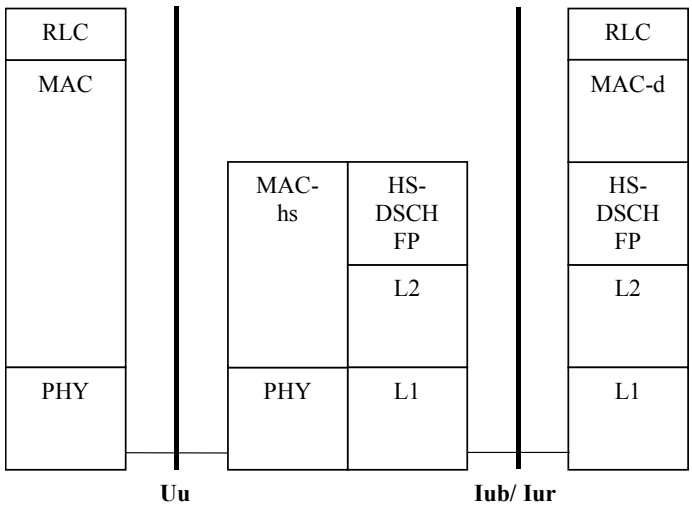

Fig. 13. Protocol architecture of HSDPA [TS25.308].

In all HSDPA flow control [TS25.435] implementations, it is the MAC-hs at Node-B that is controlling the flow over the Iub interface (see Fig. 13), i.e., MAC-hs allocates a certain number of credits per time unit for each HSDPA buffer in the RNC. Each credit allows HS-DSCH frame protocol in the RNC to send a single MAC-d [TS25.321] PDU during the allocation interval. The implementation details of HSDPA flow control are left to the network equipment manufacturers.

HSDPA frame protocol is used between the RNC and the UE: 1 to 255 MAC-d frames are packed into a single FP frame every 10 ms (other TTIs can be used as well). The number of MAC-d frames in a single HSDPA FP frame naturally depends on flow control and on the number of buffered MAC-d frames [TS25.877].

4.2 Active Queue Management

Here we describe how some well-known AQM mechanisms from the IP world could be applied in the RNC in order to reduce buffering delays. These AQM mechanisms have been introduced in Chapter 2.

Random Early Detection

Probably the best-known method for TCP performance enhancement, RED (and ECN) [Flo93, RFC3168] aims at preventing global TCP synchronization by dropping random packets when the averaged queue size exceeds the minimum threshold. The bigger the averaged queue size, the bigger the probability of a packet drop. However, RED does not usually work well if there is only a single flow or a couple of flows sharing the buffer [Såg03]. In such a case, the buffer occupancy will vary considerably, and we are forced to use the instantaneous queue length instead of a slowly averaged one. This is exactly the case with RNC and its PDCP buffers. Thus, other possible methods should be tested, too.

Adaptive RED Thresholds Based on RAB Rates, ARED

Static RED parameters can lead to decreased TCP goodput or too high IP packet delays. Variable *minTh* and *maxTh* would thus be preferred over static values. In the RNC we can simply utilize the RAB rate (using, e.g., the number of HSDPA credits received from the Node-B) and then compute *minTh* by multiplying the RAB *rate* [B/s] by desired *delay* (e.g., 0.5 s),

$$minTh = \max(7500, delay * rate). \tag{8}$$

Minimum *minTh* is limited to 7500 bytes. Maximum threshold is then given by a widely used rule of thumb [Flo],

$$maxTh = 3 * minTh. \quad (9)$$

4.3 Related Work

In [Såg03] Sågfors, Ludwig, Meyer, and Peisa have presented a technique called packet discard prevention counter (PDPC). They propose a deterministic packet dropping mechanism using a counter: after a packet drop we have to accept *N* (default: 20) packets before next packet drop can take place – unless the maximum threshold is exceeded. This technique assumes that we have adaptive *minTh* and *maxTh*. Details, however, are not provided in [Såg03]. Thus, we shall simply utilize RAB rate here as presented in the previous section. Drop from tail shall be used instead of drop from front (suggested in [Såg03]), since drop from front would be difficult to implement in the RNC if the IP packet is already segmented into RLC blocks and given a sequence number (and not at all feasible if RLC blocks have already been sent).

4.4 TTLRED

As an alternative to PDPC and RED, we propose an active queue management mechanism, where there is no need to estimate the RAB rates. In our proposal, TTLRED, we provide the IP packets with timestamps as they enter the PDCP buffer. This is somewhat similar to assigning PDU lifetimes (the remaining time period that the PDU is considered as valid) in the 2G-SGSN [TS48.018] and the lifetime packet discard idea proposed by Gurtov and Ludwig [Gur03]. Those schemes, however, have very little to do with AQM. Next, we present two alternative schemes that utilize the timestamp.

TTLRED for Incoming Packets (TTLRED1)

In our first scheme, when applying TTLRED for incoming packets, we simply replace the averaged queue size in the gentle RED algorithm [Flo] (see Fig. 14) with the packet lifetime. Whenever a packet arrives, we find the packet with the highest lifetime (current time less timestamp) from the PDCP buffer. Drop counter is also utilized; only every N^{th} packet can be dropped (or marked if we use ECN) – even if the high threshold is exceeded.

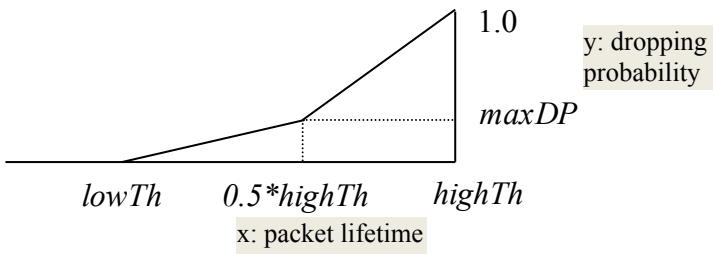

Fig. 14. TTLRED: dropping probability as a function of packet lifetime.

TTLRED for Buffered Packets (TTLRED2)

Another alternative is to utilize timers. In this approach, the packets are given a random dropping (or marking) time when they enter the PDCP buffer. Packet marking algorithm is illustrated in Fig. 15 and it is somewhat analogous to gentle RED [Flo]. Moreover, a higher lifetime is always assigned when the buffer is empty or whenever a new TCP flow is detected, as it could be a SYN packet.

```
if (U(0, 1) < maxDP)
  lifetime := U(lowTh, 0.5*highTh)
else
  lifetime := U(0.5*highTh, highTh)
```

Fig. 15. TTLRED packet marking algorithm.

The packets are checked every *T* seconds (e.g., 10 ms). If current time exceeds the packet timestamp, the packet is dropped (or marked if ECN is used and the flow supports it). Since some packets are stored both in the RLC buffer (size usually around 30 kB) and in the PDCP buffer simultaneously, TTLRED will remove only such PDCP packets that do not have any corresponding RLC blocks in flight yet.

4.5 Performance Evaluation

In order to compare the DL delays and TCP goodputs of different AQM algorithms, we use a modified version of the ns-2 simulator [NS2]. Six simulations are run in each test case in order to get small enough 95% confidence intervals. Simulation time is 1200 seconds. The different AQM mechanisms are tested under numerous different conditions. We vary the downlink bearer rate (from 64 kbps to 3.6 Mbps, DCH or HSDPA; uplink is always a 64 kbps DCH RAB), number of parallel TCP connections (one or three), TCP's advertised window (30 or 60 packets) and PDCP buffer size (30 kB or 100 kB). Moreover, different file sizes are tested (250 kB vs. 2.5 MB) in the case of single TCP connection.

Simulation Parameters

One-way core network delay between server and RNC is set to 70 ms. The only bottleneck in our system is the RNC; there are no packet drops or variable delays elsewhere in the system. AM RLC with polling- and timer-based retransmissions as well as duplicate detection and in-sequence delivery are used. HSDPA flow control is strongly simplified: the radio capacity is divided equally among all currently active flows and multiplied by 0.9. For DCH air interface packet loss, a simple Gilbert model [Gil60] is used. Average packet loss rate is 1.5%. (Since AM RLC with retransmissions is used, packet loss on the air interface is seen as additional delay.) This model is not applied with HSDPA but local retransmissions are modeled as

additional delay: with 10% probability a packet has to wait until the next scheduling period, i.e., 0 – 10 ms.

For static RED (as well as for ARED and PDPC, where applicable), we apply the following parameter values: *minTh* = 10 kB (33.3 kB with 100 kB PDCP buffer), *maxTh* = 30 kB (100 kB with 100 kB PDCP buffer), *maxDP* = 0.1, w_{AQS} = 1.0, and drop from the tail.

For TTLRED, we apply the following parameter values: *lowTh* = 0.25 s (TTLRED1) or 0.5 s (TTLRED2), *highTh* = 1.5 s (TTLRED1) or 3.0 s (TTLRED2), and *maxDP* = 0.1.

We simulate two kinds of TCP traffic: web browsing and "peer-to-peer" file downloading. For web browsing, we use HTTP/1.0 with four parallel TCP connections, whereas in file downloading, HTTP/1.1 [RFC2616] with a single TCP connection and pipelining is used. In both traffic models, session length equals simulation time and page reading time (or time between two file downloads) is six seconds. In the web browsing model, we download the main page and three inline objects per page; all these objects have a size of 65 kB. File size in the file downloading model is either 250 kb or 2.5 MB, depending on the case. In both models, we utilize TCP NewReno [RFC2582] with the ns-2 default values.

Simulation Results

Figs. 16–18 illustrate the main results (95th percentile delay and TCP goodput) of our simulations. From these, we make the following observations.

The main benefit of AQM is a decreased delay with lower RAB rates. This can be important if the same RAB carries different flows, some of which are delay-sensitive. Large PDCP buffers, narrow bandwidth, big advertised windows and tail dropping can lead to long delays.

PDCP buffers should be large enough for maximized TCP goodput. Bandwidth-delay product formulas (see, e.g., [App04]), however, should be used with caution, as PDCP buffers are not like IP router buffers (AM RLC, HSDPA flow control). 100 kB of buffer space seems to be enough even for HSDPA RABs (assumed RTT of 80 ms and theoretical maximum bandwidth of 10 Mbps). If the PDCP buffers are large enough, AQM does not increase TCP goodput, however, things would most probably be different with very small PDCP buffers (see, e.g., [Såg03]). AQM may actually slightly decrease TCP goodput, but this depends on the selected parameters.

Static RED may unnecessarily lower TCP goodput with high RAB rates, especially with big files (see Fig. 18). ARED, PDPC, and TTLRED give the best results.

TTLRED for incoming packets can result in higher delays than the other AQM schemes, since there is no protection against bigger bursts (see Fig. 17). We also noticed that if the parameters are non-optimal, TTLRED for buffered packets could result in lower TCP goodput especially with the following combination: high bandwidth, multiple TCP flows per RAB, big buffers, and big advertised windows. With multiple flows, we could think of downgrading the advertised windows somewhat – but this kind of action would violate the end-to-end principle.

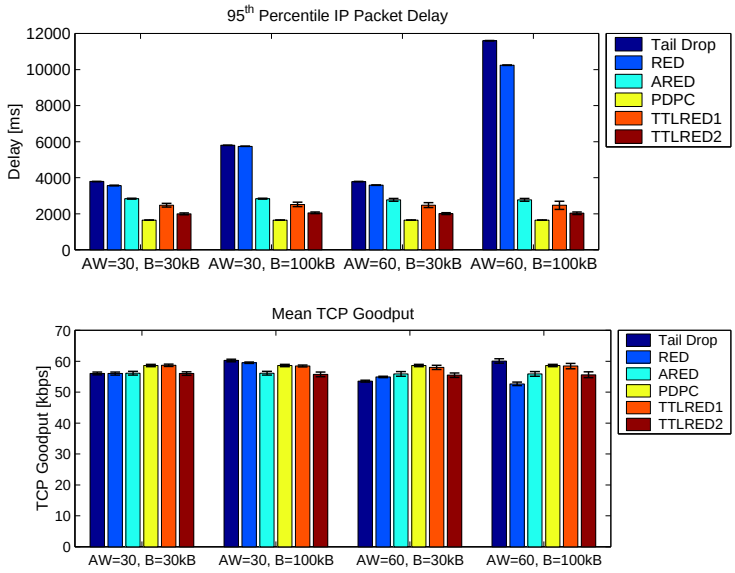

Fig. 16. R99 64 kbps, single flow: delay and goodput.

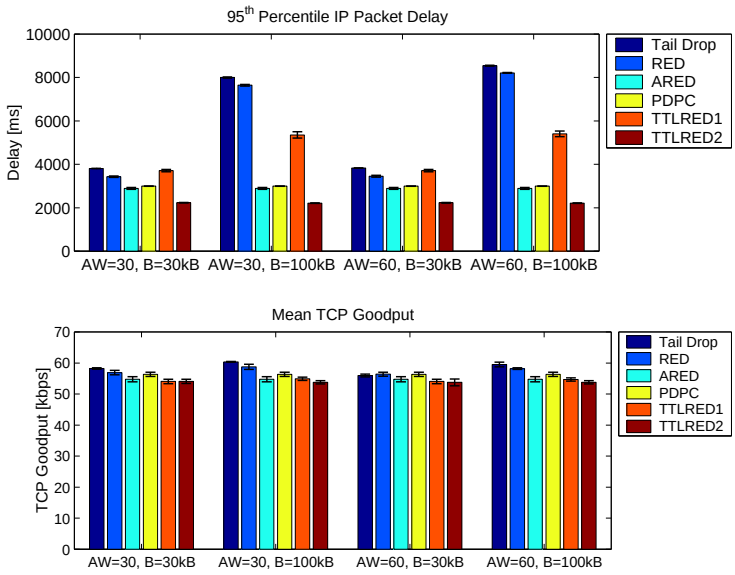

Fig. 17. R99 64 kbps, four flows: delay and goodput.

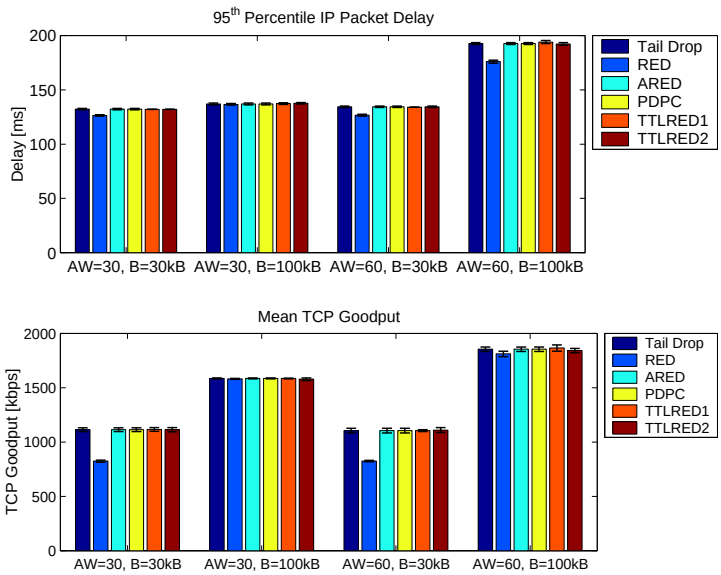

Fig. 18. HSDPA 3.6 Mbps, single flow, 2.5 MB files: delay and goodput.

4.6 Summary

We studied the end-to-end performance of a single RAB, with single or multiple TCP flows, in a WCDMA network using four different AQM schemes. We have shown that AQM is useful in reducing the queuing delays caused by the PDCP buffering. Providing enough buffer space, on the other hand, will maximize TCP goodput.

Static RED (or most probably any AQM scheme with static parameters) is not enough but we need to adjust the parameters according to the RAB rate. ARED and PDPC both utilize the RAB rate in their minimum/maximum threshold setting and they clearly outperform tail drop (i.e., no AQM). Nevertheless, it may not be straightforward to obtain the rate information for PDPC and ARED. Moreover, determining the parameters on the basis of the rate information is not easy either.

Our proposal, TTLRED, does not need any rate information, which is a very nice property. The average queue size that is normally used in packet dropping decisions is replaced with the packet lifetime. Moreover, the delay/goodput performance of TTLRED is more or less the same as the performance of ARED or PDPC. It is also worth noting that TTLRED can be used with adaptive buffer size, i.e., when the buffer starts to fill up, we can allocate more buffer space to the RAB in question. Nevertheless, further research (on TTLRED for incoming packets) may still be needed, e.g., in order to deal with sudden packet bursts in a more sophisticated way than tail drop.

5 Active Queue Management in EGPRS Networks

General packet radio service (GPRS) is a packet oriented mobile data service of the global system for mobile communications (GSM). Enhanced GPRS (EGPRS), also known as enhanced data rates for GSM evolution (EDGE), allows improved data transmission rates. EDGE is standardized by the 3GPP.

The 2G-SGSN acts as a buffer for the GPRS/EGPRS radio access network by temporarily holding downlink packets instead of forwarding them immediately if the base station controller (BSC) is not able to receive them because of, e.g., a lack of its own buffer space. The main benefit of this approach is to avoid placing too high memory requirements on the base station subsystem (BSS) network elements. However, it also means that in loaded conditions, the 2G-SGSN will be the main element in charge of handling excess downlink traffic in the radio access network (RAN). In other words, the 2G-SGSN downlink buffer is a potential traffic bottleneck, since overload may not only be caused by the 2G-SGSN or the Gb interface capacity limitations but also by cell or even mobile station (MS) congestion, which are indeed more common cases.

Measurements performed in live GPRS/EGPRS networks typically confirm that the end-to-end latency grows with the network load. Thus, some efficient mechanisms to control or reduce buffer delays for non-real time traffic are needed in 2G-SGSN to optimize both the end-user experience and the spectral efficiency. It should be noted that although the same observation applies to any other core network element (e.g., GGSN, 3G-SGSN, or backbone routers), the buffer delay issue is typically most acute, and also in a way specific to 2G-SGSN because of the standard flow control between the radio and the core network domains. Therefore, specific non-classical approaches to solving this buffer delay problem are worth investigating.

In this chapter, we review the work in Publication P3, where we present a number of well-known techniques for reducing the buffering delay in 2G-SGSN. We also introduce our own proposal, TTLRED, where the average queue size (that is normally used in packet marking or dropping decisions) is replaced with the packet lifetime.

5.1 Flow control in EGPRS

There are three different levels in EGPRS flow control (see Fig. 19). The first one is the BVC (BSSGP, i.e., base station subsystem GPRS protocol, virtual connection) flow control, which refers to the cell level. If the available buffer space in the BSC reserved for a particular BVC gets below a certain threshold, the BSC will signal the 2G-SGSN to reduce its sending rate for the traffic accessing that BVC. The second level is the MS specific flow control. Again, if the

available memory in the BSS reserved for a particular MS gets too low, the 2G-SGSN will reduce the sending rate for that particular MS. The last (optional) level is the packet flow context (PFC) that handles flows within a certain MS that have specific QoS requirements.

As specified in [TS48.018], the 2G-SGSN will apply these flow control tests to every logical link control (LLC) PDU. The flow control is performed on each LLC PDU first by the PFC flow control mechanism (if applicable and negotiated), then by the MS flow control mechanism and last by the BVC flow control mechanism.

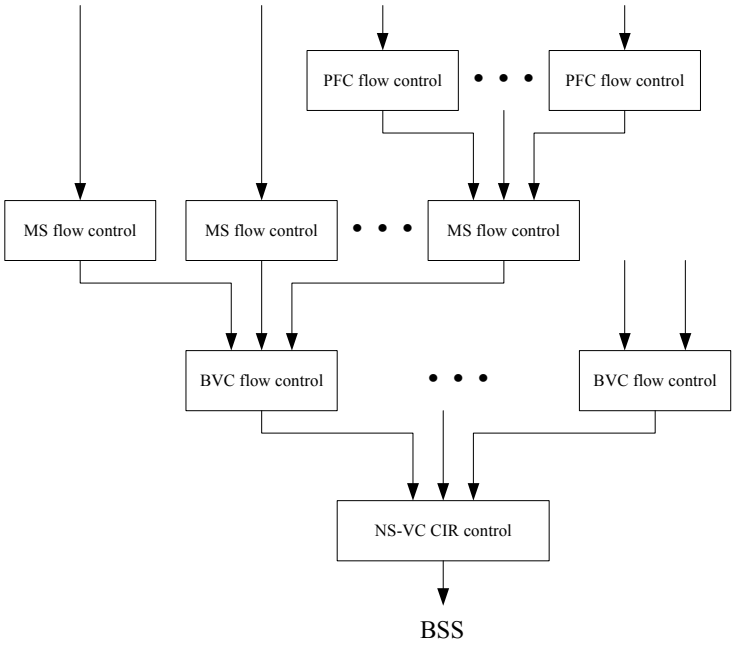

Fig. 19. Flow control levels in the 2G-SGSN applied to every LLC PDU.

This flow control approach has the benefit that it prevents downlink traffic overflow (i.e., packet drops) in the BSS and it ensures that a certain congested cell, MS or PFC, will not create unnecessary downlink buffer delay in the 2G-SGSN for other flows accessing non-congested cells, MSs or PFCs.

One way to deal with potential buffer delay is to prioritize the traffic based on how delay sensitive it is. In 2G-SGSN, the traffic from different traffic classes may be handled in separate buffers. A weighted fair queuing scheduler may then allocate a certain share of the output capacity to each buffer. Although QoS-based queuing and scheduling may lower or even eliminate buffer delays for highest priority classes (i.e., real time traffic), lowest priority classes (i.e., non-real time traffic) are then even more likely to experience long delays (depending on the traffic mix). Thus, the need for efficient AQM schemes for TCP-based traffic is even more critical.

5.2 Active Queue Management Schemes

The first way to control buffer delays in the 2G-SGSN is to introduce a pre-defined lifetime for LLC frames. The idea is very simple: After having spent a certain pre-defined time in the 2G-SGSN and/or BSC buffers, the LLC frame will be discarded. Such a mechanism is available by default in most router-like network elements in order to ensure that too old packets are removed. In the 2G-SGSN, this scheme can also help to guarantee a certain maximum buffer delay depending on the traffic class considered. The objective is to find the right trade-off between high resource utilization and optimized end-user throughput. For instance, network utilization may be affected if the packet lifetime is set too low. On the other hand, a too large lifetime may degrade TCP goodput because of high latency, which may result in TCP timeouts and retransmissions.

Splitting LLC Frame Lifetime

In order to avoid unnecessary packet drops at the BSS, LLC frames successfully sent from the 2G-SGSN to the BSS should be given at least a pre-defined minimum lifetime, i.e., the total LLC frame lifetime should be split between the 2G-SGSN and the BSS. Moreover, since flow control cannot provide any delay bounds (and there is no active queue management at the BSS), we should also introduce a pre-defined maximum lifetime for LLC frames at the BSS.

Smaller Buffer Sizes

Another way to limit buffer delays is simply to limit the buffer size. It very much resembles the previous approach although in this case the output interface speed shall be known in order to predict the maximum buffer delay. What complicates things in this respect in 2G-SGSN is the multi-layer flow control presented earlier. For instance, although the output link speed of the 2G-SGSN would allow forwarding immediately the received packets, some packets may have to be buffered because the BSC is not able to accept them. Thus, extracting a maximum buffer delay out of the 2G-SGSN buffer size is not easy. Moreover, the 2G-SGSN buffers tend to be big, which suggests that it is not wise to rely on the maximum buffer size in order to control the buffer delay. However, it should be noted that buffer sizes may be configured differently in the 2G-SGSN for each traffic class.

Random Dropping/Marking Based on Buffer Occupancy

A third, more advanced, approach is to randomly drop packets before the buffer gets full or before the packet lifetime expires. The RED algorithm [Flo93] drops arriving packets probabilistically. The probability of packet drop increases as the estimated average queue size grows. RED responds to a time-averaged queue length, not an instantaneous one. Thus, if the

queue has been mostly empty in the "recent past", RED is not likely to drop packets (unless the queue overflows). On the other hand, if the queue has recently been relatively full, indicating persistent congestion, newly arriving packets are more likely to be dropped.

ECN has later been introduced as an improvement of RED. As stated in [RFC3168], ECN allows a TCP receiver to inform the sender of congestion in the network by setting the ECN-Echo flag upon receiving an IP packet marked with the congestion experienced (CE) bit(s). The TCP sender will then reduce its congestion window. Thus, the use of ECN is believed to provide performance benefits.

RED/ECN implementation could, in principle, take into account the 2G-SGSN multi-layer flow control mechanism (see Fig. 19). That is, one instance of RED/ECN could be applied to each independent flow control entity. As an illustration, if the RED threshold in the 2G-SGSN buffer is exceeded mostly because of a few congested cells (BVC flow control), it does not necessarily mean that packets accessing other, non-congested cells (buffered in the 2G-SGSN for other reasons, e.g., because of Gb capacity limitation) should be randomly dropped by the same rules. Likewise, if a detected 2G-SGSN buffer congestion is mostly due to a few MSs, RED may not need to be applied on other non-congested MSs.

Although all the packets may not be buffered in the 2G-SGSN for the same reasons (MS vs. BVC vs. Gb congestion), all the aforementioned reasons indicate some sort of congestion. A multi-layered RED/ECN approach in the 2G-SGSN would probably add significant complexity and require additional CPU and memory resources, while the practical performance gains are not so clear. As a conclusion, our view is that the potential performance gains of applying one separate instance of RED/ECN to each independent flow control entity do not justify the required extra complexity.

5.3 Random Dropping or Marking Based on Packet Lifetime

Our proposal for congestion management at the 2G-SGSN, is to follow a time-to-live (TTL) based RED approach since, as explained above, it is not straightforward to relate 2G-SGSN buffer occupancy and buffer delay. The motivation for TTL-based RED is the fact that Gb buffer usually stores packets destined to different cells. Some cells can be more congested than others. When RED is enabled, random packet dropping is applied for all packets using that buffer – even if their destination cells are very lightly loaded. Instead of averaged queue size we shall use the packet lifetime as a basis for random packet dropping.

There are two possible implementations for a TTL-based RED approach. In the first one, the packet is checked periodically (every T seconds) and if the age of the packet (current time less

timestamp) exceeds a threshold (*lowTh*), the packet is randomly dropped[4] if one of the following conditions holds:

- $age \geq highTh$
- U(0, 1) < ((*age* - *lowTh*)/(*highTh* - *lowTh*))**maxDP*

This is very similar to RED: at *lowTh* we start to apply random packet dropping, *highTh* is the maximum packet lifetime, *maxDP* is the maximum drop probability and U(0, 1) stands for a uniformly distributed random variable. The probability that a given packet is dropped at the Nth TTL-check or earlier is given by (10).

$$\sum_{i=1}^{N}\left[\prod_{j=1}^{i-1}\left(1-\frac{j*T}{highTh-lowTh}*maxDP\right)*\frac{i*T}{highTh-lowTh}*maxDP\right]. \tag{10}$$

With the parameters (*lowTh* = 1.0 s, *highTh* = 3.0 s, *maxDP* = 1.0, and *T* = 10 ms) this means that a given packet is dropped with the probability of 0.9991 after 50 TTL-checks, which translates to 500 ms. (We calculate the sum of drop probabilities over 50 TTL-checks.)

Our second TTL-based RED implementation is somewhat simpler. In this variant, each packet is given a random lifetime (in addition to the fixed lifetime that is used both in the 2G-SGSN and in the BSC) when it enters the 2G-SGSN. As in the first implementation, the packet is periodically checked if it should be discarded. When the age of the packet exceeds its assigned lifetime, the packet is dropped (or marked). Packet marking algorithm is illustrated in Fig. 20 and it is somewhat analogous to gentle RED [Flo].

```
if (U(0, 1) < maxDP) {
  lifetime := U(lowTh, highTh)
else
  lifetime := U(highTh, 2*highTh)
```

Fig. 20. TTLRED packet marking algorithm.

5.4 Simulation model

Simulations were performed in order to evaluate the efficiency of the aforementioned AQM schemes in decreasing buffer delay and improving TCP goodput. Our simulator makes use of publicly available ns-2 [NS2] modules such as TCP (NewReno variant [RFC2582]) and traffic sources, e.g., HTTP/1.1 [RFC2616].

[4] Packet is marked instead of dropping it if ECN is used and the flow supports ECN.

GPRS Model

In our simulation model, we have implemented a GPRS agent that has four different instances: GGSN, 2G-SGSN, BSC, and cell. The cell and 2G-SGSN agents take care of LLC segmentation and reassembly as well as LLC retransmissions. Moreover, the 2G-SGSN agent inserts and removes GTP headers, and it also performs 2G-SGSN downlink queuing equipped with MS, cell and committed information rate (CIR) specific flow/rate control. The BSC and GGSN agents are really simple; the BSC agent just passes the packets, while the GGSN agent inserts and removes GTP headers.

The air interface is modeled as a special type of link, where RLC block segmentation and reassembly are done and scheduling is implemented with a combination of flow-based round-robin and time slots. The air interface model also provides MS and BVC flow control information for the 2G-SGSN.

Other Settings

In our simulations, we have five active video streaming[5] users mapped in the streaming traffic class, 15 active push to talk over cellular (PoC)[6] users mapped in the interactive Traffic Handling Priority 1 traffic class, 30 active web-browsing[7] users mapped both in the interactive THP2 and interactive THP3 traffic class, and 30 active file-downloading[8] users in the background traffic class.

A total of 48 EGPRS time slots are available for packet switched traffic. This is equivalent to, e.g., five sites having three sectors each, where each sector allocates three EGPRS timeslots for packet switched traffic. The users are distributed evenly in two cells, i.e., the cells have the same traffic mix. The cells are under the same Gb "pipe". Gb capacity is set to 2048 kbps.

We use the following AQM-parameters:

- RED/ECN: *minTh* = 15 kB, *maxTh* = 45 kB, *maxDP* = 0.2.
- TTLRED/TTLECN (the simpler implementation): *lowTh* = 0.75 s, *highTh* = 2.25 s, and *maxDP* = 0.2.

The following class queue weights are applied at the 2G-SGSN: $w_{streaming} = 900$, $w_{IA\ THP1} = 50$, $w_{IA\ THP2} = 35$, $w_{IA\ THP3} = 15$, and $w_{BG} = 5$. Priority queuing is applied at the BSS.

---

[5] Mean bit rate of 40 kbps, maximum bit rate of 80 kbps, UDP is used as the transport protocol.
[6] Constant bit rate of 8 kbps, UDP is used as the transport protocol.
[7] Main page and the 30 inline items per page all have a size of 4.91 kB, which results in a total page size of 152 kB; page reading time is 1.0 seconds; four persistent TCP connections are utilized.
[8] The same HTTP traffic model is used here, too, but now four files (each having a size of 875 kB) are downloaded; time between two downloads is 1.0 seconds; four persistent TCP connections are utilized.

5.5 Performance Evaluation

Figs. 21 and 22 illustrate the end-to-end IP packet delay and average TCP goodput experienced by end-users with various congestion control schemes. From the simulation results we can observe that the end-to-end delay (95th percentile) can be fairly high without any AQM: up to five seconds for the interactive THP2 traffic class, 14 seconds for the interactive THP3 traffic class and even 34 seconds for the background traffic class. These are definitely too high for interactive applications such as web browsing.

In most cases, RED and ECN (and their TTL-based variants) reduce the buffer delays dramatically, down to one from five seconds. RED increases the average goodput by 10–20% depending on the traffic class, while ECN performs even better (40–90%).

It should be noted that the RED/ECN (as well as the TTLRED/TTLECN) implementation in the simulator followed the basic approach recommended in the literature. In other words, there was only a single instance of RED/ECN per traffic class specific BSSGP buffer.

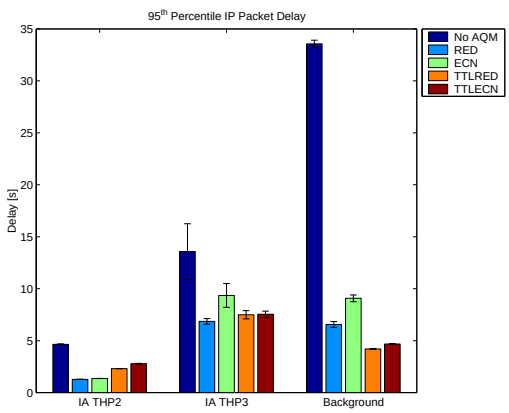

Fig. 21. End-to-end delay (95th percentile).

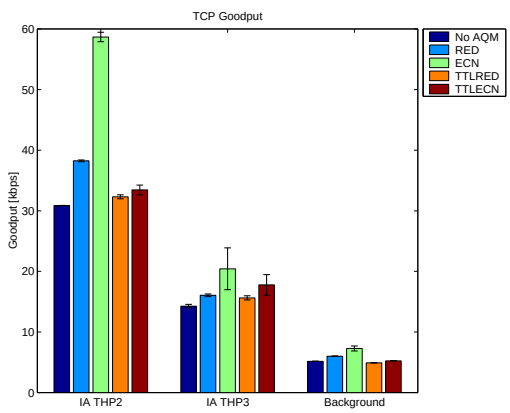

Fig. 22. TCP goodput.

5.6 Summary

In this study, various AQM schemes for the 2G-SGSN were evaluated. Although a single simulation scenario is presented here, several traffic mix and network configurations (e.g., varying number of available EGPRS timeslots) were considered during the study. The results obtained are in general well in line with those presented here.

The importance of well-implemented and configured features for optimal end-user experience and spectral efficiency was illustrated. Our simulation results suggest that in loaded conditions AQM could nearly double the average end-user goodput and reduce the buffer delay by a factor of two or three. The obvious conclusion is that appropriate congestion control mechanisms are very useful in the 2G-SGSN for handling non-real time traffic in loaded scenarios. Moreover, RED and ECN seem to be suitable active queue management schemes for the 2G-SGSN.

Although a normal Internet router buffer differs significantly from the 2G-SGSN buffer, even a simple RED implementation in that element (with only one RED instance per traffic class buffer) seems to provide dramatic improvements. It would be interesting to find out whether a more complex RED implementation would give additional gains.

Our TTL-based variants of RED (where the average queue size is replaced with the packet lifetime) could be studied further to evaluate if different parameters or logic would achieve better goodput results. Moreover, the applicability of other congestion control schemes described in the literature (e.g., explicit window adaptation [Kal98] and adaptive RED [Flo01]) could also be studied.

6 Quality of Service and Resource Management in IEEE 802.16e

WiMAX is a telecommunications technology aimed at providing wireless data over long distances in a variety of ways, from point-to-point links to full mobile cellular-type access. WiMAX is based on the IEEE 802.16 standard [802.16] and the 802.16e-2005 amendment [802.16e].

In this chapter, we present several ways to improve the quality of service in IEEE 802.16e networks. AQM is used to lower the DL delays; the performance of our TTLRED algorithm is compared to that of other AQM algorithms, e.g., PDPC. Moreover, the performance of the BS scheduling algorithms that take the MCS into account is compared to the DRR algorithm. In addition to the well-known PF scheduling algorithm, we present a modified version of the DRR algorithm (WDRR) that takes the MCS into account. A simple mapping between IEEE 802.16e data delivery services and DiffServ traffic classes and two connection admission control algorithms are also proposed. Finally, we compare different methods that an SS can use to inform the BS that it wants to resume an ertPS VoIP connection, i.e., continue receiving resources after a silence period. In all cases, simulations are used for evaluating the performance of the proposed mechanisms.

Before going into the details of our proposals, a brief introduction to IEEE 802.16e is given and an overview of existing mechanisms and solutions is provided. Then we present our contributions related to AQM (as introduced in Publication P4), BS scheduling (Publication P5), backhaul QoS (Publication P6), connection admission control (Publication P7), and ertPS VoIP resumption mechanisms (Publication P8). Finally, our IEEE 802.16e simulation model and simulation results related to all the aforementioned proposals are presented and conclusions are drawn.

6.1 IEEE 802.16e

IEEE 802.16-2004 [802.16] was introduced first and it was followed by the 802.16e-2005 amendment [802.16e]. The former is generally referred to as “Fixed WiMAX” while the latter is known as “Mobile WiMAX”. Terms 802.16d and 802.16e are being widely used, too. IEEE 802.16d uses orthogonal frequency-division multiplexing (OFDM) as the air interface technology while IEEE 802.16e uses orthogonal frequency-division multiple access (OFDMA). Fig. 23 illustrates IEEE 802.16e time division duplex (TDD) frame structure. In this thesis, we focus on IEEE 802.16e only.

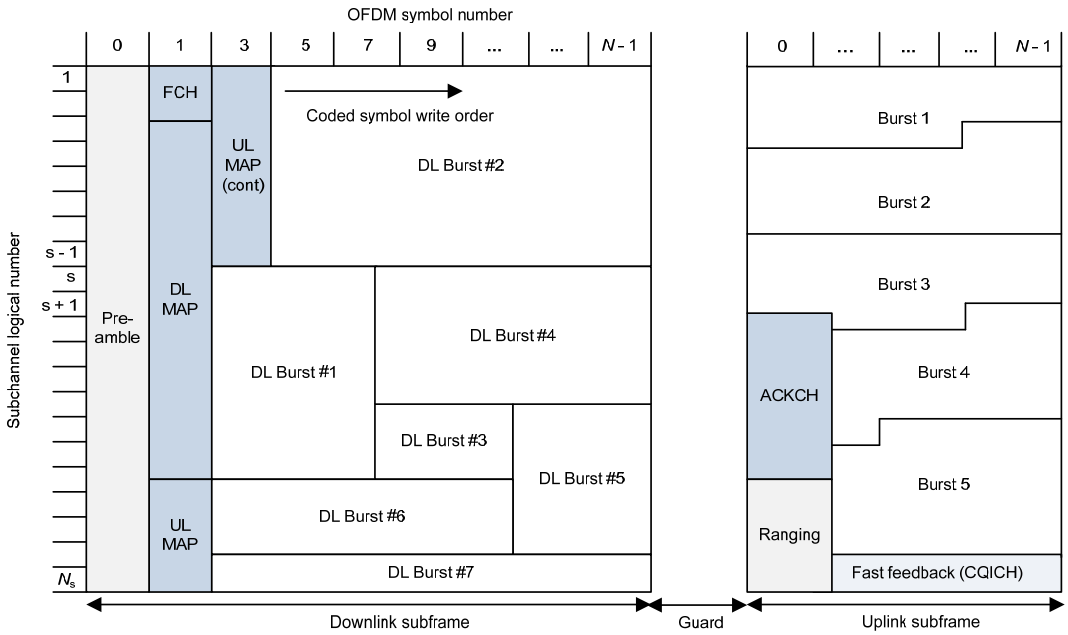

Fig. 23. Mobile WiMAX TDD Frame Structure [Wan08].

The 802.16e standard supports mesh and point-to-multipoint (PMP) operation modes. In the mesh mode, SSs communicate to each other and to the BS whereas in the PMP mode, the SSs communicate through the BS. It is very likely that the network operators will use the PMP mode to connect their customers to the Internet. Thus, the network operator can control the environment to ensure the QoS requirements of its customers.

IEEE 802.16e has five scheduling service classes. In unsolicited grant service (UGS), the BS allocates fixed-size grants periodically; UGS connections do not send any bandwidth requests. In real-time polling service (rtPS), the BS periodically polls the SS by granting one slot for sending a bandwidth request, while the goal of extended real-time polling service (ertPS) is to combine the advantages of UGS and rtPS. In ertPS, the BS continues granting the same amount of bandwidth (by default, the size of this allocation corresponds to maximum sustained traffic rate of the connection) until the ertPS connection explicitly requests a change in polling size. Extended piggyback request field of the grant management subheader can be used for this purpose. If the request size is zero, the BS may provide allocations for bandwidth request header only or nothing at all. In the latter case, contention request opportunities may be used. Non-real time polling service (nrtPS) is similar to rtPS except that connections are polled less frequently and they can use contention request opportunities. Best effort connections are never polled but they can receive resources only through contention.

For each scheduling service class there is a corresponding data delivery service class (see Table III). The service classes are defined for UL direction only whereas the data delivery service classes are defined for both uplink (UL) and downlink (DL) direction. QoS parameters are almost identical for a scheduling service and a corresponding data delivery service.

TABLE III
SCHEDULING SERVICES AND CORRESPONDING DATA DELIVERY SERVICES

Scheduling service	Corresponding data delivery service
UGS	Unsolicited Grant Service (UGS)
rtPS	Real-Time Variable-Rate Service (RT-VR)
ertPS	Extended Real-Time Variable Rate Service (ERT-VR)
nrtPS	Non-Real Time Variable Rate Service (NRT-VR)
BE	Best Effort Service (BE)

6.2 Related Work

In our IEEE 802.16e AQM studies, we apply the same algorithms as we have already applied in EGPRS and WCDMA networks, i.e., RED [Flo93] and PDPC [Såg03]. The performance of these algorithms is then compared to our own TTLRED.

In the BS scheduler studies, the main comparison is between DRR and PF [Jal00] schedulers. However, we also introduce a variant of DRR that takes MCS into account (WDRR) and compare its performance to that of PF.

Simple guidelines on mapping services to DiffServ classes have been presented in [RFC4594]. We propose here a simple mapping between IEEE 802.16e data delivery services and DiffServ traffic classes.

Two CAC schemes for OFDM wireless networks are studied in [Niy05]. The first scheme sets a threshold to limit the number of ongoing connections, and new connections are admitted as long as the total number of connections (including the incoming one) does not exceed the threshold. The second scheme admits a connection with a certain probability based on the queue status. Another new CAC scheme, called quadra-threshold bandwidth reservation (QTBR), is proposed in [Tsa07]. In QTBR, different threshold values are used for different service classes. The threshold values are determined by the number of calls of the corresponding service class in the system and the number of free channels.

In the literature, there are many recent research articles on uplink scheduling in IEEE 802.16e. For example, in [Den09] the authors propose a delay constrained uplink scheduling policy for rtPS/ertPS services and in [Soo08] the performance of UGS, rtPS, and ertPS is compared to each other. However, we are not aware of any articles specifically addressing different ertPS resumption methods and their performance.

6.3 Active Queue Management

At the IEEE 802.16e BS, all DL connections have dedicated buffers and resources are allocated per connection. There can be multiple connections between the BS and an SS. In UL direction,

however, the BS grants slots per SS (GPSS) and not per connection (GPC). The SS then decides how the resources are shared between different UL connections.

As in the case of other wireless technologies, it is likely that the radio interface [802.16, 802,16e] is the biggest system bottleneck with IEEE 802.16e, too. Thus, we might benefit from keeping the DL per-connection queues at the BS short enough by using AQM algorithms such as RED [Flo93], PDPC [Såg03], and our own TTLRED. This should result in a better TCP goodput and shorter response times. Earlier experience has shown (Publications P2 and P3) that AQM is feasible in both 2G-SGSN and RNC, which are bottlenecks similar to a BS in IEEE 802.16e. Publication P4 studies AQM in IEEE 802.16e BS.

We apply RED for connection specific buffers at the IEEE 802.16e BS in the same manner as we did with PDCP buffers in Chapter 3. PDPC is also applied in a similar fashion as in Chapter 3. However, now we use static *minTh* and *maxTh*.

In our time-to-live based RED, i.e., TTLRED, we provide the DL TCP packets with timestamps: each packet is given a random dropping time when it enters the BS. Packet marking algorithm is illustrated in Fig. 15 and it is somewhat analogous to gentle RED [Flo]. In Chapter 3 (or Publication P2), this scheme was called TTLRED2 (i.e., TTLRED for buffered packets).

6.4 BS Scheduling Algorithms

In Publication P5, we compare the performance of different IEEE 802.16e BS scheduling algorithms. In addition to the well-known DRR and PF scheduling algorithms, we present a modified version of the DRR algorithm that takes the MCS of the connection into account, just like the PF algorithm. We call this new algorithm simply weighted deficit round-robin (WDRR).

Deficit Round-Robin

In an IEEE 802.16e BS, one schedules packets, not slots. In fact, in the UL direction one does not even know the size of the head-of-line packet. Thus, the basic DRR algorithm needs to be modified slightly for our purposes. In each frame, the queue sizes are converted from bytes to slots. The number of required slots depends on the current MCS of the connection. The *quantum* parameter is now given in slots. In turn, the *deficitCounter* of each connection is increased by *quantum*. Slots are granted (not immediately; here we only construct the MAP messages) until all requests have been fulfilled or we run out of slots. Several rounds per frame can be done. If a queue drains out, the *deficitCounter* is reset to zero. It should be noted that in our variant of

DRR only those DL and UL connections that are granted the last DL and UL slots of a frame can save any deficit for the next frame; all other *deficitCounters* should be zero at this stage.

Proportional Fair

The PF scheduler [Jal00] assigns slots first to those connections that have the best ratio of current achievable rate *R(t)* to averaged rate *T(t)*. In every frame, the scheduler serves the connections in this order. The sequence described below is repeated as long as there are available slots or until there are no more requests. This should result in better throughput than with the DRR scheduler.

First, find the connection with biggest *R(t)* to *T(t)* ratio and grant as many slots as the connection needs or maximum number of slots that can be allocated at a time (scheduling slot size, similar to *quantum* in DRR). *R(t)* is the number of bytes that can be sent in a single slot; it is determined by the current MCS of the connection.

Then, update the exponentially averaged rate *T* for all connections (including those connections that do not have any data to send in this frame) *N* times. *N* is the number of slots we just allocated. If the connection was just served:

$$T(t+1) = (1 - 1/t_c) * T(t) + 1/t_c * R(t) . \tag{11}$$

Otherwise:

$$T(t+1) = (1 - 1/t_c) * T(t) . \tag{12}$$

Time constant t_c is an adjustable parameter; it determines how long we can let a flow starve. In the case of 5 ms frame length and 500 slots per (DL or UL) subframe, t_c of 10000 slots corresponds to 10000*(0.005/500) = 0.1 seconds. However, if we schedule real-time connections before BE connections, the number of DL or UL slots available for the BE class cannot be known beforehand. Thus, we should take our traffic mix into account when selecting the t_c parameter. Initial value for *T*, *T(0)*, can be set as the expected average rate divided by the expected average number of connections.

Weighted Deficit Round-Robin

Our weighted deficit round-robin (WDRR) is a simple variant of DRR, where we adjust the *quantum* size according to the connection's current MCS. We simply multiply the *quantum* by bytes per slot that the current MCS of the connection can deliver and then divide the *quantum* by six (bytes per slot for QPSK-1/2, our most robust MCS). For example, with 16QAM-3/4 we

would have three times bigger *quantum* than with QPSK-1/2. WDRR might need some additional starvation-avoidance features, e.g., a coefficient that determines the final *quantum* sizes. This, however, has not been studied in this thesis.

6.5 Backhaul QoS

Radio interface is not necessarily always the bottleneck of the system. In all wireless networks the backhaul can become a bottleneck, too. In IEEE 802.16e, one likely reason for this is the introduction of femtocells that are currently discussed actively in the research community and in the WiMAX Forum [WMF07]. The general idea is that a low cost BS is connected to the customer wired network, thus providing a local access point similar to Wi-Fi hotspots. Then it is possible that the wired connection can be of lower bandwidth when compared to the maximum bandwidth achieved at the wireless interface.

If all packets that are carried on the backhaul get best effort treatment, the bottleneck links need to be dimensioned in such a manner that the resulting buffering delays are low enough for real-time connections, too. This is called overprovisioning and it is the simplest and most straightforward approach to backhaul QoS. Naturally, this is not always feasible, since the backhaul bottleneck could be a microwave radio link or a leased line and thus the cost of overprovisioning would be prohibitive. Backhaul QoS in IEEE 802.16e is studied in Publication P6.

Differentiated Services

DiffServ [RFC2475] can alleviate the problem: we can prioritize real-time traffic over non-real time traffic. EF [RFC3246] can be used for the real-time traffic, while AF [RFC2597] or best effort can be used for the non-real time traffic. In practice, EF PHB is usually implemented as a priority queue with a rate limiter, while AF and BE queues have scheduling weights. As long as the real-time traffic load is low enough (e.g., less than 50% of the bottleneck link capacity), there is no need for immediate bottleneck link upgrades. However, admission control based on bottleneck link load is needed – otherwise all real-time connections will suffer during periods of congestion.

Traffic classification on the backhaul should naturally follow the traffic classification over the air interface. Mapping from IEEE 802.16e data delivery services to DiffServ traffic classes should be kept as simple as possible (simple guidelines on mapping services to DiffServ classes have been presented in [RFC4594]). No gains can be achieved from having more than three DiffServ classes if the BS scheduler has only three priority levels. Thus, we suggest the following mapping in Publication P6:

- UGS, ERT-VR, and RT-VR are mapped to EF
- NRT-VR is mapped to AFx (x being an integer between 1 and 4) or best effort
- BE is mapped to best effort

Having two DiffServ classes only is a solution that requires no scheduling weight tuning. If we have three DiffServ classes, we need to set appropriate scheduling weights for AFx and BE. Strict priority like weights for AFx and BE, e.g., 90:10, could be a good starting point.

Most routers that support DiffServ have also support for AQM. Usually, this means applying RED [Flo93] for the AF and BE queues. In the AF queues, we naturally need one set of thresholds per drop precedence level. If configured properly, RED keeps the router queues short (without degrading TCP goodput) and thus reduces the end-to-end delay. AQM can also be applied at the IEEE 802.16e BS DL queues, where it can be done per connection, as we have seen earlier.

6.6 Connection Admission Control

IEEE 802.16 standards do not specify any connection admission control mechanisms. However, CAC is definitely needed at least for all real-time (i.e., UGS, ertPS, and rtPS) connections – otherwise, we cannot guarantee any delay bounds or packet loss rates for these connections.

Our approach to the connection admission control problem in IEEE 802.16e is somewhat similar to the schemes proposed in [Niy05] and [Tsa07]. However, we have a more pragmatic viewpoint; the techniques proposed in Publication P7 take into account all the details of a real IEEE 802.16e system. The first algorithm utilizes the averaged number of free slots as input in admission decisions while the second method is more advanced and it tunes the admission thresholds according to current traffic load.

The number of real-time connections[9] has to be controlled in order to guarantee the QoS. This can be done at the BS, for example, by monitoring the DL queuing delays, virtual UL queue sizes and the number of free (DL and UL) slots for real-time traffic. However, it seems that only the number of free slots is a reasonable choice for UL admission control as the virtual (bandwidth request based) queue sizes may not always be accurate.

[9] As explained later, the number of active SSs has to be controlled, too.

Measurement-Based Admission Control (MBAC): Monitoring the Number of Free Slots

Our first algorithm is simple. In each frame *i*, when all real-time connections have been served, we check the number of remaining DL and UL slots for real-time connections ($freeSlots_i$) and update their exponentially weighted moving averages ($freeSlotsAv_i$). These averages are used in admission control and they are compared to our "safety margin" (e.g., 10 slots). If the averaged number of free DL/UL slots for real-time traffic is above the safety margin, we can admit the connection. w_S is a configurable averaging weight that determines how fast the average changes over time

$$freeSlotsAv_i = (1 - w_S) * freeSlotsAv_{i-1} + w_S * freeSlots_i \,. \tag{13}$$

Measurement-Aided Admission Control (MAAC): Adjusting the Limits Based on Measurements

Our second algorithm is somewhat more advanced than the first one. Since using the aforementioned averaged number of free slots for real-time traffic as such offers no protection against connections arriving in large batches, we can choose a more conservative approach instead and exercise bookkeeping with dynamically updated reservation limits for DL and UL traffic. A similar method for IP networks is proposed in Chapter 2.

Whenever a connection arrives, we check if the sum of currently reserved DL/UL bandwidth and the MRTR (minimum reserved traffic rate) of the connection is below the corresponding limit. If this is the case, the connection is admitted and the MRTR is added to the reserved DL/UL bandwidth. Naturally, the MRTR is subtracted from the reserved bandwidth when the connection is torn down.

```
if (freeSlotsAv > highTh)&&(limit < maxBw)
  limit := limit + increment
if (freeSlotsAv < lowTh)
  limit := limit * coefficient
if (limit < reservedBw)
  limit := reservedBw
```

Fig. 24. Reservation limit updating algorithm for DL and UL.

The reservation limits are updated (additive increase with parameter *increment*, multiplicative decrease with parameter *coefficient*) periodically and they are based on the averaged number of free slots for real-time traffic. Fig. 24 illustrates the updating algorithm. If the averaged number of free slots is larger than *highTh*, we adjust *limit* upwards. Similarly, if the averaged number of free slots is smaller than *lowTh*, we adjust *limit* downwards. We cannot set *limit* higher than *maxBw* or lower than currently reserved bandwidth, *reservedBw*.

6.7 Uplink Channel Access and ertPS Resumption Mechanisms

The IEEE 802.16e standard supports several mechanisms an SS can use to request uplink bandwidth. Depending on the QoS and traffic parameters associated with a service, one or more of these mechanisms may be used by the SS. Once the SS has an allocation for sending traffic, it is allowed to request more bandwidth by transmitting a stand-alone bandwidth request or piggybacking a bandwidth request on generic MAC packets [And07].

In Publication P8, we are interested in different mechanisms an ertPS VoIP connection can use to resume sending packets after a silence period (during which no packets are sent), i.e., we study different ertPS resumption mechanisms and their performance. In addition to this, we propose a limit for uplink CDMA allocations that are granted as a response to CDMA request codes.

Polling

The BS allocates dedicated or shared resources periodically to each SS. The SS can then use these resources to request bandwidth. This process is called polling. Polling may be done either individually (unicast) or in groups (multicast) [And07]. If an ertPS VoIP connection is polled regularly also during the silence period, we can send the first packet of the next talkspurt without additional delay. Some uplink resources are wasted, though. With rtPS class, this is our only alternative.

Contention Resolution

Contention resolution mechanism in IEEE 802.16e allows the SSs to send their bandwidth requests to the BS without being polled. This kind of mechanism is necessary for scheduling service classes that are polled irregularly or not at all, i.e., ertPS, nrtPS, and BE. Contention resolution parameters are the number of bandwidth request transmission opportunities per frame and backoff start/end values. The backoff start value determines the initial backoff window size, from which the SS randomly selects a number of the transmission opportunities to defer before sending the bandwidth request. If there is a collision, the backoff window is increased and the contention resolution is repeated. The SS continues to retransmit the bandwidth request until the maximum number of retransmissions expires.

In OFDMA PHY, the uplink contention comprises several phases. First, the SS sends a CDMA request code. If the code is received correctly (no collisions), the BS grants an uplink CDMA allocation, which the SS can use for sending a bandwidth request.

Contention resolution mechanism is useful, e.g., with VoIP connections that support silence suppression. We assume here that ertPS is used for these connections. During the silence periods, we can either have periodic allocations just big enough for a stand-alone bandwidth request or no polling at all. Naturally, the latter option is more bandwidth-efficient. However, in this case we need to use the contention resolution mechanism when the connection becomes active again, i.e., when the connection has a packet to send. If there are many connections that participate in contention resolution, this could result in considerable UL packet delays. Moreover, the backoff parameters are common for all connections and this may not suit well for real-time connections.

Multicast Polling

In multicast polling, the SS does not send its bandwidth requests during the common bandwidth request contention slots but during slots that have been assigned for a particular group of SSs. Multicast polling and VoIP has been studied in [Ala07]. The aforementioned paper proposes separate backoff parameters for different multicast polling groups in order to fulfill VoIP delay (and packet loss) requirements.

CQICH / Fast Feedback Channel

An alternative to contention resolution (and polling) is to use the fast feedback channel (CQICH, see Fig. 23) for informing the BS that the SS has a packet it wants to send after the silence period. The fast feedback channel is mainly used for transmitting the SNR information (e.g., every four frames) that the BS can use in link adaptation. As we cannot fit the SNR information and the ertPS resumption codeword into the same message (the length is only six bits), sending the latter may have some implications on the link adaptation. However, switching from silence period to talkspurt should be a rare event. Assuming that an average talkspurt and silence period have lengths of 1.2 seconds and 0.8 seconds, correspondingly, there should be only one ertPS resumption message per two seconds on average. When the ertPS resumption codeword arrives at the BS, we immediately grant enough slots for one VoIP packet and re-schedule the next grant, i.e., we reset the frame counter of the connection.

6.8 IEEE 802.16e Simulation Model

The basic implementation of our IEEE 802.16e module is described in detail in [Say09]. The module includes the following features: OFDM and OFDMA PHY levels, automatic repeat request (ARQ), hybrid ARQ (HARQ), transport and management connections, fragmentation, packing, ranging and bandwidth request contention periods, CDMA codes for ranging and bandwidth requests, and support for the most important MAC level signaling messages.

Additionally, the module includes several different BS schedulers and it has a simple, trace-based model[10] for link adaptation. These features are described in more detail in the following subsections.

MCS, Link Adaptation, and Errors

Modulation and coding scheme (MCS) defines how many bits can be sent in a single slot. The BS can dynamically change both the DL and UL MCS of an SS. Link adaptation is based on reported signal-to-noise ratio (SNR) values and carefully tuned transition thresholds. Naturally, we have a different set of link adaptation thresholds for HARQ and non-HARQ connections. Our error model analyzes the PDU SNR, maps it to the forward error correction (FEC) block error rate (BLER) based on the channel performance curves, and decides whether the PDU is erroneous or not. Each SS has a randomly selected trace file, where the SNR values are read from[11]. We have obtained our SNR values from system simulations. In our simulations, we model only one sector, while the trace files have been obtained from 19-cell system simulations (where the focus has been on the lower protocol layers).

BS Scheduler

The BS scheduler grants slots for the SSs and DL connections according to the QoS parameters and bandwidth request sizes of the individual connections. Uplink virtual queue sizes are updated based on bandwidth requests and received UL packet sizes. For DL connections, we use the BS queue sizes and the QoS parameters. Our scheduler assigns slots in three stages (see Fig. 25): management connections are served first, then real-time connections, and finally non-real-time connections. Different scheduling algorithms can be applied in the two latter stages.

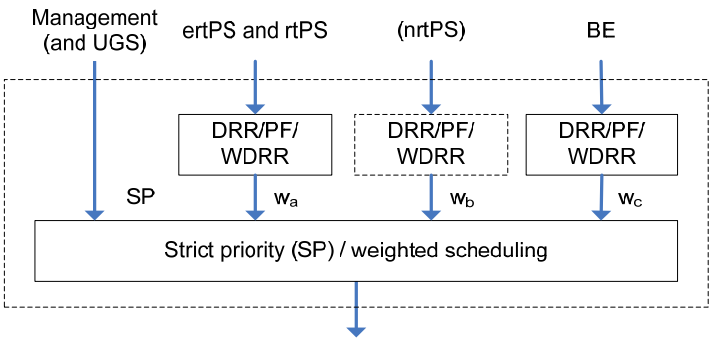

Fig. 25. Multi-stage scheduler at the BS.

[10] However, in Publications P4 and P5 we modeled link adaptation with a Markov chain, where the states represented different MCSs. The transition probabilities were obtained from system simulations. In these studies, HARQ model was still missing from our simulator. Thus, we used a fixed error rate of 10% for a 100-byte MAC PDU with non-HARQ connections (ARQ would deal with these errors) and 1% error rate with HARQ connections (it was assumed that HARQ had already corrected most errors).
[11] 60% of our traces correspond to ITU PedB model and 40% to ITU VehA model.

We have implemented support for three IEEE 802.16e data delivery services: extended real-time variable rate service (ERT-VR), real-time variable rate service (RT-VR), and best effort. ERT-VR and RT-VR connections are served before BE connections; they are assigned slots until all ERT-VR and RT-VR queues are empty or until there are no more slots left for real-time traffic. Connection admission control should take care that there are always enough slots for real-time connections and that all slots are not used for real-time connections. Moreover, rate limiters are used at the BS to enforce the minimum reserved traffic rate (MRTR) of real-time connections; excess real-time traffic gets BE treatment.

In order to waste as little bandwidth as possible, silence suppression detection at the BS is done for the ertPS connections: whenever an UL PDU is received, a connection-specific timer is started. When this timer expires, silence state is started. If we let the ertPS connections participate in contention (or if CQICH-based resumption is deployed), no polling is done during the silence state. Otherwise, periodical polling slots are granted for ertPS connections during silence periods.

However, before any connection can be granted slots, we have to serve the management traffic in every frame, i.e., we need to grant slots for the UL-MAP, DL-MAP, CQICH reports, HARQ acknowledgements, HARQ retransmissions, and CDMA uplink allocations[12]. Since all SSs contribute to this overhead, admission control for real-time connections alone is not sufficient but we have to limit the number of active SSs, too. In the case of MBAC, this is rather simple: the arrival of a new SS is treated in a similar fashion as the arrival of a new real-time connection from an SS that is already registered to the BS. If the averaged number of (UL or DL) slots is too low, the new SS is rejected. In the case of MAAC, however, we need to come up with a suitable "MRTR" for the SSs. How much resources are reserved for control traffic of a single SS should depend mostly on the CQICH report interval.

If ARQ is enabled for a connection, the following connection-internal scheduling order is applied: 1) ARQ feedback messages, 2) retransmissions, and 3) all other PDUs. We study only cases with one UL/DL transport connection per SS.

Traffic Models

Our VoIP traffic source is a simple Markov model, where both on and off period lengths are exponentially distributed with mean lengths of 1.2 and 0.8 seconds, respectively. 24 bytes of payload is sent every 30 ms during active periods. UDP (8 bytes overhead) is used as transport protocol; RTP adds 12 bytes of overhead and IPv4 20 bytes, which results in a total packet size

[12] This is not something explicitly required by the standard but rather something that makes sense.

of 64 bytes. Packet header compression (from 40 bytes to 4 bytes) is applied at the BS and the SS. All user traffic is given BE treatment except for VoIP traffic that is given rtPS or ertPS treatment in some simulations.

The variable bit rate video traffic source is simulated according to [Mag88]. The following parameters are used: mean rate of 125 kbps, maximum rate of 250 kbps; MTU (1500 bytes) sized packets are sent; UDP is used as transport protocol; IPv4 adds 20 bytes of overhead, which results in a payload size of 1500-8-20 = 1472 bytes.

Our web browsing traffic source is simplified from [Mol00]; it is necessary to limit the variability of page sizes as the TCP goodput depends on that, too. We have a main page and 30 inline items per page, all items having a size of 4.91 kB, which results in a total page size of 152 kB (this is based on our measurements). Page reading time is uniformly distributed between 1 and 5 seconds and four NewReno TCP [RFC2582] connections are utilized. The same HTTP traffic model is used for modeling file downloading, but this time only a single 250 kB file is downloaded. The time between two downloads is uniformly distributed between 1 and 5 seconds. A single NewReno TCP connection is utilized.

Simulation Methodology and Simulation Parameters

We use a modified version of the ns-2 simulator [NS2]. Six simulations are run in each case in order to obtain small enough 95% confidence intervals. Simulation time is 200 seconds. One-way core network delay between a server and the BS is set to 31 ms, using a few links with latencies ranging from 1 ms to 10 ms. This is mainly done in order to have realistic round-trip times for TCP connections, and thus realistic throughput. The only bottleneck in our scenario is the IEEE 802.16e air interface (see Fig. 26). There are no packet drops or variable delays elsewhere in the system, except in our backhaul bottleneck studies. The most important IEEE 802.16e network parameters are listed in Table IV.

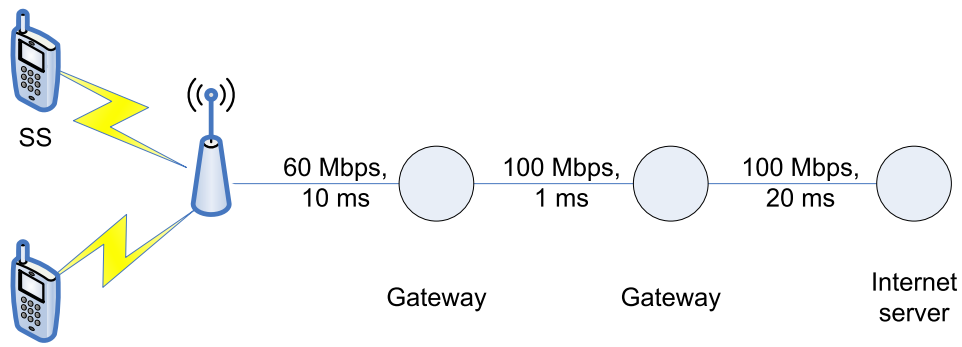

Fig. 26. Simulation topology.

TABLE IV
IEEE 802.16e RELATED SIMULATION PARAMETERS

Parameter	Value
PHY	OFDMA
Bandwidth	10 MHz
FFT size	1024
Cyclic prefix length	1/8
TTG (Transmit-receive transition gap)	296 PS
RTG (Receive-transmit transition gap)	168 PS
Duplexing mode	TDD
Frame length	5 ms
DL/UL ratio	35/12 OFDM symbols
DL/UL permutation zone	FUSC/PUSC
Channel report type and interval	CQICH, 20 ms
MAP MCS	QPSK-1/2, REP 2
Compressed MAP	Yes
Number of ranging opportunities	1
Ranging backoff start/end	0/15
Number of request opportunities	3[13]
Request backoff start/end	3/15
CDMA codes for ranging and bw requests	64/192
HARQ (CC)	For VoIP connections only
Number of HARQ channels	16
HARQ buffer size	2048 B per channel
HARQ shared buffer	Yes
Max. number of HARQ retransmissions	4
HARQ ACK delay	1 frame
PDU SN	With HARQ (no ARQ)
Fragmentation/Packing	Yes/Yes
Maximum MAC PDU size	100 bytes
ARQ	For FTP connections only
ARQ feedback types	All
ARQ block size / window size	64 bytes / 1024
ARQ block rearrangement	No
ARQ feedback frequency	5 ms
ARQ retry timer	50 ms
ARQ block lifetime	1500 ms
ARQ rx purge timeout	2000 ms
MRTR for VoIP connections	11800 bps
Max. SS/BS queuing delay for VoIP SDU	150 ms

6.9 Performance Evaluation

The questions we address in our simulations are the following: how can AQM reduce DL delays (Publication P4), how can BS scheduling algorithms that utilize MCS information improve throughput (Publication P5), how can backhaul QoS be supported cost-effectively (Publication P6), how should connection admission control be implemented (Publication P7), and what are the most effective ertPS VoIP resumption mechanisms (Publication P8).

[13] In scenarios, where multicast polling is used, there are two request opportunities for the basic contention and one request opportunity for the multicast polling group.

Active Queue Management

Each AQM algorithm is modified so that we only drop TCP packets. This may be beneficial for UDP-based variable bit rate streaming traffic – the jitter buffer at the receiver will handle the occasional bursts and drop packets if they are delayed too much. Per-connection queue size is limited to 50 packets. Assuming that all packets are of MTU size, this translates into 75 kB buffers. For RED and PDPC, we apply the following parameter values (where applicable): $minTh$ = 15 kB, $maxTh$ = 45 kB, $maxDP$ = 1.0, w_{AQS} = 1.0, and N = 20. For TTLRED, we apply the following parameter values: $lowTh$ = 0.3 s, $highTh$ = 0.9 s, and $maxDP$ = 1.0.

We simulate the following traffic mix: 5 VoIP connections, 5 video streaming connections (DL only); 10, 14, 18, 22, 26, or 30 web browsing connections and 5, 7, 9, 11, 13, or 15 file downloading connections per BS. All user traffic is given BE treatment except for VoIP traffic that is given RT-VR treatment.

Once again, we are interested in DL delay and TCP goodput. Based on our simulations (in Publication P4), we can say that the AQM algorithms reduce the delay most, when the advertised window is relatively big and when the TCP connections are long-lasting. However, non-optimal AQM parameters can lead to decreased goodput. Figs. 27 and 28 illustrate the effect of the most promising AQM algorithms, PDPC and TTLRED, on DL packet delay and TCP goodput for short-lived web browsing sessions, this time with 60-packet TCP advertised window. As the load grows, the improvement in delay is more and more obvious with both algorithms. Fig. 28 shows that TCP goodput is not really affected.

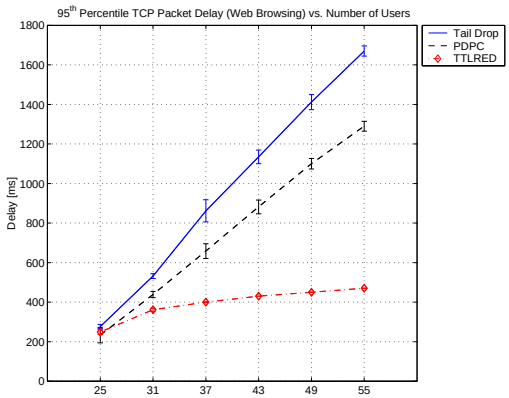

Fig. 27. 95th percentile packet delay for web browsing with different loads.

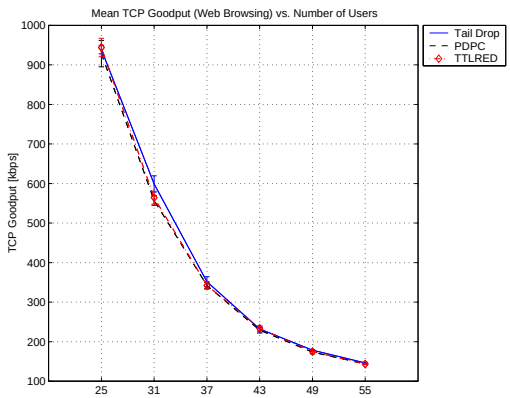

Fig. 28. TCP goodput for web browsing with different loads.

BS Scheduling Algorithms

The goal of our BS scheduling studies in Publication P5 is to find out how much scheduling algorithms that utilize MCS information (PF and WDRR) can improve MAC throughput and TCP goodput, when compared to DRR. We simulate the same traffic mix as in the previous

section. Moreover, the same service mapping is applied (all user traffic is given BE treatment except for VoIP traffic that is given RT-VR treatment).

Figs. 29–32 illustrate the main results of our simulations. Both PF and WDRR perform very well (in terms of MAC throughput and TCP goodput) against DRR. This is in line with the results of [Chi07]. Especially the good performance of WDRR is a nice surprise as this scheme should be easier to implement (and less computationally complex) than PF. The fact that PF scheduler can leave a connection without any resources for quite a long period of time (if t_c is large enough) may be a problem if ARQ timers are set to expire too soon. Moreover, sudden variations in round-trip time (RTT) might launch TCP retransmissions, and that could possibly lead to degraded TCP goodput.

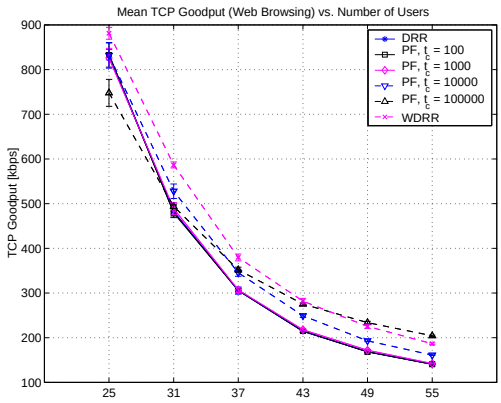

Fig. 29. TCP goodput for web browsing.

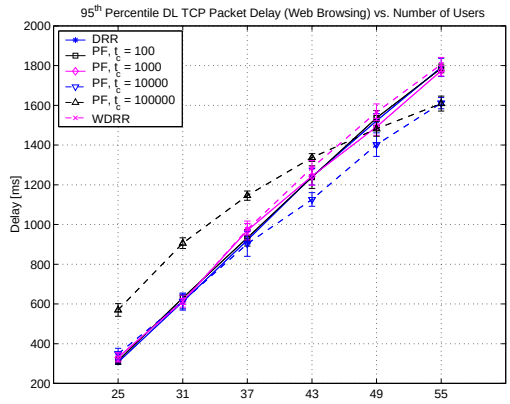

Fig. 30. 95th percentile DL TCP packet delay for web browsing.

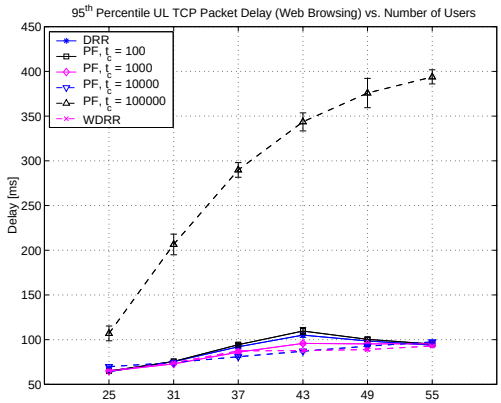

Fig. 31. 95th percentile UL TCP packet delay for web browsing.

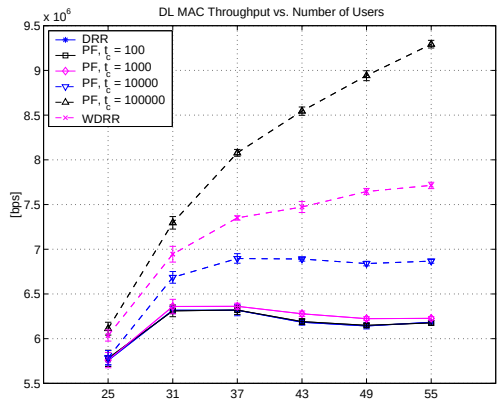

Fig. 32. DL MAC throughput.

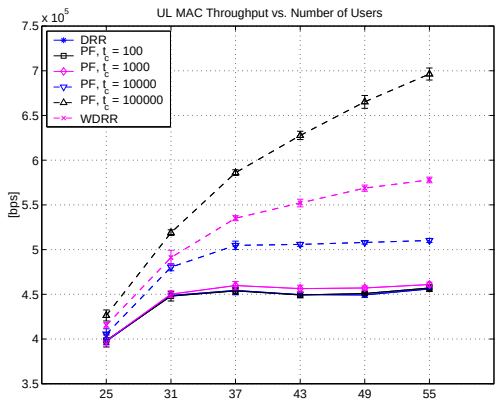

Fig. 33. UL MAC throughput.

WDRR outperforms PF with lower traffic loads, because the PF algorithm needs to have enough connections in order to achieve better relative throughput gain. When there are more connections, it is more likely that the PF algorithm always picks a connection with a good MCS.

When large t_c values are used in the PF scheduler, the price we have to pay for better TCP goodput is an increased delay. However, AQM at the BS can be used to dramatically reduce the queuing delays without sacrificing the goodput.

Backhaul QoS

In our backhaul QoS studies (Publication P6), we mostly want to verify that when DiffServ is applied on the backhaul links, and when IEEE 802.16e traffic classes are mapped to DiffServ traffic classes according to our proposal, there is no need to overprovision the backhaul links in order to support real-time VoIP. We simulate the same traffic mix as in the two previous sections. Moreover, the same service mapping is applied.

Figs. 34–36 illustrate the main results of our DiffServ simulations. In Fig. 34, BE VoIP delay grows as the link speed decreases, while in Fig. 35 EF VoIP delay stays the same as the link speed decreases. Fig. 36 illustrates an interesting phenomenon: when the number of users increases, the backhaul is not so big a bottleneck anymore; there is now more overhead (and VoIP header compression) on the air interface and this overhead does not load the backhaul. Naturally, the same phenomenon would happen if VoIP packets were given BE treatment on the backhaul (this can actually be seen in Fig. 34, where the BE VoIP delay with 10 Mbps bottleneck starts to decrease when the number of users grows big enough).

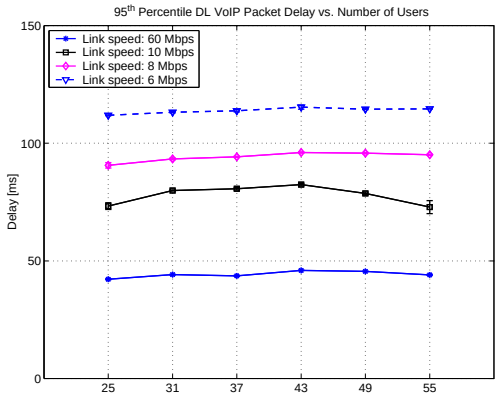

Fig. 34. BE VoIP: 95th percentile DL VoIP delay.

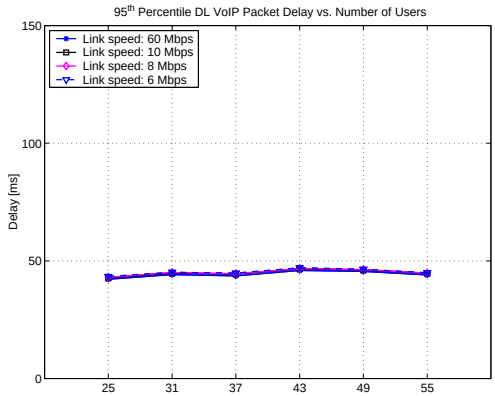

Fig. 35. EF VoIP: 95th percentile DL VoIP delay.

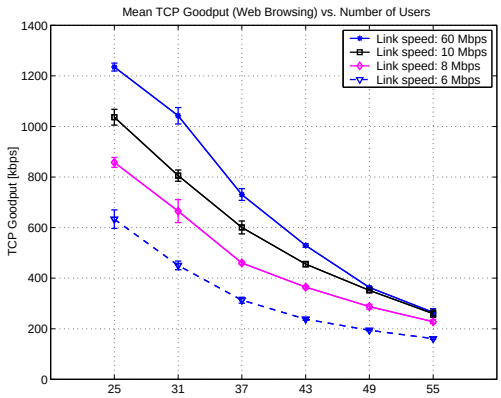

Fig. 36. EF VoIP: TCP goodput for web browsing.

Connection Admission Control

In Publication P7, we introduce two connection admission control algorithms for IEEE 802.16e. In our simulations, we validate the proposed algorithms and show that the MAAC algorithm offers protection against batch arrivals.

We simulate the following traffic mix: a variable number of VoIP connections and 10 file downloading connections. New VoIP connections arrive in the system, according to a Poisson process, with an intensity of 6.7 connections per second. However, 200 first connections arrive with an intensity of 20 connections per second, and without admission control. VoIP connection duration is exponentially distributed with a mean of 60 seconds, while the file downloading connections are active during the whole simulation run.

Depending on the simulated connection admission control method, a new VoIP connection is admitted to the network only if:

1. Average number of free DL/UL real-time slots is bigger than 25, 20, 15, or 10 (MBAC, averaging weight, $w_S = 0.001$).

2. The dynamic reservation limits allow a new connection to be admitted to the system. Here we use MAAC with the following parameters: $w_S = 0.001$, *highTh*/*lowTh* = 27/23, 22/18, 17/13, or 12/8, *increment* = 3 kbps, *coefficient* = 0.9, *maxBw* = 3 Mbps. Limit update frequency is set to 100 ms.

Figs. 37 and 38 illustrate the benefit of measurement-aided admission control. With Poisson connection arrivals, there is no real difference between MBAC and MAAC. However, when batch arrivals are introduced, MBAC cannot reject all the connections it should. Thus, too many connections are admitted and UL delays grow intolerable (see Fig. 38). MAAC does not have this problem as it utilizes bookkeeping with adaptive reservation limits instead of the number of free slots as such.

Figs. 39–41 illustrate the dynamics of MBAC and MAAC. Fig. 40 shows that the number of connections as a function of time does not follow a "saw tooth" pattern as in Fig. 39 (MBAC) but the curve is more stable. Fig. 41 shows that uplink was the bottleneck all the time.

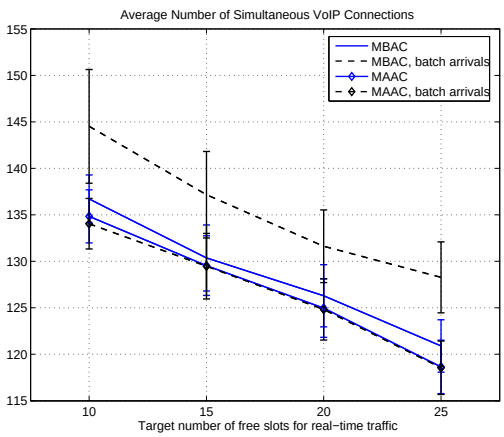

Fig. 37. MBAC vs. MAAC (ertPS VoIP): number of VoIP users.

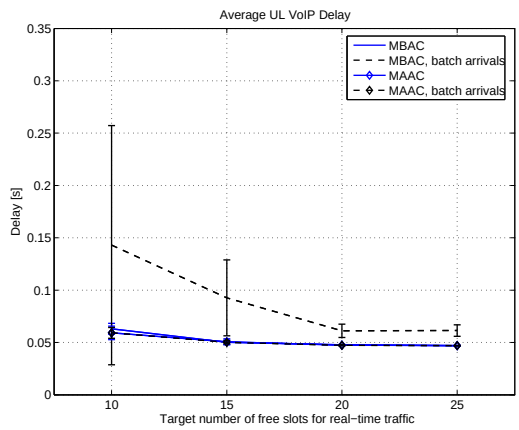

Fig. 38. MBAC vs. MAAC (ertPS VoIP): average UL VoIP delay.

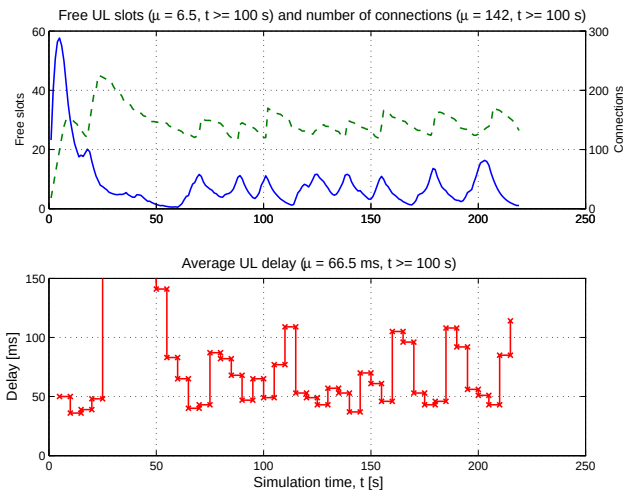

Fig. 39. MBAC, ertPS, batch arrivals, target number of free slots: 10.

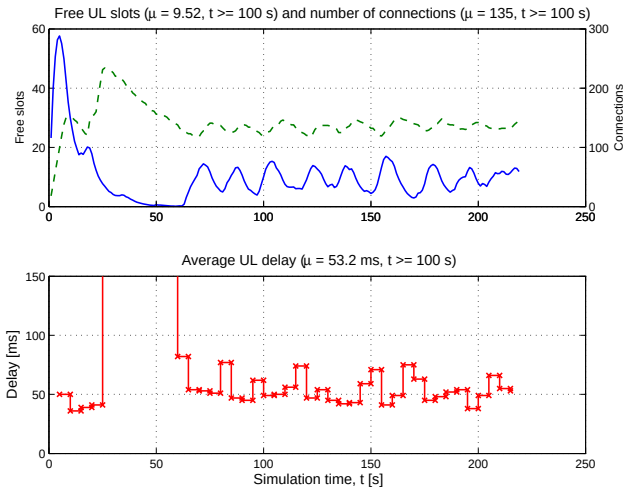

Fig. 40. MAAC, ertPS, batch arrivals, target number of free slots: 10.

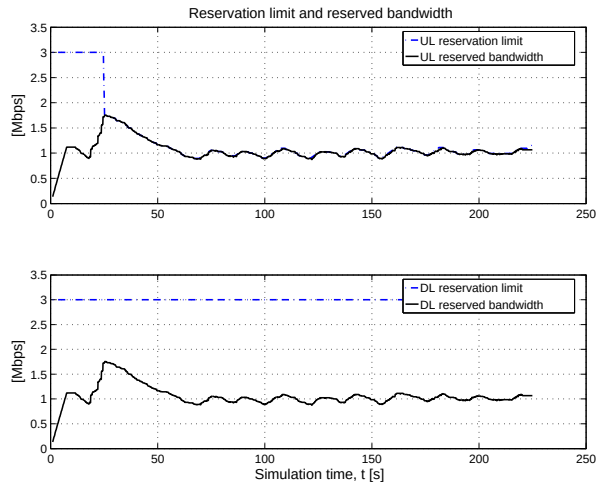

Fig. 41. MAAC, ertPS, batch arrivals, target number of free slots: 10.

Uplink Channel Access and ertPS Resumption Mechanisms

In Publication P8, we study different IEEE 802.16e uplink channel access mechanisms that can be used to activate ertPS VoIP connections after a silence period. The goal is to do this as fast and as possible, without wasting resources.

We simulate the following traffic mix: 120–130 or 95–105 VoIP connections and 10 or 50 file downloading connections per BS. All connections are active during the whole simulation run.

Even though our FTP traffic is downloading and not uploading, there are a lot of TCP acknowledgements that need to be sent upstream. This will cause a heavy load on the bandwidth request opportunities and CDMA codes that are shared with all SSs – including the ones with VoIP traffic. To better illustrate this phenomenon, additional simulation scenarios have been tested in addition to the basic one. In the second and third scenarios, we have increased the number of file downloading SSs from 10 to 50 (and decreased the number of VoIP users by 25).

In our basic scenario, there are 120 or 130 VoIP users and 10 file downloading users. As one could easily guess, ertPS with polling consumes more uplink resources (see Fig. 42)[14] than the other three mechanisms and therefore the (resumption) delays (see Fig. 43) start to grow with this mechanism.

Contention resolution is not a bottleneck in this scenario. However, had there been more connections using contention (or multicast polling), there could have been long delays due to CDMA code collisions and the uplink CDMA allocations.

CQICH based resumption uses somewhat more resources than contention based resumption. This is probably due to non-optimal link adaptation. As we speculated earlier, link adaptation may not work optimally if the SNR value is replaced with the ertPS resumption codeword. However, we also ran simulations where the SNR value and the ertPS resumption codeword were put into the same message, but this had no major effect on the results. Things could be different with larger CQICH report interval, though.

We chose not to allocate additional resources for multicast polling but one request opportunity per frame for the multicast polling group was taken from the basic contention region. In the first case, the poor performance of multicast polling based resumption was due to non-optimal request backoff parameters (start/end: 3/15). In order to limit uplink VoIP delay with multicast polling based resumption, we also applied backoff parameters different (start/end: 1/15) from the basic contention resolution parameters, as proposed in [Ala07]. The results were indeed better in the latter case. This, however, would require changes in the specification. Moreover, we did not want to apply request backoff parameters optimized for ertPS VoIP resumption for BE traffic as that would have led to a large number of collisions and thus lower TCP goodput.

In order to have meaningful results with 50 (instead of 10) file downloading users, we decreased the number of VoIP users by 25 in all cases. Thus, the amount of non-controllable UL resources (that are allocated to CQICH reports, HARQ acknowledgements etc.) stays more or less the same as in the previous scenario.

In this scenario, polling based ertPS resumption leads to excessive delays with a high number of VoIP users (see Figs. 44–45), while the other resumption mechanisms do not – except for the first multicast polling case, which suffers from non-optimal request backoff parameters. This leads us conclude that it is not the CDMA code collisions but the uplink CDMA allocations that are the reason for a bottleneck in the unicast allocations. With contention and multicast polling based ertPS resumption, the BS grants resources upon receiving the CDMA code. With polling

[14] This figure illustrates the averaged number of free uplink slots after the real-time (ertPS) connections have been served.

based resumption, however, the polling slots are granted periodically and only after the CDMA codes (from BE connections) have been responded to. If there are no slots left in the present frame, we have to wait. The same phenomenon could happen with CQICH based resumption, only with higher traffic load, as CQICH based resumption consumes less resources.

If we do not limit the number of uplink slots that can be granted as a response to CDMA codes, the BS allocates slots for sending bandwidth requests as a response to all CDMA codes it has received. Since adding CDMA allocation IEs to the UL-MAP happens before scheduling any user traffic, it is likely that the non-real time connections "steal bandwidth" from the real-time connections.

We limited the number of uplink resources that could be granted as a response to CDMA codes to ten slots, and the results changed dramatically (see Fig. 46–47). The reason for this is that now there are more slots available for ertPS connections. In this scenario, most of the SSs that participate in contention resolution have an ARQ feedback (or TCP acknowledgment) to send. Delaying ARQ feedbacks of BE connections is a better alternative than letting ertPS VoIP connections suffer. Of course, it may every now and then happen that we delay the resumption of an ertPS VoIP connection when contention or multicast polling based resumption is used. (However, the latter should be a rare event. In our simulator, the contention region for multicast polling comes before the basic contention region in the UL subframe and thus multicast polling SSs get the first CDMA allocations.) This can be seen from Fig. 47: now polling and CQICH based resumption give the best delay performance. Multicast polling with optimized backoff parameters performs well, too.

In any case, it is possible that a high number of SSs that are hosting BE connections can cause problems to ertPS and other real-time connections. This is due to CQICH reports, HARQ acknowledgements, HARQ retransmissions, and the aforementioned uplink CDMA allocations that are all granted slots before any user connection. A partial solution would be to have larger CQICH report intervals for the BE users. However, since an SS can host many connections (of different types) it might make more sense to introduce admission control for all active SSs – no matter what connections they might host.

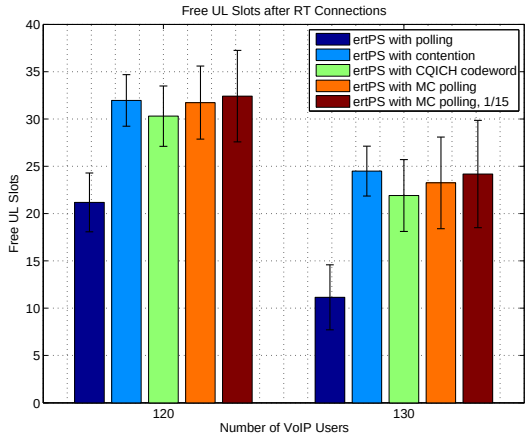

Fig. 42. Case 1: free UL slots.

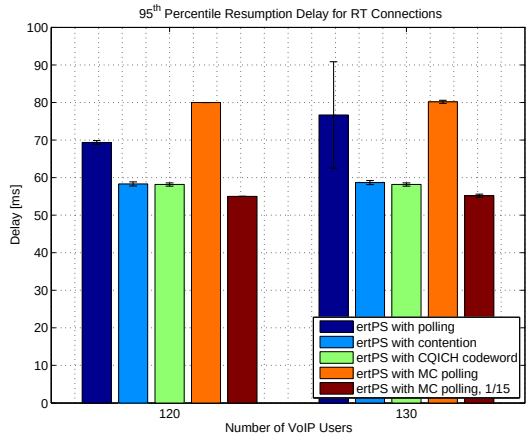

Fig. 43. 95th percentile resumption delay.

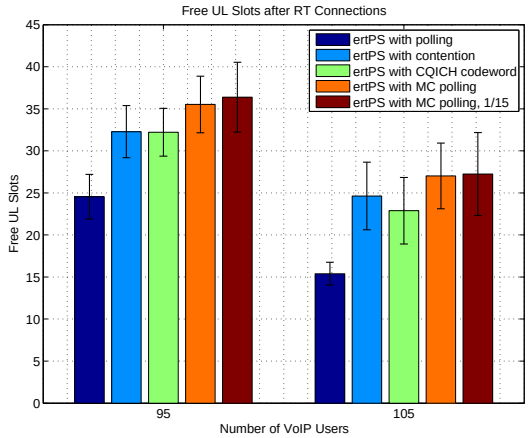

Fig. 44. Case 2: free UL slots.

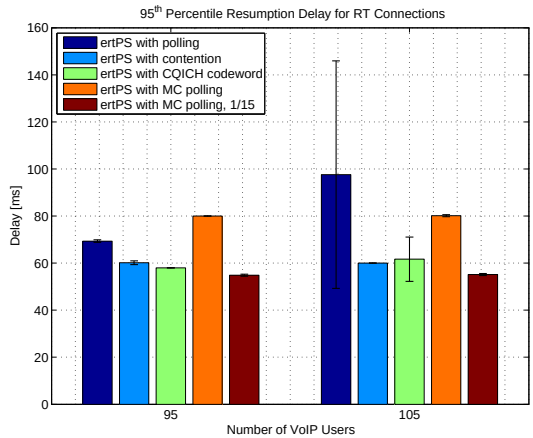

Fig. 45. 95th percentile resumption delay.

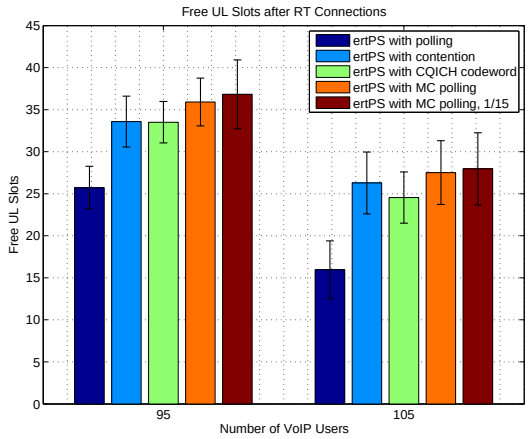

Fig. 46. Case 3: free UL slots.

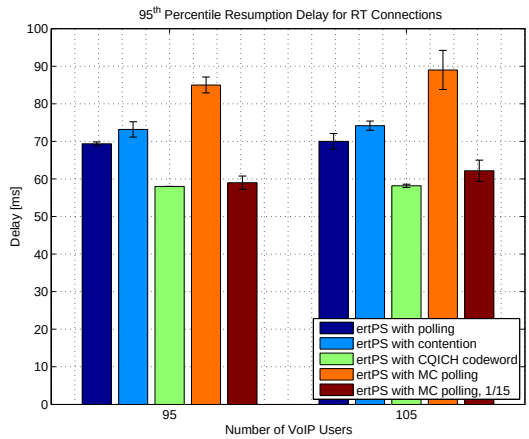

Fig. 47. 95th percentile resumption delay.

6.10 Summary

In this chapter, we studied how different AQM mechanisms (including our own proposal, TTLRED) can reduce DL buffering delays at the IEEE 802.16e, how different BS scheduling algorithms (including our own proposal, WDRR) that utilize MCS information can improve throughput, how backhaul QoS can be supported cost-effectively, how connection admission

control should be implemented (we have presented two novel CAC algorithms), and what the most effective ertPS VoIP resumption mechanisms are.

Active Queue Management

We have shown that AQM is useful in reducing the queuing delays caused by the BS DL buffering. Providing enough buffer space will maximize TCP goodput, however, as AQM helps to reduce the DL queuing delays of TCP connections, we shall have better user experience in web surfing and in other semi-real time activities. Moreover, less buffer memory is needed when queues are kept short. TCP goodput is not increased, given that the DL buffer size and TCP advertised window are sufficient.

It is worth mentioning that badly configured AQM algorithms can even reduce the goodput as a result of excessive packet drops. Thus, RED, PDCP, or most probably any AQM scheme with static parameters is not enough but we should adjust the parameters according to the available bandwidth, which keeps changing all the time. Thus, we do not find these methods feasible. TTLRED, on the other hand, is simple to configure, does not require parameter adjustments (when the queuing delays are short enough, the algorithm is effectively disabled), and performs well in all the conditions that it was tested in.

Since we apply AQM to individual per-connection queues at the BS, it makes sense to treat this feature as a connection parameter, just like ARQ or the QoS class. Indeed, a provider can turn AQM on and off for a particular connection without compromising the IEEE 802.16e specification. However, since AQM is not a part of the IEEE 802.16e QoS profile, there is no way to negotiate it with the customer. Thus, whether AQM is applied for a given connection or not should depend on the QoS class selected. AQM could be applied to BE and nrtPS connections only (scheduling should guarantee low enough DL queuing delays for the more real-time connections). However, we should still check the packet header and drop TCP (or TCP-friendly) packets only. It is not reasonable to apply AQM to traffic that does not react to packet drops.

BS Scheduling Algorithms

We have presented a performance comparison of different schedulers for IEEE 802.16e BS. Our simulations show that the PF scheduler is clearly a better choice for BE traffic than the DRR scheduler. However, more studies are still needed on, e.g., the impact of different ARQ timer values.

WDRR is an interesting alternative to PF as it is somewhat simpler in implementation. As WDRR seems to outperform PF when the number of connections is low, it could be feasible to combine PF and WDRR so that there would be a certain threshold (e.g., the number of users or connections), after which the scheduling algorithm would change from WDRR to PF.

For VoIP and other real-time traffic, DRR is still the best choice. It is not acceptable to let VoIP connections starve every now and then (when the most robust MCSs are used) just because that would lead to better MAC throughput. With PF scheduling, VoIP delay could grow intolerable if the number of VoIP connections were significant.

Backhaul QoS

We have shown that there is no need for major overprovisioning of the backhaul but one can (and should) apply well-known IP traffic management methods in order to control the bottleneck link loads. DiffServ can help; if VoIP and other real-time traffic are given priority over non-real time traffic, there is no need for immediate bottleneck link upgrades. Additionally, the real-time traffic load of the bottleneck link should be taken into account in VoIP admission control decisions.

We think that mapping from IEEE 802.16e data delivery service classes to DiffServ traffic classes should be kept as simple as possible; no gains can be achieved by having more than three DiffServ classes if the BS scheduler has only three priority levels. Moreover, network management becomes increasingly difficult when there are more classes.

AQM can be applied both for the connection-specific DL buffers at the BS and for the traffic aggregates in backhaul routers. AQM reduces delay considerably, especially when applied at the BS.

Connection Admission Control

We have presented two simple-to-implement measurement-based connection admission control methods for real-time services in IEEE 802.16e networks. Our simulations show that the proposed methods lead to the more efficient use of scarce radio resources than purely parameter-based connection admission control mechanisms with conservative limits for the number of connections admitted.

Since we cannot assume that connections always arrive in the system according to a Poisson process, we should combine the benefits of parameter-based and measurement-based admission control in order to manage batch arrivals. We call this method measurement-aided admission control. If the connections arrive according to a Poisson process, MAAC can admit as many

VoIP connections as MBAC. However, if there are batch arrivals, MAAC blocks connections more aggressively than MBAC and thus delays are better controlled. On the basis of our results, we can conclude that it would be unwise to use MBAC instead of MAAC.

Moreover, we have shown that a well-designed ertPS implementation is a more bandwidth efficient solution for silence suppression capable VoIP than rtPS. However, if the VoIP client does not support silence suppression and the packet size varies considerably (this is the case with, e.g., Skype), rtPS might be a better choice.

Different Channel Access and ertPS Resumption Mechanisms

We have presented different uplink channel access and ertPS resumption mechanisms for VoIP traffic in IEEE 802.16e systems. Our simulation studies indicate that we can have more ertPS VoIP connections (or better QoS), if we get rid of polling during the silence periods. However, this can result in long delays if the SS has to participate in contention after each silence period.

Multicast polling, with appropriate request backoff parameters, can lower the resumption delays. However, when there are many VoIP connections, we might need more multicast polling groups, which would take resources from the basic contention. Of course, multicast polling does not bring any gains if the basic contention is not a bottleneck.

Multicast polling group members (i.e., those SSs that host VoIP connections) send their bandwidth request CDMA codes in a dedicated multicast polling region, which prevents these codes from colliding with the codes sent by the BE connections. In our implementation, we have also prioritized uplink CDMA allocations based on the contention region: multicast polling group members always get the first uplink CDMA allocations.

CQICH can also be used for ertPS resumption. With CQICH based resumption, delays are lower when compared to contention based resumption – assuming that contention is a bottleneck. However, with large CQICH reporting intervals, this approach could cause some problems to link adaptation: if we use the CQICH message for resumption, we cannot update the SNR in the same CQICH message.

If the CQICH reporting interval is short enough, our recommendation is to use CQICH based ertPS VoIP resumption. If that is not the case, multicast polling with its own request backoff parameters should be used. With contention based resumption, we cannot guarantee low ertPS VoIP resumption delays unless the number of other connections participating in contention is somehow limited.

Moreover, we have observed some issues that make resource management and connection admission control in IEEE 802.16e quite challenging. A common approach is that CQICH reports, HARQ acknowledgements, HARQ retransmissions, and CDMA uplink allocations are always granted slots before any real-time connection. Therefore, it seems that admission control for real-time connections only is not sufficient but that, in addition to connection admission control, we should have admission control for the SSs when they are entering the network.

7 Author's Contribution

Publication P1: This paper is a joint work of the authors. The simulation studies were conducted and the paper was written by the present author.

Publication P2: This paper is a joint work of the authors. The simulation studies were conducted and the paper was written by the present author.

Publication P3: This paper is a joint work of the authors. The simulation studies were conducted by the present author and the paper was co-written by Renaud Cuny and the present author.

Publication P4: This paper is a joint work of the authors. The simulation studies were conducted and the paper was written by the present author.

Publication P5: This paper is a joint work of the authors. The simulation studies were conducted and the paper was written by the present author.

Publication P6: This paper is a joint work of the authors. The simulation studies were conducted and the paper was written by the present author.

Publication P7: This paper is a joint work of the authors. The simulation studies were conducted and the paper was written by the present author.

Publication P8: This paper is a joint work of the authors. The simulation studies were conducted and the paper was written by the present author.

The additional ns-2 code used in the simulation studies for Publications P1–P3 was written solely by the present author, while the IEEE 802.16e code, which was used in the simulation studies for Publications P4–P8, was a joint effort between the Telecommunication laboratory of University of Jyväskylä (JYU) and the present author. JYU delivered the basic IEEE 802.16e module and the present author added the features of interest (e.g., AQM, scheduling algorithms, and CAC) on top of the basic version.

8 Conclusions

The work for this thesis has been done over a long period of time (from the year 2002 up to the present date) and different technologies have been studied. However, in all the publications that are included in this thesis, we have focused our attention on QoS and resource management issues in MAC/IP layers and above.

There are many similarities between IP, EGPRS, WCDMA, and IEEE 802.16e network elements. IP packets are stored in the PDCP buffers at the RNC, as well as in the IEEE 802.16e BS and SS buffers. Moreover, the LLC PDU buffering at the 2G-SGSN utilizes mechanisms from the IP world. In all these bottlenecks, similar AQM mechanisms, such as RED, ECN, PDPC, and our own contribution TTLRED/TTLECN, can be used in order to reduce the DL delay and in some cases even to improve the TCP goodput. We have shown this in our AQM publications (Publications P2–P4).

It is also interesting to see that almost identical CAC algorithms can be applied in DiffServ-capable IP access networks (Publication P1 presents our modified bandwidth broker framework and adaptive CAC algorithms) and at IEEE 802.16e base stations (Publication P7 presents our algorithms, MBAC and MAAC). In the former case, one just has to first gather the link load information from the IP routers before the bandwidth broker can utilize it in admission decisions. In the latter case, the load information (number of free slots after real-time connections have been served) is already available. In both cases, our measurement-aided admission control algorithm adjusts the reservation limits, which provides protection against connections arriving in batches. This would not be possible if we simply used the current load in admission decisions.

In Publication P6, we speculate that the backhaul could become a bottleneck in IEEE 802.16e networks in the future, when radio capacity is increased. We have proposed that well-known IP QoS mechanisms such as DiffServ should be used in order to avoid costly overprovisioning of the backhaul. We present a simple mapping between IEEE 802.16e data delivery services and DiffServ traffic classes. Moreover, we propose that the IEEE 802.16e BS should take backhaul load into account it its admission decisions. Thus, radio and backhaul load would both have to be low enough before we could admit a new connection. This proposal combines the ideas presented in Publications P1 and P7.

In addition to AQM and CAC, we studied different IEEE 802.16e BS scheduling algorithms in Publication P5 and uplink channel access mechanisms in Publication P8. In the former study, we show that PF scheduling offers superior spectral efficiency when compared to DRR, though

in some cases at the cost of increased delay. Moreover, we propose a variant of DRR, where the *quantum* value depends on the MCS that is currently used. This scheme also proved to be spectrally more efficient than DRR.

The topic of our last publication (Publication P8) is uplink delays using different ertPS resumption mechanisms. We have shown that there are several ways to implement ertPS in an efficient manner, i.e., so that during the silence periods no (or very few) UL slots are granted. The problem here is how to implement the resumption after the silence period so that as little delay as possible is introduced. Granting one polling slot periodically during the silence period is a simple solution. However, the polling slots eat too much capacity. If contention is used, the problem is that there is no separation between VoIP connections that want to send a packet after a silence period and (BE) TCP connections that want to send an acknowledgement or data packet. Thus, CDMA codes from VoIP and BE SSs can collide and delays can easily grow too big.

In multicast polling, we have a dedicated contention region for a certain group (e.g., VoIP users) of SSs. Thus, the CDMA codes from VoIP and BE users cannot collide. However, a dedicated contention region eats capacity from the basic contention region. Moreover, the standard [802.16] mandates that the same set of request backoff parameters should be used with both basic contention and multicast polling. This does not encourage us to use multicast polling for ertPS VoIP resumption, unless the standard is amended so as to allow one to apply separate request backoff parameters for multicast polling.

Since there are bigger problems with other ertPS resumption mechanisms, it seems that using CQICH reports is the best resumption mechanism. However, this mechanism has one drawback, too: the link adaptation might suffer a little when the SNR information is replaced by the ertPS resumption codeword.

The common target of our contributions presented in Publications P1–P8 was to study the applicability of different traffic management mechanisms in order to avoid overprovisioning. Connection admission control, active queue management, advanced scheduling algorithms, and efficient ertPS VoIP resumption schemes make it possible to provide sufficient QoS with limited bandwidth resources. These mechanisms are not needed if there is plenty of cheap bandwidth available. However, we believe that despite recent advances in wireless technologies, bandwidth will continue to be a scarce resource in future wireless systems as well.

References

[802.16] Air interface for fixed broadband wireless access systems. IEEE Standard 802.16, Jun. 2004.

[802.16e] Air interface for fixed broadband wireless access systems – amendment for physical and medium access control layers for combined fixed and mobile operation in licensed bands. IEEE Standard 802.16e, Dec. 2005.

[Ala07] O. Alanen, "Multicast Polling and Efficient VoIP Connections in IEEE 802.16 Networks," in Proc. IEEE/ACM MSWiM 2007, Chania, Greece, Oct. 2007, pp. 289–295.

[And07] J. Andrews, A. Ghosh, and R. Muhamed, Fundamentals of WiMAX: Understanding Broadband Wireless Networking. Prentice Hall, 2007.

[App04] G. Appenzeller, I. Keslassy, and N. McKeown, "Sizing Router Buffers," in Proc. SIGCOMM 2004, Portland, Oregon, USA, Aug.–Sep., 2004.

[Arm00] G. Armitage, Quality of Service in IP Networks: Foundations for a Multi-Service Internet. Indianapolis, Indiana, USA: Macmillan, 2000.

[Bol99] V. Bolotin, "Characterizing Data Connection and Messages by Mixtures of Distributions on Logarithmic Scale," in Proc. 16th Int. Teletraffic Congr., Edinburgh, U.K., Jun. 1999, pp. 887–894.

[Bre00] L. Breslau, S. Jamin, and S. Shenker, "Comments on the Performance of Measurement-based Admission Control Algorithms," in Proc. IEEE INFOCOM 2000, Tel-Aviv, Israel, Mar. 2000, pp. 1233–1242.

[Chi08] Chingyao Huang, Hung-Hui Juan, Meng-Shiang Lin, and Chung-Ju Chang, "Radio Resource Management of Heterogeneous Services in Mobile WiMAX systems," IEEE Wireless Communications, vol.14, no. 1, pp.20-26, Feb. 2007.

[Den09] D.J. Deng, L.W. Chang, C.H. Ke, Y.M. Huang, and J.M. Chang, "Delay Constrained Uplink Scheduling Policy for rtPS/ertPS Service in IEEE 802.16e BWA Systems," International Journal of Communication Systems, vol. 22, no. 2, pp. 119–133, Feb. 2009.

[Flo] Sally Floyd's RED page. [Online]. Available: http://www.icir.org/floyd/red.html

[Flo01] S. Floyd, R. Gummadi, and S. Shenker, "Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management," Aug. 2001. [Online.] Available: http://www.icir.org/floyd/papers/adaptiveRed.pdf

[Flo93] S. Floyd and V. Jacobson, “Random Early Detection Gateways for Congestion Avoidance,” IEEE/ACM Trans. Netw., vol. 1, no. 1, pp. 397–413, Aug. 1993.

[Gil60] E.N. Gilbert, “Capacity of a Burst-Noise Channel,” Bell Syst. Tech. J., vol. 39, pp. 1253–1266, Sep. 1960.

[Gur03] A. Gurtov and R. Ludwig, “Lifetime Packet Discard for Efficient Real-Time Transport over Cellular Links,” ACM Mobile Computing and Communications Review, vol. 7, no. 4, pp. 32–45, Oct. 2003.

[Jal00] A. Jalali, R. Padovani, and R. Pankaj, “Data Throughput of CDMA-HDR: A High Efficiency-High Data Rate Personal Communication Wireless System,” in Proc. IEEE VTC2000-Spring, Tokyo, Japan, May 2000, pp. 1854–1858.

[Kal98] L. Kalampoukas, A. Varma, and K.K. Ramakrishnan, “Explicit Window Adaptation: A Method to Enhance TCP Performance,” in Proc. IEEE INFOCOM 1998, San Francisco, CA, USA, Apr. 1998, pp. 242–251.

[Kar06] J. Karhula, “WiMAX-verkon simulointi NS2-verkkosimulaattorilla,” Master’s Thesis, University of Jyväskylä, Finland, Feb. 2006.

[Kaw02] R. Kawahara and N. Komatsu, “Dynamically Weighted Queueing for Fair Bandwidth Allocation and its Performance Analysis,” in Proc. IEEE ICC 2002, New York, USA, Apr.–May 2002, pp. 2379–2383.

[Kes88] S. Keshaw, “REAL: A Network Simulator,” Technical Report: CSD-88-472, University of California at Berkeley, Dec.1988.

[Kil02] K. Kilkki and J. Ruutu, “Selection of QoS Mechanisms Based on Operator’s Business Objective,” in Proc. 16th Nordic Teletraffic Seminar, Espoo, Finland, Aug. 2002, pp. 313–324.

[Kil99] K. Kilkki, Differentiated Services for the Internet. Indianapolis, Indiana, USA: Macmillan, 1999.

[Mag88] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J. Robbins, “Performance Models of Statistical Multiplexing in Packet Video Communications,” IEEE Trans. Commun., vol. 36, no. 7, pp. 834–844, Jul. 1988.

[Mai09] G. Maier, A. Feldmann, V. Paxson, and M. Allman, “On Dominant Characteristics of Residential Broadband Internet Traffic,” to appear in Proc. Internet Measurement Conference 2009, Chicago, USA, Nov. 2009.

[Mis00] V. Misra, W. Gong, and D. Towsley, "Fluid-Based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED," in Proc. SIGCOMM 2000, New York, USA, Aug.–Sep. 2000, pp. 151–160.

[Mol00] M. Molina, P. Castelli, and G. Foddis, "Web Traffic Modeling Exploiting TCP Connections' Temporal Clustering through HTML-REDUCE," IEEE Netw., vol. 14, no. 3, pp. 46–55, May-Jun. 2000.

[Moy98] J. T. Moy, OSPF: Anatomy of an Internet Routing Protocol. Reading, Massachusetts, USA: Addison-Wesley, 1998.

[Niy05] D. Niyato and E. Hossain, "Connection Admission Control Algorithms for OFDM Wireless Networks," in Proc. IEEE Globecom 2005, St. Louis, Missouri, USA, Nov. 2005, pp. 2455–2459.

[NS2] UCB/LBNL/VINT. (2003) Network Simulator – ns (Version 2). [Online]. Available: http://www.isi.edu/nsnam/ns/index.html

[RFC1157] J. Case, M. Fedor, M. Schoffstall, and J. Davin, "A Simple Network Management Protocol (SNMP)," RFC 1157, May 1990.

[RFC1633] R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: An overview," Request for Comments 1633, Jun. 1994.

[RFC2475] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," RFC 2475, Dec. 1998.

[RFC2581 M. Allman, V. Paxson, and W. Stevens, "TCP Congestion Control," RFC 2581, Apr. 1999.

[RFC2582] S. Floyd and T. Henderson, "The NewReno Modification to TCP's Fast Recovery Algorithm," RFC 2582, Apr. 1999.

[RFC2597] J. Heinänen, F. Baker, W. Weiss, and J. Wroclawski, "Assured Forwarding PHB Group," RFC 2597, Jun. 1999.

[RFC2616] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext Transfer Protocol – HTTP/1.1," RFC 2616, Jun. 1999.

[RFC2638] K. Nichols, V. Jacobson, and L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet," RFC 2638, Jul. 1999.

[RFC3168] K. Ramakhrisnan, S. Floyd, and D .Black, "The Addition of Explicit Congestion Notification (ECN) to IP," RFC 3168, Sep. 2001.

[RFC3246] B. Davie, A. Charny, J. C. R. Bennett, K. Benson, J. Y. Le Boudec, W. Courtney, S. Davari, V. Firoiu, and D. Stiliadis, "An Expedited Forwarding PHB," RFC 3246, Mar. 2002.

[RFC4594] J. Babiarz, K. Chan, and F. Baker, "Configuration Guidelines for DiffServ Service Classes," RFC 4594, Aug. 2006.

[Såg03] M. Sågfors, R. Ludwig, M. Meyer, and J. Peisa, "Queue Management for TCP Traffic over 3G Links," in Proc. IEEE Wireless Communications and Networking Conference (WCNC), New Orleans, Lousiana, USA, Mar. 2003, pp. 1663–1668.

[Say06] A. Sayenko, O. Alanen, J. Karhula, and T. Hämäläinen, "Ensuring the QoS Requirements in 802.16 Scheduling," in Proc. IEEE/ACM MSWiM 2006, Torremolinos, Spain, Oct. 2006, pp. 108–117.

[Say07a] A. Sayenko, O. Alanen, and T. Hämäläinen, "Adaptive Contention Resolution Parameters for the IEEE 802.16 Networks," in Proc. Qshine 2007, Vancouver, Canada, Aug. 2007.

[Say07b] A. Sayenko, V. Tykhomyrov, H. Martikainen, and O. Alanen, "Performance Analysis of the 802.16 ARQ Mechanism," in Proc. IEEE/ACM MSWiM 2007, Chania, Greece, Oct. 2007, pp. 314–322.

[Say09] A. Sayenko, O. Alanen, H. Martikainen, V. Tykhmyrov, A. Puchko, and T. Hämäläinen, "WINSE: WiMAX NS-2 Extension", in Proc. 2nd International Conference on Simulation Tools and Techniques, Rome, Italy, Mar. 2009.

[Sch98] O. Schelén, "Quality of Service Agents in the Internet," Ph.D. Dissertation, Luleå University of Technology, Luleå, Sweden, 1998.

[Shi02] H. Shimonishi, I. Maki, T. Murase, and M. Murata, "Dynamic Fair Bandwidth Allocation for DiffServ Classes," in Proc. IEEE ICC, New York, USA, Apr. – May 2002, pp. 2348–2352.

[Shr96] M. Shreedhar and G. Varghese, "Efficient Fair Queueing Using Deficit Round-robin," IEEE/ACM Trans. Netw., vol. 4, pp. 375–385, Jun. 1996.

[Soo08] J.W. Soo, "Performance Analysis of Uplink Scheduling Algorithms for VoIP Services in the IEEE 802.16e OFDMA System," Wireless Personal Communications, vol. 47, no. 2, pp. 247–263, Oct. 2008.

[Tei99] B. Teitelbaum, S. Hares, L. Dunn, R. Neilson, V. Narayan, and F. Reichmeyer, "Internet2 QBone: Building a Testbed for Differentiated Services," IEEE Netw., vol. 13, no. 5, pp. 8–16, Sep.–Oct. 1999.

[TS23.107] 3GPP, "3GPP TS 23.107, V5.9.0; Technical Specification Group Services and System Aspects; Quality of Service (QoS) concept and architecture (Release 5)," Jun. 2003.

[TS25.308] 3GPP, "3GPP TS 25.308, V5.4.0; Technical Specification Group Radio Access Network; High Speed Packet Access (HSDPA); Overall description; Stage 2 (Release 5)," Mar. 2003.

[TS25.321] 3GPP, "3GPP TS 25.321, V5.5.0; Technical Specification Group Radio Access Network; Multi Access Control (MAC) protocol specification (Release 5)," Jun. 2003.

[TS25.322] 3GPP, "3GPP TS 25.322, V5.5.0; Technical Specification Group Radio Access Network; Radio Link Control (RLC) protocol specification (Release 5)," Jun. 2003.

[TS25.323] 3GPP, "3GPP TS 25.323, V6.0.0; Technical Specification Group Radio Access Network; Packet Data Convergence Protocol (PDCP) specification (Release 6)," Dec. 2003.

[TS25.435] 3GPP, "3GPP TS 25.435, V5.5.0; Technical Specification Group Radio Access Network; UTRAN Iub Interface User Plane Protocols for Common Transport Channel data streams (Release 5)," Jun. 2003.

[TS25.877] 3GPP, "3GPP TS 25.877, V5.1.0; Technical Specification Group Radio Access Network; High Speed Downlink Packet Access: Iub/Iur protocol aspects (Release 5)," Jun. 2002.

[TS48.018] 3GPP TS 48.018. 3rd Generation Partnership Project; Technical Specification Group GSM, EDGE Radio Access Network; General Packet Radio Service (GPRS); Base Station System (BSS) – Serving GPRS Support Node (SGSN); BSS GPRS Protocol (BSSGP) (rel'6).

[Tsa07] K.F. Tsang, L.T. Lee, H.Y. Tung, R. Lam, Y.T. Sun, and K.T. Ko, "Admission Control Scheme for Mobile WiMAX Networks," in Proc. IEEE International Symposium on Consumer Electronics, ISCE 2007, Dallas, Texas, USA, Jun. 2007, pp. 1–5.

[Wan01] H. Wang, C. Shen, and K. G. Shin, “Adaptive-weighted Packet Scheduling for Premium Service,” in Proc. IEEE ICC, Helsinki, Finland, Jun. 2001, pp. 1846–1850.

[Wan08] F. Wang, A. Ghosh, C. Sankaran, P.J. Fleming, F. Hsieh, and S.J. Benes, “Mobile WiMAX Systems: Performance and Evolution,” IEEE Communications Magazine, vol. 46, no. 10, pp. 41–49, Oct. 2008.

[Wig03] J. Wigard, T. Kolding, K. Pedersen, H. Holma, and P. Mogensen, “High Speed Downlink Packet Access (HSDPA) for WCDMA,” Nov. 2003. [Online]. Available: http://www.nokia.com/research

[WMF08] WiMAX Forum, “Requirements for WiMAX Femtocell Systems,” version 0.3.0, Jul. 2008.

[Won08] Y.J. Won, M-J Choi, B-C Park, J.W. Hong, H-W Lee, C-K Hwang, J-H Yoo, “End-User IPTV Traffic Measurement of Residential Broadband Access Networks," in Proc. IEEE NOMS Workshops, Salvador, Bahia, Brazil, Apr. 2008, pp. 95–100.