Department of Communications and Networking

Adaptive Scheduling in Cellular Access, Wireless Mesh and IP Networks

Johanna Nieminen (neé Antila)

Aalto University

DOCTORAL
DISSERTATIONS

Aalto University publication series
DOCTORAL DISSERTATIONS 39/2011

Adaptive Scheduling in Cellular Access, Wireless Mesh and IP Networks

Johanna Nieminen (neé Antila)

Doctoral dissertation for the degree of Doctor of Science in Technology/ in Architecture (Doctor of Philosophy) to be presented with due permission of the School of Electrical Engineering for public examination and debate in Auditorium S4 at the Aalto University School of Electrical Engineering (Espoo, Finland) on the 20th of May 2011 at 12 noon (at 12 o'clock).

Aalto University
School of Electrical Engineering
Department of Communications and Networking

Supervisor
Professor Riku Jäntti

Instructor
Professor Riku Jäntti

Preliminary examiner
Professor Seong-Lyun Kim (Yonsei University, Korea) and Professor Timo Hämäläinen (University of Jyväskylä, Finland)

Opponent
Professor Seong-Lyun Kim (Yonsei University, Korea) and Professor Tapani Ristaniemi (University of Jyväskylä, Finland)

Aalto University publication series
DOCTORAL DISSERTATIONS 39/2011

ISBN 978-952-60-4115-5 (pdf)
ISBN 978-952-60-4114-8 (printed)
ISSN-L 1799-4934
ISSN 1799-4942 (pdf)
ISSN 1799-4934 (printed)

Aalto Print
Helsinki 2011

Finland

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

Aalto University

Abstract

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Author
Johanna Nieminen (neé Antila)

Name of the doctoral dissertation
Adaptive Scheduling in Cellular Access, Wireless Mesh and IP Networks

Publisher School of Electrical Engineering

Unit Department of Communications and Networking

Series Aalto University publication series DOCTORAL DISSERTATIONS 39/2011

Field of research Communications technology

Manuscript submitted 12 November 2010 **Manuscript revised** 8 March 2011

Date of the defence 20 May 2011 **Language** English

☒ **Monograph** □ **Article dissertation (summary + original articles)**

Abstract

Networking scenarios in the future will be complex and will include fixed networks and hybrid Fourth Generation (4G) networks, consisting of both infrastructure-based and infrastructure-less, wireless parts. In such scenarios, adaptive provisioning and management of network resources becomes of critical importance. Adaptive mechanisms are desirable since they enable a self-configurable network that is able to adjust itself to varying traffic and channel conditions. The operation of adaptive mechanisms is heavily based on measurements. The aim of this thesis is to investigate how measurement based, adaptive packet scheduling algorithms can be utilized in different networking environments.

The first part of this thesis is a proposal for a new delay-based scheduling algorithm, known as Delay-Bounded Hybrid Proportional Delay (DBHPD), for delay adaptive provisioning in DiffServ-based fixed IP networks. This DBHPD algorithm is thoroughly evaluated by ns2-simulations and measurements in a FreeBSD prototype router network. It is shown that DBHPD results in considerably more controllable differentiation than basic static bandwidth sharing algorithms. The prototype router measurements also prove that a DBHPD algorithm can be easily implemented in practice, causing less processing overheads than a well known CBQ algorithm.

The second part of this thesis discusses specific scheduling requirements set by hybrid 4G networking scenarios. Firstly, methods for joint scheduling and transmit beamforming in 3.9G or 4G networks are described and quantitatively analyzed using statistical methods. The analysis reveals that the combined gain of channel-adaptive scheduling and transmit beamforming is substantial and that an On-off strategy can achieve the performance of an ideal Max SNR strategy if the feedback threshold is optimized. Finally, a novel cross-layer energy-adaptive scheduling and queue management framework EAED (Energy Aware Early Detection), for preserving delay bounds and minimizing energy consumption in WLAN mesh networks, is proposed and evaluated with simulations. The simulations show that our scheme can save considerable amounts of transmission energy without violating application level QoS requirements when traffic load and distances are reasonable.

Keywords Scheduling, Modulation, Transmit Beamforming, QoS, Energy, Simulation, Implementation

ISBN (printed) 978-952-60-4114-8 **ISBN (pdf)** 978-952-60-4115-5

ISSN-L 1799-4934 **ISSN (printed)** 1799-4934 **ISSN (pdf)** 1799-4942

Location of publisher Espoo **Location of printing** Helsinki **Year** 2011

Pages 263 **The dissertation can be read at** http://lib.tkk.fi/Diss/

Aalto-yliopisto

Tiivistelmä

Aalto-yliopisto, PL 11000, 00076 Aalto www.aalto.fi

Tekijä
Johanna Nieminen (neé Antila)

Väitöskirjan nimi
Adaptiivinen vuoronjako solukkoverkoissa, langattomissa mesh-verkoissa ja IP-verkoissa

Julkaisija Sähkötekniikan korkeakoulu

Yksikkö Tietoliikenteen- ja tietoverkkotekniikan laitos

Sarja Aalto University publication series DOCTORAL DISSERTATIONS 39/2011

Tutkimusala Tietoliikennetekniikka

Käsikirjoituksen pvm 12.11.2010 **Korjatun käsikirjoituksen pvm** 08.03.2011

Väitöspäivä 20.05.2011 **Kieli** Englanti

☒ **Monografia** □ **Yhdistelmäväitöskirja (yhteenveto-osa + erillisartikkelit)**

Tiivistelmä

Tulevaisuudessa tietoverkkoja sovelletaan mitä moninaisimmissa skenaarioissa. Skenaariot käsittävät toisaalta kiinteitä IP-verkkoja ja toisaalta neljännen sukupolven (4G) hybridiverkkoja, joihin kuuluu sekä infrastruktuurillisia että infrastruktuurittomia osia. Tällaisissa skenaarioissa verkkoresurssien adaptiivinen provisiointi ja hallinta muodostuvat erittäin tärkeiksi. Adaptiiviset mekanismit ovat hyödyllisiä, sillä niiden avulla voidaan toteuttaa itsekonfiguroituva verkko, joka pystyy sopeutumaan muuttuviin liikenne- ja kanavatilanteisiin. Adaptiiviset mekanismit hyödyntävät toiminnassaan mittausinformaatiota. Väitöskirjassa tutkitaan, miten mittauspohjaisia, adaptiivisia vuoronjakomekanismeja voidaan soveltaa erilaisissa verkkoympäristöissä.

Väitöskirjan ensimmäisessä osassa kuvataan uusi adaptiivinen, viiverajoitettu suhteelliseen hybridiviiveeseen perustuva vuoronjakoalgoritmi (DBHPD). Algoritmin suorituskyky testataan perusteellisesti sekä simulaatioilla että mittauksilla FreeBSD-prototyyppireitittimessä. Tulokset osoittavat, että DBHPD johtaa huomattavasti hallitumpaan differentiointiin kuin staattiset kaistanjakoon perustuvat algoritmit. Prototyyppireitittimen mittaustulokset osoittavat lisäksi, että DBHPD voidaan toteuttaa helposti käytännössä, mikä johtaa pienempään prosessointikuormaan kuin tunnettu CBQ-algoritmi.

Väitöskirjan toisessa osassa keskustellaan 4G-hybridiverkkoskenaarioiden asettamista erityisvaatimuksista vuoronjaolle. Aluksi kuvataan ja analysoidaan matemaattisia menetelmiä käyttäen 3.9G- ja 4G-verkoissa mekanismeja, joissa vuoronjako yhdistetään keilanmuodostukseen. Analyysin perusteella voidaan todeta, että yhdistetyllä kanava-adaptiivisella vuoronjaolla ja keilanmuodostuksella pystytään saavuttamaan merkittävää hyötyä, ja että on-off-strategia saavuttaa ideaalisen maksimaaliseen signaalikohinasuhteeseen perustuvan strategian suorituskyvyn, kun vastaanottajan palautteen kynnysarvo optimoidaan. Lopuksi kehitetään WLAN mesh -verkkoihin uusi energia-adaptiivinen vuoronjako- ja jononhallinta-algoritmi EAED, jonka tarkoituksena on säilyttää viiverajat ja minimoida energiankulutus. Algoritmi testataan useissa simulaatioskenaarioissa. Simulaatiot osoittavat, että EAED-algoritmilla voidaan säästää merkittävästi energiaa sovellusten palvelunlaatuvaatimuksia rikkomatta olettaen, että liikennekuorma ja etäisyydet ovat kohtuulliset.

Avainsanat Vuoronjako, Modulaatio, Keilanmuodostus, Palvelunlaatu, Energia, Simulaatio, Implementaatio

ISBN (painettu) 978-952-60-4114-8 **ISBN (pdf)** 978-952-60-4115-5

ISSN-L 1799-4934 **ISSN (painettu)** 1799-4934 **ISSN (pdf)** 1799-4942

Julkaisupaikka Espoo **Painopaikka** Helsinki **Vuosi** 2011

Sivumäärä 263 **Luettavissa verkossa osoitteessa** http://lib.tkk.fi/Diss/

Preface

The first part of this thesis was written as part of the IroNet project while I was working in the Networking Laboratory at the Helsinki University of Technology. The second part of the thesis was written when I was with the Nokia Research Center, first as a GETA graduate school student via the HUT Communications Laboratory, and later as a full-time research leader and Principal researcher.

I would like to express my gratitude to all the co-authors: Mika Husso, Prof. Jyri Hämäläinen, Marko Luoma, Olli-Pekka Lamminen, Prof. Riku Jäntti, Prof. Jukka Manner, Antti Paju, Harri Paloheimo, Taneli Riihonen, Prof. Risto Wichman and Prof. Antti Ylä-Jääski. This thesis would not have been possible without their active cooperation on several intriguing topics.

Special thanks go to Prof. Riku Jäntti, who has been supervising my postgraduate studies after my Licentiate thesis. His strong mathematical background has greatly complemented the measurement-based algorithm design and his timely comments have been especially helpful when writing the final manuscript of the thesis.

I would also like to thank all my colleagues at Nokia and especially Mauri Honkanen, Harri Paloheimo, Jarkko Kneckt, Mika Kasslin and laboratory directors Petteri Alinikula and Hannu Kauppinen for the possibility of conducting radio research in the industry.

Funding is essential for long term research work. I sincerely appreciate the privilege of receiving financing from several sources: The GETA graduate school, projects and individual grants from Nokia and TES.

Finally, I would like to express my deep gratitude to my parents and family, my husband Tuomas and children Sara, Aada and Eerik for their love and extra energy required to complete this thesis.

Contents

List of Abbreviations

AC	Access Category
ACK	Positive Acknowledgement
ADC	Absolute Delay Constraints
AF	Assured Forwarding
AIFS	Arbitrary Interframe Space
ALTQ	Alternate Queuing
AMC	Adaptive Modulation and Coding
AP	Access Point
AR	Access Router
ARQ	Automatic Repeat reQuest
ASIC	Application-specific Integated Circuit
B3G	Beyond 3G
BEB	Binary Exponential Backoff
BEP	Bit-Error Probability
BER	Bit-Error Ratio
BLER	Block Error Rate
BI	Backoff Interval
BS	Base Station
BPA	Back Propagation Algorithm
BPSK	Binary Phase Shift Keying
CBQ	Class Based Queueing
CBR	Constant Bit Rate
CCK	Complementary Code Keying
CCQ	Cisco Custom Queueing
CD-EDD	Channel Dependent Earliest Due Date
CDF	Cumulative Distribution Function
CIR	Carrier-to-Interference Ratio

CLD	Cross Layer Design
CN	Core Network
CoMP	Coordinated multi-point transmission
CPC	Common Power Control
CQI	Channel Quality Information
CPU	Central Processing Unit
CSI	Channel State Information
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTS	Clear to Send
CW	Contention Window
DBHPD	Delay-bounded Hybrid Proportional Delay
DBPSK	Differential Binary Phase Shift Keying
DCF	Distributed Coordination Funtion
DDC	Downlink Dedicated Channel
DDP	Delay Differentiation Parameter
DFS	Delay Fair Scheduling
DHS	Decreasing Hazard Rate
DiffServ	Differentiated Services
DIFS	DCI Interframe Space
DMS	Dynamic Modulation Scaling
DPCCH	Dedicated Physical Control Channel
DQPSK	Differential Quadrature Phase Shift Keying
DRR	Deficit Round Robin
DSCH	Downlink Shared Channel
DSSS	Direct Sequence Spread Spectrum
DTN	Delay Tolerant Network
EAED	Energy Aware Early Detection
EDCF	Enhanced Distributed Coordination Function
EDD	Earliest Due Date
EEN	Explicit Energy Notification

EF	Expedited Forwarding
eNodeB	E-UTRAN NodeB
EPC	Evolved Packet Core
ER	Exponential Rule
ETSI-SMG	European Telecommunications Standard Institute-Special Mobile Groups
E-UTRAN	evolved UMTS Terrestrial Radio Access Network
EWMA-pe	EWMA Estimator based on Proportional Error of the Estimate
EWMA-r	EWMA Estimator with Restart
FB	Foreground Background
FBI	Feedback Indicator
FCFS	First Come First Served
FC-ID	Flow-Class Identifier
FD	Frequency Domain
FDD	Frequency Division Duplex
FDPS	Frequency Domain Packet Scheduling
FEC	Forward Error Correction
FER	Frame Error Rate
FIFO	First In First Out
FPLS	Fair Packet Loss Sharing
FRF	Frequency Reuse Factor
FRS	Fair Resource Scheduling
FSM	Feedback Signalling Message
FT	Fair Throughput
FTP	File Transfer Protocol
3G	Third Generation
3GPP	Third Generation Partnership Project
4G	Fourth Generation
GPS	Generalized Processor Sharing
HARQ	Hybrid ARQ
HFSC	Hierarchical Fair Service Curve

HPD	Hybrid Proportional Delay
HSPA	High Speed Packet Access
HSDPA	High Speed Downlink Packet Access
HSUPA	High Speed Uplink Packet Access
HTB	Hierarchical Token Bucket
HTTP	Hypertext Transfer Protocol
ICI	Inter-Cell Interference
ICIC	Inter Cell Interference Coordination
IETF	Internet Engineering Task Force
iHSPA	Internet HSPA
IMT	International Mobile Telecommunications
IMT-A	International Mobile Telecommunications Advanced
IntServ	Integrated Services
IPC	Independent Power Control
IP	Internet Protocol
ITU	International Telecommunication Union
ITU-R	International Telecommunication Union - Radio
IWFQ	Idealized Wireless Fair Queueing
JoBS	Joint Buffer Management and Scheduling
LTE	3G Long Term Evolution
LTE-A	3G Longer Term Evolution-Advanced
LTFS	Long-Term fairness Server
MAC	Medium Access Protocol
MAP	Mesh Access Point
Max SNR	Maximum SNR
MCS	Modulation and Coding Scheme
MGW	Mobile Gateway
MIMO	Multiple Input Multiple Output
MISO	Multiple-Input Single-Output
MME	Mobility Management Entity

MMSE	Minimum Mean Square Error
MP	Mesh Point
MPLS	Multi Protocol Label Switching
MR	Mobile Router
MRC	Maximal-Ratio Combining
MT	Mobile Terminal
MUP	Most Urgent Packet
NACK	Negative Acknowledgement
Node B	Base Station
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiple Access
OVSF	Orthogonal Variable Spreading Factors
PA	Power Amplifier
PAPR	Peak-to-Average Power Ratio
PCF	Point Coordination Function
PCI	Precoding Control Information
PDF	Probability Distribution Function
PF	Proportional Fair
P-GW	Packet Data Network Gateway
PHB	Per Hop Behavior
PLP	Packet Loss Probability
PN	Pseudo Noise
P2P	Peer to Peer
PRR	Packet Round Robin
PRIQ	Priority Queuing
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
QWPF	QoS Aware Weighted Proportional Fair
RB	Resource Block

RF	Radio Frequency
RDC	Relative Delay Constraints
RED	Random Early Detection
RMLP	Recurrent Multilayer Perceptron Network
RNC	Radio Network Controller
RNS	Radio Network Subsystem
R99	Release 99
RR	Round Robin
RS	Relay Station
RTS	Ready to Send
RU	Resource Unit
SBFA	Server Based Fairness Approach
SC-FDMA	Single Carrier FDMA
SCFQ	Self Clocked Fair Queuing
SCS	Scheduling Candidate Set
SEL	Selection Combining
SF	Spreading Factor
S-GW	Serving Gateway
SIFS	Short Interframe Space
SIMO	Single-Input Multiple-Output
SLA	Service Level Agreement
SINR	Signal to Interference and Noise Ratio
SNR	Signal to Noise Ratio
SPRT	Shortest Remaining Processing Time
SRS	Sounding Reference Symbol
TB	Token Bucket
TBER	Transmission Bit-Error Ratio (BER)
TBF	Transmit Beamforming
TBTT	Beacon Transmission Time
TCP	Transmission Control Protocol

TD	Time Domain
TDMA	Time Division Multiple Access
TDCDMA	Time Division-Code Division Multiple Access
TDD	Time Division Duplex
TDPS	Time Domain Packet Scheduling
TES	Transform Expand Sample
TPC	Transmission Power Control
TTI	Transmit Time Interval
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
UTRAN	UMTS Terrestrial Radio Access Network
VLAN	Virtual Local Area Network
VoIP	Voice over IP
VBR	Variable Bit Rate
WCDMA	Wideband Code Division Multiple Access
WDM	Wave Division Multiplexing
WFQ	Weighted Fair Queuing
WFS	Wireless Fair Service
WLAN	Wireless Local Area Network
WNI	Wireless Network Interface
WF^{2}Q	Worst Case Weighted Fair Queuing
WPS2	WPS Extensions
WRED	Weighted Random Early Detection
WRR	Weighted Round Robin
WWW	World Wide Web
ZF	Zero Forcing

List of Symbols

$\max(\cdot)$	maximum
$\min(\cdot)$	minimum
$\arg\max(\cdot)$	maximizing argument of $(\cdot)$
$\arg\min(\cdot)$	minimizing argument of $(\cdot)$
$exp(\cdot)$	exponential function
$E[\cdot]$	expectation operator
$E\{\cdot\}$	expectation operator
$\bar{(\cdot)}$	mean operator
$f(\cdot)$	probability distribution function of $(\cdot)$
$f(\cdot\vert\cdot\cdot)$	probability distribution function of $(\cdot)$ with the condition of $\cdot\cdot$
$F(\cdot)$	cumulative distribution function of $(\cdot)$
$P(\cdot)$	probability of an event
$u(\cdot)$	step function
$\Upsilon(\cdot,\cdot\cdot)$	incomplete gamma function
$(\cdot)^{sup}$	the supremum
$(\cdot)^{inf}$	the infinum
$p(\cdot)$	probability function of $(\cdot)$
$N(\cdot,\cdot\cdot)$	normal distribution
Δ	delta function
Γ	Gamma function
Q	Q function
$\frac{\partial x}{\partial y}$	partial derivative
$(\cdot)^*$	complex conjugate
j	imaginary unit
$\propto$	proportional to
$[\cdot]^{-1}$	inverse of a matrix

$[\cdot]^H$	Hermitian transpose (complex conjugate transpose) of a matrix
$\|\cdot\|$	Euclidean norm
$\lceil x \rceil$	the smallest integer not less than x.
$\widehat{\mathbf{x}}$	dominant eigenvector of A in $\underset{\mathbf{x}\,:\,\|\mathbf{x}\|^2=1}{\mathtt{argmax}}\mathbf{x}^\dagger\mathbf{A}\mathbf{x}$
a	factor for tuning the impact of delay
a_i^k	arrival time of the k^{th} packet of class i
$a_i(t)$	amount of traffic arrivals from class i at time t
abs_factor_i	absorption factor for class i
A	bandwidth efficiency factor
A	all transmission loss components
A_i	arrival curve of class i
AC	number of access categories
$AIFS[AC]$	arbitrary inter frame space per AC
$AIFSN[AC]$	arbitration IFS number
α	filtering coefficient of an estimator
α_{short}	short term weighting factor
α_{long}	long term weighting factor
b	modulation level
β	factor taking into account elements such as filter non-idealities
b_{i-max}	lag bound for lagging flow i
B	system bandwidth (Hertz)
B	SNR efficiency factor
B_i	backlog of class i
$B(i)_h$	value for previous backoff timer at hop h
$B(t)$	denotes the set of currently backlogged classes
BI	random backoff interval
C	link capacity
C	wireless link capacity in an ideal channel

C_E	factor
C_R	factor
$\frac{C}{I}_i$	current Carrier-to-Interface ratio (CIR) of user i
CW_{min}	minimum backoff period
CW_{max}	maximum backoff period
CW	contention window
$CW_{min}[AC]$	minimum backoff period per AC
$CW_{max}[AC]$	maximum backoff period per AC
$cycle_i$	period for restarting the EWMA estimator for class i
δ_i	Delay Differentiation Parameter (DDP) of class i
$\bar{d}_i$	average queuing delay of class i packets
$\bar{d}_i(m)$	average delay of class i when m packets have departed
d_i	period of class i
d_i	worst-case delay-bound of class i
d_{max}	delay bound in the highest class
$\tilde{d}_i(m)$	normalized average queuing delay of class i when m packets have departed
$D_{ack,h}$	duration of ACK at hop h
D_h	total delay at hop h
D_i	delay of class i
$D_i(m)$	sequence of class i packets that have been served
$D_i(kT)$	head-of-line packet delay for flow i
$\tilde{D}_{i,s}(t)$	projected delay of class i at time s
$\bar{D}_{i,s}$	time averaged projected delay of class i over a period $\tilde{T}_{i,s}$
D_i^k	deadline of kth packet in class i
$D_{k,h}$	total delay of packet k at hop h
$D_{mac,h}$	estimated waiting time of all data packets in the node's buffers at hop h due to MAC layer mechanisms
D_{max}	maximum delay
$D_{max,i,k}$	maximum delay for user i's flow k
DT_i	delay threshold of flow i

$e_j[s]$	parameter that is back propagated to the preceding layers of a neural network
e_k^-	a priori estimate error
e_k	a posteriori estimate error
ϵ	constant
E	absolute error function at the output layer of a neural network
E_{bit}	energy consumed for transmitting one bit
E_k	new estimate
$\tilde{E}_{k,h}$	energy consumption of packet k at hop h
E_{long}	effective bandwidth
f	transfer function
f_i	load fraction produced by class i
F_0	predefined CDF level
$F(x_s)$	objective function
g	weighting coefficient
γ	weighting factor
γ	received SNR
$\bar{\gamma}$	received SNR averaged over active users set in a longer time scale
$\check{\gamma}$	SNR that is needed in the Round Robin strategy to reach the predefined CDF level
$\hat{\gamma}$	SNR that is needed in the Max SNR strategy to reach the predefined CDF level
γ_0	predefined SNR threshold
γ_m	received SNR from antenna m
$\gamma(P^{\text{out}})$	SINR needed to achieve a given outage probability P^{out}
γ_i	EWMA filter base weight in class i
$\gamma_a P_{loss}^*$	upper loss threshold
$\gamma_b P_{loss}^*$	lower loss threshold
g_i	service rate for class i
$g_i(t)$	instantaneous service rate for class i
G	SNR gain

G_n	SNR gain for Mode n
h	complex valued channel coefficient
h	index of a hop
h_m	complex valued channel coefficient for antenna m
$\mathbf{h}$	vector of complex valued channel coefficients
H	constant matrix
$\mathbf{H}_n$	channel matrix associated with the receive antenna n
i	index of a service class
i	index of an access category
i	index
$I_j[k]$	combined input of the j-th perceptron in layer k
$IFS(i)_h$	the next inter frame space in access category i at hop h
j	index of a service class
j	index
k	index of a packet
k	index of a neural network layer
k	time
k	frame
k	number of bits in scheduling feedback
K	number of TTIs
k^*	optimal value for the number of codes
k_i	differentiation coefficient
K	integer constant
K	threshold for dropping packets
K	blending factor
K	number of mobile users
l	output layer of a neural network
λ	flow arrival intensity
λ_i	average rate of user i
l_i	current lead of flow i

l_{i-max}	lead bound for leading flow i
$l_i(t)$	amount of traffic arrivals and dropped traffic from class i at time t
lc	learning coefficient
m	number of packet departures
m	modulation and coding scheme
M	number of antennas
M	number of channels
M	number of packet arrivals at the transmitter
n	multiplier for base EWMA weight
n	thermal noise
n	transmit beamforming mode number
n	number of receivers
now	current time
N	number of service classes
N	number of bits in a frame
N_0	noise power spectral density
N_FC	available capacity in the system
N_{max}	bin size
N_{po}	number of bits for transmission power feedback in the feedback message
N_{ph}	number of bits for phase feedback in the feedback message
N_R	number of receive antennas
N_{rp}	number of bits in the feedback word in Mode 1
N_s	spreading factor
N_q	average queue length
ϕ_i	service weight for class i
ψ_m	signal phase from antenna m
ϕ_m	adjusted signal phase from antenna m
$p_{i,s}$	fraction of dropped traffic during a time interval $[t_0, s]$
$p\_samples_i$	counter for the number of departed packets in class i

p_tresh	threshold for the number of departed packets
$P_e(k)$	probability that an RU is discarded at the receiver because of transmission errors when k codes are used in the same time slot
P_E	power consumed in electronic circuitry
P_i	scheduling priority for flow i
P_k^-	a priori prediction of the estimate error covariance
P_k	a posteriori state estimate of the estimate error covariance
P_N	noise power
P_{tx}	power consumed by the Power Amplifier for transmission
P_{loss}^*	packet loss bound
$\bar{P}_{loss}$	measured average loss
P^{out}	outage probability
$Pdus_h$	total number of data pdus in the node's buffers at hop h
q	required queue size
q	feedback BER
q_i	physical queue size of class i
q_idle_i	time when class i goes idle
$qlen_i$	queue length of class i
Q	process noise covariance
Q	scheduling policy
R_c	fixed chip rate
ρ	load
ρ	SNR in the channel
$\bar{\rho}$	measured traffic intensity
$\bar{\rho}$	desired mean load in the output link
$\tilde{\rho}$	target traffic intensity
r	service rate of the GPS server
r	time-fraction assignment
r	received signal
$r_i(t)$	allocated service rate to class i at time t

R	lowest possible service rate
R	measurement noise covariance
R_i	stationary service rate of flow i
R_i^{in}	input curve of class i
R_i^{out}	output curve of class i
$\tilde{R}_{i,s}^{out}(t)$	projected output curve of class i at time s
R_s	service rate
R_S	symbol rate
$\bar{R}_k$	short term mean of the maximal traffic envelopes
$\bar{R}_T$	mean of the maximal traffic envelopes
R^{out}	outage rate
s	time
s	transmitted symbol
s_i	mean packet size of class i
S	service time
$S_i(\tau, t)$	the amount of traffic serviced for class i in an interval (τ, t)
$SIFS_h$	SIFS at hop h
σ_k	short term standard deviation of the maximal traffic envelopes
σ_T	standard deviation of the maximal traffic envelopes
t	time
t_i	random arrival time of packet i
t_0	beginning of the current busy period
t_{curr}	current time
t_{in}	packet arrival time in the highest class queue
t_{safe}	safety margin for the delay bound
τ	basic measurement period
T	length of the frame
T	time when the last estimate was calculated
T	measurement epoch
$T(i)$	expectation value for the number of correctly delivered

	RUs when i codes are used
T_{bit}	average time to transmit one bit
$T_{m,h}$	actual time for transmitting all the packets with MCS m at hop h
$T \cdot \tau$	period for collecting measurement data
$\tilde{T}_{i,s}$	time period
u	optional control input
$\widehat{u}_m$	amplitude weight of antenna m
$U_i k$	performance that would be experienced by user i
	if it was scheduled at time k
U	performance vector of all users
U	medium utilization
U_{max}	maximum medium utilization
v	matrix of measurement error
v	random variable following normal probability distribution
v^*	optimal value for time-fraction assignment
v_l	current output value of a neural network
w	random variable following normal probability distribution
$\mathbf{w}$	codebook of complex transmit weights
$\widehat{\mathbf{w}}$	vector of transmit weights
$\widehat{w}_m$	component of the transmit weight vector
$\tilde{w}_i(m)$	normalized head waiting time of class i when m packets have departed
w	system noise
$w_i(k)$	QoS weighting factor for user i's flow k
$w_i(\tau)$	energy function of packet i
$w_{ji}[k]$	weighting function of the connection between the i-th perceptron
	from layer (k-1) and the j-th neuron from layer k
W	workload (bits) created by a single user instance in class i
$\bar{W}$	average waiting time of all users
W_i	waiting time for user i
WL	window length

$x_j[k]$	current output of the j-th perceptron in layer k
ξ	parameter used for threshold optimization
X	random file size
X_k	new observation
x	time
x	state of a discrete-time process
x_k^-	a priori prediction of the state of a discrete-time process
x_k	a posteriori state estimate of the system state
y	final receiver output
y_l	desired output value of a neural network
$y(v)$	function for time-fraction assignment
z	measurement
z_k	measurement

List of Figures

List of Tables

1 Introduction

1.1 Background

For decades, networking research has been driven by increasing consumer demands for higher bitrates and lower delays. Architectures for providing a better service to more important customers have been proposed as one solution to meet stringent QoS requirements. At the moment, the class-based Differentiated Services (DiffServ) [30] architecture seems to be the most promising solution due to its simplicity and scalability. With DiffServ, customers can be classified based on different criteria, such as their willingness to pay or their traffic type. Networking scenarios will be complex and will include fixed networks and hybrid Fourth Generation (4G) networks consisting of both infrastructure-based and infrastructure-less, wireless parts. In order to meet the requirements set for telecommunication in this complex environment, differentiation alone will not be sufficient: the underlying networking technologies must also be drastically improved. In 4G networks, major changes have been designed on layers 1 and 2 of the ISO OSI model to increase data rates and spectral efficiency. Time-frequency operation, channel aware scheduling, adaptive modulation and coding (AMC) and multiantenna techniques can be mentioned as the most important advancements.

While vast amounts of effort have been put into providing services and meeting the requirements of western consumers, certain parts of the world have remained completely disconnected. As the telecommunication markets are beginning to saturate, new growth potential can also be identified in areas that are currently disconnected, where basic connectivity at an affordable price could make a difference. Providing a low price, low bit rate and low energy communications architecture, in which energy can be gracefully traded off against QoS would be an important step towards fully taking advantages of this emerging potential. Using such an architecture, traffic like voice over IP (VoIP) calls or video conferencing could become a reality in developing countries or in emergency situations in rough environments.

In scenarios, where various networking technologies and service requirements may overlap or contradict each other, adaptive provisioning and management of the network resources becomes a key tool for handling its increasing complexity. Adaptive mechanisms are desirable as they enable a self-configurable network that is able to adjust itself to different conditions, such as changes in traffic trends, load fluctuations due to time-of-day effects, channel quality variations or the incorporation of new customers. The operation of adaptive mechanisms relies on measurements: the state of the network is monitored either off-line or on-line to produce an estimate of a desired quantity, such as link utilization, average packet loss, queuing delay or signal to noise ratio (SNR). These measurements can be used on different network control time scales, such as routing and load balancing, admission control, inter-cell interference coordination, packet scheduling and multi-antenna techniques.

Adaptive Quality of Service (QoS) provisioning requires new kinds of functionality from the network elements. Link resources have to be divided between service classes and class based routing or load balancing can be supported as an option. One of the most important components for resource allocation is a packet scheduler that determines the service order of the packets. In current fixed Internet Protocol (IP) network routers, the scheduling algorithms are static and allocate resources based on estimated traffic loads offered to different classes and the required quality level. The estimate is usually based on traffic history, from which an average load is calculated in order to predict the future load. In reality, however, the loads of different classes vary quite significantly on a short timescale due to traffic bursts, as well as on a longer timescale due to traffic trends. If the resource allocation is performed in a static manner, the scheduling algorithm will not be able to adapt to dynamic load conditions. Adaptive scheduling also plays an important role in 4G networks. In LTE-Advanced networks, a receiver can provide regular feedback to the Base Station (BS) by encoding measured channel state information (CSI) in the feedback message, indicating both power and phase information. The BS can utilize this information when applying channel aware scheduling and transmit beamforming. In wireless mesh networks, measurements can be utilized for predicting future delays and selecting certain modulation and coding schemes to preserve the delay bound while minimizing energy consumption.

When designing adaptive scheduling algorithms that may contain computationally demanding estimation procedures the evaluation process should ideally include both simulations and real measurements. If the evaluation is based only on simulations, where traditional performance metrics are used as an evaluation criteria, there is a risk of drawing overly optimistic conclusions. This is because an algorithm that behaves well in simulations may not have the desired results in the real world. In the worst case scenario, it may not even be possible to implement the algorithm in a real-time environment if it uses overly complex computation and estimation procedures. Thus, an algorithm should only be put into commercial use only after its implementation complexity, as well as its resulting performance in a real implementation, has been assessed.

1.2 Research problem

In this thesis, adaptive scheduling algorithms for fixed Internet (IP protocol -based backbone infrastructure) and hybrid 4G networks consisting of wireless infrastructure-based and infrastructure-less networks, are analyzed and proposed. In the first part of the thesis, which deals with handling scheduling in fixed IP networks, the main goals are to propose a delay-based scheduling algorithm Delay Bounded Hybrid Proportional Delay (DBHPD), evaluate it using ns2-simulations and measurements, and to implement it in a FreeBSD based prototype router. The objective is to investigate what kind of performance advantage can be achieved with adaptive scheduling, compared to conventional or pseudo-adaptive scheduling algorithms, and to find out whether the DBHPD algorithm can be implemented with a reasonable level of complexity. Simulations are also used to analyze alternative delay estimators, to examine whether or not the estimator itself should be adaptive. Another major goal is to compare the measurement results with simulation results in order to see how well real implementations of the selected algorithms correspond to the theoretical simulation models of these algorithms.

In the second part of the thesis joint scheduling and transmit beamforming in 4G networks are quantitatively analyzed using statistical methods. The objective is to study upper and lower scheduling gain bounds when On-off scheduling is applied together with antenna selection or Mode1 and Mode2 transmit beamforming. Round Robin (RR) and Maximum SNR (MaxSNR) scheduling are used as a reference. Another objective is to track the impact of feedback errors on On-off strategy and to analyze the upper bound of achievable outage rate. Following on from this, a novel cross-layer energy-adaptive scheduling and queue management framework, EAED (Energy Aware Early Detection), is proposed for minimizing energy consumption and preserving delay bounds in Wireless LAN (WLAN) mesh networks, and evaluated with simulations. The goal is to evaluate the performance of EAED with real-time traffic in realistic WLAN Mesh network simulation scenarios and to discuss possible use- cases for the EAED framework.

1.3 Structure of the thesis

The thesis is structured as follows: Chapter two presents networking technologies relevant to the thesis. Chapter three describes requirements for scheduling algorithms in different environments. In chapter four, the conventional algorithms most relevant to the thesis are presented and their capabilities and implementation complexity are discussed. The fifth and sixth chapter present the adaptive scheduling algorithms, the estimation theory and the delay estimation approach used in this thesis. The next four chapters focus on evaluating the adaptive scheduling algorithms and estimators in various scenarios through the use of both simulations and measurements. Chapters 11 and 12 describe channel aware scheduling algorithms and multi-antenna technologies for Third Generation (3G) and 4G networks while chapter 13 presents results of the performance of joint channel aware scheduling and transmit beamforming. Chapters 14 and 15 concentrate on the development and evaluation of an energy efficient scheduling algorithm for wireless 802.11s mesh networks. Finally, chapter 16 concludes the thesis.

1.4 Author's contribution

The author strongly contributed to the ideas presented in the first part of this thesis, covering novel scheduling algorithms and estimators, simulation and measurement scenarios and parameterization of the different setups. Considering the simulation studies, the author conducted the simulator code implementation and actual simulations and wrote the related publications [20], [21]. Regarding the measurement studies, the publications [120] and [119] were also mostly written by the author. Except for the kernel profiling, the measurements were mostly conducted and analyzed by the author.

In the second part of the thesis, the author strongly contributed to the ideas related to scheduling in 4G hybrid architectures with the emphasis on energy-adaptive scheduling and queue management for WLAN mesh networks. The author was one of the main authors in the publication analyzing challenges for packet scheduling in 4G hybrid networks [78] and primary author in [121] where also the simulator code implementation and simulation studies were conducted by the author. The author is the second inventor of an energy-adaptive packet dropper that is one of the ideas presented in publication [121]. A US patent will be granted soon for this innovation. Regarding the analysis of joint scheduling and transmit beamforming in 4G networks the author participated by commenting on the results, proposing ideas for future work and writing the final manuscript of accepted publication [80].

2 Networking technologies

In a broad sense, all the topics covered in this thesis fall under the umbrella of adaptive scheduling for the Internet. While originally the term Internet referred to the fixed Internet Protocol (IP)-based backbone infrastructure, nowadays the term is used in a very wide sense to describe a "network of networks", where different kinds of access networks, ranging from fixed and wireless home and office networks to sensor networks, are interconnected as depicted in Figure 2.1. These networks do not even necessarily have to implement IP protocol, since the trend is for almost any network to also be an "Internet". The Internet Engineering Task Force (IETF) is very active when it comes to the standardization of Internet technologies. Their most important standardization activities currently revolve around new or modified transport protocols, routing mechanisms, mobility, and security issues. Due to certain fundamentally different characteristics, this thesis solves scheduling problems separately for fixed Internet (IP-based backbone infrastructure) and wireless infrastructure-based and infrastructure-less networks.

2.1 Fixed networks

Fixed Internet originated in military and academic circles, where it was used mainly for information exchange and was dominated by simple e-mail, remote access and file transfer services. Fixed Internet runs IP protocols over various layer-two networking technologies, such as Multi Protocol Label Switching (MPLS), Ethernet and Wave Division Multiplexing (WDM). The rise of the World Wide Web (WWW) boosted the use of the Internet as a commercial, public, multiservice network that supports a variety of applications and customers. Over recent decades, Quality of Service (QoS) architectures and mechanisms to enable applications and customers to be treated differentially in heterogeneous scenarios, has become an important area of Internet development. A service

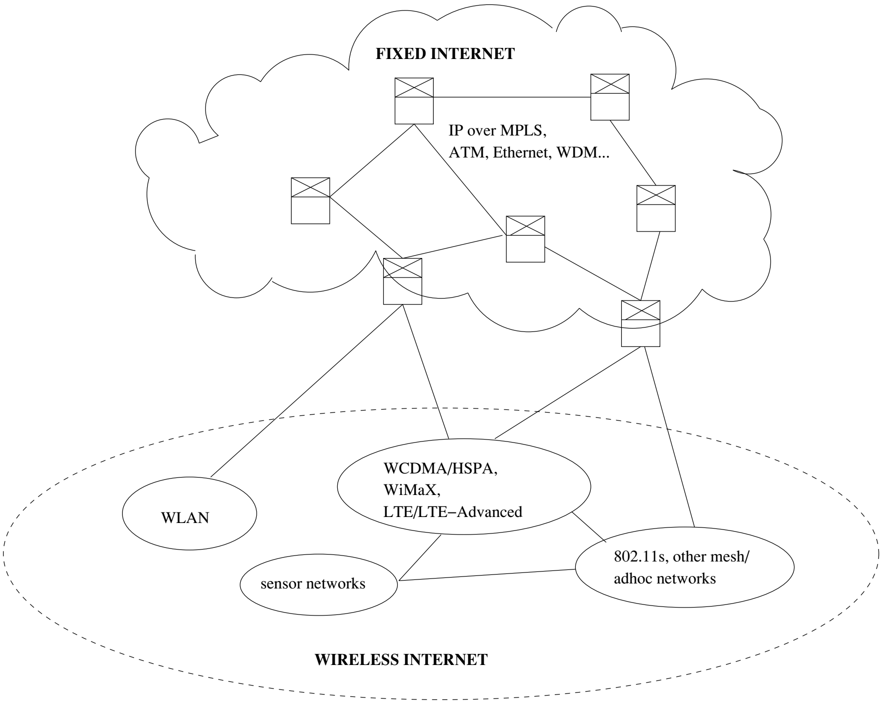


Figure 2.1: The Internet

architecture provides a general framework for service provisioning by abstracting the services and functionality of the network. However, network operators have the freedom to choose what services they offers to customers and what mechanisms are used for implementation. The first service architecture to be proposed was the Integrated Services architecture [32], which aims to provide absolute per-flow guarantees through resource reservation. However, it has not been widely accepted due to its high implementation complexity and poor scalability. Differentiated Services architecture [30] is a more scalable solution for QoS provisioning: traffic is divided into a limited number of forwarding classes and resources are then allocated to these classes. Two Per-Hop Behaviors (PHBs) have currently been standardized: Expedited Forwarding (EF) [42] and Assured Forwarding (AF) [74], but other feasible approaches, such as relative differentiated services [46] have also been proposed.

2.2 Radio networks

2.2.1 3G

The Universal Mobile Telecommunications System (UMTS) radio access scheme was standardized by the European Telecommunications Standard Institute-Special Mobile Groups (ETSI-SMG) in 1998. It operates at a frequency of 2 GHz with 5 MHz frequency bands and supports data rates up to 2 Mb/s. The convergence trend has resulted in the formation of a common IMT (International Mobile Telecommunications)-family for the development of mobile technologies and nowadays International Telecommunication Union - Radio (ITU-R) concepts for the IMT-2000 are included in the UMTS.

Figure 2.2 shows an overview of the UMTS network architecture. The architecture consists of the following main elements: User Equipment (UE), the UMTS Terrestrial Radio Access Network (UTRAN) and the Core Network (CN). The UTRAN includes several

Radio Network Subsystems (RNSs), which in turn consist of a Radio Network Controller (RNC) that controls a number of Node Bs (base stations).

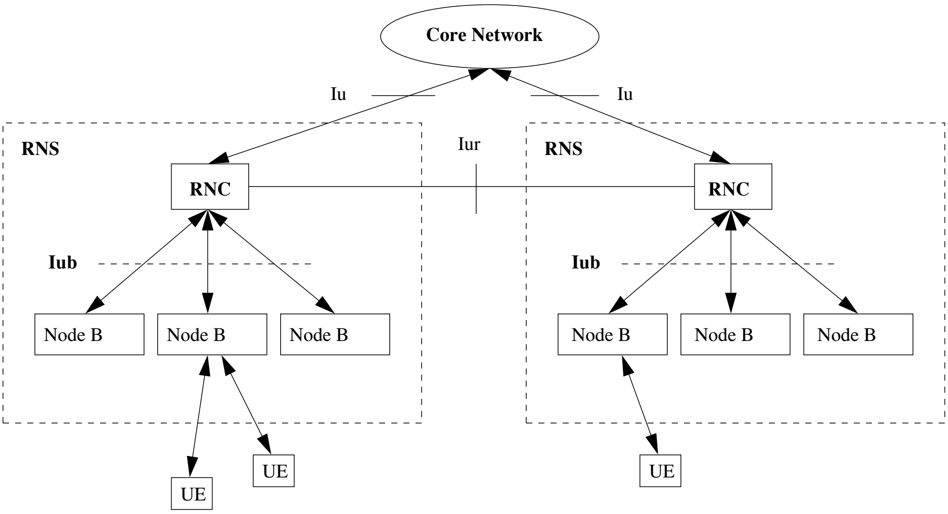


Figure 2.2: Overview of a UMTS network

Two radio interfaces have been defined for the UMTS Terrestrial Radio Access Network (UTRAN): the frequency division duplex (FDD) and the time division duplex (TDD). The FDD mode uses the Wideband Code Division Multiple Access (WCDMA) access method, standardized by Third Generation Partnership Project (3GPP) in [1], while TDD utilizes Time Division-Code Division Multiple Access (TD-CDMA), a time and code division multiple access combination. The FDD mode is suitable for applications with a data rate of below 384 kbps. The TDD mode, on the other hand, can support data rates as high as 2Mb/s, due to the more efficient use of spectrum.

FDD radio interface

Three types of downlink transport channels are supported by UMTS: common, dedicated and shared channels. These channels are divided into frames of 10 ms, and each frame is in turn divided into 15 time slots. The FDD mode uses separate radio frequencies for uplink and downlink. The multiple access method used is the WCDMA direct sequence spread spectrum method, where each user is assigned one or more code sequences that are used to scramble the data before transmission. The Spreading Factor (SF) is the number of code bits that are used to code a single bit of information. By spreading the information before transmission, many users are able to transmit data simultaneously on the same frequency channel, provided that the code sequences are orthogonal and, as a result the cross-correlation of the codes is zero [41]. In order to isolate these codes in such propagation conditions where orthogonality is hard to preserve, Pseudo-Noise (PN) scrambling sequences are used in addition.

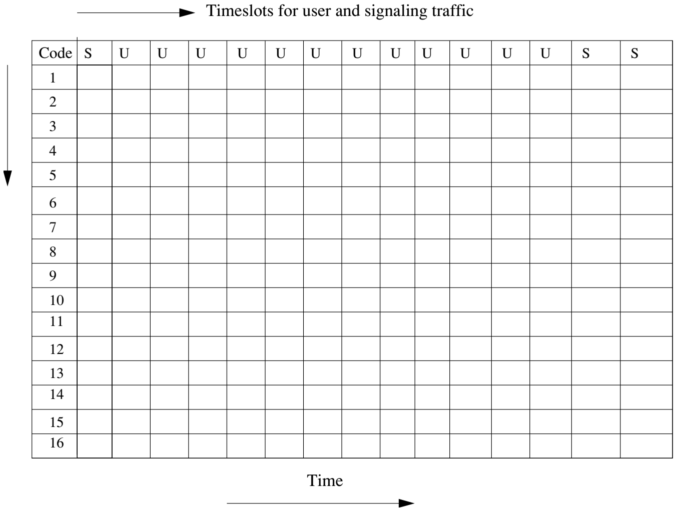
Common channels are only suitable for transmitting small amounts of data, such as signaling traffic. The Downlink Dedicated Channel (DCH) uses a fixed spreading factor for each user, determined by the maximum transmission rate that they are allocated. Thus, DCH is not efficient for variable rate traffic either. The Downlink Shared Channel (DSCH), on the other hand, can share capacity efficiently among many users, since the user can be allocated a different rate during each frame. This can be achieved by assigning dynamically Orthogonal Variable Spreading Factors (OVSF) to users.

In FDD mode the spreading factor for a user can range from 256 to 4 in the uplink and from 512 to 4 in the downlink. It can be observed that the service rate depends directly on the spreading factor, since the symbol rate can be written as R_c/N_s, where R_c is the fixed chip rate and N_s is the spreading factor. If $R = R_c/512$ is the lowest possible service rate, then the possible service rates R_s for the connections can be written as [41]:

$$R_s = K * R = \frac{K * R_c}{512}, K = 1, 2, 4, 8, ... \tag{2.1}$$

TDD radio interface

In TDD mode, both uplink and downlink transmissions can use the same frequency. Some of the time slots are used for transmission and others for reception. In TDD mode the spreading factor can range from 16 to 1. Thus, if the spreading factor is 16, up to 16 simultaneous transmissions can be made in the same time slot [41]. Figure 2.3 shows the time-code matrix of the resulting radio channel. It is assumed that the first and the last three time slots are reserved for signaling traffic, while the rest of the time slots are used by both uplink and downlink user traffic.



Code	S	U	U	U	U	U	U	U	U	U	U	U	U	S	S
1															
2															
3															
4															
5															
6															
7															
8															
9															
10															
11															
12															
13															
14															
15															
16															

Figure 2.3: A time-code matrix for the TD-CDMA mode

HSPA

High Speed Packet Access (HSPA) [75], [113] is an amendment of WCDMA that adds new transport and physical channels and enables resource allocation for bursty data on a shorter time-scale in a power efficient manner, both in the uplink (HSUPA) and downlink

(HSDPA). A HSDPA scheme can achieve peak user data rates of 14 Mbps in the 5MHz channel (3GPP Release 5). If Multiple Input Multiple Output (MIMO) is applied, data rates of up to 20 Mbps can be supported. HSUPA supports data rates of up to 5.76 Mbps (3GPP Release 6). HSPA achieves performance gains by using multi-code transmission on several parallel code channels and applying adaptive modulation and coding, as well as fast scheduling with a granularity of 2 ms (duration of the transmission time interval). HSPA is integrated into the WCDMA standard, referred to as WCDMA/HSPA, but it can also be deployed independently as Internet HSPA (iHSPA) or Evolved HSPA. Evolved HSPA provides data rates of up to 56 Mbit/s on the downlink and 22 Mbit/s on the uplink, using MIMO and higher order modulation (64QAM). Although HSPA can stretch the performance of WCDMA, it does not meet the requirements of most demanding usage scenarios, since underlying spread spectrum communications set certain limits for increasing the bandwidth while maintaining low levels of inter-symbol interference.

2.2.2 3.9G

LTE

Figure 2.4 shows an overview of the 3G Long Term Evolution (LTE) network architecture which consists of evolved UMTS Terrestrial Radio Access Network (E-UTRAN) and Evolved Packet Core (EPC). The following main elements are supported in the architecture: E-UTRAN Node B (eNB) and Mobility Management Entity (MME) including Serving Gateway (S-GW) and Packet Data Network Gateway (P-GW). eNB is responsible for radio interface related functions while MME manages mobility, identity and security related aspects.

The following requirements were set for the first version of LTE, 3GPP Release 8 [7]: support for peak rates of 100 Mbps in downlink and 50 Mbps in uplink, as well as increased cell-edge bit rates, a RAN latency of less than 10 ms, two to four times the spec-

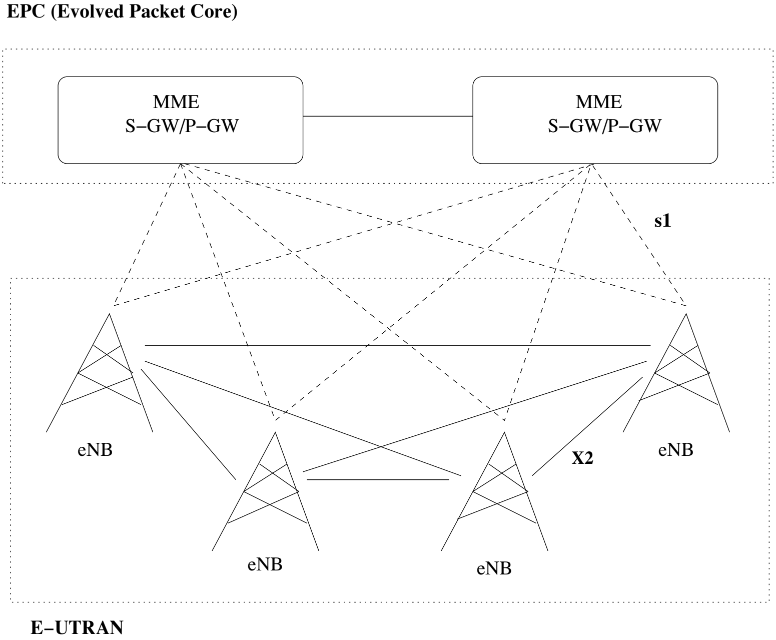


Figure 2.4: The LTE network architecture

trum efficiency of WCDMA/HSPA (3GPP Release 6), support for FDD and TDD modes, support for scalable bandwidths of up to 20 MHz, support for interworking with legacy networks and cost-effective migration from said networks. Time-frequency operation, channel aware scheduling, multi-antenna techniques and inter-cell interference coordination are the most important technologies applied to meet these demanding requirements.

The OFDMA multiple access method is applied to the LTE downlink. Subcarriers have a spacing of 15 kHz, and each subcarrier is adaptively modulated with an optimal Modulation and Coding Scheme (MCS), which is chosen from Quadrature Phase Shift Keying (QPSK), 16-Quadrature Amplitude Modulation (16-QAM) or 64 QAM combined with turbo coding. LTE allocates resources for users in both the time domain (TD) and frequency domain (FD), whereas WCDMA and HSPA only operate in the TD. Subcarriers are grouped into resource blocks (RBs), consisting of 12 adjacent subcarriers [96], see Figure 2.5. In the TD, each RB has a duration (time slot) of 0.5 ms, equivalent to a duration of 6 or 7 OFDM symbols, depending on the form of cyclic prefix used. Two time slots form a sub-frame with duration or transmit time interval (TTI) of 1 ms.

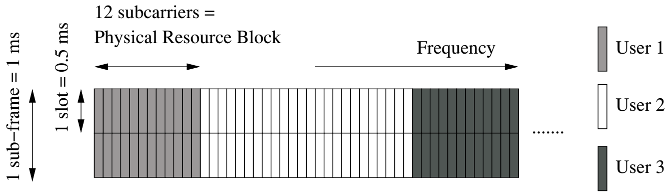


Figure 2.5: Allocation of PRBs in LTE

Single Carrier FDMA (SC-FDMA) is used in the uplink. SC-FDMA is a pre-coded version of OFDMA, in which frequency allocations must be contiguous. SC-FDMA is used in the uplink due to the fact that contiguous frequency allocation results in less Radio Frequency (RF) power variation within a single OFDM symbol, and therefore yields a smaller peak-to-average power ratio (PAPR), meaning that the mobile terminal's power consumption is reduced, leading to a simpler power amplifier (PA) design, as well as improved cell-edge performance and coverage.

2.2.3 Wireless LAN

Wireless LANs can be deployed for short-range communications in various environments. The most well-known WLAN scenarios are ad-hoc networks, infrastructure networks and hotspots. In ad-hoc mode, mobile nodes form a peer-to-peer like network where all the nodes are directly connected to each other. The ad-hoc mode is useful in situations where fixed network services are not needed, for example when employees are working from remote locations. In the infrastructure mode, mobile nodes are connected to an access point (AP) that is responsible for arbitrating channel access and performing tasks related to network management. The infrastructure mode is mostly used in company and home WLANs. Hotspots provide a WLAN service for a fee or for free and are used in places like airports, coffee shops, public meetings and hotels.

Standards and physical layer properties

The basic IEEE 802.11 [82] WLAN standard was developed in 1997. It was later enhanced to provide higher data rates with more advanced physical layer mechanisms. 802.11b and 802.11a standards were confirmed in 1999 and 802.11g was ratified in 2003. 802.11n is a further extension based on 802.11a/g, which adds the use of multiple antennas and double width channels (40MHz) to the physical layer and radically increases achievable data rates. The standard was ratified in 2009, but not all the details have been finalized yet. Table 2.1 summarizes the main properties of 802.11b, 802.11g, 802.11a and 802.11n standards in terms of multiplexing schemes, frequency, data rates and typical indoor and outdoor ranges. The indoor and outdoor ranges (m) correspond to the ranges between the maximum and minimum data rates. In the case of Orthogonal Frequency Division Multiplexing (OFDM), the system uses 52 subcarriers to divide the spectrum.

Each subcarrier is modulated with Binary Phase Shift Keying (BPSK), Quadrature Phase Shift Keying (QPSK) or 16- or 64- Quadrature Amplitude (QAM) modulation, depending on the data rate [31]. OFDM is known to persistently suffer from undesirable multipath fading, since it uses many orthogonal narrowband subcarriers to transmit the information. On the other hand, the Direct Sequence Spread Spectrum (DSSS) uses a relatively wide band for information transmission, leading to different multipath behavior at different frequencies. In the cases of DSSS, Differential Binary Phase Shift Keying (DBPSK), Differential Quadrature Phase Shift Keying (DQPSK) or Complementary Code Keying (CCK), modulation is used to produce different date rates [31].

Table 2.1: Wlan 802.11 standards

	802.11b	**802.11g**	**802.11a**	**802.11n**
Multiplexing	DSSS	OFDM	OFDM	OFDM
Frequency	2.4 GHz	2.4 GHz	5 GHz	2.4 and 5 GHz
Data Rates	11, 5.5, 2 and 1 Mbps	54, 48, 36 24, 18, 12, 9 and 6 Mbps	54, 48, 36 24, 18, 12, 8 and 6 Mbps	upto 72.2, 150 Mbps
Indoor Range	30-91 m	30-91 m	12-30 m	70-230 m
Outdoor Range	120-460 m	120-460 m	30-305 m	250-820 m

Medium Access mechanisms

Wireless LANs utilize similar access mechanisms to 802.3 Ethernet networks. The basic WLAN access strategy is called carrier sense multiple access with collision avoidance (CSMA/CA). Two basic access mechanisms have been designed for WLANs: a Distributed Coordination Function (DCF) and a Point Coordination Function (PCF). A DCF is a distributed access mechanism that utilizes CSMA/CA, which can be deployed in both ad-hoc and infrastructure scenarios. A PCF is a centralized mechanism designed for the infrastructure mode. It utilizes a contention-free polling based medium access method to arbitrate channel access. Neither DCFs nor PCFs are capable of providing Quality of Service by traffic prioritization. In order to enable QoS support, the DCF mechanism has been extended by adding a prioritization function. The resulting access mechanism is called an Enhanced Distributed Coordination Function (EDCF).

DCF The basic idea of a DCF medium access mechanism is as follows [137]: The WLAN station first senses the medium in order to determine if its free. If the medium is found to be idle, the station may transmit, after a certain time period, the DCI interframe space (DIFS), has elapsed. The DIFS is used to ensure that the station has enough time to detect a message that may have just been sent by another station. If the medium is found to be busy, the station first has to wait a short time, corresponding to the DIFS, followed by a random backoff interval (BI) before it can start transmitting. BI is defined as follows [127]:

$$BI = Rand(CW_{min}, CW_{max}) \times SlotTime, \tag{2.2}$$

where CW_{min} and CW_{max} are the minimum and maximum backoff periods. The difference between CW_{min} and CW_{max} is called the Contention Window (CW), which indicates the range for possible backoff intervals. The random backoff timer is used to minimize the probability of two or more stations transmitting at the same time. However, if collisions still occur, the backoff timer is doubled and the collision avoidance procedure is repeated. This kind of backoff strategy is called Binary Exponential Backoff (BEB). The receiving side is responsible for sending an ACK frame for a received frame. The ACK frame can be sent only after a short interval space (SIFS) corresponding the time of the receiver to process the frame and pass it to the MAC sublayer. SIFS plus the time for the ACK to propagate back to the sender is smaller than DIFS in order to avoid collisions with ACK frames among the sending stations. The sending station has to wait for the ACK to arrive before it can start sensing the medium for a new transmission [137]. Figure 2.6 depicts the two-way handshake DCF procedure described above.

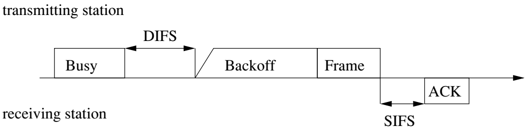


Figure 2.6: Two-way DCF handshake procedure

Besides the physical carrier sensing approach (two-way handshake), medium sensing can also be performed using virtual carrier sensing, (four-way handshake), which exchanges short Ready To Send (RTS) and Clear To Send (CTS) reservation frames before a transmission attempt. Virtual carrier sensing is especially useful when there are hidden terminals in the network that may not be able to hear other terminals. Figure 2.7 depicts the four-way handshake DCF procedure when one frame is transmitted. Network Allocation Vector (NAV) can be used to reserve the medium for several consecutive frame transmissions. It is updated upon RTS/CTS frame receptions and indicates the remaining time before medium becomes free.

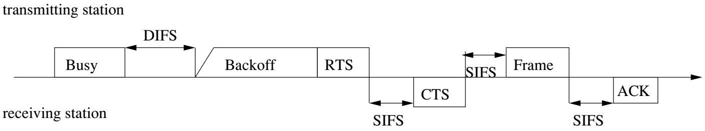


Figure 2.7: Four-way DCF handshake procedure

Enhanced Distributed Coordination Function (EDCF) DCFs and PCFs cannot provide any kind of QoS differentiation. Thus, the DCF has been improved in order to enable different access category priorities (ACs). The improved DCF is referred to as an Enhanced Distributed Coordination Function (EDCF) ([72], [118]). With the EDCF, each packet is assigned a priority tag before entering the MAC layer. In the MAC layer, this priority value is mapped in one of four access category First In First Out (FIFO) queues. In this way, real-time traffic such as VoIP and Video can be directed to higher priority queues. In practice, the EDCF prioritizes traffic by modifying the following parameters: $CW_{min}[AC]$, $CW_{max}[AC]$ and $AIFS[AC]$, where $AIFS[AC]$ is an arbitrary inter-frame space used to replace DIFS in DFS. These parameters are announced periodically by the QoS-enabled access point in so-called beacon frames. It is intuitively clear that lower values for $CW_{min}[AC]$, $CW_{max}[AC]$ and $AIFS[AC]$ result in shorter channel access delays. The AIFS parameter of the EDCF is calculated as follows:

$$AIFS[AC] = SIFS + AIFSN[AC] \times SlotTime, \tag{2.3}$$

where $AIFSN[AC]$ is the so-called arbitration IFS number that is used to differentiate the AIFS values between ACs.

2.2.4 4G

The 3G RAN cannot be scaled up to meet the requirements set for telecommunications over the next decade. As a result, a novel Fourth Generation (4G) RAN design was needed, with major changes applied to layers 1 and 2 of the ISO OSI model. The 4G concept can be introduced in various ways. A high-level perspective can be chosen based on either the generation of communication technology, or on the technology itself. The Fourth Generation can be seen as an evolution of 3G or part of the "Beyond 3G" (B3G) development [25, 154], as a parallel to other legacy and potential future access technologies [154], as a comprehensive access solution that enables the use of current and future technologies and solutions [134, 150], or as a uniform congregation of high-speed seamless access technologies and intelligent agents [27]. The final 4G solution is not complete, but significant advances have been made with 3GPP to design features for LTE-Advanced, which is currently the most promising concrete solution for future 4G.

Various requirements have been identified for 4G networks [116,124,134,150], presenting a demanding target for the 4G concept. With the high frequencies (e.g. 5 GHz) and bit rates required by the 4G RAN, cell sizes are in the order of a few hundred of meters.

It follows on from this that the large number of access points required to provide full coverage may not be economically feasible, considering future manufacturing and site installation costs. Even though most of the requirements involve the radio part, the higher layers are also affected. For example, handoff latencies, routing information convergence time and throughput are affected by the number of hops in the network. 4G networks are supposed to support services such as real-time voice and video conferencing, video streaming, interactive games, and best effort web browsing. QoS requirements for these services in 4G networks are much more challenging compared to 3G networks, and high requirements are set for scheduling.

The type of multiple access method chosen will have a major effect on system performance. First of all, a decision needs to be made of whether to use one or several carriers for both the uplink and downlink directions. There are many factors to consider here but, in general, the multicarrier option offers better spectral efficiency in the wide band case. On the other hand, the single carrier option has the advantage of more efficient power usage at the transmitting end. Therefore, the proposed solution for this area is to use a single carrier in the uplink direction and multiple carriers in the downlink direction. With the multicarrier option, the channel access method could be based on Orthogonal Frequency Division Multiple Access (OFDMA). OFDMA divides the bandwidth into orthogonal tones that can carry multiple data symbols in parallel. OFDMA is preferred over CDMA since it can better preserve the orthogonality at the receiver, due to the fact that OFDMA uses sinusoids, which are eigenfunctions of a time-invariant linear system [99]. OFDMA also benefits from multiuser diversity, since multipath effects are highly frequency selective. Therefore, not all subcarriers are likely to be affected in the same manner. Furthermore, while TDMA and CDMA are designed for a fixed system bandwidth, OFDM supports bandwidth scaling whereby the subcarrier bandwidth can be scaled to any power of two, assuming that subcarrier bandwidth and symbol length have been defined. However, there are strong arguments for also having TDMA components in the system design, such as energy consumption, packet data and resource allocation. If the wide band is reserved for a single user and no data is transmitted, bandwidth and energy are wasted. In terms of terminal relaying, system complexity and manufacturing costs would be reduced if transmitters and receivers had common solutions, such as multiple access schemes, when applicable. However, solutions for the issues presented here have not yet been standardized for 4G.

In addition to the advanced multiple access method and coding and modulation schemes, the obvious ways to improve reliability and data rates are to increase transmission power and bandwidth, which may not always be possible. However, these improvements are required in 4G networks. They could be achieved by using cross layering and the spectral, temporal and spacial dimensions of diversity in communication [117]. We will analyze these techniques in more detail in Chapter 12.

4G Hybrid relaying scenarios

Hybrid architectures [107] are networks that include both single hop and multi-hop networking. In our case the hybrid architecture comprises both infrastructure and ad hoc parts.

Even though research is still being carried out into hybrid architectures, they are believed to have potential for next-generation wireless networks [58]. Relaying, on the other hand, has been included in a Third-Generation Partnership Project (3GPP) draft standard. Figure 2.8 illustrates the hybrid extensions to 4G scenarios presented in [124]. The following networking approaches that utilize relays are described:

1. Access point (AP) coverage is extended with a Relay Station (RS) structure. Connections between the AP and Relay Stations (RS) are point-to-point. Direct connections between MTs are also possible (A-B)

2. The multi-hop relay network can benefit from cooperative strategies (C-D-E), or

assist with ad-hoc network traffic optimization (F-G-H).

3. An isolated ad-hoc network is connected (I-J-K) to the 4G RAN via Mobile Gateways (MGW).

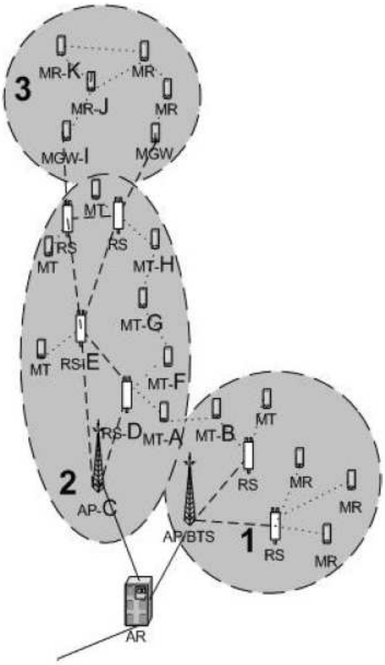


Figure 2.8: Hybrid 4G Scenarios

The functions of the Access Router (AR) are to aggregate traffic towards the backbone network, , and to route traffic between sub-networks. The Access Point (AP) is analogous to a base station providing radio coverage, and possesses advanced capabilities such as support for various radio technologies. The Relay Station (RS) is a simplified AP in terms of cost, with similar tasks but equipped only with wireless connections. The Relay Station is analogous to a radio coverage extender in 2G or WLAN networks. However, the relay function is designed to route traffic on layers 2 or 3, in addition to simply receiving and amplifying the RF signal. On the physical layer, this requirement means that more advanced forwarding methods, such as decode-and-forward or estimate-and-forward, have to be used. The RS can be designed to be either fixed, movable or mobile. The Mobile Terminal (MT) is a 4G compatible device with a high data rate and multiple antennas. It is capable of operating in both licensed and unlicensed frequency bands. The Mobile Router (MR) is a mobile terminal with routing capability.

Limitations to cooperative relaying strategies do exist. For example, amplify-and-forward is considered as a simple cooperative protocol [97]. From the system perspective, the problem is that received interference is also amplified and forwarded by the relaying node. As can be seen in Figure 2.8, AP C could transmit to RS E directly, or use RS D as an intermediary. Using D as an intermediary, and increasing transmission power at D to tackle deteriorating channel conditions, may not improve the Signal-to-Noise-Ratio (SNR) at the destination node E. This results in the SNR of the signal received by relay D from the source node C falling under the unrecoverable limit. Instead of increasing the transmission power, Automatic Repeat Request (ARQ) mechanisms can be used to overcome channel variation defects. Both cooperative methods and ARQ both have an effect on scheduling in the system. The majority of differences between existing 3G and forthcoming 4G networks lie in changes to layers 1 and 2, in the form of higher bit rates, new frequencies, modulation and frame structure, to mention just a few [134].

LTE-Advanced

In the previous section we presented several relatively general requirements and possible technical solutions for 4G systems. At the moment, there are two proposals for the concrete future 4G solution: LTE-Advanced [6] and IEEE 802.16m. LTE-Advanced is the candidate technology created by the 3rd Generation Partnership Project (3GPP) to fulfill the IMT-Advanced (International Mobile Telecommunications Advanced) requirements specified by the ITU (International Telecommunication Union). 3GPP and ITU-R have worked on LTE Advanced since 2009 as part of 3GPP Release 9, and standardization was begun on Release 10 in 2010. LTE-Advanced can be seen as evolution of LTE technology that already meets most of the 4G requirements. LTE-Advanced adds additional features, such as support for peak data rates, but most of the underlying technology is the same. IEEE 802.16m has received less attention lately and is not likely to become the widely adopted final 4G solution.

The main additional technical requirements for LTE-Advanced are support for peak data rates of 1 Gbps in downlink and 500 Mbps in uplink, bandwidth scalability up to 100 MHz, increased spectral efficiency up to 15 bps/Hz in UL and 30 bps/Hz in DL, along with improved cell edge capacity and decreased user and control plane latencies [6]. Like LTE, LTE-Advanced also applies OFDMA in the downlink and SC FDMA in the uplink. The additional requirements are solved mainly with the following technologies:

- *Scalable bandwidth/carrier aggregation* where bandwidth of an LTE-Advanced terminal is increased with OFDM, either in a contiguous or non-contiguous manner
- *Multicarrier operation* where resource allocation, MIMO, link adaptation, hybrid ARQ (HARQ) etc. are performed per carrier, and where feedback per carrier is also required.
- *Asymmetric uplink/downlink spectrum allocation* where different amounts of spectrum can be allocated to the uplink and downlink
- *Coordinated multi-point transmission (CoMP)* where system capacity is increased with frequency reuse and the coordination of scheduling and multisite beamforming.

In addition, one key technology that will be applied in LTE-Advanced is cognitive radio, which is able to utilize the spectrum resources in a dynamic and flexible manner, both within a single wireless technology and across heterogeneous technologies. Another major advancement will be the use of mesh networks to increasing capacity, coverage and spectral efficiency.

Mesh networks

Wireless mesh/adhoc networks are autonomous, infrastructureless systems whose topology changes dynamically due to movements of the MTs. Due to their decentralized nature, easy and fast deployment, inexpensiveness and reliability, mesh networks are considered especially attractive in scenarios where there is no network infrastructure available, as they can also be used for offloading the infrastructure based network (see Section LTE-Advanced). Some of the proposed mesh networking solutions are based on a TDMA-like allocation of resources, while others utilize more advanced techniques such as OFDMA. Currently, OFDMA-based IEEE802.11s is considered as one of the most promising practical technical solutions for mesh networking. IEEE802.11s [81] is a draft IEEE802.11 amendment for mesh networking that defines a way for wireless devices to interconnect in an ad-hoc manner. Standardization work started in 2005 and has proceeded to a second letter ballot. The standard defines the technical properties of mesh networking, while the Wi-Fi Alliance's Mesh Marketing group defines possible business cases and the WPS Extensions (WPS2) group may define additional security features and perform operation tests.

A 802.11s network consists of Mesh Points (MPs) as depicted in Figure 2.9. MPs are mobile terminals that may voluntarily perform multi-hop mesh networking functions such as traffic forwarding and routing. In addition, MPs may have access point (AP) capability (MAP), or they may act as mesh portals that connect to other external networks. MPs find other MPs and networks through active or passive scanning. In active scanning, probe request frames are used to explicitly request information, while passive scanning is based on listening to beacons. Every MP transmits its own beacon frame during each beacon transmission time (TBTT), which is typically 100ms. After the other MPs are found, MPs can authenticate and associate with each other to create peer links and start data transmission.

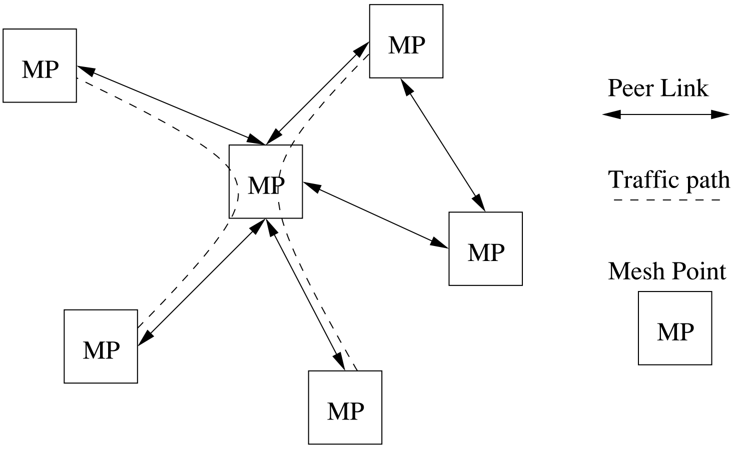


Figure 2.9: 802.11s mesh network

3 Requirements for scheduling

Scheduling in fixed IP networks is about determining the service order of packets in the output link of a router. The packets may be served from a single queue, according to the First Come First Served (FCFS) principle, or there may be several queues that are subjected to some form of service differentiation, as depicted in Figure 3.1. In wireless networks, the task of the scheduling algorithm in the base station is to decide which user can transmit on the channel at a given time. As an example, Figure 3.2 depicts a simple model for a wireless channel in which there is a single cell with one base-station and several users. The users transmit data to the base-station via a wireless channel, one user at a time (a TDMA-like network).

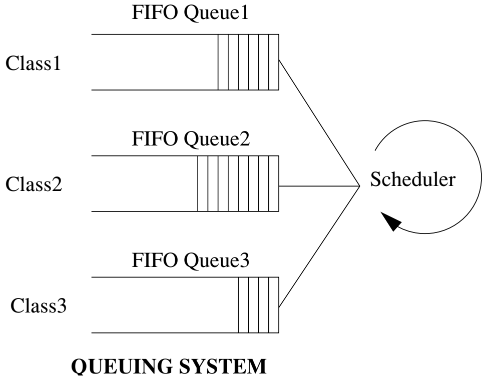


Figure 3.1: Scheduling in fixed IP networks

3.1 Desirable properties of conventional scheduling disciplines

Conventional algorithms are parameterized in a static fashion based on a priori estimates of offered loads. Thus they are mostly suitable for scenarios where loads are either expected to be relatively constant or where another QoS mechanism such as routing is used

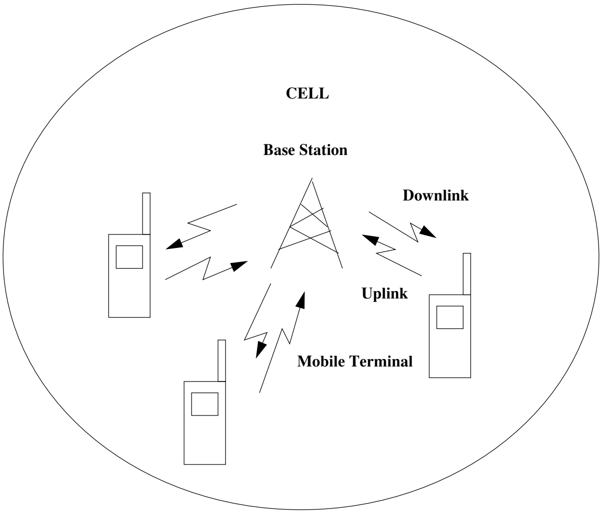


Figure 3.2: Simple model for a wireless channel

to take care of the effects of changing traffic dynamics. The desirable properties of conventional scheduling disciplines are closely related to whether deterministic, statistical or relative type guarantees are made to the end user. However, there are also some general desirable properties common to all scheduling disciplines:

- *Flexibility*: The scheduling discipline must not optimize performance from a single application's point of view but should rather be able to accommodate applications with varying traffic characteristics and performance requirements.

- *Simplicity*: The scheduling discipline should be both conceptually and mechanically simple [143]. Conceptual simplicity enables a tractable mathematical analysis of worst case bounds and performance parameters. Mechanical simplicity, on the other hand, allows the efficient implementation of the scheduling discipline at high speed.

If the performance guarantees are deterministic or statistical, the following desirable properties are identified further in [157] and [143]:

- *Protection*: Real network environment is not static. Thus, the scheduling discipline should be able to protect and satisfy the performance requirements of well behaved users, also in the presence of different sources of variability, such as best effort traffic, badly behaved users and network load fluctuations [157].

- *Efficiency*: When performance guarantees are deterministic or statistical, a connection admission control policy is used to decide, whether to admit a new flow into the network. The more flows that can be accepted into the network without violating the end-to-end performance guarantee of any other flow, the larger the schedulable region of the discipline [143] and the higher the network utilization.

Scheduling discipline criteria aimed at relative differentiation are not that strict, since service guarantees are considerably less stringent compared with deterministic or statistical guarantees. Besides flexibility and simplicity, the most important criteria are [47]:

- *Predictability*: Class i should consistently receive a better (or at least no worse) service than class $i - 1$ with regard to a selected performance metric. This also applies to short timescales and the presence of class load variations.

- *Controllability*: If quality spacing between traffic classes is quantified, the scheduling discipline should be able to maintain and adjust the specified spacing based, for example, on pricing even on short timescales [104].

3.2 Requirements for adaptive scheduling algorithms

The adaptive QoS provisioning approach requires functionality in network routers to be rethought. As stated in the previous section, scheduling algorithms in current routers are static and rely heavily on parametrization. The parametrization is based on the required quality level and an off-line estimate of the traffic loads offered to different classes. The main drawback in this approach is that once the parameters have been set, they can not be modified even if the traffic profiles change dramatically. In reality, the loads of different classes can vary quite a lot on a short timescale due to traffic bursts, and also on a longer timescale due to traffic trends. If resource allocation is performed in a static manner, the scheduling algorithm will not be able to adapt to dynamic load conditions. In the worst case scenario, this could lead to a situation where a higher quality class receives worse service than a lower quality class. In the future network environment, therefore, an ideal scheduling algorithm should be adaptive and capable of dynamically adjusting the class resources, so that the policy chosen by the operator is implemented, regardless of traffic conditions.

Adaptivity poses new requirements for scheduling algorithms. The most important requirements for the development of adaptive scheduling algorithms are:

- *Robustness*: The adaptive algorithm should operate properly regardless of the traffic and network conditions. Thus, the algorithm should not rely on pre-defined parameters, but rather utilize real-time measurement information on different timescales in order to be truly adaptive.

- *Low implementation complexity*: Low complexity is a crucial objective in measurement-based algorithms since real-time measurements and estimations cause additional overheads. In the worst case scenario, calculating an estimate might take such a long time that current information is not available when it is needed. Preferably, mechanisms, at least on a packet level, should be implemented in hardware. It is obvious that efficient and cheap real implementation is not possible if the algorithms use fancy, overly complex estimators.

- *Stability*: Measurement-based algorithms are often closed-loop systems, because the amount of allocated resources affects the amount of offered traffic, and vice versa, in networks with feedback-based traffic. An algorithm that provides perfect results for open-loop traffic may result in chaotic instability for something like TCP traffic. Thus, possible interactions with higher layers should be considered when determining the operation timescales of the algorithm.

3.3 Requirements for wireless scheduling algorithms

The previous section set out the requirements for conventional and adaptive scheduling algorithms in fixed IP networks. These requirements also have to be met when scheduling is performed in wireless networks. However, due to the special characteristics of the wireless medium and wireless devices, additional requirements arise that must be taken into account. The main reason why the scheduling problem becomes much more challenging, compared to scheduling in fixed IP networks, is the time varying channel conditions that are caused by the attenuation of the signal between transmitter and receiver. Attenuation is the result of pathloss, which is a decrease in transmission power proportional to the square of the distance traveled, and other fading and interference effects, such as ([53], [108], [41]):

- *Path-loss variation, slow log-normal shadowing and fast multipath-fading* Shadowing is caused by obstacles such as water and buildings that can block the propagation of radio waves. Path loss and shadowing are examples of slow fading effects, where a significant movement of the receiver is required to produce a variation in the signal strength. Multipath, on the other hand, is a fast fading effect where signal strength variation can even be observed in distances separated by half a wavelength [63]. Multipath fading is the result of the signal scattering from surfaces such as glass and metal, so that multiple copies of the signal combine constructively or destructively in the receiver, having taken different paths to get there.

- *Inter-user and inter-cell interference* Inter-cell interference (ICI) results from two or more users located in different cells trying to use the same channel at the same time. Inter-user interference refers to the case where two or more users located in the same cell interfere with each other. For example, these kinds of situation could occur in WCDMA networks if code words are not orthogonal.

- *Background noise* Noise is present in both fixed and wireless networks and further limits the communication range. It comes from several natural sources, such as the sun, the thermal vibrations of atoms in conductors and black body radiation.

As a result, two users with the same amount of resources do not necessarily experience the same performance, since the one user may have a good channel while the other user may experience several channel errors.

The objectives for an ideal wireless scheduling algorithm are very much the same as the objectives for an ideal wireline scheduling algorithm. However, the following important features characteristic of wireless networks have to be additionally considered ([53], [108], [78]):

- *Link utilization:* Since radio resources are limited, channel capacity must be efficiently utilized. The transmission slot should only be assigned to flows with error-free channels. Alternatively, adaptations have to be made to prevailing channel conditions using advanced methods such as adaptive transmission, link adaptation and MIMO.

- *Graceful service degradation:* If a flow that has received too much service due to having a good channel has to subsequently compensate for the flows that are transferred from a bad channel to a good channel, the service degradation should be smooth.

- *Complex network scenarios and interoperability issues:* As presented in Section 2.2.4, hybrid architectures pose additional scheduling challenges. As an example, potential performance gains and limitations of cooperative strategies must be considered.

- *Energy consumption:* Energy consumption should be kept to a minimum in order to save the batteries of the Mobile Station. This is particularly important when low-power devices are used to forward traffic in a mesh network. Energy consumption is also becoming an increasingly important factor, allowing BSs to minimize carbon dioxide emissions and OPEX.

- *Receiver complexity:* When utilizing advanced methods such as MIMO for enhancing scheduling performance, it must be considered that receiver complexity also increases as the amount of physical layer information grows.

4 Conventional scheduling algorithms

Various packet scheduling algorithms have been proposed for quality differentiation in fixed IP networks over recent decades. Examples of these algorithms are Priority Queuing, Earliest Due Date (EDD), Generalized Processor Sharing (GPS), Weighted Fair Queuing (WFQ) [125], Worst Case Weighted Fair Queuing (WF^{2}Q) [26], Self Clocked Fair Queuing (SCFQ) [62], Deficit Round Robin (DRR) [141], Cisco Custom Queuing (CCQ), Hierarchical Token Bucket (HTB) and Class Based Queuing (CBQ) [56]. The common denominator shared by these algorithms is the fact that they rely heavily on static parameterization and thus are not able to adapt to changing traffic dynamics. The hierarchical HTB and CBQ algorithms contain some adaptive logic but they are not truly adaptive in the sense that their redistribution capacity rules are completely heuristic.

In this chapter some of the most important conventional scheduling algorithms relevant to this thesis are reviewed and compared with each other in terms of their capabilities and implementation complexity. The algorithms are classified into three groups: bandwidth sharing algorithms, hierarchical bandwidth sharing algorithms and deadline-based algorithms. Bandwidth sharing and deadline-based algorithms are purely static, whereas hierarchical bandwidth sharing algorithms may use some heuristic rules to redistribute excess bandwidth in an intelligent way.

4.1 Bandwidth sharing algorithms

The main aim of bandwidth sharing algorithms is to provide exact max-min weighted fair sharing for different flows or traffic classes [152]. Generalized Processor Sharing (GPS) is a theoretical, fluid-based reference model of a fair-queuing algorithm that meets this goal. GPS assumes that traffic is infinitely divisible and that different flows can be served simultaneously. However, as an ideal algorithm it cannot be implemented in practice, since packets are always serviced as entities, one at a time, in a real IP-router. Thus, different practical packet-per-packet approximations of GPS have been proposed, which try to emulate the GPS system as accurately and simply as possible while still treating packets as entities. This section presents the principles of GPS as well as two low complexity packet-based GPS approximations, the Deficit Round Robin (DRR) and Cisco Custom Queuing (CCQ) algorithms.

4.1.1 GPS

The ideal Generalized Processor Sharing (GPS) algorithm operates as follows: each class i is assigned a weight ϕ_i, reflecting the amount of resources that should be allocated to the class. If the service rate of the GPS server is r and there are N classes being served, then for any two backlogged classes, i and j,

$$\frac{S_i(\tau,t)}{S_j(\tau,t)} = \frac{\phi_i}{\phi_j}, \tag{4.1}$$

where $S_i(\tau, t)$ denotes the amount of traffic serviced for class i in an interval (τ, t) [125]. Thus, in any interval (τ, t) class i receives service with a rate

$$g_i \geq \frac{\phi_i}{\sum_{j=1}^{N} \phi_j} r. \tag{4.2}$$

This corresponds to the situation where all classes are backlogged during the interval. However, if some classes are not backlogged, the excess bandwidth is distributed among the backlogged classes in proportion to their weights. Then, the instantaneous service rate can be expressed as

$$g_i(t) = \frac{\phi_i}{\sum_{j \in B(t)} \phi_j} r, \tag{4.3}$$

where $B(t)$ denotes the set of classes that are currently backlogged.

GPS is considered to be an attractive scheduling discipline because it has many desirable properties. Firstly, it treats the classes fairly by servicing each of them at a rate equal to, or greater than their guaranteed rate. Secondly, if the incoming traffic is leaky bucket-constrained, it can be shown [125] that strict bounds exist for the worst-case network queuing delay. Thirdly, the classes can be treated in different ways by varying their weights. For instance, if there are two classes with weights $\phi_1 = 1$ and $\phi_2 = 0$, GPS is reduced to strict priority scheduling. On the other hand, if all classes are assigned equal weights, GPS behaves as a uniform processor sharing system.

4.1.2 DRR

The Deficit Round Robin (DRR) is a static, frame-based scheduling algorithm that aims to emulate the ideal Generalized Processor Sharing (GPS) [125] algorithm. Other scheduling algorithms for emulating GPS have also been proposed, such as the Worst Case Weighted Fair Queuing (WF^{2}Q) [26] and Self Clocked Fair Queuing (SCFQ) [62] algorithms. It was decided to present DRR in this thesis, mainly due to the fact that it is considerably simpler to implement and that it has the ability to take variable packet sizes into account. With the DRR, each class i is assigned a weight ϕ_i. In each service round, the scheduler divides a frame of N bits among the classes in proportion to their weights. The resulting number of bits reserved for a certain class is called a quantum. The DRR also associates each class with a deficit counter that keeps track of the quantum not used by the class in previous rounds. Thus, packets can be transmitted from a certain class as long as there are enough bits left either in the quantum or in the deficit counter.

4.1.3 CCQ

Cisco Custom Queuing (CCQ) is presented here as an example of a scheduling algorithm that is currently implemented in commercial Cisco network routers. Conceptually CCQ is very similar to DRR: it allocates a certain number of bytes to each queue and then serves these queues in a round robin fashion. However, CCQ does not use a deficit counter, meaning that classes cannot utilize resources that should have been allocated to them in a previous round. In Cisco routers, 17 output queues are used for each network interface. Queue number 0 is a system queue that has a strict priority over the other queues handled by CCQ or any other scheduling algorithm. It is assumed that the system queue stores urgent and important packets, such as keepalive packets and packets used for signaling purposes.

4.2 Hierarchical bandwidth sharing algorithms

In hierarchical bandwidth sharing algorithms the traffic classes are organized in a tree structure, as depicted in Figure 4.1. In general, there are three types of classes in the hierarchy: a root class, intermediate classes and leaf classes. The idea is that the bandwidth resources can first be divided between the intermediate classes and then between the leaf classes.

4.2.1 CBQ

Class Based Queuing (CBQ) [56] is the most well known hierarchical bandwidth sharing algorithm. It has been implemented both in prototype and commercial routers. CBQ can be considered as a pseudo-adaptive scheduling algorithm in the sense that it provides heuristic rules for borrowing capacity, if a class is running out of resources. With CBQ the roles of the different hierarchy levels are as follows:

- **Root class** contains the link resource that is to be divided among the traffic classes.
- **Leaf classes** represent the actual traffic classes that are served by the link.
- **Intermediate classes** are responsible for sharing resources among the leaf classes.

The intermediate classes act as parents for the leaf classes, allowing the leaf classes to borrow resources.

In practice, the root class represents the total link capacity while intermediate classes could, for example, be agencies of a company and leaf classes could be protocols used within the agencies. The hierarchy can also be used for efficient application differentiation by using separate intermediate classes for real-time and non-real-time traffic and separating leaf classes for individual applications.

CBQ resource allocation is enforced with two different schedulers: one general and one for link sharing.

- **The general scheduler** is all that is required when there are enough resources available for all leaf classes.
- **The link sharing scheduler** is required when some of the leaf classes get congested. In this case, the link sharing scheduler enforces rules to allow the congested classes to borrow resources from the parent classes.

The actual way in which the general scheduler and the link sharing scheduler are implemented has not been defined. The idea is that any rate-based scheduler, such as WFQ, WRR or DRR can be used as the general scheduler, and the borrowing rules used by the link sharing scheduler can also vary between implementations.

Formal link sharing guidelines for CBQ

Before the formal link sharing guidelines for CBQ are defined, a brief explanation will be given of the following concepts: *overlimit*, *underlimit*, *at-limit*, *satisfied* and *unsatisfied*.

A class is defined as overlimit if, during a certain interval, it receives more resources than allowed by its link-sharing bandwidth. It is defined as underlimit if it receives less resources, and otherwise it is defined as at-limit. The status of the class is defined by an *estimator* that measures the bandwidth consumption of the classes over a predetermined interval. The estimator can be implemented in different ways, depending on the desired estimation timescale. If a class is underlimit and has a persistent backlog, it is said to be *unsatisfied*, otherwise it is *satisfied*.

The formal link sharing guidelines can then be formulated as follows:

A class can continue unregulated (i.e served only by the general scheduler) if:

1. The class is not overlimit, or
2. The class has a non-overlimit ancestor that has no unsatisfied descendants.

If these conditions are not met, the class must be regulated by the link sharing scheduler. According to so-called alternate link sharing guidelines, a regulated class can return to an unregulated status when either:

1. The class becomes underlimit, or
2. The class has an underlimit ancestor with no unsatisfied descendants.

Practical approximations of the link sharing guidelines for CBQ

In the formal link sharing guidelines, both the limit status of the parents and the satisfied status of the descendants of the parent classes have to be examined. This is very time consuming and inefficient for practical implementation. As a result, practical approximations for the formal link sharing rules have been developed. Some of the most well known approximations are ancestor-only link sharing and top level link sharing.

- **Ancestor-only link sharing** allows overlimit classes to borrow capacity from their parents, as long as the parents are underlimit. This means that root class resources are shared with the leaf classes through the intermediate classes.
- **Top level link sharing** is similar to ancestor-only link sharing, except for the fact that borrowing can only proceed to a defined top level in the tree. Resources cannot be borrowed above this level.

However, it should be noted that even the practical approximations cause considerable resource overhead when implemented in real life, because the status of the classes has to be frequently estimated and checked.

4.2.2 HTB

A Hierarchical Token Bucket (HTB) is a packet scheduler implemented in the latest versions of the Linux kernel. The basic idea behind HTB is similar to CBQ in the sense that the scheme is class-based and allows borrowing from parent classes. However, instead of borrowing capacity, HTB uses the concepts of tokens and buckets for borrowing. With HTB, each class has a *rate* parameter, which defines a guaranteed bandwidth for the class, and a *ceil* parameter that defines an upper limit for the sending rate of a class. The tokens are borrowed in *quantums*, *cburst* noting the size of the *ceil* bucket, i.e. the maximum number of bytes that can be lent in a single round.

HTB uses different borrowing rules for the leaf classes and for the intermediate and root classes. The borrowing rules used for the leaf classes are:

- *sending rate* < *rate*: the class will dequeue a certain number of bytes, corresponding to the number of available tokens.
- *rate* < *sending rate* < *ceil*: the class tries to borrow tokens from its parent class. If the parent has enough tokens, the leaf class dequeues up to *cburst* bytes.
- *sending rate* > *ceil*: packets cannot be dequeued from this class.

Correspondingly, the borrowing rules for the intermediate and root classes are:

- *sending rate* < *rate*: the intermediate class can lend tokens to child classes.
- *rate* < *sending rate* < *ceil*: the intermediate class does not have tokens of its own to lend, but it tries to borrow from its parent class and lend these tokens to competing child classes.
- *sending rate* > *ceil*: the intermediate class can not borrow from its parent and thus will not lend tokens to the child classes.

In summary, when a child class needs to borrow a token, it will request one from its parent class. If the parent class cannot borrow one, the request is passed hierarchically up the class tree until a token can be borrowed from some level or the root class is reached.

4.3 Deadline-based algorithms

It has been analytically proved that the bandwidth sharing algorithms presented in the previous section are able to provide a finite delay-bound. However, providing delay guarantees with these algorithms requires the assignation of a class with a large service weight. This easily leads to underutilization of resources, since a delay-sensitive class does not necessarily need much bandwidth. Thus, a more efficient approach to providing delay-bounds is to serve the delay-sensitive classes based only on their deadlines. This section introduces the best known deadline-based algorithm, the Earliest Due Date (EDD).

4.3.1 EDD

The earliest Due Date (EDD) [108], also known as EDF (Earliest Deadline First), is a classic example of a deadline-based scheme where packets are scheduled based on the earliest-deadline-first principle [152]. EDD was originally designed for serving individual flows, but it can also be applied to class based differentiation.

Working with the assumption that the traffic arriving in each class is periodic and using d_i to denote the period for class i, the EDD algorithm works simply as follows: upon arrival of the k^{th} packet of class i at the router at time a_i^k, the packet is stamped with a deadline

$$D_i^k = a_i^k + d_i, \tag{4.4}$$

i.e. the sum of its arrival time and period. The packets are then served in the numerical order of their deadlines. Notice that, in reality, the arriving traffic is not periodic; the purpose of the period is only to describe the expected inter arrival time of packets.

4.4 Size-based scheduling

For elastic traffic, file size information can be used to minimize flow delays with size-dependent scheduling mechanisms. [12] introduces size-based schedulers for wireless downlink data channels, but they can also be applied to fixed IP networks. The random file size is denoted by X, the stationary service rate of flow i by R_i, and the flow arrival intensity by λ. The system is modeled with a M/G/1 queue with service time $S = X/E[R_i]$ and load $\rho = \lambda E[S]$. The stability is given by $\rho < 1$.

4.4.1 FB and SRPT

The Foreground Background (FB) policy always serves the flow that has received the least amount of service. Out of the non-anticipating policies, the FB policy is known to minimize the mean flow delay when service time distribution belongs to the decreasing hazard rate (DHR) property class [12]. On the other hand, the Shortest Remaining Processing Time (SRPT) policy minimizes the mean flow delay by always serving the flow with the smallest remaining processing time.

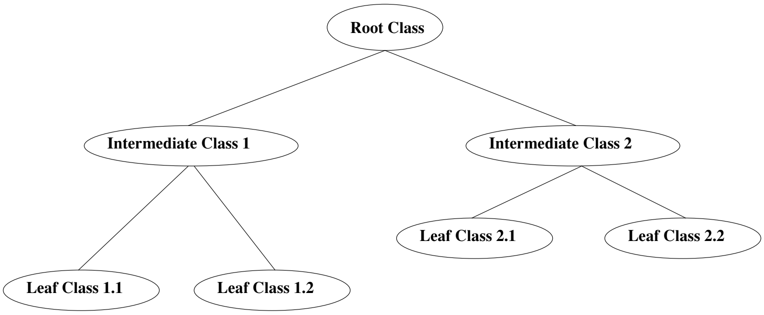


Figure 4.1: Hierarchical structure for bandwidth sharing

5 Adaptive scheduling algorithms

5.1 Introduction

Adaptive provisioning at the packet level can be based on different quantities such as offered load, packet delay or packet loss. Most of the approaches proposed so far have adopted the idea of adjusting the bandwidth shares of the classes in a rate based scheduling algorithm. This is a natural solution, since the majority of the conventional, static scheduling algorithms already implemented in network routers are rate-based (see the bandwidth sharing and hierarchical bandwidth sharing algorithms in Chapter 4). However, a significant problem with rate-based schedulers is that provisioning delay guarantees may result in a waste of resources, since a large bandwidth share must be allocated to a traffic class in order to maintain a low delay. An alternative approach for conventional rate-based scheduling is to use packet delay measurements directly in resource allocation decisions. This guarantees that it is not necessary to over-provision the classes.

5.2 Algorithms based on capacity adaptation

5.2.1 Adaptive WFQ based on effective bandwidths

Moore et al. [115] have developed an adaptive scheduling algorithm that adapts resource allocation based on the calculated theoretical equivalent bandwidth that would be required to support the delay-bound or loss rate. The algorithm uses the WF^2Q^+ scheduler as a basis and adjusts the weights of the classes. The algorithm can perform three kinds of adaptations:

1. calculates the capacity required to preserve a delay-bound within a certain proba-

bility

2. calculates the capacity required to maintain a loss-rate with a given buffer size
3. calculates the buffer size required to preserve a loss-rate when the service rate is known.

The authors use the E-KQ traffic envelope estimator [92] for long term effective bandwidth calculations with a given loss rate and queue size. It is assumed that the basic measurement period used is τ and that measurement data is collected during $T \cdot \tau$ seconds. The effective bandwidth E_{long} is calculated using the following equation:

$$\bar{R}_T + \alpha_{long}\sigma_T = E_{long}, \tag{5.1}$$

where $\bar{R}_T$ and σ_T are the mean and standard deviation of the maximal traffic envelopes. The required queue size q can be calculated utilizing the information about the short term mean and standard deviation of the traffic envelope, $\bar{R}_k$ and σ_k, that reflect the burstiness of the traffic.

$$\max_{k=1,2,\ldots,T} k\tau(\bar{R}_k + \alpha_{short}\sigma_k - E_{long}) = q. \tag{5.2}$$

5.2.2 Adaptive WFQ based on M/M/1/K queue analysis

Liao et al. ([106]) have defined a dynamic provisioning method that aims to provide a delay guarantee and differentiated loss assurance for the traffic classes. Their method consists of two parts: a node provisioning algorithm and a core provisioning algorithm. The node provisioning algorithm uses measurement data to predict SLA violations and adjusts the class weights of a WFQ scheduler in order to prevent transient service level violations. The node provisioning algorithm communicates with the core provisioning algorithm by notifying it about severe and long lasting SLA violations. The core provisioning algorithm then adjusts the rate regulation at the network edge.

The appropriate service weights for the classes are computed with the help of an analytic M/M/1/K model, where K is a threshold after which packets are dropped. The maximum delay D_{max} is proportional to K, since delay guarantees are provided by dropping packets that exceed the threshold K. Given a packet loss bound P^*_{loss} for a class, the node provisioning algorithm aims to keep the measured average loss $\bar{P}_{loss}$ below P^*_{loss}. If P^*_{loss} is very small, it is difficult to obtain an accurate packet loss measurement quickly. Thus, measurement accuracy is improved with the help of the average queue length N_q as follows:

$$N_q = \frac{\rho}{1-\rho}(\rho - (K+1)P_{loss}). \tag{5.3}$$

The set point in the control algorithm is the target traffic intensity $\tilde{\rho}$, which is calculated as:

$$\tilde{\rho} = (\rho^{sup} + \rho^{inf})/2. \tag{5.4}$$

Measured traffic intensity $\bar{\rho}$ is used as the feedback signal. ρ^{sup} and ρ^{inf} are obtained from the upper loss threshold $\gamma_a P^*_{loss}$ and the lower threshold $\gamma_b P^*_{loss}$.

The node provisioning algorithm can provide the following control actions:

1. If $\bar{N}_q(i) > N_q^{sup}(i)$, reduce traffic intensity to $\tilde{\rho}(i)$.

2. If $\bar{N}_q(i) < N_q^{inf}(i)$, increase traffic intensity to $\tilde{\rho}(i)$.

Traffic intensity can be reduced either by increasing the service weights or by reducing the arrival rate. Correspondingly, traffic intensity can be increased by decreasing the service weights or by increasing the arrival rate. The arrival rate can be decreased or increased by the core provisioning algorithm that changes the parameters in the edge traffic conditioners. However, the details of the core provisioning algorithms have been omitted here.

5.2.3 Joint buffer management and scheduling (JoBS)

Christin et al. ([39]) have proposed a Joint Buffer Management and Scheduling (JoBS) mechanism that provides both absolute and proportional differentiation of packet loss and delay. The refined version of the algorithm is also able to provide absolute and relative bandwidth guarantees. Contrary to previous approaches, buffer management and scheduling decisions are interdependent in JoBS, i.e. the service rates and the amount of traffic to be dropped are determined in a single decision. This decision is based on a non-linear optimization problem that can be solved heuristically.

$a_i(t)$ and $l_i(t)$ denotes the amount of traffic arrivals and dropped traffic from class i at time t, and $r_i(t)$ denotes the allocated service rate to class i at time t. In addition, A_i, R_i^{in} and R_i^{out} denote the arrival, input and output curves of class i. The vertical and horizontal distances between the input and output curves represent the backlog B_i and the delay D_i of class i, as shown in Figure 5.1.

In order to form an appropriate rate allocation for the classes, the scheduler calculates a projection of current backlogged traffic delays. In the calculation it is assumed that service rates are not changed after time s and there will be no further packet arrivals or drops. The projected delay of class i at time s, $\tilde{D}_{i,s}(t)$ is defined as

$$\tilde{D}_{i,s}(t) = \max_{t-s<x<t} \{x | \tilde{R}_{i,s}^{out}(t) \geq R_i^{in}(t-x)\}. \tag{5.5}$$

In the following subsections we present the absolute and relative delay and packet loss QoS constraints and the objective function for the original JoBS algorithm [39].

QoS constraints

The Absolute Delay Constraints (ADC) guarantee that the projected delays of class i do not exceed a worst-case delay-bound d_i:

$$\max_{s<t<s+\tilde{T}_{i,s}} \tilde{D}_{i,s}(t) \leq d_i. \tag{5.6}$$

The Relative Delay Constraints (RDC) enforce proportional delay differentiation between classes. Here, the term delay means the average projected delay $\bar{D}_{i,s}$, a time averaged projected delay for class i over a period $\tilde{T}_{i,s}$. Since some flexibility is allowed in the proportional differentiation, the RDC constraints can be written as

$$k_i(1-\epsilon) \leq \frac{\bar{D}_{i+1,s}}{\bar{D}_{i,s}} \leq k_i(1+\epsilon), \tag{5.7}$$

where $0 \leq \epsilon \leq 1$ and $k_i > 1$ is a differentiation coefficient. The absolute and relative QoS loss rate constraints for the are formulated in the same way as the delay constraints. The loss rate $p_{i,s}$ is defined here as the fraction of traffic that is dropped during the time interval $[t_0, s]$, where t_0 is the beginning of the current busy period.

Objective function

If the presented QoS constraints and the system's boundary conditions can be fulfilled, optimization will be performed with respect to the objective function $F(x_s)$. The objective function is formulated as

$$F(x_s) = \sum_{i=1}^{Q}(r_i(s) - r_i(s^-))^2 + C^2 \sum_{i=1}^{Q} l_i(s), \tag{5.8}$$

where C is the link capacity. The objective function aims to avoid changing the current service rate allocation and dropping traffic.

It is clear that this kind of non-linear optimization problem is too complex to be solved in a real implementation. Therefore, the authors have developed a heuristic algorithm to approximate the optimization problem. The idea is to divide optimization into smaller problems that are computationally less expensive.

5.3 Algorithm based on delay adaptation

The main problem with the adaptive approaches presented in the previous section is that they rely on optimization problems or analytic models that may be too complex and expensive to implement in practice. Our goal has been to develop an algorithm that has a low complexity and that is truly measurement based, requiring no optimization problems or analytic models to be solved. The idea is that the algorithm should be easy to implement in router hardware. Another major difference between our approach and the other proposed adaptive scheduling algorithms is that, in our approach, adaptation is totally based on queuing delay rather than capacity. Delay-based provisioning is justified, since most traffic nowadays is time-critical to some extent. For example, web-users expect constant feedback on the document retrieval process. Another important advantage of delay-based algorithms is that it is not necessary to over-provision resources in order to guarantee short delays. With rate-based algorithms, a large service weight has to be allocated to real-time classes in order to provide a delay-bound.

5.3.1 Our approach: delay-bounded HPD (DBHPD)

In [20], we proposed a delay-bounded HPD (DBHPD) scheduling algorithm for combined absolute and proportional delay differentiation. In this algorithm, the most delay sensitive class is assigned an absolute delay bound. If this bound is on the verge of being violated, a packet is dispatched directly from this class; otherwise the operation is based on the delay ratios between classes according to [47].

The delay-bounded HPD (DBHPD) algorithm first checks whether or not the package in the highest class (class 0) is about to violate its deadline. d_{max} denotes the delay bound in the highest class, a safety margin for the delay bound is denoted by t_{safe}, the arrival time of the packet in the highest class queue by t_{in}, and t_{curr} denotes the current time. The packet in the highest class queue is considered to be violating its deadline if

$$t_{in} + d_{max} < t_{curr} + t_{safe}. \tag{5.9}$$

If delay violation is not occurring, the algorithm takes into account the delay ratios between the other classes. $\bar{d}_i$ denotes the average queuing delay of class i packets and δ_i denotes the Delay Differentiation Parameter (DDP) of class i. The average delay ratio in the two classes i and j should equal the DDPs ratio in these classes

$$\frac{\bar{d}_i}{\bar{d}_j} = \frac{\delta_i}{\delta_j}, \quad 1 \leq i, j \leq N, \tag{5.10}$$

assuming that $\delta_1 < \delta_2 < \ldots < \delta_N$. In [47] this is interpreted so that the normalized average delays of traffic classes must be equal, i.e.,

$$\tilde{d}_i = \frac{\bar{d}_i}{\delta_i} = \frac{\bar{d}_j}{\delta_j} = \tilde{d}_j, \quad 1 \leq i, j \leq N. \tag{5.11}$$

When the server becomes free at time t, the DBHPD algorithm selects a packet from a backlogged class j for transmission, with the maximum normalized hybrid delay [47]:

$$j = \arg\max\,(g\tilde{d}_i(m) + (1-g)\tilde{w}_i(m)), \tag{5.12}$$

where $\tilde{d}_i(m)$ and $\tilde{w}_i(m)$ denote the normalized average queuing delay and the normalized head waiting time (i.e. the waiting time for the first packet) of class i when m packets have departed and $0 \leq g \leq 1$ is a weighting coefficient. Thus, the algorithm utilizes measurements of both short and long term queuing delays ($\tilde{d}_i(m)$ and $\tilde{w}_i(m)$) in the scheduling decisions. The operation of the algorithm depends largely on how the average delay $\bar{d}_i(m)$ is calculated, because it determines the amount of history that is incorporated into the scheduling decisions. In the following subsections we will present a general estimation theory and a simple and practical estimator for $\bar{d}_i(m)$.

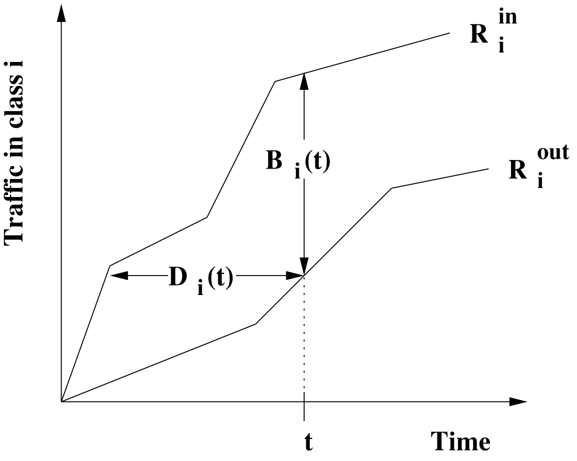


Figure 5.1: Backlog and delay in JoBS

6 Estimating delay in the DBHPD algorithm

In Chapter 5 we stated that the operation of the DBHPD algorithm depends largely on how the average delay $\bar{d}_i(m)$ is estimated. In this chapter the theoretical foundations of the most common estimation approaches are first provided and their limitations are discussed. Finally, basic algorithms for estimating delay in the DBHPD algorithm are presented.

6.1 Estimation theory

Measurements and estimations are used at many levels of network control. Examples are link utilization estimation for routing, arrival rate estimation for traffic conditioning and delay estimation for packet scheduling. The desirable properties of an estimator depend on its application. However, there are some common properties that should be shared by ideal estimators: stability, agility and low implementation complexity. Stability means that the estimator should provide a smoothed, long term estimate of the desired quantity. Agility, on the other hand, refers to the ability of the estimator to follow recent changes with a sufficient degree of accuracy. The best known estimators utilized in network control are the Token Bucket (TB) estimator and the Exponential Weighted Moving Average (EWMA) estimator [156]. The main benefit of these estimators is their simplicity. However, they cannot be tuned to be both stable and agile, since they rely on a static parameter (a measurement window or a weighting factor) that determines the amount of history to be incorporated in the estimate. Some estimators based on Kalman filters or neural networks have been developed to overcome this problem. However, these estimators are often too complex to be implemented in router software/hardware and are presented here mainly as an example of how complex and theoretically sophisticated estimators may lead to heavy calculations.

6.1.1 Token Bucket (TB)

The Token Bucket (TB) estimator filters the data with a rectangular weighting function. When a new observation X_k is obtained, the new estimate E_k is calculated as follows:

$$E_k = \frac{E_{k-1} * WL + X_k}{now - T + WL}, \tag{6.1}$$

where WL is the window length, now is the current time and T is the time when the last estimate was calculated. The window length determines the stability and agility of the TB estimator: the estimator is stable with a long window length.

6.1.2 Exponential Weighted Moving Average (EWMA)

The Exponential Weighted Moving Average (EWMA) estimator is similar to the TB filter, except for the fact that the EWMA has an exponential weighting function. The new estimate E_k is updated with

$$E_k = \gamma * X_k + (1 - \gamma) * E_{k-1}, \tag{6.2}$$

where $0 \ \leq \gamma_i \leq 1$. The term γ is the weighting factor or filter gain, which corresponds to the window length WL of the TB estimator. Both the TB and EWMA estimators are extremely simple to implement, since they only require one update equation and one value (E_{k-1}) to be stored in memory.

6.1.3 Kalman filters

Kalman filters are known to be optimal for the estimation of a linear system in the sense that they minimize mean squared estimation error. A discrete Kalman filter estimates the state x of a discrete-time process that follows the linear stochastic difference equation [153]

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}, \tag{6.3}$$

where u is an optional control input and w represents system noise. Since the system state x is not known, it must be derived indirectly with the help of measurement z:

$$z_k = Hx_k + v_k, \tag{6.4}$$

where H is a constant matrix and v is a matrix of measurement error. Variables v and w are assumed to be independent and follow normal probability distributions as follows:

$$p(w) \sim N(0, Q), \tag{6.5}$$

$$p(v) \sim N(0, R), \tag{6.6}$$

where Q is process noise covariance and R is measurement noise covariance. In practice, R can be obtained by measuring the measurement noise variance, but Q is more difficult to determine since the form of the process is not usually known. One option is to tune R and Q by using off-line measurements.

The actual discrete Kalman filter algorithm is divided into two steps: a time update step and a measurement update step. The time update step calculates the a priori prediction for the system state and the estimate error covariance. These a priori predictions are denoted by x_k^- and P_k^-. The measurement update step generates an a posteriori state estimate of the system state and estimate error covariance; x_k and P_k. P_k^- and P_k are defined as

$$P_k^- = E[e_k^- e_k^{-T}], \tag{6.7}$$

$$P_k = E[e_k e_k^T], \tag{6.8}$$

where e_k^- and e_k are the a priori and a posteriori estimate errors. In order to obtain the a posteriori estimate of the system state, x_k is calculated as a linear combination of the a priori estimate x_k^- and a weighted difference between the measurement z_k and a measurement prediction Hx_k^- as follows:

$$x_k = x_k^- + K(z_k - Hx_k), \tag{6.9}$$

where K is a blending factor that minimizes the a posteriori error covariance P_k. A form of K that minimizes P_k is

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}. \tag{6.10}$$

The discrete Kalman filter algorithm equations for the two update steps can be summarized as:

Discrete Kalman filter time update equations:

$$x_k^- = A x_{k-1} + B u_{k-1} \tag{6.11}$$

$$P_k^- = A P_{k-1} A^T + Q \tag{6.12}$$

Discrete Kalman filter measurement update equations:

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}. \tag{6.13}$$

$$x_k = x_k^- + K(z_k - H x_k), \tag{6.14}$$

$$P_k = (I - K_k H) P_k^- \tag{6.15}$$

A stationary Kalman filter actually becomes an EWMA filter. The non-stationary case, however, is significantly more complex than TB and EWMA, since it requires several update equations and matrix operations, such as inverse matrix computation, which are difficult to implement in router software/hardware in real-time.

Some practical examples of the use of Kalman filters for estimation include [29], [19] and [145]. In [29] Kalman filters are applied to estimate the number of competing terminals in a 802.11 network. In [19], a Kalman filter based estimator is used to forecast capacity requirements of interdomain links in fixed IP networks. The [145] paper applies Kalman filters to traffic matrix estimation.

6.1.4 Filters based on neural networks

Neural networks are nonlinear systems that are capable of learning. Their learning property makes neural networks attractive alternatives for many network control applications, since Internet traffic processes are extremely hard to predict using conventional estimators. Several different types of neural networks exist, such as perceptron networks and recurrent neural networks. In perceptron networks, the task of a perceptron is to combine several inputs using a weighted addition. The resulting output can be further processed by a transfer function. The perceptrons in a neural network are arranged in three types of layers: input, output and hidden layers. The data is fed into the input layer and the hidden layers process the data. Finally, the output layer returns the processed data. A neural network operates in two phases: a learning phase and a recalling phase. In the learning phase, the weighting functions of the perceptrons are adjusted so that they respond properly to certain types of input. Depending on the purpose of the neural network, millions of samples may be required to train the network. In the recalling phase the neural network generates proper responses to the inputs.

In order to learn the optimal weights to be used by the perceptrons, the network calculates the error between the measured output and the desired output in the output layer. The back-propagation algorithm (BPA) is used to propagate the error to all the preceding layers via existing connections. The weighting factors in each perceptron are then adapted based on the error.

For example, the following notations can be used in the back-propagation algorithm:

- $x_j[k]$: The current output of the j-th perceptron in layer k
- $w_{ji}[k]$: Weighting function of the connection between the i-th perceptron from layer (k-1) and the j-th neuron from layer k
- $I_j[k]$: The combined input of the j-th perceptron in layer k

In a neural network that utilizes a back-propagation algorithm, the output of a perceptron can be written as

$$x_j[k] = f\{\sum_i (w_{ji}[k].x_i[k-1])\} = f\{I_j[k]\}, \tag{6.16}$$

where f is a selected transfer function. The absolute error function E at the output layer is defined as

$$E = \frac{1}{2}\sum_l (y_l - v_l)^2, \tag{6.17}$$

where y_l and v_l represent the desired and current output values. The parameter that is back-propagated to the preceding layers is

$$e_j[s] = -\frac{\partial E}{\partial I_j[k]}. \tag{6.18}$$

The weighting function in each perceptron is then adapted so that the absolute error decreases. This is achieved by changing the weights in the opposite direction to the gradient vector:

$$\Delta w_{ji}[k] = -lc\frac{\partial E}{\partial w_{ji}[k]} = lc.e_j[k].x_i[k-1], \tag{6.19}$$

where lc is a learning coefficient.

Neural networks have been applied in many practical traffic classification and estimation problems. In [22], multilayer perceptron classifier networks are used for classifying Internet traffic without any online analysis of packet headers. In [45], an adaptable neural-network model has been applied to recursive nonlinear traffic prediction and the modeling of online and offline MPEG video sources. [43] uses a Recurrent Multilayer Perceptron Network (RMLP) for large-scale IP traffic matrix estimation. Like Kalman filters, neural network filters add considerable complexity compared to TB and EWMA, since they require the calculation of partial derivatives and the recomputing of neural network parameters with back-propagation.

6.2 Practical delay estimation in the DBHPD algorithm

The theoretical foundations of the most common estimation approaches (TB, EWMA, Kalman filter and filters based on neural networks) were presented in the previous section, where it was stated that the neural network and Kalman filters are difficult to tune. In this section we will present the original sum estimator for delay estimation proposed by Dovrolis in [47], as well as a practical EWMA filter for computing $\bar{d}_i(m)$. We will also discuss how the parameters of the EWMA filter should be selected in order to make it feasible in practice.

6.2.1 Simple sum estimator

In the original form [47], the average delay of class i after m packet departures, $\bar{d}_i(m)$, is calculated by a simple sum estimator as follows: The sequence of class i packets that have been served is denoted by $D_i(m)$ t, and the delay of the m'th packet in $D_i(m)$ by $d_i(m)$. Then, assuming that at least one packet has departed from class i before the m'th packet

$$\bar{d}_i(m) = \frac{\sum_{m=1}^{|D_i(m)|} d_i(m)}{|D_i(m)|}. \tag{6.20}$$

However, this kind of calculation to infinity is not feasible in practice, since the counter for the sum of delay values easily overflows when enough packets have departed from a certain class. Additionally, we do not want to incorporate infinite history into the estimator and thus into the scheduling decisions, because historical information becomes irrelevant after some point.

6.2.2 EWMA estimator

A simple approach to eliminating the overflow problem in the sum estimator is to update $\bar{d}_i(m)$ in each packet departure with exponential smoothing as follows:

$$\bar{d}_i(m) = (\gamma_i d_i(m) + (1 - \gamma_i)\bar{d}_i(m-1)), \tag{6.21}$$

where $0 \leq \gamma_i \leq 1$. Now, calculation to infinity is not required and the amount of history can be determined by selecting the γ_i parameters, which may be the same or different for each traffic class. If the γ_i parameters are chosen so that calculations are performed with powers of two, the estimator operations can be performed by simple bit-shifting, which is considerably more efficient than heavy multiplications and divisions. In fact, this estimator becomes a kalman filter, if γ_i is selected properly. We have also developed the basic EWMA estimator further in [21], so that the filtering coefficients of the estimation algorithm can be adaptively adjusted depending on the actual traffic characteristics of the service classes.

7 Adaptive Scheduling Simulations in fixed IP Networks

In the previous chapters, conventional and adaptive scheduling algorithms as well as delay estimators were presented and analyzed at a qualitative level. We introduced our approach to adaptive scheduling, the Delay-Bounded HPD (DBHPD) algorithm, which is completely measurement-based and does not require analytical models to be solved. The next four chapters concentrate on evaluating the DBHPD algorithm using both simulations and measurements. This chapter describes the topology, traffic mixes and other relevant parameters used in the network simulation stage, in which DBHPD is compared with the Deficit Round Robin (DRR) static bandwidth sharing algorithm, and also presents the simulation results.

7.1 Performance evaluation of the DBHPD and DRR algorithms

7.1.1 Goals of the simulation study

In our earlier work we evaluated the DBHPD algorithm with high abstraction level simulations and showed that it performs better than static scheduling algorithms. The objective of this simulation study is to compare the performance of static and adaptive provisioning methods with ns2-simulations in a more realistic setup, with a considerably larger topology and more advanced traffic models. The scheduling algorithms that we investigated were the Deficit Round Robin (DRR) [141] and the delay-bounded HPD (DBHPD) [20]. DRR was used as a benchmark, since it is one of the most simple and practical GPS approximations in commercial use. DRR is also used as general scheduler in the well-known CBQ algorithm. In our earlier studies we have also simulated one type of adaptive Deficit Round Robin, but it could not compete with DBHPD. Other adaptive scheduling algorithms could have been considered as well but they either aim at different type of QoS differentiation making comparison with DBHPD meaningless or implementing these algorithms would be too complex due to heavy optimization and estimation procedures. Since the final goal in this thesis is to compare the algorithms in both simulations and measurements only algorithms that could also be implemented were chosen. We have implemented both the DRR and the delay bounded HPD algorithm in ns2 simulator and conducted several simulations to evaluate their performance in this more realistic setup.

7.1.2 Simulation topology

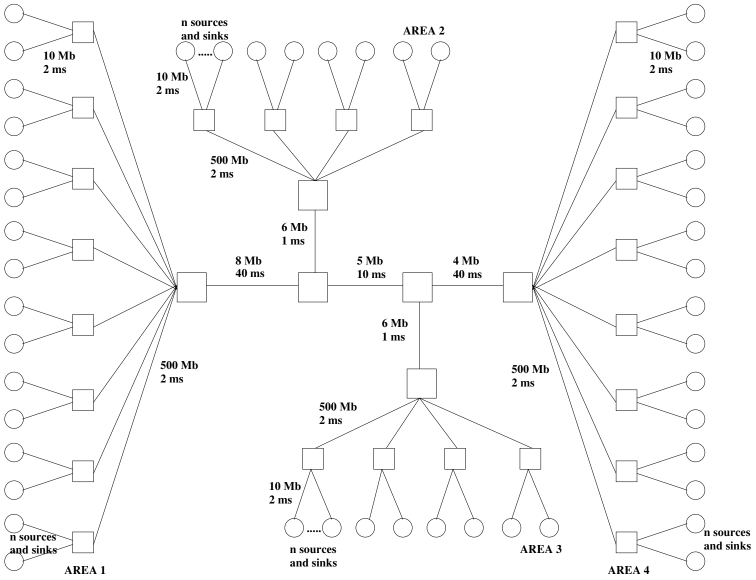


Figure 7.1: Topology.

Figure 7.1 shows the topology used in our simulations. It should be noted that the topology is restricted to a single domain and that topology parameters (link bandwidths and delays) do not necessarily represent any specific network technology. However, the topology and parameters have been selected to capture the most fundamental characteristics of real networks: multiple bottleneck links, paths with low and high delays [1], cross traffic, bidirectional traffic, etc. [56]. In Figure 7.1 the topology is divided into four separate areas. In the simulation, the traffic flows from Area1 to Area4, Area2 to Area3, and correspondingly from Area4 to Area1 and from Area3 to Area2. The path between Area1 and Area4 is a high delay path, while the path between Area2 and Area3 has a low delay.

7.1.3 Traffic mixes

We used five different traffic types in the simulations, namely FTP, HTTP, Video, VoIP and control traffic (small and large control messages). The generation of HTTP-traffic is based on the webcache model implemented in ns2. In this model it is assumed that a HTTP session consists of a number of page requests, each possibly containing several objects. We also used the webcache model for FTP-traffic, except that with FTP there is no reading time between page requests, and a page always contains one object. Video traffic generation is based on a real trace of mpeg4 coded BBC news, from which the traffic stream has been regenerated using a Transform Expand Sample (TES) process. VoIP traffic is generated with a simple model representing a continuous conversation with talk spurts and silence periods, while control traffic is created by sending either large, individual control messages at regular intervals or deterministic bursts of a few, short packets. Table 7.1 shows the number of FTP and HTTP clients, as well as the video, VoIP and control traffic sources belonging to these areas. Note that, for example, when there are 8 HTTP clients in Area1, there will correspondingly be 8 HTTP servers in Area4.

[1] Propagation delay is 98 ms in the long-delay path and 20 ms in the short-delay path.

Table 7.1: Number of traffic sources in different areas

Application	**Area1**	**Area2**	**Area3**	**Area4**
FTP clients	4	2	2	4
HTTP clients	8	4	4	8
Video clients	4	2	2	4
VoIP clients	5	2	2	5
Control sources	2	2	2	2

7.1.4 General simulation parameters

In the simulation scenarios we have kept the simulation topology and the number of traffic classes fixed and changed the traffic load levels and the traffic shares of different applications. Traffic is mapped into different classes according to the guidelines provided by our previous research. According to this research the best results could be achieved by dividing the traffic into four classes: first class for real-time applications sending short packets (VoIP etc.), second class for real-time applications sending larger packets and more bursty traffic (video etc.), third class for applications sending short TCP flows (Web etc.), and fourth class for applications sending long TCP flows (FTP etc.). We used two scenarios for the relative traffic shares of different applications: (FTP: 9%, WWW: 71%, Video: 9%, VoIP: 10%, Control: 1%) and (FTP: 29%, WWW: 40%, Video: 20%, VoIP: 10%, Control: 1%). The first scenario corresponds to the situation today and the second scenario reflects the situation in the future, when the amount of multimedia and peer-to-peer traffic is expected to be larger. We used three average total load levels, 75%, 80% and 85% (measured from the most congested bottleneck link), to test the algorithms.

We dimensioned the total buffer size in the routers to 230 packets, of which 200 packets were allocated for elastic traffic and 30 for real-time traffic. We tested the performance of the scheduling algorithms using DropTail and RED queue management in order to see if there is less difference between them when more advanced queue management methods are used. The parametrization of the DRR and the delay bounded HPD algorithms was following: In DRR, the real-time traffic classes were over-provisioned by a factor of two, and the excess capacity was divided between classes meant for elastic traffic, in proportion to their expected load shares. In the delay-bounded HPD algorithm we set d_{max} to 5 ms and the target ratio for delays between consecutive classes to 4. The safety margin in Eq. 5.9 was set to $d_{max}/10$, and g in Eq. 5.12 was set to 0.875.

7.1.5 Collected results

We performed five statistically independent iterations of each simulation scenario. The simulation time in each iteration was 3600s, including a 400s warm-up period. We used the simulations to measure bottleneck link utilization and average aggregate throughput, packet loss and end-to-end delay for each traffic type. In addition, we measured object throughput and its variability for web and FTP traffic. Object throughput results are seldom presented in research papers. However, in our opinion object throughput is an important performance metric, especially from the user's point of view, since the user constantly expects to receive some feedback on the process of a document retrieval (i.e. the loading of independent objects).

7.2 Simulation results

In this section the results from the simulations are presented. The DRR and DBHPD algorithms are compared to each other by investigating the object throughput and packet losses for FTP and HTTP, and end-to-end delays and packet losses for Video and VoIP traffic. Link utilizations, recorded from the common 5Mb bottleneck link (Figure 7.1), are also used for comparison. The results were averaged over all the clients belonging to the same edge area. To better assess the variability of object throughput in a single iteration, we also recorded the relative standard deviation (rsd) of the object throughput for FTP and HTTP. The rsd value reflects the short term variability of the object throughput, which is also a relevant performance metric for the end user, who expects constant quality of service.

7.2.1 Performance of the DRR and the DBHPD algorithms

In Tables 7.2, 7.3 and 7.4 the above results are shown for every algorithm combination, using application mix 1 with a mean load of 80%. The results with RED are only shown on Table 7.2, since the differences between these and the DropTail results were so small. From Table 7.2 and Table 7.3, it can be seen that the delay-bounded HPD algorithm is able to provide considerably better service for HTTP-traffic, in terms of packet losses and object throughput in both the high and the low delay paths. In addition, according to the rsd values, object throughput suffers less variability with delay-bounded HPD, resulting in a more constant quality of service from the user's perspective. The absolute difference between DBHPD and DRR, in terms of HTTP object throughput, may not seem that large but it should be taken into account that 71% of the total traffic is HTTP, meaning that the weighted throughput gain is considerable.

Naturally, we cannot provide a considerably better service to one service class without degrading the performance of other classes. With delay-bounded HPD, the performance of the FTP traffic is worse than with the static DRR algorithm. However, FTP packet loss is still nearly zero with DBHPD, which is more important for FTP traffic than a short delay. In addition, since FTP traffic is mapped to the best-effort class that, by default, should not be provided with any quality of service, we consider it more important to provide a good service to the HTTP traffic, which is interactive and more delay sensitive. It should also be noted that if we want FTP traffic to receive more resources from the delay-bounded HPD algorithm, this could be achieved by setting the DDP values in a different way.

From the result tables it can also be observed that the end-to-end delays for Video and VoIP are somewhat less with DRR than with delay-bounded HPD. This is due to a high over-provisioning factor used by the DRR for real-time traffic (in a rate-based algorithm, low delay can only be guaranteed by allocating a large service weight). However, a difference of a few milliseconds in end-to-end delay is not relevant in practice. All in all, the real-time traffic losses for both algorithms are almost zero (this is why they are not shown in the result tables) and the end-to-end delays in each edge area are perfectly tolerable. Thus, both the delay-bound in DBHPD and capacity over-provisioning in DRR are proper means to guarantee that the real-time traffic gets through with minimal delays and losses. We also experimented with different initial provisioning values for the DRR algorithm. Instead of using an over-provisioning factor of 2 for the real-time traffic, we only used an over-provisioning factor of 1.1. With this provisioning, the packet losses and end-to-end delays suffered by real-time traffic increased as expected, but there was only a negligible improvement for the HTTP and the FTP traffic. In order to improve the performance of HTTP traffic, which suffered large losses with the DRR algorithm, we should use under-provisioning for the real-time classes. This, however, is not feasible since the real-time traffic service was on the verge of becoming intolerable, even with 10% over-provisioning.

It is often argued that active queue management mechanisms, such as RED, can reduce the problems of static scheduling algorithms by dropping packets in advance, before congestion situations occur. However, all the result tables show that RED only provides minor improvements, compared to the performance of the simple DropTail system. We investigated the effect of RED with two different parameter sets, but found that its advantages were negligible with both sets. On the other hand, the results prove that considerable performance advantages can be achieved if the scheduling algorithm is selected properly.

Note that most of the traffic in application mix 1 (71%) consists of short HTTP flows. When the traffic loads of different classes are biased in this way, resource allocation is a challenging task. It is not possible to provide the required service for the enormous num-

Table 7.2: Object throughput statistics, application mix 1

	Object throughputs (kbps)							
	AREA1		**AREA2**		**AREA3**		**AREA4**	
DRR-DROP	**mean**	**rsd**	**mean**	**rsd**	**mean**	**rsd**	**mean**	**rsd**
HTTP	14.6±2.0	1.0	48.2±7.9	1.1	48.6±7.9	1.1	14.6±2.0	1.0
FTP	906.4±9.6	0.52	2192.0±39.3	0.41	2100.2±119.0	0.4	881.7±21.0	0.56
DRR-RED	**mean**	**rsd**	**mean**	**rsd**	**mean**	**rsd**	**mean**	**rsd**
HTTP	14.8±1.8	0.99	48.9±7.2	1.04	49.2±7.4	1.05	14.8±1.8	0.99
FTP	905.1±14.0	0.53	2202.8±37.5	0.4	2055.8±131.7	0.41	878.1±30.5	0.55
HPD-DROP	**mean**	**rsd**	**mean**	**rsd**	**mean**	**rsd**	**mean**	**rsd**
HTTP	17.6±1.1	0.87	55.7±5.2	0.96	56.4±5.1	0.96	17.5±1.1	0.87
FTP	523.0±52.4	0.55	1062.9±109.7	0.43	992.9±102.4	0.48	535.7±63.1	0.55
HPD-RED	**mean**	**rsd**	**mean**	**rsd**	**mean**	**rsd**	**mean**	**rsd**
HTTP	17.8±0.9	0.86	56.7±4.7	0.94	57.0±4.7	0.95	17.8±1.0	0.86
FTP	530.9±48.8	0.52	1037.3±132.1	0.44	964.4±115.3	0.49	516.9±68.7	0.56

Table 7.3: Packet loss statistics, application mix 1

	Packet loss (%)			
	AREA1	**AREA2**	**AREA3**	**AREA4**
DRR-DROP	**mean**	**mean**	**mean**	**mean**
HTTP	6.1±2.4	3.5±1.6	4.8±1.9	6.7±2.4
FTP	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0
HPD-DROP	**mean**	**mean**	**mean**	**mean**
HTTP	2.2±0.8	1.7±0.7	2.2±0.8	2.8±1.0
FTP	0.1±0.0	0.1±0.1	0.2±0.2	0.1±0.1

Table 7.4: End-to-end delay statistics, application mix 1

	End-to-end delay (ms)			
	AREA1	**AREA2**	**AREA3**	**AREA4**
DRR-DROP	**mean**	**mean**	**mean**	**mean**
Video	107.2±0.4	27.2±0.3	27.0±0.2	106.5±0.4
Voip	101.5±0.1	22.6±0.0	22.6±0.1	101.9±0.1
HPD-DROP	**mean**	**mean**	**mean**	**mean**
Video	111.5±1.2	31.5±1.9	30.2±2.0	111.9±2.0
Voip	107.4±0.3	25.9±0.1	25.5±0.3	106.9±0.4

ber of web flows without severely degrading the quality of other traffic types. Tables 7.5 and 7.6 show the results for application mix 2, where the loads of different traffic types are more even. The delay results are not shown, since they are very similar to the results in Table 7.4. In this case, the losses in each class are considerably smaller compared with application mix 1, even with the static DRR algorithm. However, as the mean total load increases from 80% to 85%, the differences between the DRR and the delay-bounded HPD algorithms again become more distinctive. The HTTP traffic losses with DRR are nearly 10 %, compared to 0.6 % at most with the delay-bounded HPD algorithm. The FTP performance is slightly worse with a delay-bounded HPD, but the difference is much less than with application mix 1, since HTTP traffic does not totally dominate this traffic mix.

Table 7.5: Object throughput statistics, application mix 2 with different loads

	Object throughputs (bps)							
	AREA1		**AREA2**		**AREA3**		**AREA4**	
DRR-80	**mean**	**rsd**	**mean**	**rsd**	**mean**	**rsd**	**mean**	**rsd**
HTTP	15.5±1.9	0.92	49.4±7.6	1.04	49.4±7.9	1.05	15.4±1.8	0.92
FTP	490.2±69.3	0.66	952.1±147.8	0.58	1032.4±155.4	0.55	503.4±60.8	0.64
DRR-85	**mean**	**rsd**	**mean**	**rsd**	**mean**	**rsd**	**mean**	**rsd**
HTTP	11.1±1.8	1.07	34.0±6.7	1.24	34.3±7.2	1.26	11.0±1.9	1.07
FTP	425.4±51.0	0.66	798.8±88.8	0.60	859.8±139.5	0.61	426.0±63.1	0.68
HPD-80	**mean**	**rsd**	**mean**	**rsd**	**mean**	**rsd**	**mean**	**rsd**
HTTP	18.8±0.5	0.8	60.1±4.0	0.88	59.8±3.7	0.89	18.8±0.5	0.8
FTP	445.0±66.5	0.67	843.0±139.4	0.61	910.9±141.3	0.59	452.6±60.6	0.66
HPD-85	**mean**	**rsd**	**mean**	**rsd**	**mean**	**rsd**	**mean**	**rsd**
HTTP	17.5±0.5	0.81	52.2±3.5	0.91	52.1±3.8	0.91	17.5±0.5	0.81
FTP	358.9±48.7	0.7	628.5±69.8	0.66	673.2±113.0	0.68	353.8±51.8	0.72

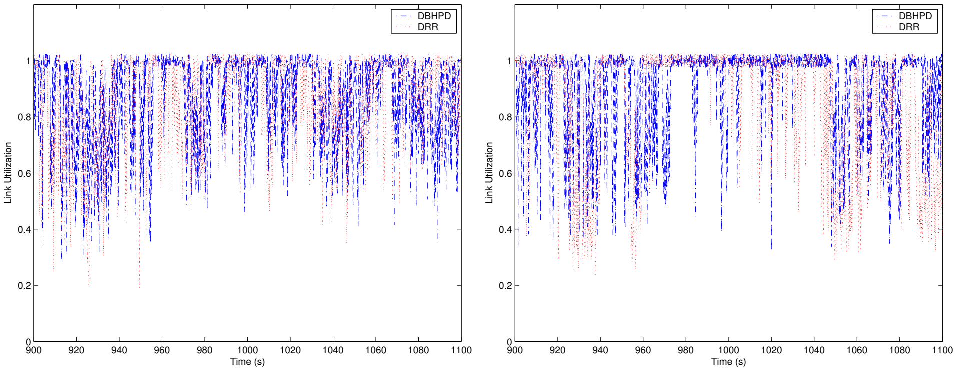
Figure 7.2 shows snapshots of link utilization from the common bottleneck link with a 5Mb capacity, for both application mixes with a total offered load of 85 %. According to these figures DBHPD provides slightly better utilization, at least for application mix 2, in which the class traffic shares are more evenly distributed. It should also be noted that DRR resulted in higher packets losses, especially with HTTP, and in more retransmissions as a result. Thus, a larger part of the DRR utilization consists of retransmissions, compared

Table 7.6: Packet loss statistics, application mix 2 with different loads

	Packet loss			
	AREA1	**AREA2**	**AREA3**	**AREA4**
DRR-80	**mean**	**mean**	**mean**	**mean**
HTTP	2.8±1.9	1.7±1.4	2.5±1.2	3.5±1.4
FTP	0.5±0.2	0.6±0.4	0.6±0.3	0.5±0.2
DRR-85	**mean**	**mean**	**mean**	**mean**
HTTP	6.7±2.2	4.3±1.6	7.0±2.3	9.2±2.7
FTP	0.7±0.2	0.9±0.3	0.9±0.3	0.7±0.3
HPD-80	**mean**	**mean**	**mean**	**mean**
HTTP	0.1±0.1	0.1±0.1	0.2±0.1	0.3±0.1
FTP	0.6±0.4	0.7±0.5	0.8±0.3	0.7±0.4
HPD-85	**mean**	**mean**	**mean**	**mean**
HTTP	0.3±0.1	0.3±0.1	0.4±0.2	0.6±0.2
FTP	0.9±0.3	1.2±0.4	1.3±0.3	1.2±0.4

with DBHPD utilization, which is mainly original data packets.

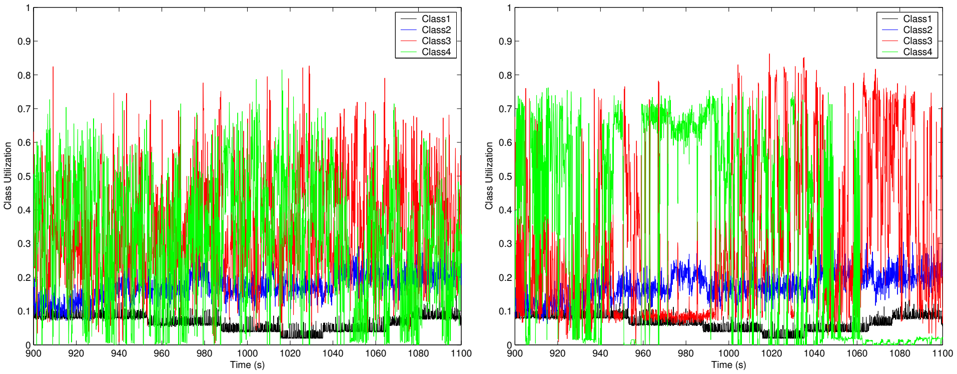
Figure 7.3 depicts the utilizations within individual traffic classes. Due to high packet losses with HTTP, DRR results in a more variable offered load process, since the TCP performs backoff more frequently. This is also reflected in the utilization figures, from which it can be observed that DBHPD provides more steady utilizations for the classes than DRR.



(a) Application mix 1

(b) Application mix 2

Figure 7.2: Bottleneck link utilization with 85 % total offered load



(a) DBHPD, application mix 2

(b) DRR, application mix 2

Figure 7.3: Utilizations within traffic classes for DBHPD and DRR with 85 % total offered load

8 Robust delay estimation for DBHPD

In the previous chapter we used simulations to evaluate the DBHPD algorithm with a simple EWMA estimator. In this chapter we will propose more robust delay estimators for the DBHPD algorithm [21] and compare four possible estimation approaches: simple sum, simple EWMA, EWMA with restart (EWMA-r) and EWMA based on the proportional error of the estimate (EWMA-pe). Our basic idea is to maintain the simplicity of the EWMA estimator, but to modify it in such a way that the estimator operates properly in different regions.

8.1 Simple Sum and EWMA estimators

In its original form [47], the average delay of class i after m packet departures was calculated by a simple sum estimator as described by Eq. 6.20. We stated that calculation to infinity is not feasible in practice, since the counter for the sum of delay values easily overflows as enough packets are departed from a certain class. A simple approach to eliminate the overflow problem in the sum estimator is to apply exponential smoothing according to Eq. 6.21. After doing this, calculation to infinity is not required and the amount of history can be determined by selecting the γ_i parameters. We believe that separate γ_i parameters should be used for each traffic class, since the characteristics of the traffic in each class can be totally different. In principle, the value of γ_i should be related to the regeneration period of the class queue, since it reflects the timescales of arriving traffic. However, since determining the regeneration period would require additional measurements, we propose the use of a fixed system parameter, namely the queue size, to determine γ_i. q_i denotes the physical queue size of class i. Then, γ_i can be determined by an approximation function:

$$\gamma_i(q_i) = \frac{1}{N * \sqrt{q_i} * ln(q_i)}. \tag{8.1}$$

If the queue size is small, it can be assumed that the scheduling decision should not depend too much on history, and that the γ_i value will be higher. The square root and logarithm functions and the number of classes N are used in Eq.(8.1) for scaling the γ_i values to a reasonable range, assuming that queue lengths in a router can range approximately from 10 packets to 10000 packets. It should be noted that Eq. 8.1 is not an exact, analytically derived expression. However, it provides a good guideline for setting the γ_i values. The value of γ is depicted as a function of the queue length in Figure 8.1, assuming 4 service classes. We also argue that the g parameter in Eq. 5.12 should be separate for each traffic class, since the g and γ parameters together determine to what extent the scheduling decision depends on history or the current situation. We propose that the g parameters should be related to the δ parameters defined in Eq. 5.10, which reflect the policy of the operator and the real urgency of the packets. The parameters could be set as follows:

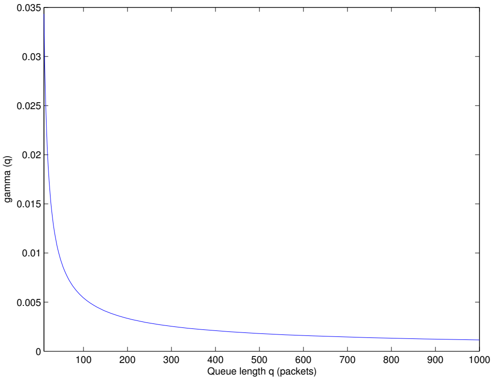


Figure 8.1: The gamma function.

$$g_1 = 0.75, \tag{8.2}$$

$$g_i = g_{i-1} * \frac{\delta_{i-1}}{\delta_i} \tag{8.3}$$

If, according to the policy, the packets in one of the classes are urgent, more weight will be given to the current measurement values than to the history. In practice, the delay-bounded class (class 0) will be used for extremely time-critical traffic, such as VoIP and network control traffic. Class 1, on the other hand, will be used for less urgent real-time traffic, such as video. For video traffic it would be reasonable for the scheduling decision to be 75% based on the current waiting time and 25% on history.

8.2 EWMA estimator with restart (EWMA-r)

One problem with both the simple sum and the EWMA estimator is that they do not take into account the times when the queue becomes idle. If the queue of a certain traffic class is idle for a long time, the delay history of that class should not be taken into account. This is because otherwise, a class that recently became active and that experiences only slight congestion would be served, and would steal capacity from other classes just because it had long delays in its history. Thus, when a queue becomes active after an idle period, the EWMA estimator should be reset. The idea is that after being reset, an average of the queuing delay is calculated as fast until a certain threshold of packets, p_tresh, is departed. After the threshold is reached, the delay is smoothed again with the low pass filter, where the γ_i parameters are determined by Eq. 8.1. Smoothing is not performed below the threshold because the smoothed value lags too far behind the real delay experienced if it is started too early. We set p_tresh to $0.25 * q_i$.

In order to know when to restart the estimator, it must be determined when the queue has been empty for long enough. Simply restarting each time the queue becomes empty would result in unstable behavior, especially if the incoming traffic is bursty. Thus, we define a variable called $cycle_i$ for each class, which indicates when the estimator will be restarted:

$$cycle_i = \frac{abs\_factor_i * q_i * s_i}{C}, \tag{8.4}$$

where s_i is the mean packet size of the class, C is the link capacity and abs_factor_i is an absorption factor. When the absorption factor is 1, $cycle_i$ tells how long it would take to serve the queue of this class if it was full to the link's capacity.

The queue length of class i is denoted by $qlen_i$, the time when class i goes idle by q_idle_i. Then, EWMA-roperates as follows in traffic class i:

```
**Initialization:**
$lowpass\_delay_i = 0.0, p\_samples_i = 0.0, sample\_sum_i = 0.0, qlen_i = 0, q\_idle_i = 0.0$
**Upon each packet departure:**
**if** $p\_samples_i < p\_tresh_i$ **then**
    $p\_samples_i + = 1$
    $sample\_sum_i + = d_i$
    $lowpass\_delay_i = sample\_sum_i / p\_samples_i$
**else**
    $lowpass\_delay_i = \gamma_i * d_i + (1 - \gamma_i) * lowpass\_delay_i$
**end if**
**if** $qlen_i == 0$ **then**
    $q\_idle_i = now$
**end if**
**Upon each packet arrival:**
**if** $q_i == 0$ **then**
    $idle\_period = now - q\_idle_i$
**end if**
**if** $idle\_period \geq cycle_i$ **then**
    $lowpass\_delay_i = 0.0, p\_samples_i = 0.0, sample\_sum_i = 0.0$
**end if**
```

8.3 EWMA estimator based on the proportional error of the estimate (EWMA-pe)

The EWMA estimator with restart provides an updated estimate for the delay when the queue becomes active after an idle period. However, the γ_i and g_i parameters that determine the timescale of the algorithm are still both static. This means that the estimator of each class can be tuned to be either agile or stable, but not both. At some point in time the traffic may be bursty and smooth at other points, even if it is assumed that there is only one traffic type in each class. Therefore, a single filter may not be suitable for the estimation, even if the parameter selection is well argued.

One alternative is to also change the memory of the estimator, γ_i (determined by Eq. 8.1) packet per packet based on how much the predicted queuing delay deviates from the real queuing delay value. The memory of the estimator is adapted as follows:

$$\bar{d_i}(m) = (n * \gamma_i d_i(m) + (1 - n * \gamma_i)\bar{d_i}(m-1)), \tag{8.5}$$

where $0 \leq \gamma_i \leq 1$ is the base weight of class i and n is a multiplier for the base weight. The idea is that the base weight $(n = 1)$ determines the longest possible memory for the estimator. The base weight estimator is suitable when there are only small changes in the traffic process and the system is stable. However, if the traffic process is more variable, the value of n is increased and, as a result, the estimator becomes more aggressive. The value of n is determined by observing the proportional error of the estimated average value $\bar{d_i}(m)$, compared to the actual measured delay value $d_i(m)$.

$$n = \begin{cases} 7, & \text{if } 0.4\bar{d_i}(m) > d_i(m) > 1.6\bar{d_i}(m); \\ 6, & \text{if } 0.4\bar{d_i}(m) \leq d_i(m) \leq 1.6\bar{d_i}(m); \\ 5, & \text{if } 0.5\bar{d_i}(m) \leq d_i(m) \leq 1.5\bar{d_i}(m); \\ 4, & \text{if } 0.6\bar{d_i}(m) \leq d_i(m) \leq 1.4\bar{d_i}(m); \\ 3, & \text{if } 0.7\bar{d_i}(m) \leq d_i(m) \leq 1.3\bar{d_i}(m); \\ 2, & \text{if } 0.8\bar{d_i}(m) \leq d_i(m) \leq 1.2\bar{d_i}(m); \\ 1, & \text{if } 0.9\bar{d_i}(m) \leq d_i(m) \leq 1.1\bar{d_i}(m). \end{cases} \tag{8.6}$$

The selection of the regions for different values of n in Eq. 8.6 determines to what extent the estimator reacts to both small and large, sudden changes.

8.4 Simulations

We implemented all four delay estimators, along with the DBHPD algorithm, in the ns2-simulator. We then tested each estimator with three traffic mixes: pure CBR-traffic, pure Pareto-ON-OFF traffic and mixed traffic from several real applications. Different traffic mixes are used in the evaluation, as the performance of the estimators depends largely on the characteristics of incoming traffic. We want to ensure that our results are applicable to more than one particular traffic type.

8.4.1 General simulation parameters

The topology used in the simulations of traffic mixes 1 and 2 is depicted in Figure 8.2. The topology used for traffic mix 3 is the same, except that each client and server has a separate access link. We have kept the topology simple, since the aim is not to collect end-to-end performance results but to investigate the queuing delay and delay estimate time-series in the bottleneck link. For this purpose, a complicated topology would add little value.

Table 8.1 shows the parameters assigned to the scheduler in the bottleneck link. The delay bound for class 0 is 5 ms and the target ratio for delays between consecutive classes is 4. The parameters s and $absorption\_factor$ are only used in the EWMA estimator with restart. All the other parameters of the different estimators can be directly derived from the parameters presented in Table 8.1.

The theoretical total offered load in the bottleneck link is 1.0 with traffic mix 2 and 0.8

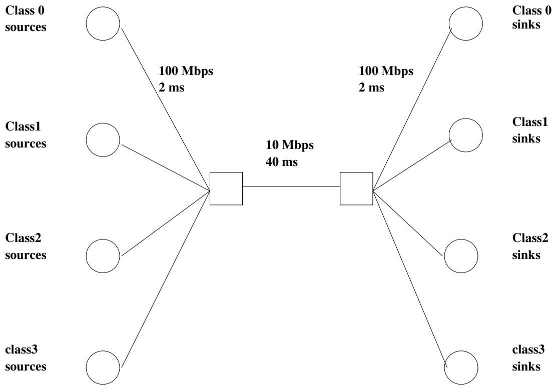


Figure 8.2: The topology with traffic mixes 1 and 2.

Table 8.1: Scheduler parameters

Class	$d_{max}(s)$	δ	**q (packets)**	**s (bytes)**	**absorptionfactor**
0	0.005	0.015625	15	200	1.0
1	-	0.0625	15	750	1.0
2	-	0.25	100	1000	1.0
3	-	1.0	100	1000	1.0

with traffic mix 3. The reason for using a smaller total load with traffic mix 3 is that the TCP retransmissions will increase the theoretical load. With traffic mix 1, the load shares of the classes change over time. The packet sizes for traffic mixes 1 and 2 are as follows: (class 0: 200 bytes, class 1: 750 bytes, class 2: 1000 bytes, class 3: 1000 bytes). With traffic mix 3, the packet sizes are determined by the applications.

8.4.2 Traffic mix 1

Traffic mix 1 represents the most simple setup, where the load in each class consists of the traffic sent by a single CBR-source. Thus, the incoming stream during the active periods are deterministic and easy to predict. We alternated the load level of each source periodically, so that at one point in time the source sent traffic at a high, constant speed, while at another the source was idle. We wanted to see how the estimators reacted during the transition periods, when the load level was changed dramatically. The total load, along with the loads of the individual sources, are shown in Figure 8.3 as a function of the simulation time. The total simulation time is only 160s in this scenario, since the traffic process is fully predictable.

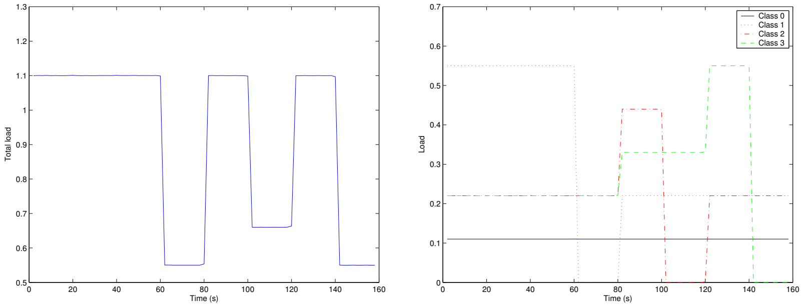


Figure 8.3: Total load and class loads with traffic mix 1

8.4.3 Traffic mix 2

Traffic mix 2 consists of 15 multiplexed Pareto-ON-OFF sources for each traffic class. The parameterization of the sources is shown in Table 8.2. It has been shown that multiplexing several Pareto-ON-OFF sources results in self-similarity, which is a fundamental characteristic of Internet traffic. Predicting self-similar traffic is considerably more difficult than predicting CBR traffic, since self-similar traffic is correlated over several time scales. The traffic shares of each class are set as follows: (class 0: 0.1, class 1: 0.2, class 2: 0.4, class 3: 0.3). The simulation time with traffic mix 2 is 2000 s.

Table 8.2: Parameters for the Pareto sources

class	on_time (s)	off_time (s)	shape_on	shape_off
0	1.0	1.0	1.2	1.2
1	0.4	0.4	1.5	1.5
2	0.6	0.6	1.4	1.4
3	0.8	0.8	1.3	1.3

8.4.4 Traffic mix 3

With traffic mix 3 we use 'real' applications to produce a realistic mix of Internet traffic. The mix consists of five different traffic types: FTP, HTTP, Video, VoIP and control traffic (small and large control messages). FTP transfers are also used to represent P2P traffic, which is becoming popular on the Internet. Control and VoIP traffic are mapped to class 0 while Video is mapped to class 1, HTTP to class 2 and FTP to class 3. The generation of HTTP-traffic is based on the webcache-model implemented in ns2. In this model it is assumed that a HTTP session consists of a number of page requests, each possibly containing several objects. We also used the webcache-model for FTP-traffic, although there is no reading time between page requests with FTP, and pages always contain one object. Video traffic generation is based on a real trace of mpeg4 coded BBC news, from which the traffic stream has been regenerated by using a Transform Expand Sample (TES) process. In the simulations the traffic flows in both directions: there is one HTTP client and server (constantly creating new page requests), one FTP client and server, two control traffic sources and sinks, and 10 video and VoIP sources and sinks on both sides of the network. The simulation time with traffic mix 3 is 1200s, and the different applications have the following percentage shares of the traffic: (FTP: 9%, HTTP: 71%, Video: 9%, VoIP: 10%, Control: 1%). This corresponds to the current situation, where the majority of traffic is HTTP. However, it is thought that the amount of P2P traffic will increase dramatically in the future.

8.5 Results

This section shows the simulation results for the different estimators. Snapshots of both the instantaneous queuing delay time-series and the estimated queuing delay time-series are shown, so that the quality of the estimators can be evaluated. It should be noted that the estimated delays in the figures are calculated according to Eq. 5.12, i.e. the effect of both the head of line packet delays and the long term delays is taken into account. This is because the scheduling decision is based on the joint effect of these delays, not only on the filtered delay $\bar{d}_i(m)$. Both the instantaneous and estimated queuing delays are real values; they have not been normalized with δ_i.

8.5.1 Traffic mix 1

Figures 8.4, 8.5, 8.6 and 8.7 depict the instantaneous and estimated queuing delays with different estimators in simulation scenario 1, in which the incoming traffic is pure CBR. Figure 8.4 (a) and Figure 8.5, show that the simple sum estimator leads to false scheduling decisions, particularly during times when a class becomes active after an idle period. Since the sum estimator incorporates an infinite history, the estimate can have a high value even when real queuing delays are nearly zero. This means that a packet is served from a class that has virtually no congestion. Figure 8.4 (b) shows the behavior of the EWMA estimator. For class 1 it leads to a considerably better estimate than the simple sum estimator. However, this is due to the fact that a separate g parameter is selected for each class, defined in Eq. 8.3. For classes 2 and 3, which have small g parameters and thus only give a small amount of weight to the head of line packet delay, the results closely resemble the simple sum estimator. Therefore, the only advantage that the EWMA estimator has over the simple sum estimator is its lower implementation complexity. Figure 8.6 and 8.7 depict the instantaneous and estimated delay values with the EWMA-r and EWMA-pe estimators. It is obvious that both of these estimators lead to accurate predictions and are able to take idle periods into account. Thus, the scheduling decision follows the selected differentiation policy.

8.5.2 Traffic mix 2

The purpose of simulation scenario 1 was to show how the estimators perform when the classes become active after an idle period. However, since the traffic in scenario 1 was CBR, the queuing delays during the active periods remained almost constant. Thus, a good estimator always matches the actual queuing delays very closely. In scenario 2, the traffic mix consists of Pareto-ON-OFF sources that produce variable rate traffic over several timescales. With this kind of traffic mix the responsiveness of the estimators to sudden bursts and longer term variations can be explored. Ideally, the estimator should follow the changes in queuing delays, but not react too aggressively to sudden peaks. This scenario studied the performance of the most promising estimators, the EWMA-r and the EWMA-pe.

Figures 8.8, 8.9 and 8.10 show the behavior of the EWMA-r estimator on two timescales: 40 seconds (approximately 67000 packet departures, assuming a mean packet size of 750 bytes) and 5 seconds (approximately 8300 packet departures). On the 40 second timescale the estimator seems to follow the instantaneous queue length quite closely for all traffic classes. However, on the 5 second timescale it can be observed that the estimator smoothes the values of classes 1 and 2 quite roughly at some points and ignores the changes in the queuing delays. In class 1 the estimated delay matches the real queuing delay, even on shorter timescales, because of the high value for the g parameter (0.75).

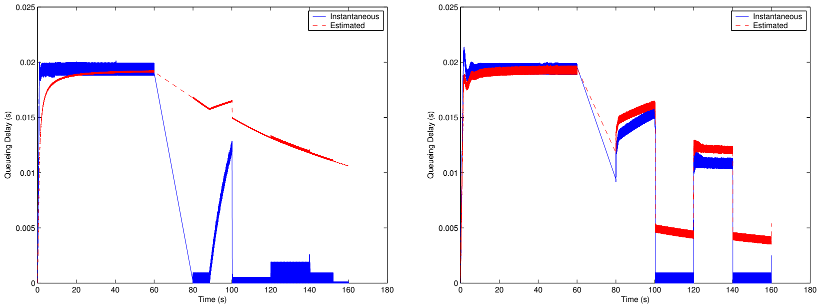


Figure 8.4: Instantaneous and estimated delays for class 1 with the simple sum (a) and EWMA (b) estimators

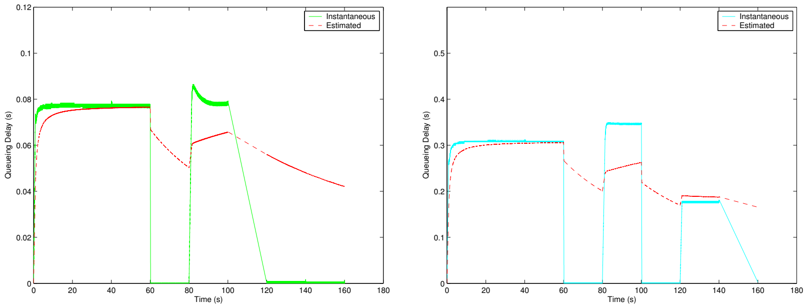


Figure 8.5: Instantaneous and estimated delays for classes 2 and 3 with the simple sum estimator

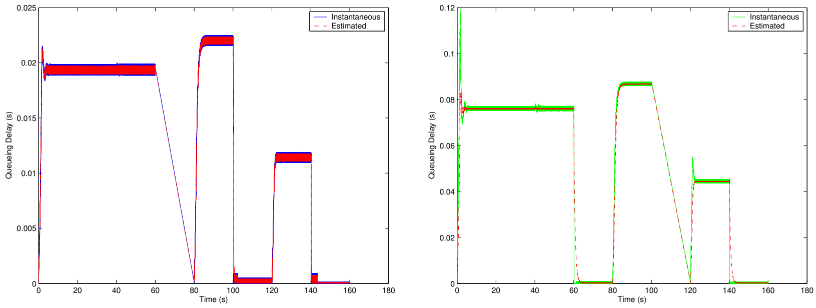


Figure 8.6: Instantaneous and estimated delays for classes 1 and 2 with the EWMA-r estimator

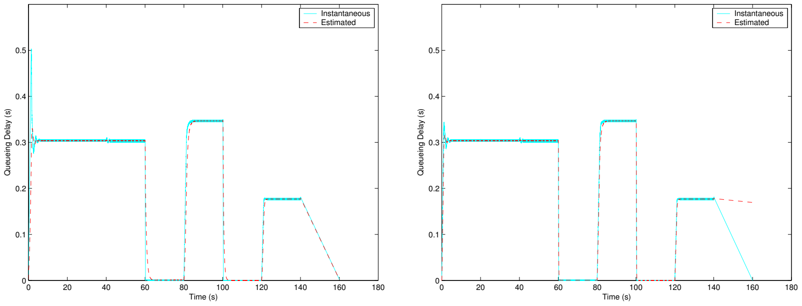


Figure 8.7: Instantaneous and estimated delays for class 3 with the EWMA-r (a) and EWMA-pe (b) estimators

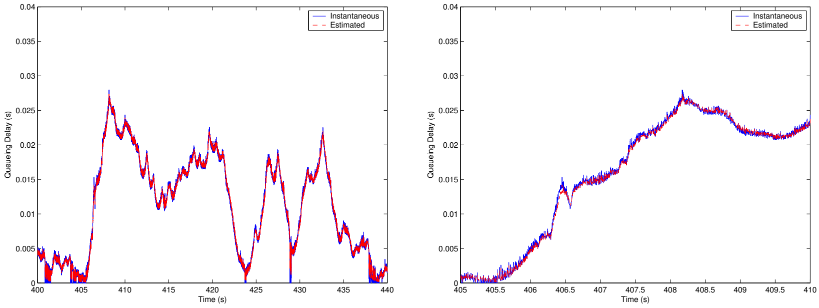


Figure 8.8: Instantaneous and estimated delays for class 1 with the EWMA-r estimator on two timescales

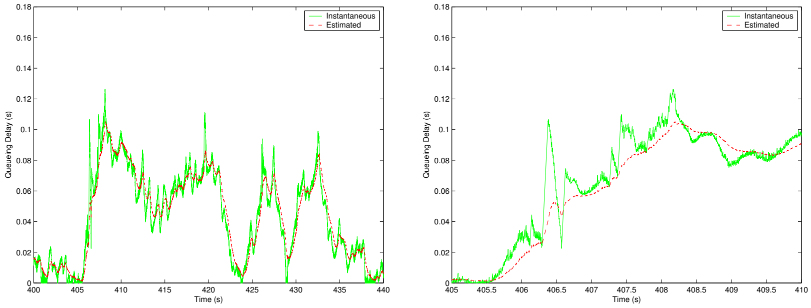


Figure 8.9: Instantaneous and estimated delays for class 2 with the EWMA-r estimator on two timescales

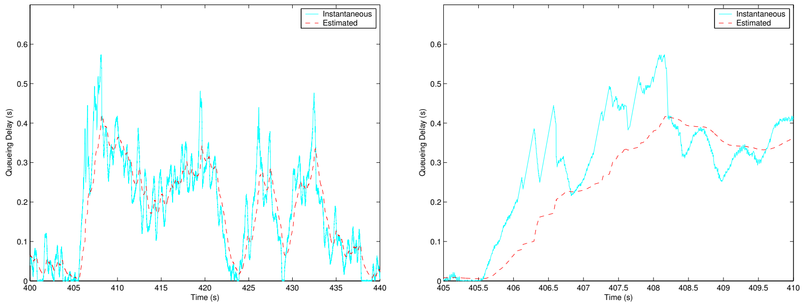


Figure 8.10: Instantaneous and estimated delays for class 3 with the EWMA-r estimator on two timescales

Figures 8.11, 8.12 and 8.13 show the corresponding results for the EWMA-pe estimator. This estimator follows the queuing delay changes more carefully, even on the shorter timescale, for each class without being too aggressive. This is because the weighting factor of the estimator is adaptively increased if there are large changes to the queuing delay.

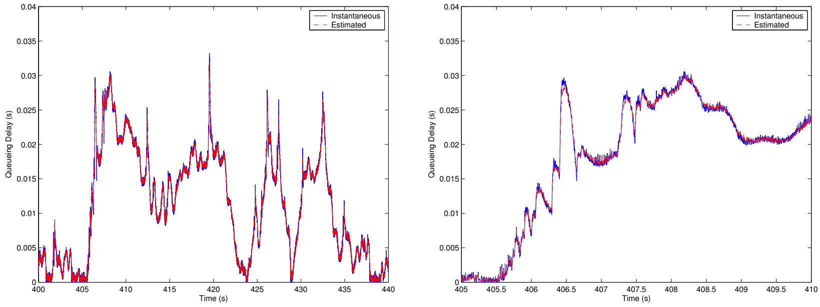


Figure 8.11: Instantaneous and estimated delays for class 1 with the EWMA-pe estimator on two timescales

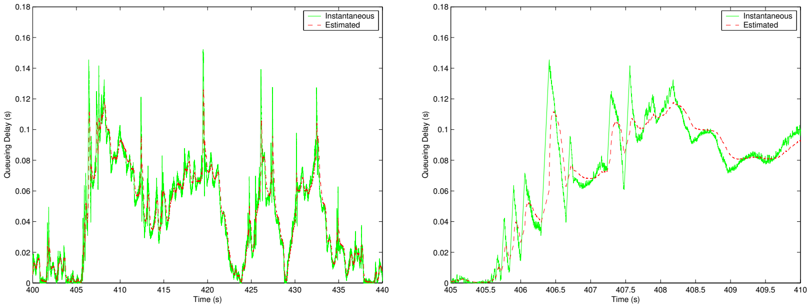


Figure 8.12: Instantaneous and estimated delays for class 2 with the EWMA-pe estimator on two timescales

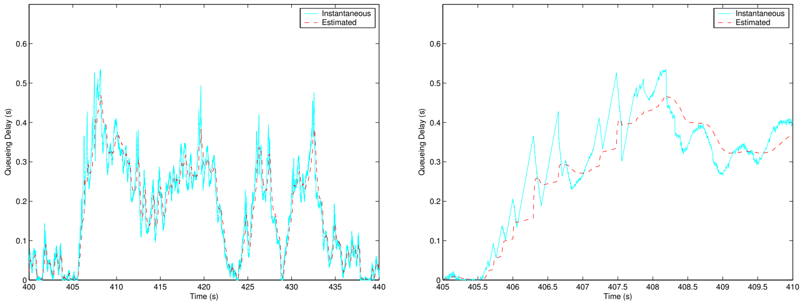


Figure 8.13: Instantaneous and estimated delays for class 3 with the EWMA-pe estimator on two timescales

8.5.3 Traffic mix 3

The simulation results with traffic mix 2 showed how the estimators respond to traffic bursts and longer term variations. However, it is still important to examine how the estimators behave with real traffic, where a substantial part of the total traffic is carried on top of the TCP-protocol.

Figures 8.14, 8.15 and 8.16 depict the instantaneous and estimated delay time-series in this scenario for classes 2 and 3 with the simple sum, EWMA-r and EWMA-pe estimators. The figures for class 1 are not shown, since in the EWMA-r and EWMA-pe estimators the instantaneous and estimated delay time-series are very close to each other due to the high value of the g parameter. The time-scale used is relatively long (60 seconds). Again, the simple sum estimator is either considerably above or below the instantaneous queuing delay value. The effects of false scheduling decisions can especially be seen in class 3 (FTP traffic): the delays in class 3 remain lower with a simple sum estimator than with the EWMA-r and EWMA-pe estimators, since the estimated value is much larger than it should be. Therefore, the scheduling algorithm decides that a packet must be selected from class 3 because it has long delays. This wrong conclusion also reflects on the other classes, which correspondingly suffer somewhat longer delays during these periods. However, it should be noted that the absolute delay values returned by the different estimators cannot be directly compared with each other, since they result in different packet loss patterns and, as a result, different queuing delays. In conclusion, the EWMA-pe estimator also gives the best estimation result with this traffic mix, in the sense that it is agile but still sufficiently stable.

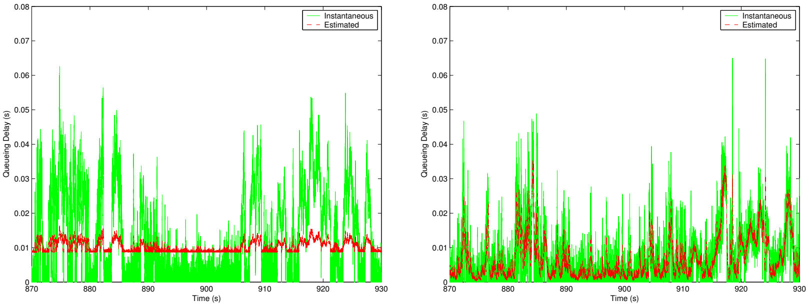


Figure 8.14: Instantaneous and estimated delays for class 2 with the sum (a) and EWMA-r (b) estimators

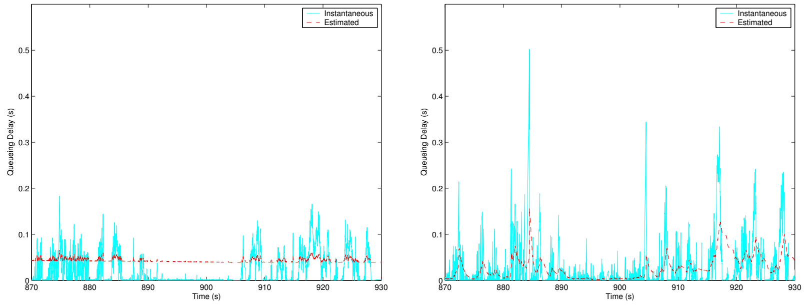


Figure 8.15: Instantaneous and estimated delays for class 3 with the sum (a) and EWMA-r (b) estimators

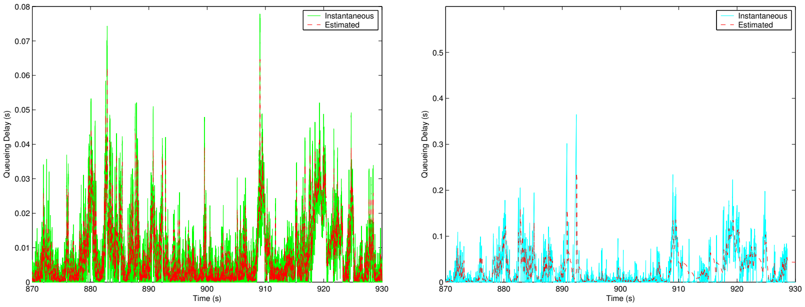


Figure 8.16: Instantaneous and estimated delays for classes 2 and 3 with the EWMA-pe estimator

9 Measuring Adaptive Scheduling in fixed IP Networks

So far we have only evaluated the DBHPD algorithm with simulations that use traditional performance metrics such as throughput, delay and packet loss as their criteria. However, using only traditional performance metrics as evaluation criteria may lead to false conclusions, since an algorithm that behaves well in simulations may not return the same results in the real world. In fact, it may be impossible to implement the algorithm in a real-time environment if it uses overly complex computation and estimation procedures. Thus, an algorithm can only be considered valuable after its implementation complexity, as well as its resulting performance in a real environment has been assessed.

The DBHPD algorithm has not been implemented in a prototype or commercial router before. This thesis shows the first working implementation of the DBHPD algorithm in a FreeBSD-based ALTQ prototype router and compares it with the Class Based Queueing (CBQ) hierarchical bandwidth sharing algorithm [120]. In simulation evaluations, DBHPD was compared with the static Deficit Round Robin (DRR) algorithm that statically divides capacity between service classes. From the simulation results, it was obvious that DBHPD performed better when compared to DRR. Thus, it is more relevant to compare the DBHPD algorithm to an algorithm that uses some kind of adaptation in its scheduling decisions. CBQ is a pseudo-adaptive algorithm that uses DRR as a general scheduler and heuristic rules for borrowing capacity between service classes. This chapter provides a short description of ALTQ and describes the implementation of the DBHPD and CBQ algorithms, as well as the measurement topology and the general parameters used in the measurements. Finally, this chapter presents the results of the measurements.

9.1 Implementation and measurements of the DBHPD and CBQ algorithms

9.1.1 Goals of the implementation and measurement study

The objectives of the implementation and measurement study are to describe the first working implementation of the DBHPD algorithm, discuss the difficulties encountered during the implementation and compare the performance of the DBHPD algorithm with the CBQ algorithm through the use of measurements.

9.1.2 Implementation of the measured scheduling algorithms

We used the Alternated Queuing (ALTQ) package [36] to implement the DBHPD algorithm. ALTQ is a general Quality of Service framework for BSD systems. It contains basic scheduling, queue management and rate control functionalities which can be assembled together in various ways to realize different types of QoS networks. ALTQ currently supports the following scheduling algorithms: Weighted Fair Queuing (WFQ) [125], Class Based Queuing (CBQ) [56], Hierarchical Fair Service Curve (HFSC) [147] and Joint Buffer Management and Scheduling (JoBS) [39]. Of these algorithms, CBQ is the most well-known and most widely used in measurement papers. CBQ can be considered as an adaptive scheduling algorithm in the sense that it provides heuristic rules for borrowing capacity if a class is running out of resources. Therefore, it is reasonable to use CBQ as a baseline when evaluating the performance of the DBHPD algorithm. In our work, we left the existing CBQ scheduling block in ALTQ unchanged and modified the priority queuing to implement DBHPD scheduling.

9.1.3 Implementation of the CBQ algorithm

We utilized the existing implementation in the ALTQ-package for CBQ. In this implementation, the general scheduler is either the Weighted Round Robin (WRR) or the Packet Round Robin (PRR). When looking at the implementation of these schedulers in greater detail, WRR also contains a deficit counter and, therefore is actually close to being a Deficit Round Robin (DRR) [141]. We used WRR, as it allows greater capacity allotment flexibility and accuracy. Possible queue management algorithms in ALTQ are tail-drop, RED and RIO. We decided to only use the DropTail algorithm, since we wanted to see the effect of pure scheduling. However, we realize that queue management increases the performance of TCP traffic and, as a result, the effect of different algorithms should be evaluated in the future.

Actual implementations of the CBQ borrowing mechanism contain a lot of approximations, which degrade the accuracy with which CBQ operates and also cause performance penalties. ALTQ contains two modes for borrowing:

1. **Borrow:** A class is allowed to borrow if its ancestor class is underlimit. This means that the system is only work-conserving among sub-trees that share common resources.
2. **Efficient:** A class is allowed to borrow even if its ancestor class is not underlimit. Such a situation may occur if borrowing has taken place to the extent that the system is in suspension (e.g. no packets are allowed to be sent if the timer has not expired), even though the link has free resources. The efficient mode takes a packet from the first scanned queue and passes it onto the link. This makes the whole system work-conserving, but can cause unexpected behaviour for classes with a small capacity fraction.

Details of the CBQ implementation can be found in [37]. We have chosen to only use simple borrowing in our measurements in order to achieve a more predictable performance and more controllable top-level borrowing. The top-level has been defined so that real-time and non-real-time leaf classes are only allowed to borrow from their own parent. Our CBQ link-sharing hierarchy is presented in Figure 9.1.

9.1.4 Implementation of the delay-bounded HPD algorithm

The delay-bounded HPD algorithm has not been implemented in a real or prototype router before. We have implemented the algorithm on the basis of the priority queue structure in ALTQ (ALTQPRIQ). The DBHPD algorithm relies heavily on the measurement of time. Time is a difficult concept for operating system kernels, as it deals with floating point numbers. The majority of operating systems do not allow floating point operations in the kernel space, as they lead to diminished performance. This can be worked around by using two interconnected integers; in case of time, one for the full seconds and the other for microseconds. The manipulation of two interconnected integer values always takes some extra time when compared to a single integer. In our implementation, the head-of-line packet delay is obtained by utilizing the kernel *gettime* routine twice: firstly when the packet is enqueued in the queue and secondly when the scheduling decision is made by polling the head-of-line packets. Usually, however, this happens more than twice, as there are several packets competing for the link resource and the queuing time calculation has to be performed each time a scheduling decision is made.

In the differentiation model, the long term average delay $\bar{d}_i(m)$ was calculated using the simple sum estimator. With simple sum estimation, the system uses an infinite memory to count the average value. Infinite memory in a real router is an impossible concept, as the actual kernel is based on either 32 bit or 64 bit registers. This would eventually lead to a register overflow. Therefore the long term delay for the implementation is obtained using an Exponential Weighted Moving Average (EWMA) filter, according to Eq. 6.21. Now calculation to infinity is not required and the amount of history can be determined by selecting the γ parameter. The weighting coefficients are normally between zero and one, but floating point arithmetics in kernel space should be avoided. Therefore, all operations are scaled by the magnitude of the number range of the coefficients. DBHPD requires a lot of time calculations, therefore we streamlined the calculation to contain only binary shifting. This means that weighting coefficients and differentiation parameters are powers of two, allowing multiplication by shifting the actual delay value by the power of a coefficient. Operating with integers increases efficiency but also reduces the granularity of parameters. However, the deviation from ideal operation is still acceptable. For future experimentation, we have developed even more sophisticated estimators for the DBHPD algorithm in [21] but these estimators have not yet been implemented in ALTQ.

9.2 Measurement setup

The measurement topology used in our test is shown in Figure 9.2. The topology consists of a router with connections into two individual networks:

1. **Server network:** QuickTime Video server, Web server, FTP server and VoIP clients

2. **Client network:** QuickTime Video clients, Web clients, FTP clients and VoIP clients

The interconnecting networks were 100Mbps Ethernet VLANs implemented with a single ethernet switch. The "in between" router was a PC running a FreeBSD-4.5 operating system on a 433MHz Intel Celeron architecture, with 128MB of memory. The FreeBSD kernel was compiled to have a 1kHz clock resolution in order to obtain sufficiently high granularity for internal interrupt handling.

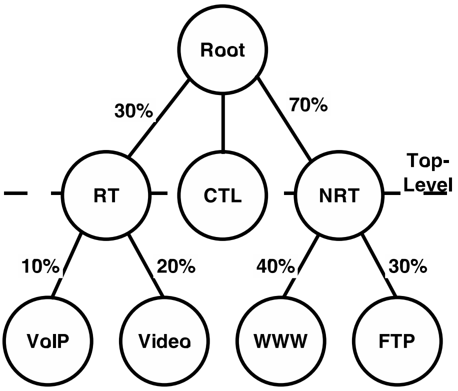


Figure 9.1: CBQ link-sharing hierarchy

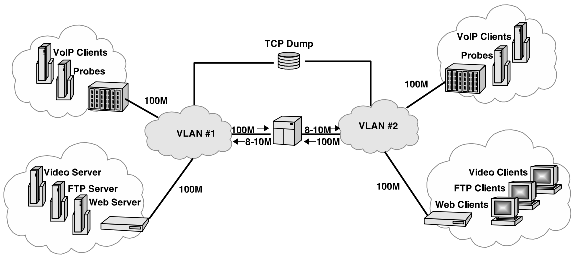


Figure 9.2: Measurement network topology.

We used four different applications for the measurements: FTP, HTTP, QuickTime Video Streaming and VoIP. FTP, HTTP and QuickTime Video Streaming traffic was produced by Spirent's Avalanche (client side) / Reflector (server side) test appliances. VoIP traffic was emulated with Spirent's AX/4000 broadband test system by defining a suitable traffic generation pattern.

Three user instances were defined in Avalanche: one for FTP, one for HTTP and one for Video Streaming. The FTP instance requests one file with a size of 1MB, the HTTP instance loads 40 www-pages with a short break after 5 pages, and the Video instance streams a 60s VBR video-clip at an average speed of 200kbps. HTTP version 1.0 was used for www-transfers, meaning that a separate TCP-connection was established for each object within a page. However, the pages used in the measurement only contained a single object. The page sizes were defined from a list with 40 entries, with a mean geometric distribution of 16.5kB. The mean value and distribution represent the actual size distribution of real web-pages containing several objects. The list was repeated for each simulated user from top to bottom.

VoIP calls were created in AX/4000 by defining a burst of 13 packets arriving every 20ms. During the burst, packets are sent at a constant rate of 30Mbps rather than back-to-back. This pattern is an approximation of 13 simultaneous VoIP calls between two ports of AX/4000.

Traffic was mapped into different classes so that VoIP used the first, delay-bounded class, Video used the second class, HTTP used the third class and FTP used the fourth class. The different applications had the following relative traffic share percentages: (FTP: 30%, WWW: 40%, Video: 20%, VoIP: 10%). The desired FTP, HTTP and Video traffic loads was produced by defining how many users/hour avalanche created. In addition, the user generation pattern may be flat, stair-step, bursty or sinusoid in order to create more load fluctuations. We have used a flat pattern for HTTP and Video and a sinusoid pattern for FTP. $\bar{\rho}$ denotes the desired mean load in the output link, f_i denotes the load fraction produced by class i, C the link capacity (bits/s), and the workload (bits) created by a single user instance in class i is given by W. The required number of simusers/hour in class i can be calculated as:

$$N = \frac{\bar{\rho} * f * C}{W} * 3600s \qquad (9.1)$$

Since the operation of the DBHPD algorithm is based on packet delays, we were particularly interested in recording the distribution of delays in each traffic class. This type of information is not provided in Avalanche/Reflector and, as a result, a packet capturing tool such as tcpdump would be required to track the packets generated by FTP and HTTP. However, it is also possible to obtain the delay distributions indirectly by defining a measurement probe for each traffic class. Suitable probes are small packets that are sent in continuous intervals and that have accurate time-stamps. We used this approach, since AX/4000 provides efficient facilities for recording traffic stream delay distributions. Each probe sends a 46 byte packet every 10 ms, resulting in a 50kbps CBR stream (raw IP-packets). The delay distributions of the probes do not correspond exactly to the distributions of the actual application traffic, but the probes provide an approximation that is accurate enough for our purposes. In [40] the accuracy of deterministic sampling techniques was judged to be reasonably good if samples were taken with a relatively high frequency. In the case of inter-arrival time distributions, every 128th packet was seen to produce an error of less than 1% error of the original distribution. Time driven sampling was not judged to be as good as packet driven sampling, as its probe distribution is not uniform during the burst periods. In our case we use time driven sampling, but a high load level and and a relatively small distance between probes (every 50th packet on the link on average and every 200th packet of the class in the worst case scenario) lowers the error caused by time driven operation. Link utilizations, application throughputs and packet losses are derived from tcpdump files captured from both sides of the router, as shown in Figure 9.2.

9.3 Provisioning parameters and measurement scenarios

CBQ was provisioned by simply allocating each class a bandwidth share corresponding to the load fraction of that class. The bandwidth fractions of the classes are shown in Figure 9.1. With the delay-bounded HPD, the delay-bound (d_{max}) in Eq. 5.9 was set to 5ms, the safety margin (t_{safe}) to 0, the long-range delay EWMA coefficient (γ) in Eq. 6.21 to $1/2^{10}$ (in practice, this corresponds to a filtering coefficient of 0.001), the long-term short-term weighting coefficient (g) in Eq. 5.12 to 0.875 and the delay ratios to 4. Queue lengths in both algorithms were set to 15, 15, 80 and 200 packets. The settings for the delay-bounded HPD correspond closely to the parameters used in our previous simulation studies, e.g. [20].

Both algorithms were tested in underload (90$\%$ mean theoretical load), overload (100$\%$ mean theoretical load) and heavy overload (110$\%$ mean theoretical load) situations. The load level of the system was varied by changing the emulated link speed within the scheduling units of router interfaces. Theoretically, there should be slightly more than 9Mbps of offered traffic in the system. Therefore, in the measurements with a 90$\%$ load level, the link speed was set to 10Mbps, whereas it was set to 8Mbps for the 110$\%$ load level measurements. CBQ was tested with both borrowing and no borrowing. When borrowing is on, CBQ uses heuristics to distribute excess resources among the classes, up to the top sharing level, which is the parent class of the leaf classes. This parent is the intermediate class used in i) realtime for VoIP and Video traffic, and in ii) non-realtime for WWW and FTP traffic.

9.4 Measurement results

This section presents the results of the CBQ and DBHPD measurements. Shown here are the link utilizations and cumulative delay distribution functions for each class, as well as the application throughputs and packet losses for both algorithms. When comparing the results of different load levels it should be remembered that the difference in load level was achieved by changing the link capacity. This causes a lower throughput with the same offered traffic. The delays consist of queuing delay, processing delays and propagation delay, of which queuing delay is the dominant component.

9.4.1 Scenario 1: Theoretical link load 90%

Figure 9.3 shows the link utilizations for DBHPD and CBQ with a theoretical link load of 90%. With CBQ, borrowing is used in order to heuristically divide extra capacity. It can be seen from the figure that DBHPD is capable of providing full link capacity for the elastic TCP traffic, while CBQ cannot provide the same level of adaptivity. This is due to the fact that part of the capacity is dedicated to real-time traffic, which does not use its full share of resources.

It should be remembered that the link capacity adjustment performed in the measurements to generate different load levels was based on a token bucket rate control at dequeue events. When the packet buffer in a NIC driver underflows, causing a packet to be sent to the network, an interrupt is generated and the dequeue method is called. The execution of this method is delayed by a time that is calculated based on the link capacity assignment and the size of the served packet. Software-based implementation of the token bucket rate control is prone to small errors caused by the low resolution of the kernel timer. This error causes the actual link capacity to vary around the correct link capacity (in Figure 9.3 utilization is over 100% at some points, although in theory this should not be possible).

Tables 9.1 and 9.2 show application throughputs and packet losses in this scenario. The throughput for FTP and HTTP was recorded by averaging the throughputs of individual objects. This results in huge variances for HTTP, as some pages are very small (they fit into a single packet) and some are considerably larger. DBHPD provides more throughput

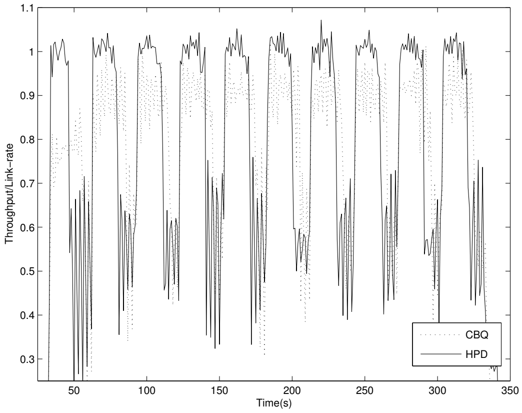


Figure 9.3: Link utilizations with HPD and CBQ scheduling (Load 90%).

Table 9.1: Statistics for HPD with delay bound (Load 90%)

	Throughput kbps		Loss %	
Traffic	**Mean**	**Dev**	**Mean**	**Dev**
FTP	2451	1524	0	0.03
WWW	4642	32588	0	0
Video	209	0.4	0.2	0.2
VoIP	-	-	0	-

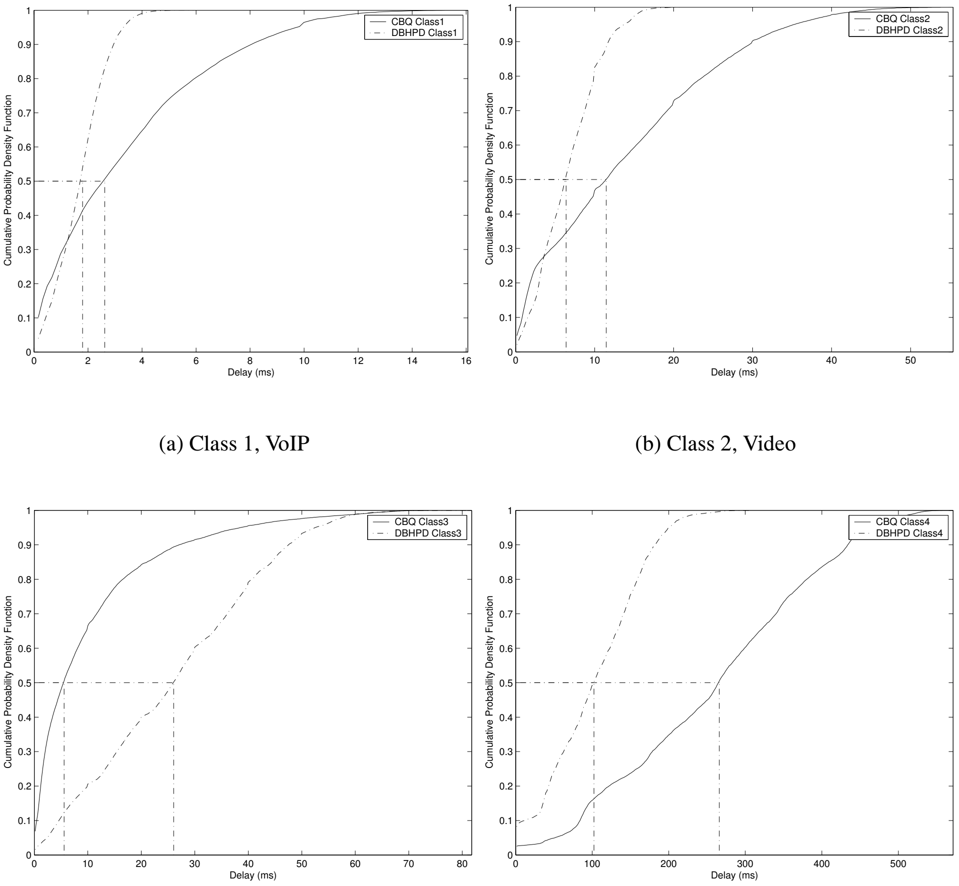
Table 9.2: Statistics for CBQ with borrow (Load 90%)

	Throughput kbps		Loss %	
Traffic	**Mean**	**Dev**	**Mean**	**Dev**
FTP	1041	687	0	0
WWW	5359	17005	0	0
Video	208	0.8	0.8	0.5
VoIP	-	-	0.1	-

for FTP, due to its capability to adjust to time varying load levels. On the other hand, CBQ allocates considerably more resources to HTTP, even though FTP should use a larger share of the resources. It should be noted that, on this occasion, the theoretical provisioning goal was 4Mbps for HTTP and 3Mbps for FTP. It can be seen that DBHPD is clearly closer to the goal of a 25% difference in capacity allocation - even though DBHPD operates on delay, not on capacity. CBQ's deviation from the target capacity is due to the nature of the implementation. Borrowing capacity in a packet-per-packet manner is a computationally intensive task. Therefore, approximations are made in the calculations by only allowing borrowing in certain time-windows. This easily leads to the starvation of low capacity classes. Video throughput is practically the same with both algorithms. VoIP throughputs are not shown since VoIP traffic was produced with AX/4000, which cannot be analyzed with tcpdump. However, delay distributions and packet losses reflect voice quality accurately enough. Figure 9.4 presents the delays for CBQ and DBHPD in this scenario. Previous notions on borrowing become obvious when the delay distributions and medians of these algorithms are compared. When CBQ is used, HTTP traffic in class 3 receives most of the network capacity, which can also be seen in the very short delays. CBQ and DBHPD do not create a large difference between real-time traffic classes, but in general DBHPD provides a much smaller delay variance. This is important for real-time communications, which require play-out compensation buffers that are dimensioned relative to the delay variance. From the delay distributions we can conclude that DBHPD provides a much more reasonable service than CBQ, because CBQ offers extra capacity to a class with the highest link-share, regardless of borrowing. We believe that the implementation of CBQ algorithm is so complex that the accuracy of its operation is far from its theoretical operation.

9.4.2 Scenario 2: Theoretical link load 100%

Figure 9.5 shows the link utilizations for DBHPD and CBQ with a theoretical link load of 100%. In CBQ, borrowing is used in order to heuristically divide extra capacity. It can



(a) Class 1, VoIP

(b) Class 2, Video

(c) Class 3, WWW

(d) Class 4, FTP

Figure 9.4: Delay distributions for CBQ and DBHPD (Load 90%)

Table 9.3: Statistics for HPD with delay bound (Load 100%)

	Throughput kbps		Loss %	
Traffic	**Mean**	**Dev**	**Mean**	**Dev**
FTP	1736	1826	0	0.02
WWW	2723	16142	0	0
Video	208	1	0.7	0.5
VoIP	-	-	0	-

Table 9.4: Statistics for CBQ with borrow (Load 100%)

	Throughput kbps		Loss %	
Traffic	**Mean**	**Dev**	**Mean**	**Dev**
FTP	852	580	0.02	0.06
WWW	5023	32896	0	0
Video	198	5	6.5	3.2
VoIP	-	-	6.7	-

be observed that both algorithms result in approximately the same link utilization, with DBHPD providing a slightly more steady utilization.

Tables 9.3 and 9.4 show application throughputs and packet losses in this scenario. DBHPD provides more throughput for FTP and Video, while CBQ again allocates more resources to HTTP. The packet losses are clearly lower with real-time traffic using DBHPD. This is due to the fact that, in the 100% scenario, the actual load is a little over 100%, mainly caused by video traffic sending slightly more than its allocated capacity. With CBQ this leads to packet losses in a real-time traffic subclass. On the other hand, DBHPD is capable of adapting to the packet loss using capacity from elastic traffic.

Figure 9.6 present the delays for CBQ and DBHPD in this scenario. By comparing delay distributions and medians of these algorithms, it is obvious that DBHBD is able to provide delay differentiation in accordance with the specified provisioning parameters. The delay

Table 9.5: Statistics for HPD with delay bound (Load 110%)

	Throughput kbps		**Loss %**	
Traffic	**Mean**	**Dev**	**Mean**	**Dev**
FTP	953	728	0.03	0.1
WWW	602	1360	0.02	1
Video	205	3.4	2.4	1.8
VoIP	-	-	0	-

bound in the highest class is not violated, the ordering of the rest of the classes is correct and the specified delay ratio between classes is preserved. However, CBQ distributes delays in an irrational way between the classes and furthermore, class ordering is violated between class 2 and class 3, since class 3 experiences lower delays.

The shape of the distributions show that the delay distribution of DBHPD is quite linear, while the distribution of CBQ is more concave. Another important observation is that the distributions of CBQ are considerably wider, especially for class 1 and class 4. This implies that DBHPD is able to provide a more predictable service in terms of delays.

9.4.3 Scenario 3: Theoretical link load 110%

Figure 9.7 shows the link utilizations for DBHPD and CBQ with a theoretical link load of 110%. Since the system is pathologically congested, both scheduling algorithms provide operations that are strictly bounded by their link-sharing rules. With CBQ, capacity is allocated to each leaf class and with DBHPD, operation is mainly based on proportional delay constraints.

Tables 9.5 and 9.6 show the application throughputs and packet losses in this scenario. As can be expected, the packet losses are considerably higher and throughput is lower than in the 100% load case.

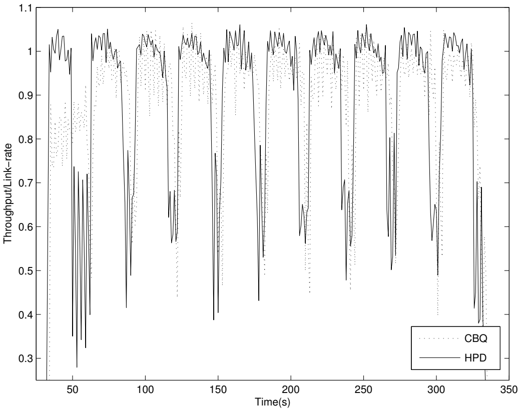
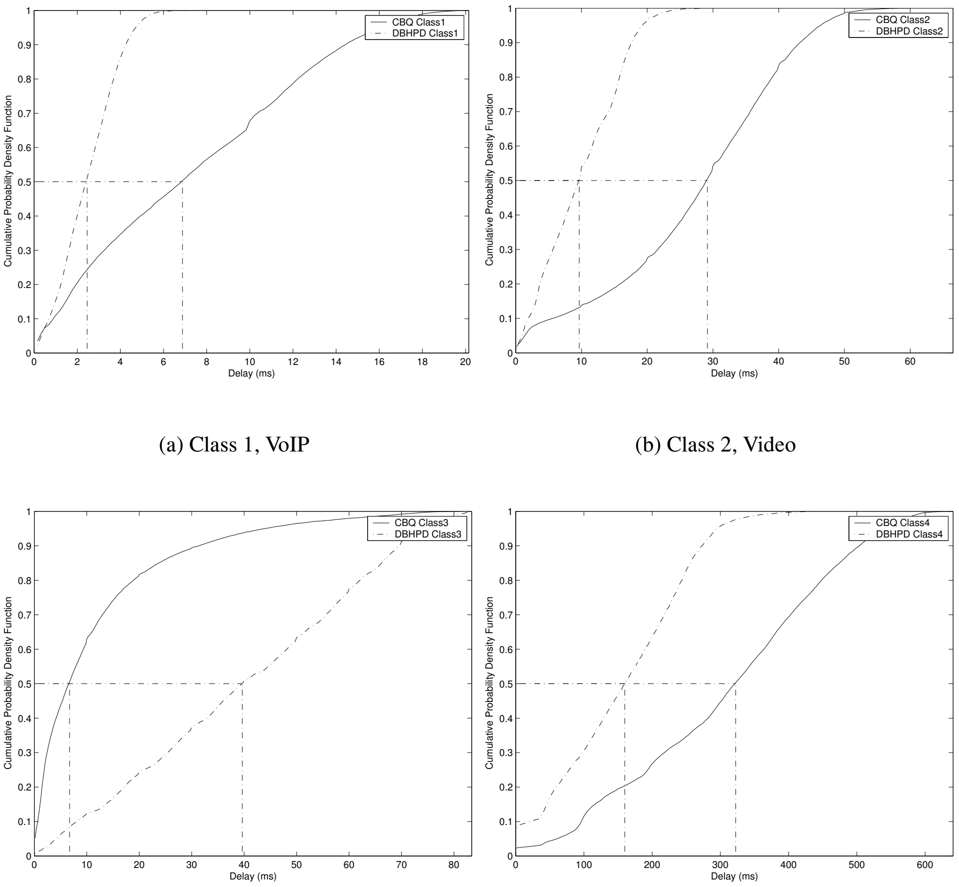


Figure 9.5: Link utilizations with HPD and CBQ scheduling (Load 100%).

Table 9.6: Statistics for CBQ with borrow (Load 110%)

	Throughput kbps		**Loss %**	
Traffic	**Mean**	**Dev**	**Mean**	**Dev**
FTP	652	426	0.07	0.1
WWW	3527	13567	0	0.1
Video	182	16	16.7	9.8
VoIP	-	-	18.4	-



(a) Class 1, VoIP

(b) Class 2, Video

(c) Class 3, WWW

(d) Class 4, FTP

Figure 9.6: Delay distributions for CBQ and DBHPD (Load 100%)

Figures 9.8 show the delay distributions for DBHPD and CBQ with borrow when the theoretical offered load stands at 110%. This corresponds to the heavy overload case. It can be seen that the shapes of the distributions remain very similar but the mean and maximum delays increase considerably, since the system is in a pathological congestion situation. In addition, the gap between the DBHPD and CBQ delay medians is significantly larger compared to previous scenarios.

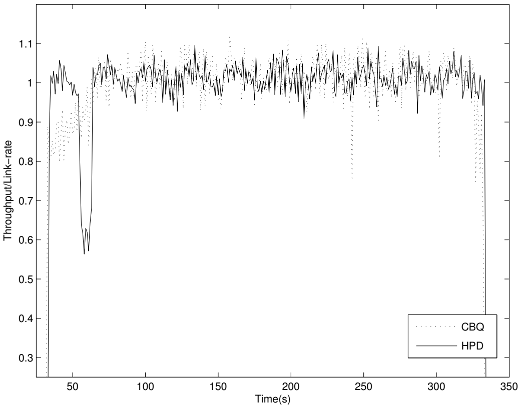
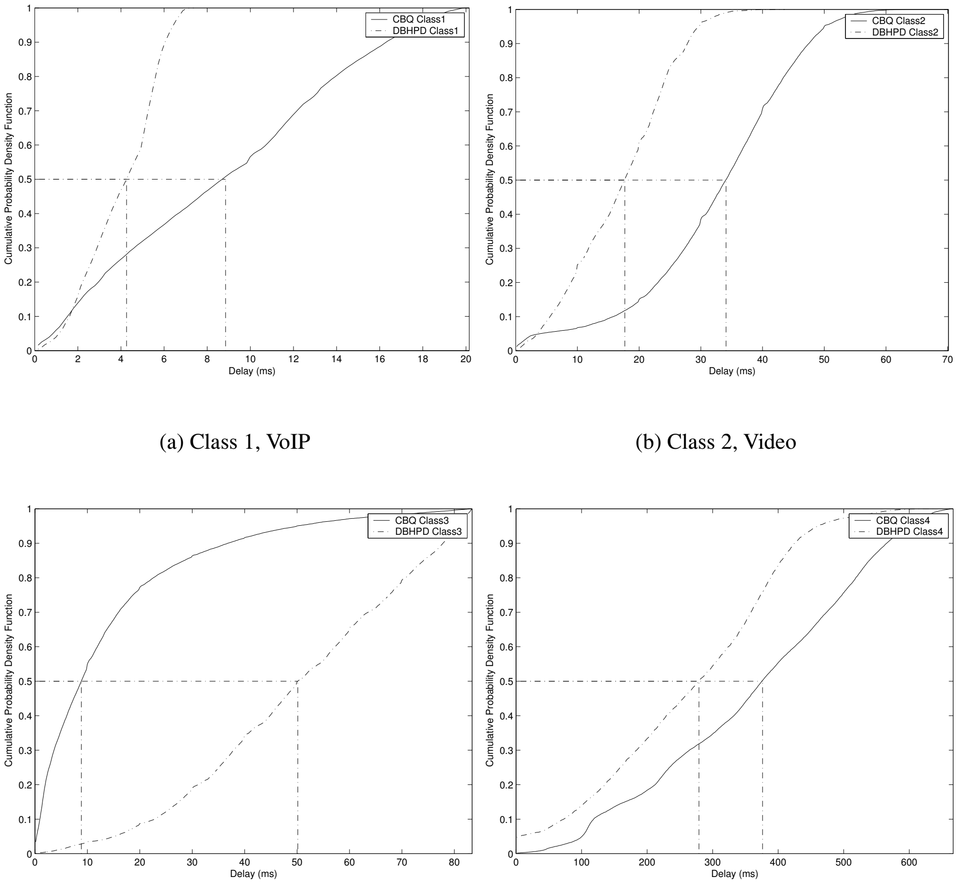


Figure 9.7: Link utilizations with HPD and CBQ scheduling (Load 110%).



(a) Class 1, VoIP

(b) Class 2, Video

(c) Class 3, WWW

(d) Class 4, FTP

Figure 9.8: Delay distributions for CBQ and DBHPD (Load 110%)

10 Comparison of Measurement and Simulation Results

The previous chapter presented the results of the DBHPD and CBQ algorithm measurements. The goal of this chapter is to show results of simulations that have been performed in setups that correspond to the measurement scenarios, and to compare how well the simulation results match the measurement results [119]. Another objective is to present kernel profiling for the implementation of both DBHPD and CBQ in terms of achievable throughput, time consumed by enqueue and dequeue operations, as well as the number of function calls used in these operations.

The motivation behind this study stems from the fundamental difference between simulations and real router measurements: in simulations, router operations such as enqueue and dequeue may not consume any time while the effect of packet processing can be considerable in the real world. The processing overheads of router operations have been modeled in some network situations, but all router operations are performed with zero delay in ns2. Especially in the case of CBQ, both the enqueue and dequeue operations involve heavy estimation procedures and heuristic rules, resulting in several function calls. In addition, the code and scheduling rules themselves may not be the same as those used in the simulator, since some operations cannot even be implemented as such in the kernel. Thus, it is to be expected that queuing delays are not necessarily distributed in a similar fashion to the measurements.

10.1 Simulation setup

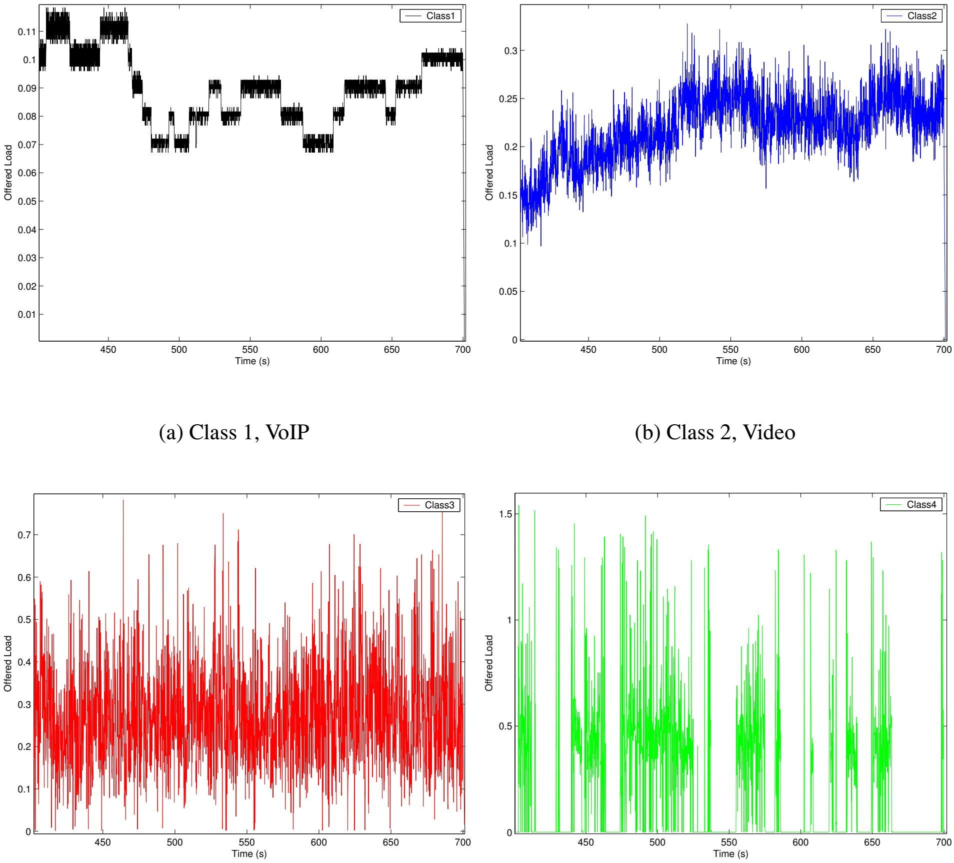
The network topology used in the simulations corresponds to the measurement topology in Figure 9.2, except for the fact that VLANs and measurement probes are not used in the simulations. The provisioning parameters for the algorithms are the same as in the measurements.

The only relevant difference between the measurements and simulations is the traffic process: the webcache model used in the simulations of FTP and HTTP traffic deviates from the FTP and HTTP traffic creation process used in Avalanche. There are also minor differences in the VoIP and Video traffic patterns. The differences in the load patterns are mainly caused by the fact that Avalanche offers only very basic patterns for starting new sessions. For instance, in the measurements, the HTTP and FTP session arrival processes are flat or sinusoid with a constant phase, while inter/arrivals are drawn from an exponential distribution in the ns2 webcache model session. A flat or sinusoid session arrival process used in the measurements clearly results in more periodic and deterministic behavior. Some differences in the load processes may also be caused by different TCP implementations in ns2 and Avalance.

Figure 10.1 and Figure 10.2 show the offered load patterns of different traffic types in the simulations and in the measurements for a desired average offered load of 90%. Due to these differences in the load patterns within individual traffic classes, the simulation and measurement results can only be compared at a relatively coarse level.

10.2 Results of the simulation and measurement comparison

The aim of this section is to analyze similarities and differences between the results of simulations and measurements of DBHPD and CBQ algorithms. The measured queuing delay distributions, obtained indirectly by utilizing traffic probes, have already been presented for both algorithms in the previous chapter on measurement. In this section, the measurement results are based on the data obtained directly from tcpdump, thus providing more accuracy.



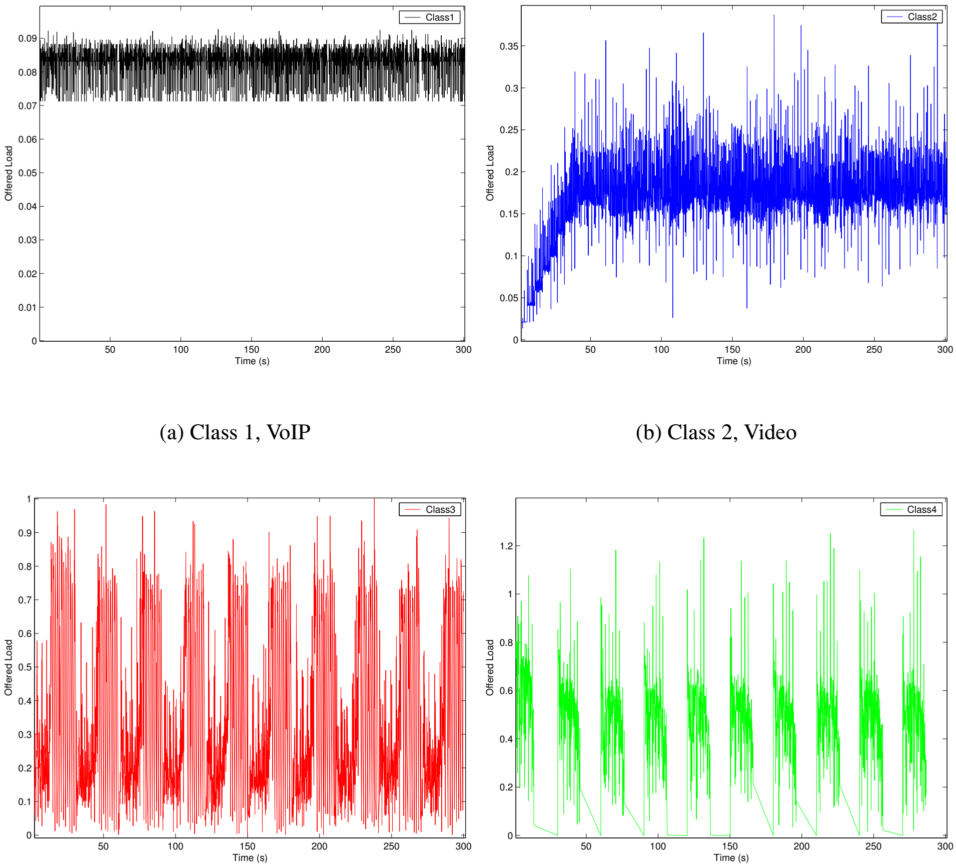
(a) Class 1, VoIP

(b) Class 2, Video

(c) Class 3, WWW

(d) Class 4, FTP

Figure 10.1: Offered class loads in the simulations (Total offered load 90%)



(a) Class 1, VoIP

(b) Class 2, Video

(c) Class 3, WWW

(d) Class 4, FTP

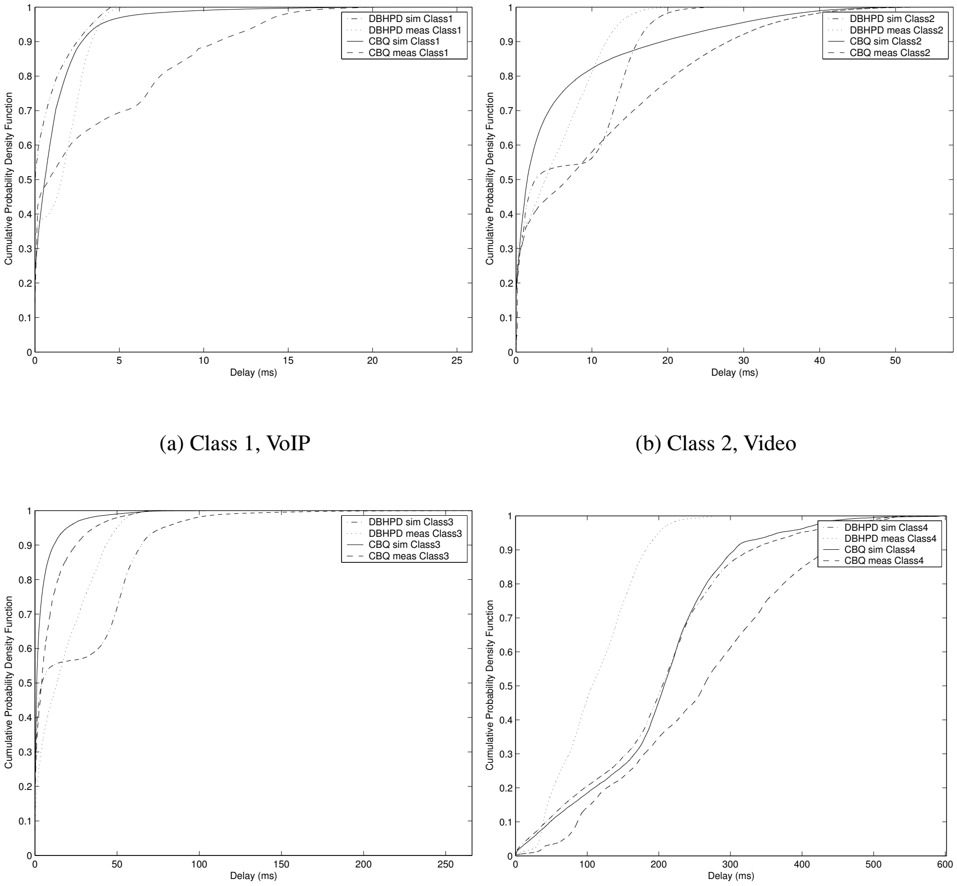
Figure 10.2: Offered class loads in the measurements (Total offered load 90%)

Figure 10.3 presents the delay distributions for each traffic class at target load level of 90%. The ranges of the delay values seem to be relatively close to each other for both algorithms, but there are clear deviations in the exact delay distributions for both algorithms. The deviations can be partly explained by the differences in the load processes. However, most of the difference is likely to be due to the simplifications used for the real implementation, as well as overhead caused by the estimation procedures. With CBQ, simulations tend to provide shorter delays for each class when compared with measurements. This is natural, since demanding borrowing operations occur frequently with a 90% load, which cause extra delays for the packets in a real router. It should also be noted that measurement operations performed before dequeue, such as packet header analysis, classification and enqueue, add some extra delays that are not modeled in simulations. As we already concluded from the measurement results in the previous chapter, CBQ provides shorter delays for the HTTP class at the expense of the real time classes. Even with a relatively small load, the maximum delay for VoIP is 5 ms with DBHPD and 20 ms with CBQ. The maximum delay for Video is 20 ms with DBHPD and nearly 50 ms with CBQ.

Figure 10.4 and Figure 10.5 show the corresponding delay distributions with higher loads of 100% and 110%. From the distributions, CBQ's problem of reversing the service order between class 2 and 3 can be observed for both simulations and measurements. Thus, even though CBQ might provide shorter delays than DBHPD for some classes, the overall service differentiation is not as predictable when the offered load increases.

The results also show that as the offered load exceeds 100%, the deviation between the CBQ's measurement and simulation results decreases. This is due to the fact that, at a high load, each traffic class uses its fixed share of resources and thus expensive borrowing operations do not occur. The algorithm mainly operates as the DRR general scheduler, which is a basic bandwidth sharing algorithm.

Tables 10.1, 10.2 and 10.3 present the average packet loss statistics with loads of 90%, 100% and 110%. The packet losses for the DBHPD algorithm seem to be larger in the



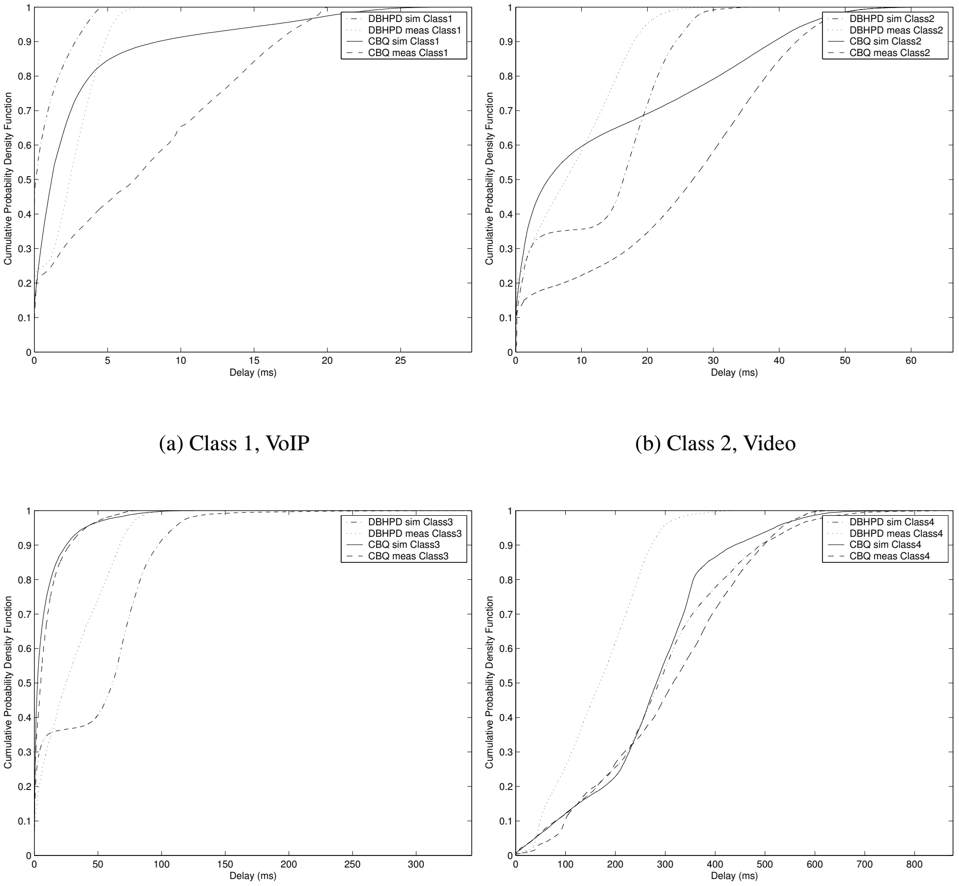
(a) Class 1, VoIP

(b) Class 2, Video

(c) Class 3, WWW

(d) Class 4, FTP

Figure 10.3: Delay distributions for CBQ and DBHPD simulations and measurements (Load 90%)



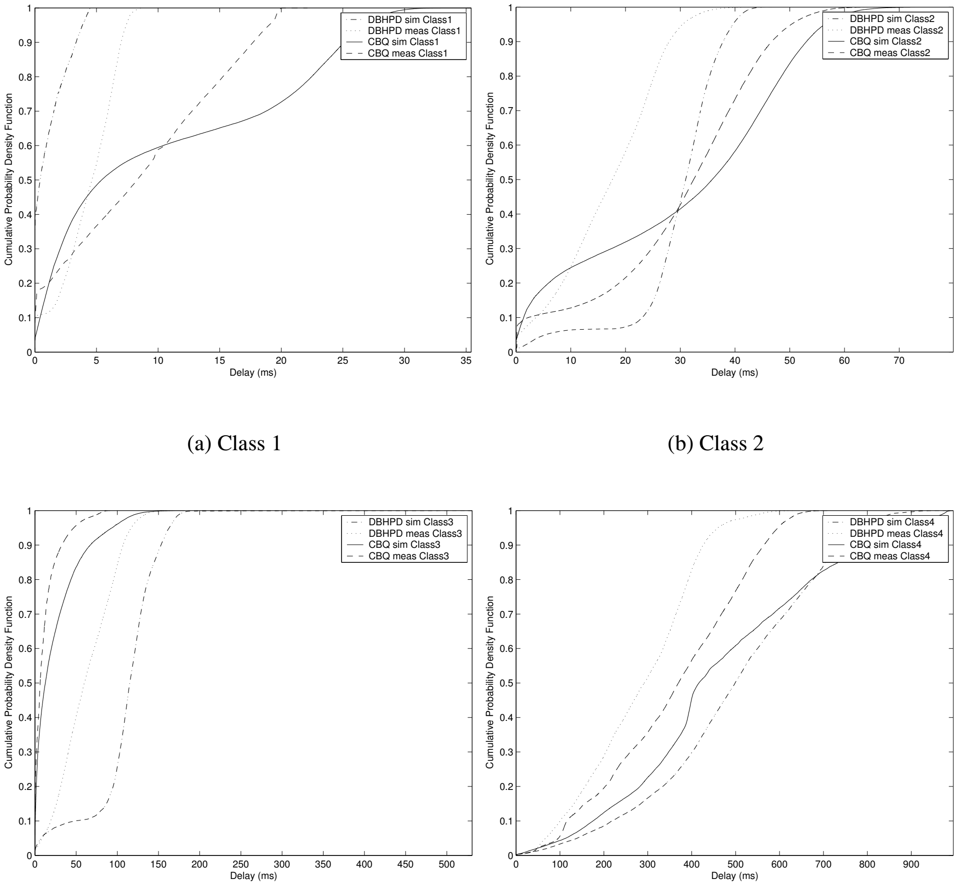
(a) Class 1, VoIP

(b) Class 2, Video

(c) Class 3, WWW

(d) Class 4, FTP

Figure 10.4: Delay distributions for CBQ and DBHPD simulations and measurements (Load 100%)



(a) Class 1

(b) Class 2

(c) Class 3

(d) Class 4

Figure 10.5: Delay distributions for CBQ and DBHPD simulations and measurements (Load 110%)

Table 10.1: Loss statistics for DBHPD and CBQ with a 90% load

	Loss (DBHPD) %		**Loss (CBQ)** %	
Traffic	**Mean (sim)**	**Mean (meas)**	**Mean (sim)**	**Mean (meas)**
FTP	0.1	0	0.07	0
WWW	0.1	0	0.01	0
Video	1.5	0.2	2	0.8
VoIP	0	0	0.07	0.1

Table 10.2: Loss statistics for DBHPD and CBQ with a 100% load

	Loss (DBHPD) %		**Loss (CBQ)** %	
Traffic	**Mean (sim)**	**Mean (meas)**	**Mean (sim)**	**Mean (meas)**
FTP	0.2	0	0.2	0.02
WWW	0.8	0	0.03	0
Video	1.8	0.7	5.2	6.5
VoIP	0	0	1.4	6.7

simulations, especially with a 110% load, suggesting that the real total offered load has most likely been higher in the simulations. However, packet losses with CBQ are considerably lower in the simulations. This can be explained by the borrowing operations that cause extra delays - and thus packet loss - in the measurements, which may not correspond to the CBQ borrowing model.

Figure 10.6 and Figure 10.7 show the bandwidth allocations for the traffic classes with an offered load level of 100%, for both simulations and measurements. From these figures it is evident that DBHPD allocates more bandwidth to the real-time classes, while CBQ provides more capacity to the non-real-time classes. This corresponds to the delay results presented earlier, where DBHPD guaranteed short delays for the real-time classes and CBQ clearly provided shorter delays to HTTP traffic at the expense of VoIP and Video traffic.

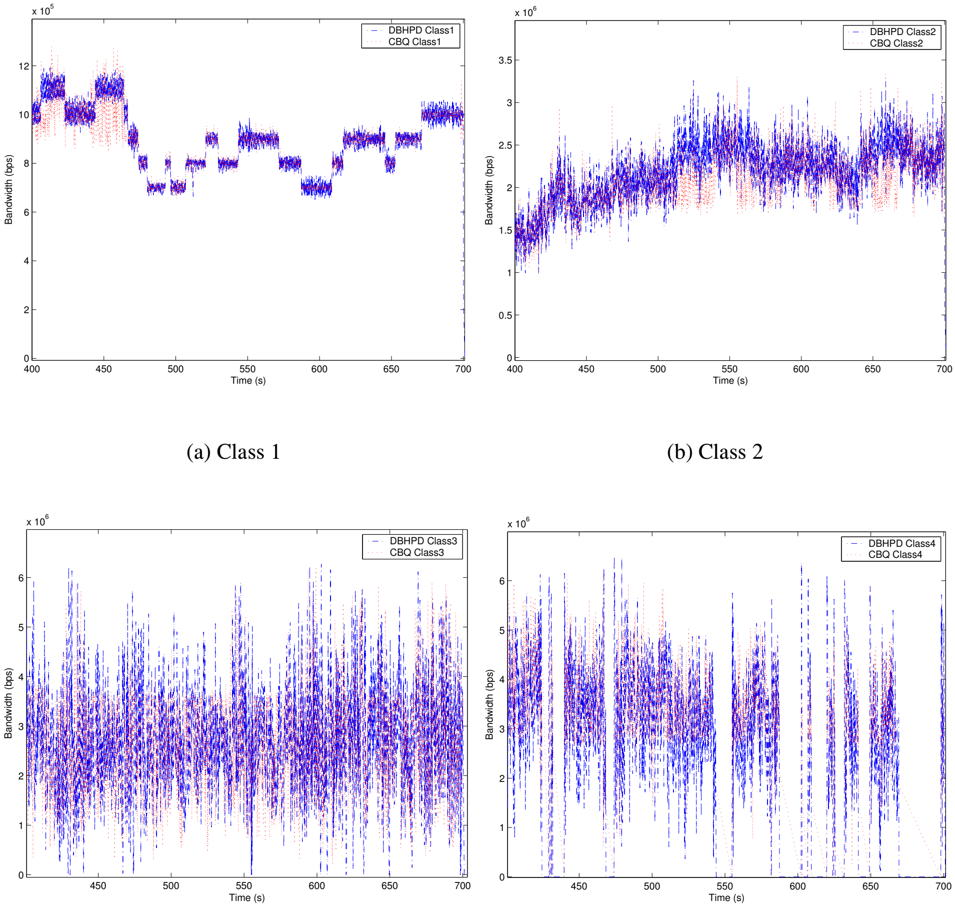
Table 10.3: Loss statistics for DBHPD and CBQ with a 110% load

	Loss (DBHPD) %		Loss (CBQ) %	
Traffic	**Mean (sim)**	**Mean (meas)**	**Mean (sim)**	**Mean (meas)**
FTP	0.7	0.03	0.5	0.07
WWW	7.7	0.02	0.3	0
Video	4.3	2.4	13.3	16.7
VoIP	0	0	7.4	18.4

Particularly in the CBQ measurements, it should be noted that there are clear darker bands near the guaranteed minimum capacity for each class, suggesting that CBQ operates as a basic bandwidth sharing algorithm at these points. DBHPD does not provide a guaranteed bandwidth, but its capacity allocation fluctuates according to the incoming load and queuing delays. In simulations the CBQ bands are not that clear. This is most likely due to the fact that the offered load profiles are somewhat different and the implemented algorithm does not operate exactly according to the model used in the simulations.

We have also performed initial kernel profiling for the implementation of both DBHPD and CBQ in ALTQ, to investigate throughput and resource overheads that can be achieved by these algorithms. The processor speed was 1.3GHz to ensure that processor capacity is not a bottleneck. The resulting throughputs for simple FIFO (First In First Out), PRIQ (Priority Queueing), CBQ and DBHPD are shown in Table 10.4. It can be seen that throughput is highest with the FIFO and PRIQ algorithms, since they do not use any estimation or time measurement procedures for scheduling. Correspondingly, DBHPD is able to achieve better throughput than CBQ due to its simpler implementation.

In order to examine how many more resources are consumed by CBQ compared to DBHPD, 1000000 TCP/IP packets with a size of 110 bytes were sent to be processed by both algorithms at a rate of 15000 packets/s. Table 10.5 shows the time consumed by enqueue and dequeue operations, as well as the number of function calls used by these algorithms. Results for PRIQ are also shown for comparison. The cumulative time used



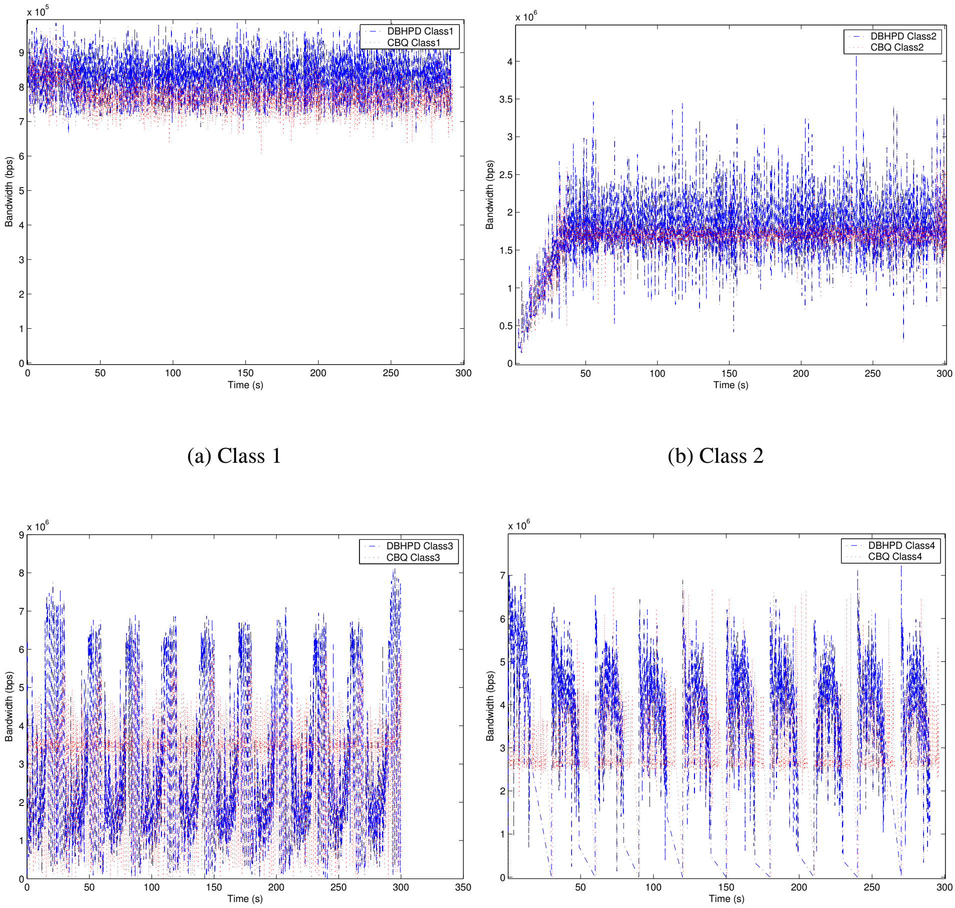
(a) Class 1

(b) Class 2

(c) Class 3

(d) Class 4

Figure 10.6: Bandwidth distributions for CBQ and DBHPD in simulations (Load 100%)



(a) Class 1

(b) Class 2

(c) Class 3

(d) Class 4

Figure 10.7: Bandwidth distributions for CBQ and DBHPD in measurements (Load 100%)

Table 10.4: Achievable throughput

Scheduler	Throughput (Mbps)	Frames/s	Relative
FIFO	85.53	66 221	1.000
PRIQ	84.38	65 322	0.987
CBQ	65.31	50 528	0.764
DBHPD	70.51	54 579	0.824

Table 10.5: Resource consumption for dequeue and enqueue operations

	Cumulative time (s)		Function calls		
Scheduler	Enqueue	Dequeue	Enqueue	Dequeue	Lost packets
PRIQ	0.25	0.07	3998480	2998860	380
CBQ	0.35	0.87	7996792	13994386	401
DBHPD	0.18	0.24	6997254	5997648	392

for enqueue and dequeue operations, as well as the number of function calls, are in line with the throughput results and confirm the observation that the CBQ dequeue function results in significantly greater overheads than DBHPD's dequeue. PRIQ has the smallest overhead, but it does not provide reasonable differentiation like DBHPD.

11 Scheduling Algorithms for 3G and 4G Networks

11.1 Design principles for 3G scheduling algorithms

11.1.1 Algorithms based on WFQ emulation

Many scheduling algorithms designed for wireless networks try to emulate the WFQ fair queuing algorithm for fixed networks in a wireless domain. Though originally designed for TDMA networks, the ideas behind the scheduling algorithms presented in this section can also be adapted to the TD-CDMA mode in 3rd generation networks. In these algorithms it is assumed that a flow can experience either a good channel or a channel with errors. The basic idea of the emulation is to compare the service received by a flow to a corresponding error-free service. The error-free service is the service that the flow would have received at a certain time if the channels had been error-free. The flow can either be leading, lagging or in sync, depending on whether it receives too many, too few or just enough resources when compared to the error-free service. A special compensation model is used to compensate for the lagging flows at the expense of the leading flows as soon as the lagging flows experience an error-free channel ([53], [28], [110]).

Idealized Wireless Fair Queueing (IWFQ)

Idealized Wireless Fair Queuing (IWFQ) ([53], [28]) simulates the wireline WFQ or WF2Q algorithm in the background in order to determine the error-free service for the flows. The algorithm assigns a service tag to each flow, which corresponds to the finish tag of the head of line slot, calculated according to the corresponding wireline WFQ equation. When a transmission slot becomes free, IWFQ schedules the flow with the smallest service tag from among all backlogged flows that perceive a good channel.

The compensation model in IWFQ functions as follows [28]: if a flow perceives an error-free channel and receives service, its service tag is increased. However, if a flow perceives a bad channel, its service tag is held constant, thus giving it a better chance to transmit when the channel later becomes error-free. In IWFQ the service tag of a leading flow i is not allowed to increase above a lead bound l_{i-max} and, correspondingly, the service tag of a lagging flow cannot decrease below a lag bound b_{i-max}. However, in spite of these restrictions leading flows might be starved for a long time when the lagging flows perceive a good channel and can begin to transmit. Thus, IWFQ does not meet the important objective of graceful service degradation.

Server Based Fairness Approach (SBFA)

The IWFQ algorithm presented in the previous subsection was only able to emulate WFQ or WF2Q algorithms. A Server Based Fairness Approach (SBFA) ([53], [28]) is a more generic framework for adapting any wireline fair queueing algorithm to the wireless domain. With a SBFA, a fraction of the channel bandwidth is reserved explicitly for compensation. Thus, the concept of a leading flow is not needed.

Instead of a leading flow, the SBFA uses the Long-Term Fairness Server (LTFS) concept, a virtual flow that is allocated a service weight corresponding to the explicitly reserved bandwidth. If the scheduling algorithm that is being emulated selects a certain flow for transmission, this flow is only allowed to send if it perceives a good channel. Otherwise, the slot that would have been assigned to the flow is queued into the LTFS flow, and another flow with a good channel is selected for transmission. The previously rejected flow has another chance to transmit if, at some later stage, the scheduling algorithm selects the head of line slot from the LTFS flow for transmission, and if this slot's original session has a good channel [28].

Since the SBFA does not explicitly use leading flows, the service degradation is graceful.

However, SBFA is not able to provide short-term fairness or worst-case delay bounds due to the use of the LTFS flow.

Wireless Fair Service (WFS)

The Wireless Fair Service (WFS) ([53], [28]) is based on the modified wireline fair queuing algorithm, which can provide decoupling between bandwidth and delay. Thus, flows requiring a short delay do not necessarily have to be allocated a large service weight if high bandwidth is not required.

As with IWFQ, the WFS flow i also has a lead bound l_{i-max} and a lag bound b_{i-max}. The idea of WFS is that the leading flows relinquish their slots in proportion to their lead (l_i/l_{i-max}, where l_i is the current lead of flow i), and that these slots will then be fairly allocated among the lagging flows [28].

Out of the three described algorithms, IWFQ, SBFA and WFS, the WFS algorithm operates best in the sense that it is able to provide both short-term and long-term fairness, delay and throughput bounds, graceful service degradation and, in addition, decoupling between delay and bandwidth.

11.1.2 Opportunistic scheduling

The wireless WFQ emulations presented in the previous sections consider the channel to be in either a good or bad state. However, considering two channel states is overly simple, especially for data services that have continuous utility functions. Opportunistic scheduling based on estimation is an important class of wireless scheduling algorithms, which are able to consider channel state as a stochastic process.

Opportunistic Transmission Scheduling with Resource-Sharing Constraints

[15] provides an example of opportunistic scheduling algorithms for TDMA or TD-CDMA networks, aimed at maximizing overall throughput and, at the same time, providing fairness for the users. The authors characterize the time-varying channel conditions using a stochastic model: user i is associated with process U_ik, where U_ik denotes the performance that would be experienced by user i if it was scheduled at time k. This performance measure can, for example, be throughput, which is the function of the user's SINR, or power consumption.

The goal of the proposed algorithm is to maximize the average system throughput while taking into account the time-varying channel conditions of the users. When user i is selected to be scheduled, the system is rewarded with the value U_{ik} . The scheduling policy is denoted by Q and the performance vector of all users by U. Formally, the problem can be expressed as follows [15]:

$$max_Q E(U_{Q(U)}) \tag{11.1}$$

As an example, let us consider the resulting scheduling policy in a two-class case. The task of the policy is to determine the time-fraction assignments $r1$ and $r2$ for the users. Define a function $y(v) = P\{U1 + vU2\}$. An optimal v^* exists, so that for any > 0

$$y(v^* - \epsilon) \leq r_1 \leq y(v^*) \tag{11.2}$$

The optimal v^* is determined by the distribution of U. However, U must be estimated, since it is not known. Figure 11.1 shows the block diagram for the estimation of v^* [15].

11.1.3 Specific algorithms for 3G networks

When compared to TDMA networks, scheduling is more challenging in 3rd generation networks due to the fact that different users may send data that use different coding sequences in the same frequency/time-slot. In this section, a few algorithms for both the

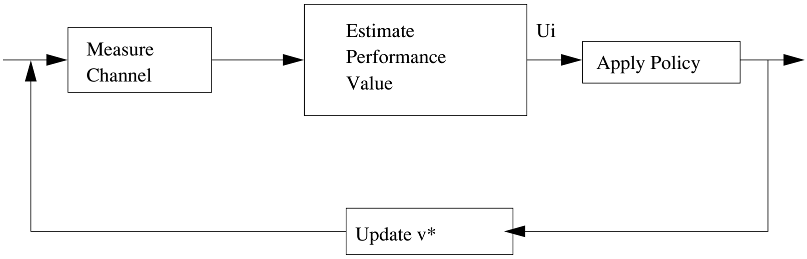


Figure 11.1: Block diagram for the estimation of v^*

UTRA-FDD mode and the UTRA-TDD mode are presented. Some of these algorithms perform scheduling independent of code assignment tasks, while others consider code assignment as a part of a scheduling problem. In terms of performance, the goal can be to simply maximize total data throughput or to provide QoS in terms of metrics, such as packet loss or delay. Some algorithms rely on heavy analytical optimization, while others use measurement information and estimation techniques for allocating the resources. As with scheduling in fixed IP networks, the optimization problems usually become overly complex as the number of factors to be considered increases. Therefore, measurement based estimation strategies have been proposed for feasible implementation.

With WCDMA, scheduling decisions are performed in the RNC, making it difficult to react to short term channel variations. Thus, scheduling decisions are made on a relatively long time scale (in multiples of a 10 ms frame, typically at least 100 ms) and other radio resource management algorithms, such as fast power control, are used to compensate for the channel variations on shorter time scales.

FDD mode scheduling algorithms

Fair Resource Scheduling (FRS) [111] and Delay Fair Scheduling (DFS) [144] are examples of UTRA-FDD mode algorithms that consider code assignment as an integral part of scheduling. FRS has been designed for supporting non real-time services on downlink shared channels (DSCH). It is based on the wireline WFQ algorithm, where each flow is assigned a service weight. Depending on the actual service received, the flows are marked as leading, lagging or in sync. A special compensation model is used to allocate resources to the lagging flows when they perceive a good channel, as explained in the previous section. The basic FSR scheme is able to prioritize between different service classes, provide fairness among flows, take into account the radio link conditions of the flows and assign a power limit to each flow and to the base station power budget.

DFS is a modification of the FRS scheduling algorithm that is able to provide differentiation based on the delay requirements of the flows. The delay threshold of a flow is denoted by DT, indicating the maximum allowed packet delay for preserving the QoS at a specified level. At the start of each frame k, the DFS scheme calculates a priority P_i for each flow i:

$$P_i = \frac{D_i(kT)}{DT_i} \geq 0, k = 0, 1, 2, ... \tag{11.3}$$

$D_i(kT)$ denotes the head-of-line packet delay for flow i and T is the length of the frame (10ms).

After calculating the priorities, DFS performs a proper rate assignment for the flows by emulating GPS resource allocation and taking into account the priorities and the available N_{FC} capacity in the system. As the rates are assigned, corresponding OVSF codes are allocated to the flows.

The first stage of the rate assignment procedure is to assign a zero rate to each flow. Then, starting from the flow with the highest priority, the following steps are followed [144]:

- The service rate R_s of flow i is set to $2R$ if R_s is currently zero and $N_{FC} - 2R \geq 0$.
- While flow i has more packets in the queue than could be transmitted during the next frame with rate R_s, and $N_{FC} - 2R \geq 0$, R_s is doubled. N_{FC} is set to

$N_{FC} - R_s$.

The rate assignment process is finished when either $N_{FC} = 0$ or when all flows can transmit their packets in the queue with the assigned rates during the next frame.

The OVSF codes are assigned from the left of the OVSF code tree, so that the first code that supports the assigned rate is selected. To eliminate interference, the ancestors and descendants of the code are blocked until it is no longer being used.

TDD mode scheduling algorithms

Fair Packet Loss Sharing (FPLS) In [79], the authors propose a Medium Access Protocol (MAC) along with fair packet loss sharing (FPLS) for TD-CDMA networks. Only the scheduling part of the algorithm is handled here. The goal of FPLS is to maximize resource utilization and the number of users supported by the system, and to share packet losses among the users based on their QoS requirements. The QoS requirement can either be a certain Bit-Error Ratio (BER) or a certain maximum delay. In this case, a BER caused by wireless transmission is called transmission BER (TBER) and a BER caused by buffer overflow or the maximum delay being exceeded is referred to as packet loss probability (PLP).

When performing the scheduling decision for the next frame, FPLS first considers the most urgent packets (MUPs). The MUPs have to be scheduled in the next frame, because otherwise the maximum delay will be exceeded and all these packets will be dropped. In order to meet PLP requirements for the users, the numbers of dropped MUPs for each user have to be controlled. Other packets can only be selected for transmission if all MUPs have already been assigned for the next frame.

The authors propose a bin-packing scheduling algorithm for controlling the TBER and

PLP requirements. Bin-packing in its original form is a well-known combinatorics problem. The idea is to pack a set of blocks into a minimum number of bins, assuming that the blocks cannot be divided. The time slots are considered as bins and the packets as blocks. The bin size is N_{max} code slots and the block size is the number of code slots required by a packet [79].

Traffic and Interference Adaptive Scheduling The traffic and interference adaptive scheduling algorithm proposed in [41] has been designed to deliver TDD mode Internet traffic using the DSCH channel. The goal of the algorithm is to maximize total data throughput by minimizing interference, in order to efficiently utilize radio resources. However, the algorithm is not capable of providing differentiated Quality of Service to the traffic flows.

The algorithm operates as follows: firstly, the time slots available for user traffic are divided into uplink and downlink parts. Then, for each part, the number of Resource Units (RUs) that can be used per time slot are calculated. Ideally, the number of RUs corresponds directly to the number of codes. However, in reality simultaneous transmissions may cause interference, despite the orthogonality of the codes. Therefore, the task of the scheduling algorithm is to allocate the RUs so that the interference level does not exceed a certain threshold.

As a performance metric, [41] uses $P_e(k)$, the probability that an RU is discarded at the receiver because of transmission errors when k codes are used in the same time slot. By using this metric, the algorithm finds an optimal value k^* for the number of codes that maximizes the throughput:

$$T(k^*) = \max\{T(i), i = 1, ..., 16\} \tag{11.4}$$

where $T(i) = (1 - P_e(i)) * i$ (the expected value for the number of correctly delivered RUs when i codes are used). After solving the optimal k, the same number of RUs is allocated to all active flows.

In the optimization problem presented here, it is assumed that the level of system interference with a certain number of codes is known. However, this is not the case in reality and efficient methods for estimating $P_e(k)$ are required. The estimators for $P_e(k)$ utilize the information about erroneous RUs from the previous frames. In the uplink direction the base station can verify the correctness of the RUs sent by the users. In the downlink direction, the so-called Acknowledged Mode Procedure must be used in order to receive feedback about the erroneous frames.

It should be noted that if the number of RUs is decreased due to the estimation result, there will be an additional deterministic loss, because this means that a smaller amount of data can be transmitted in the channel. However, if the expected decrease in the RU error probability provides a greater gain compared to the deterministic loss, it is worth decreasing the number of RUs. Correspondingly, if the number of RUs is increased, this can potentially result in a higher throughput or in a significant increase in error probability. The estimation algorithms aim to balance these risks in order to enable the optimal number of RUs to be chosen. The estimation algorithm can rely on worst-case assumptions about increasing the number of RUs, or alternatively it can filter the errors exponentially in some way. However, the details of the estimation algorithms are omitted here.

11.1.4 HSPA scheduling

With HSPA scheduling, decisions are made in NodeB/eNodeB for both the uplink and downlink, allowing a fast reaction to channel variations [136]. With HSPA, a scheduling resolution of 2 ms (the duration of one subrame) is applied. Channel quality information (CQI) plays an important role in scheduling. With HSPA, UE provides uplink CQI in the form of the highest MCS that can be used to receive the data, with the desired QoS including SINR and BLER targets. If multi-antenna transmission is used, additional feedback such as Precoding Control Information (PCI) has to be provided.

A thorough simulation analysis of several channel-adaptive scheduling algorithms that use CQI feedback can be found in [112]. Both the overall system performance and the individual user quality of service (QoS) constraints are studied. In Section 11.2.2 we provide mathematical formulations of both these and other channel adaptive scheduling algorithms in LTE. The basic ideas in these algorithms are the same as those used for HSPA; the main difference is the scheduling resolution.

Size-based scheduling can also be combined with channel-adaptive HSPA scheduling. [100] presents heuristic algorithms that utilize both flow sizes and channel knowledge. According to their results, a significant performance gain can be obtained by using size information in addition to channel information.

11.2 Design principles for 3.9G/4G scheduling algorithms

In 4G networks the basic task of allocating time slots to service classes or individual users is very similar to 3G networks. The algorithm could, for example, be some sort of modified Fair Queuing algorithm presented in the previous sections. Yet, 4G networks should be able to bring considerable enhancements to reliability and throughput when compared to 3G networks. Furthermore, networking and interoperability solutions must allow users more flexibility. Energy efficiency will be a critical issue, especially in hybrid scenarios where low-battery devices are used to forward traffic. As we have stated in [78], it is obvious that pure scheduling is not able to meet these requirements, so significant improvements and changes will need to be made to layers 1 and 2 of the ISO OSI model. Cross layering should be used to tie the different layers together so that the scheduling algorithm can exploit and modify the information and algorithms available in the lower layers. Advanced methods such as link adaptation, adaptive transmission and MIMO provide large amounts of physical layer information that should be taken into account for scheduling. The structure of the wireless channel also has to be considered (i.e. how long frames are used, the number of time slots per frame, etc.), as well as the channel access method used.

11.2.1 Cross-layering in wireless scheduling

Optimizing algorithms across layers is known as Cross Layer Design (CLD) [87, 133]. Cross layering may either be revolutionary or evolutionary [23]. In a revolutionary approach an existing protocol stack is completely redesigned, whereas with an evolutionary approach the existing stack is only modified so that some layers can exchange information. The evolutionary approach is preferred due to compatibility and economic issues. Still, cross-layering may cause unintentional loops and instability effects if a parameter is modified on several layers [87]. CLD also breaks the modularity and abstraction of layers and makes it more difficult to redesign and review protocols.

In 4G packet scheduling, cross-layering information mainly from the physical, link and MAC layer could be utilized. In existing systems the task of the physical layer is to transmit the bits at a certain power level and to provide the following information: coding/modulation, bit-error rate and transmit power. The link/MAC layer, on the other hand, provides error-free transmission with the help of Forward Error Correction (FEC) and Automatic Repeat reQuest (ARQ). The link/MAC layer provides information on the FEC scheme, the number of retransmitted frames and the time when the medium is available for transmission. In 4G systems, the amount of channel information needed for link adaptation, adaptive transmission and MIMO is much greater in comparison to existing systems. Evidently, it is challenging to transfer this large amount of channel information to the MAC layer and to send processed control information from the MAC to physical layer. Moreover, to be useful in fast fading conditions, information transfer and processing should be carried out in a time frame of milliseconds.

The physical and link/MAC layers could act, for example, in the following way: If the Bit Error Rate (BER) or Frame Error Rate (FER) in the physical layer exceeds a certain threshold, the link/MAC layer could instruct the physical layer to increase the transmit power, or the power could be decreased while increasing error correction. However, it should be noted that a better error correction method in itself increases power consumption. This results from the fact that channel coding techniques increase the redundancy of the transmitted data, again leading to an increase in overall power.

The scheduler element could interact with the physical and MAC layers in the following ways: the scheduler could ask the physical layer to adapt or even change the modulation, depending on the QoS requirements of the traffic streams and the channel conditions. In the MAC layer the scheduler could give instructions to change the frame transmission priority or to adapt FEC/ARQ methods [133].

Spatial diversity and MIMO are key technologies considered in 4G system design, which bring better power and interference control benefits. The use of these technologies also affects scheduling and feedback mechanisms.

In existing mobile assisted network models, the infrastructure network decides the optimal rate for adaptation schemes based on the information collected from the MTs [2]. If MTs became MRs they would be required to carry out similar channel estimations on their own. Optionally, in the case of hybrid networks, the AP or RS could assist MRs, for example, by providing optimal power control information. However, this is not a viable option with a high terminal mobility rate and/or in dynamic fast fading channel conditions, as the information becomes stale very quickly. On the other hand, this could be a feasible solution to counter slow or shadow fading. Therefore, distributed rate adaptation algorithms that are optimized for MRs should be considered for hybrid networks.

11.2.2 Algorithms for LTE/LTE-Advanced

Like HSPA, LTE scheduling decisions are also performed in NodeB/eNodeB for both uplink and downlink, allowing a fast reaction to channel variations ([136]). With LTE scheduling, the resolution is 1 ms and scheduling occurs in the domains of both time and frequency. Thus, CQI also has to be provided in terms of both time and frequency resolution [96]. Since both uplink and downlink scheduling is performed in eNodeB, information also has to be available about subcarrier states, even when there is no ongoing transmission in the uplink. For this purpose LTE uses channel sounding, whereby UE transmits sounding reference symbols (SRS) periodically for the eNodeB channel quality estimation.

With LTE, most schedulers operate in two phases: time domain packet scheduling (TDPS) and frequency domain packet scheduling (FDPS). Firstly, TDPS creates a list of users to be considered during the next scheduling period and FDPS allocates the RBs to users in this scheduling candidate set (SCS). Average CQI is often used with TDPS, while FDPS utilizes CQI per subcarrier. The performance gain of LTE scheduling comes from increased diversity and flexibility when operating in these two domains. [128], [89], [107] and [98] study the performance gain of FD scheduling compared with pure TD scheduling in various scenarios. Based on the reported results, average system throughput and cell-edge user bit rate can be improved by about 40% compared to a TD scheduler that does not utilize the CQI information of different subcarriers. Figure 11.2 depicts the essential components of a time-frequency scheduler.

Both TDPS and FDPS can apply various scheduling algorithms depending on the desired outcome. The most important schedulers are Round Robin (RR), Maximal signal to interference ratio (MaxSIR), Proportional fair (PF) ([18], [151], [76]), QoS aware weighted proportional fair (QWPF), Fair throughput (FT), channel dependent earliest due date (CD-EDD) and Exponential rule (ER) [138]. The RR scheduler simply allocates an equal share of resources to all users, without considering their channel states. The FT scheduler al-

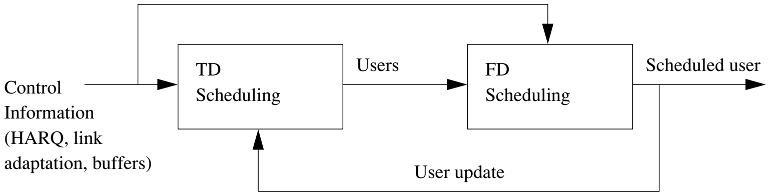


Figure 11.2: Time-frequency LTE scheduler

locates resources so that, on average, different users get the same throughput. Table 11.1 shows the scheduling functions for the more advanced channel aware schedulers. $\frac{C}{I}_i$ and λ_i denote current Carrier-to-Interface ratio (CIR) and average rate of user i; $w_i(k)$ is a QoS weighting factor for user i's flow k; W_i and $\bar{W}$ are the waiting time for user i and the average waiting time for all users; $D_{max,i,k}$ is the maximum delay for user i's flow k and a is a factor for tuning the impact of delay.

Table 11.1: Typical scheduling functions for time-frequency scheduler

Scheduler	**Scheduling function**
MaxSIR	$\frac{C}{I}_i$
PF	$\frac{\frac{C}{I}_i}{\lambda_i}$
QWPF	$\frac{\frac{C}{I}_i}{\lambda_i} \cdot w_i(k)$
CD-EDD	$w_i(k) \cdot \frac{\frac{C}{I}_i}{\lambda_i} \cdot \frac{W_i}{D_{max,i,k}}$
ER	$\frac{\frac{C}{I}_i}{\lambda_i} \cdot exp(\frac{aW_i - a\bar{w}}{1+\sqrt{a\bar{W}}})$

The above schedulers are only able to allocate resources between users, not at the flow-class identifier (FC-ID) level. In [61] the authors present the design for a LTE scheduler within a DiffServ context that operates on two layers: an inter-FC-ID scheduler that first sorts the FC-IDs in function of their policies, and an intra-FC-ID scheduler that performs scheduling within a given class. Other proposed LTE schedulers include [129] and [24], where a Hybrid ARQ is used together with scheduling to further improve performance. [107] considers the state of UE buffers as well as channel state in order to reduce PLR and to maintain fairness and high system throughput. [135] presents a queue- and channel-aware low-complexity scheduler that is able to support different types of QoS.

The scheduling algorithms described so far have only considered scheduling in the LTE downlink. In [101] and [16] PF FDPS for a 3GPP LTE Uplink are proposed. Contiguous frequency allocation, required by the SC-FDMA used in LTE uplinks, limits the degrees of freedom in scheduling. The authors show that the FD scheduling problem becomes NP-hard with this constraint and propose practical heuristic algorithms to solve the problem. Other proposals for LTE uplink scheduling include [35], [34] and [102], which describe schedulers with different optimization metrics and complexities.

As well as channel-aware scheduling, interference mitigation also has an important role, especially at the cell edge or when the network is heavily loaded, when interference increases mainly due to two reasons: as the UE moves further from its eNodeB, the strength of the received signal decreases. At the same time, inter-cell-interference (ICI) increases when the UE approaches the neighboring eNodeBs. With LTE, ICI occurs when there is a collision between resource blocks that are used simultaneously by two or more cells. Inter Cell Interference Coordination (ICIC) has been studied in several papers and aims at reducing this collision probability, as well as the resulting SINR degradation. In ([105], [94], [14]), ICIC has been defined as an optimization task, whose objective is to maximize the multi-cell throughput, subject to power constraints, inter-cell signaling limitations, fairness objectives or minimum bit rate requirements. The optimization problems often become complex and, as a result, 3GPP has studied feasible and intuitive heuristics using system simulations [5]. ICIC time scales range from milliseconds (fully synchronized scheduling) to several days (static planning). The time scales chosen largely depend on the desired performance versus the signaling overhead that is allowed. Enhanced frequency reuse is an important class of ICIC mechanisms. It can be used to reduce the ICI, for example, by using a frequency reuse factor (FRF) greater than one in cell-edge regions and a FRF of one near the cell-center [90]. Other modifications to this scheme include [3] and [4]. In these schemes, the assumption is that the traffic load remains stable within a cell. However, this is not the case in reality and, as a result, dynamic frequency reuse schemes have been suggested ([52], [95]) in which the bandwidth is divided into sub-bands so that the cell center and cell edge use different bands and bands can be borrowed either within a cell or between cells.

12 Multiantenna technologies and channel aware scheduling in 4G networks

The use of multiple antennas is a key physical layer technology that can be used to radically enhance the performance and spectral efficiency of wireless systems. Denote by B and C the system bandwidth (Hertz) and capacity (bps) for the wireless link in an ideal channel. Further denote by ρ the SNR in the channel. According to Shannon's law (Eq. 12.1), capacity is proportional to the logarithm of transmission power meaning that simply increasing power finally results in diminishing returns. However, by using several antennas capacity can be increased linearly while transmitting on the same frequency [63], [57]. The desired system efficiency can be further improved by combining multiantenna technologies with channel-aware schedulers which divide the radio resources between multiple users [93], [149]. Both approaches are already used, for example, in high speed downlink packet access (HSDPA), in the extension of wideband code division multiple access (WCDMA) and in 3G Long Term Evolution (LTE) [8], [10], [9], [11]. Moreover, further development of these technologies will be an important component of LTE-Advanced design efforts [6].

$$C = B \times log_2(1 + \rho) \tag{12.1}$$

12.1 Spatial diversity with multiantenna technologies

Depending on whether multiple antennas are used at the receiving or transmitting end, or both, multiple antenna systems can be known as Single-Input Multiple-Output (SIMO), Multiple-Input Single-Output (MISO) or Multiple-Input Multiple-Output (MIMO). Multiple antenna techniques can be used for two types of diversity, namely spatial diversity and spatial multiplexing. With spatial diversity, reliability and range is increased by sending or receiving copies of the same data stream along different spatial paths between the transmitter and receiver antennas. With spatial multiplexing, on the other hand, independent data streams are sent simultaneously along different spatial paths, resulting in a multiplexing gain. Table 12.1 shows achievable capacity with different diversity methods [63].

Table 12.1: Achievable capacity assuming an ideal channel

Method	**Capacity (bits/sec)**
SISO	$Blog_2(1+\rho)$
Diversity (1xN or Nx1)	$Blog_2(1+\rho N)$
Diversity (NxN)	$Blog_2(1+\rho N^2)$
Multiplexing	$BNlog_2(1+\rho)$

12.1.1 Receive diversity

Figure 12.1 depicts a 2x2 system used for receive diversity. s and r respectively denote the transmitted symbol and received signal, while h is a complex valued channel coefficient describing the signal's amplitude and phase shift between selected antenna pairs. n represents thermal noise. When using multiple antennas for reception, the easiest method is to select the antenna with the strongest signal (SNR) and ignore the other signals. This method is called selection combining (SEL). A more advanced method is to add the signals together so that they combine without the undesirable effects of multi-path fading. In practice, the signals are delayed until they are in the same phase. The signals can also be weighted by their SNR so that larger weight is given to a signal with better SNR, which is equivalent to scaling the signals by their magnitude before adding them. This method is called Maximal-ratio combining (MRC) and involves adding the signals based on MRC results to the overall SNR, which is the sum of the component SNRs. Operation of MRC can be analytically expressed as:

$$y = \sum_{k=1}^{M} w_k \cdot r_k, \tag{12.2}$$

where y denotes the final receiver output and $w_k = h_k * /||h||$, where h_k* denotes the complex conjugate of the corresponding channel coefficient.

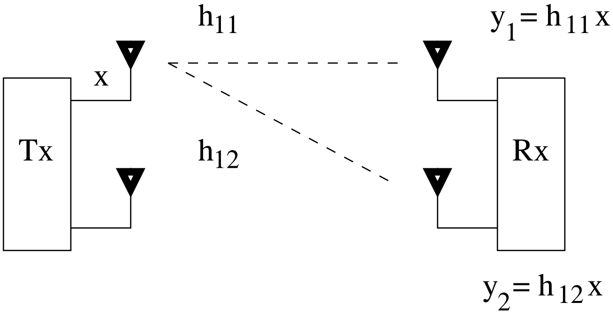


Figure 12.1: 2x2 system for receive diversity

12.1.2 Transmit diversity

Figure 12.2 depicts a 2x2 system used for transmit diversity. With transmit diversity, the equivalent to SEL in antenna selection, the best antenna is chosen for transmitting the packet. The MRC equivalent is transmit beamforming, where the transmitter precodes (delays) and weights the signals so that the transmit power allocated to each spatial path is based on the SNR. In order to be able to precode and weight the signals, the transmitter must be aware of the channel state. This requires regular feedback from the receiver. Transmit beamforming mobile stations encode channel state information (CSI) into the feedback message that is sent to the BS through an uplink control channel. Note that the term CSI refers to feedback that contains both power and phase information, while the term channel quality information (CQI) is commonly used for signal power related feedback. For example, with HSDPA the dedicated physical control channel (DPCCH) contains a feedback indicator field (FBI) field that carries CSI to the BS [8]. Current HSDPA may employ only two transmit antennas but we consider general M antenna case.

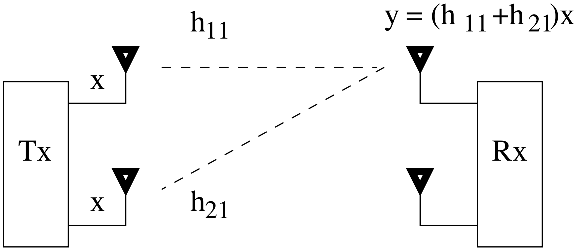


Figure 12.2: 2x2 system for transmit diversity

Transmit beamforming methods

This section provides an analytical description of the most important transmit diversity methods, antenna selection, Mode1 and Mode2. Received SNR from antenna m is denoted by γ_m. In the analysis we assume that the complex channel coefficients $h_m = \sqrt{\gamma_m} e^{j\psi_m}$ $(m = 1, \cdots, M)$ are *i.i.d.* zero–mean Gaussian random variables that are perfectly known at the receiver. Feedback words are composed of short–term CSI, available at the transmitter without errors or latency. Note that the effects of feedback errors and feedback latency on the performance of transmit beamforming have been previously studied in [38,65,123].

The received signal in the mobile station is given by:

$$r = (\mathbf{h} \cdot \mathbf{w})s + n = \Big(\sum_{m=1}^{M} w_m h_m \Big) s + n, \tag{12.3}$$

where s is the transmitted symbol, n refers to zero–mean complex additive white Gaussian noise, and the vector $\mathrm{w} \in \mathbf{W}$ refers to the codebook of complex transmit weights, such that the sum power constraint $\|\mathrm{w}\|^2 = \sum_{m=1}^{M} |w_m|^2 = 1$ is satisfied.

In the analysis, we will compare the following three practical transmit beamforming methods:

Antenna selection. In this method $\mathbf{w} = (0, .., 0, 1, 0, .., 0)$, where the non–zero component indicates the best channel in terms of received power. Hence,

$$|\mathbf{h} \cdot \widehat{\mathbf{w}}| = \max\{|h_m| : 1 \leq m \leq M\}.$$

The quantization set has M vectors and, as a result, only $\lceil \log_2(M) \rceil$ feedback bits are required, where $\lceil x \rceil$ denotes the smallest integer, not less than x.

Mode 1. Here antenna specific feedback is applied in order to adjust the phases of the component signals from different antennas with respect to a reference antenna [64]. Without disregarding the general nature of this analysis, the first antenna was chosen as the reference. For each of the other antennas, the feedback word consists of N_{rp}-bit information on the state of each relative phase between the reference antenna and the other $M-1$ antennas. Feedback bits are determined using the algorithm:

$$|h_1 + \widehat{v}_m h_m| = \max\{|h_1 + v_m h_m| : v_m = e^{j2\pi n/2^{N_{\text{rp}}}}\}, \tag{12.4}$$

where $0 \leq n \leq 2^{N_{\text{rp}}} - 1$, and the components of the transmit weight $\widehat{\mathbf{w}}$ are of the form:

$$\widehat{w}_m = \frac{1}{\sqrt{M}} \begin{cases} 1, & m = 1, \\ \widehat{v}_m, & m > 1. \end{cases}$$

A natural generalization for the above scheme is to weight transmit amplitudes in addition to adjusting phase differences, giving rise to a method that is known as Mode 2.

Mode 2. In Mode 2, three bits are reserved for adjusting the antenna phases relative to the reference antenna and one bit is dedicated to transmitting power feedback. This leads to the circular 16-quadrature amplitude modulation (16-QAM) constellation illustrated in Figure 12.3

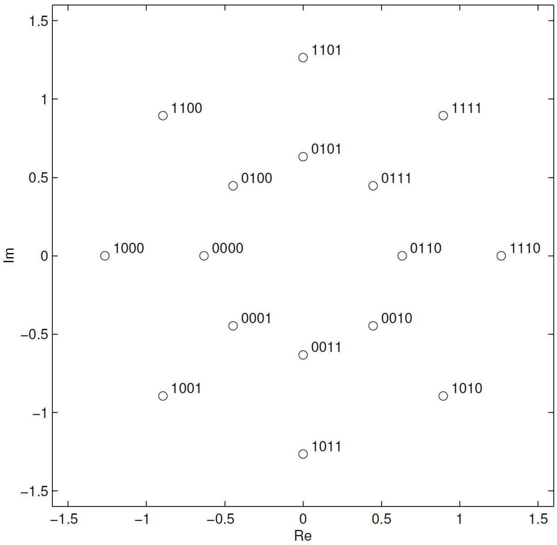


Figure 12.3: Weight states (w_2) in Mode 2. [77]

The feedback word corresponds to the Gray labeled signal state closest to weight w_2. These labels are transmitted to the BS in the FSM_{ph} field of the uplink signal. The Feedback Signaling Message (FSM) is a part of the FBI field of the DPCCH. Each feedback message contains N_{po} + N_{ph} bits, and one bit being transmitted in each slot results in a feedback bit rate of 1500 bps.

In principle, the problem of finding an optimal w_2 (or w_n in the n-antenna case), so as to maximize received SNR, can be formulated as follows. First, a matrix is defined as **R** := $\sum_{n=1}^{N_R} \mathbf{H}_n^\dagger \mathbf{H}_n$, where N_R is the number of receive antennas and $\mathbf{H}_n$ the channel matrix associated with the receive antenna n. Then, the optimal weight vector w_2 (or w_n) is the dominant eigenvector of R in:

$$\underset{\mathbf{W}\,:\,||\mathbf{W}||^2=1}{\texttt{argmax}}\,\mathbf{W}^\dagger\mathbf{R}\mathbf{W} \tag{12.5}$$

A more thorough analysis of Mode 1 and Mode 2 has already been presented in [64].

The main benefit of antenna selection is the low feedback overhead, which makes the algorithm attractive from the practical implementation point of view. However, antenna selection is sensitive to feedback errors [65]. Mode 1 and 2 are straightforward extensions of the two-antenna UMTS CL transmit diversity Mode 1 and 2. Mode 2 is included only in Release 99 (R99) of the UMTS two-antenna system, while Mode 1 also appears in later specification releases. Finally, we note that the length of the feedback word in Mode 1 is $(M-1)N_{\mathrm{rp}}$, while in Mode 2 additional $\lceil \log_2(M!) \rceil$ order bits are needed.

12.1.3 Spatial multiplexing

Figure 12.4 depicts a 2x2 system used for spatial multiplexing. Different streams will combine if they are multiplexed in the same channel, and the receiver has to properly separate and decode them. The two best known schemes for spatial multiplexing are Direct-Mapped MIMO and Precoded MIMO. With Direct-Mapped MIMO, transmit power is divided equally between different antennas. Using a channel matrix $\mathbf{H}$, the received signal vector ($\mathbf{r}$) can be expressed as:

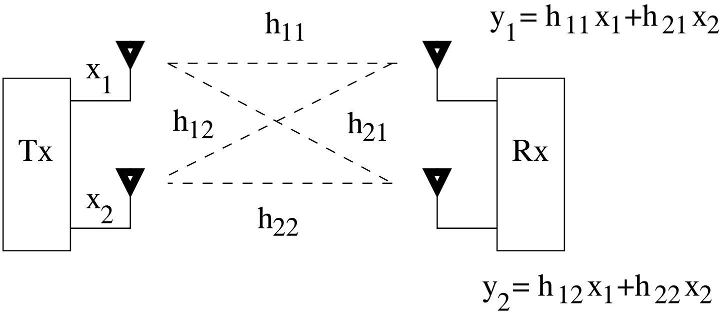


Figure 12.4: 2x2 system for spatial multiplexing

$$\mathbf{r} = \mathbf{H}\mathbf{s} + \mathbf{n}, \tag{12.6}$$

where $(\mathbf{s})$ is the vector of transmitted symbols and $(\mathbf{n})$ is the noise vector. In order to decode the different streams, this linear system can be solved to estimate $\mathbf{s}$ from:

$$\mathbf{H}^{-1}\mathbf{r} = \mathbf{s} + \mathbf{H}^{-1}\mathbf{n}. \tag{12.7}$$

This is equivalent to projecting $\mathbf{r}$ in a direction that is orthogonal to the channel coefficients of the other streams, i.e forcing the other signals to zero. This kind of MIMO receiver is called a Zero-Forcing (ZF) receiver. However, the problem with a ZF receiver is that H is almost non-invertible when different spatial paths are correlated and thus the noise term is amplified compared to the noise in the original system. The Minimum Mean Square Error (MMSE) detector is a modification of ZF, which strikes a balance between zeroing other streams and amplifying noise. Both ZF and MMSE receivers have a low computational complexity due to their linearity. With Precoded MIMO, the transmission power is not evenly distributed between different paths; instead a larger portion of the total power can be given to higher capacity paths. In this way, a greater theoretical MIMO capacity can be achieved when compared with Direct Mapped MIMO. However, since Precoded MIMO requires precoding at the transmitter and shaping at the receiver, the transmitter must be aware of the channel coefficients, as with transmit diversity techniques. This increases implementation complexity.

12.2 Channel aware scheduling with transmit beamforming

Both transmit beamforming and channel-aware scheduling provide performance benefits, even when there is scarce CSI in the transmitter [70], [69]. Methods that rely on heavily quantized channel information are especially important in FDD systems like HSDPA, LTE and LTE-Advanced, where different frequency bands are employed for the uplink and downlink and a separate feedback channel is required. Furthermore, the capacities of control channels are necessarily limited, and large amounts of control information imply large latencies in feedback signaling, which may limit the operation point of the system to slow mobile velocities.

This thesis considers downlink transmission systems, as shown in Figure 12.5. It is assumed that radio resources are shared between K mobile users, that the base station (BS) employs M-antenna transmit beamforming and that each mobile station (MS) has a single receive antenna. For transmit beamforming feedback and feedback needed for scheduling strategies, mobile stations are required to estimate the different channels M from pilot signals that are multiplexed to the data. Such a common code-multiplexed pilot structure is employed, for example, in HSDPA systems.

We assume that users are homogeneous in terms of QoS requirements. Scheduling decisions are based on temporal fairness in the sense that each user obtains an equal share of transmit resources over time. MS sets the feedback bit based on the relative SNR $\gamma/\bar{\gamma}$. Here γ refers to an instantaneous SNR in the MS, while $\bar{\gamma}$ denotes the SNR averaged over an active user set on a longer time scale. $\bar{\gamma}$ may vary between user sets in different geographical locations, although the base station transmission power is fixed. It should also be noted that temporal fairness is not providing fair throughput for users, since MSs near the cell center may have an access to higher data rates than users that are located on the cell edge.

12.2.1 Scheduling strategies

This thesis takes into consideration a comparison between the following three scheduling strategies: Round Robin, On-Off and Maximum SNR. Although *Maximum SNR* or *On-Off scheduling* are not inherently fair, there are some ways to make them approximate fairness through proper system design. For example, power control can balance average user SNRs, scheduling can be performed separately for subgroups of equidistant users, or the CQI may be defined so that it measures the instantaneous SNR relatively to the average SNR.

Round Robin strategy. This is a baseline approach where the served user is either randomly selected from among K active users, or the selection is performed according to a fixed sequence. The system does not take into account users' channel states and there is user-specific feedback that only relates to transmit beamforming.

On-Off strategy. In this approach, the transmitter either selects the served user randomly, or according to a fixed sequence based on the set of users that have reported a positive acknowledgment (ACK) during the last transmit time interval (TTI). The one-bit ACK/negative ACK (NACK) is set at the MS during each TTI, based on a comparison between the received SNR γ and a predefined SNR threshold γ_0 such that:

$$\begin{cases} \text{ACK}, \gamma > \gamma_0, \\ \text{NACK}, \gamma \leq \gamma_0, \end{cases} \tag{12.8}$$

where $\gamma_0 = \xi\bar{\gamma}$. In (12.8), ξ is a parameter used for threshold optimization, while $\bar{\gamma}$ is the average SNR of an active user set over a longer time scale and may vary between user sets in different geographical locations. The threshold γ_0 is the same for all competing users and is set in the BS on a long-term basis. The scheduled user is selected from the set of users that send an ACK.

If none of the users sent a positive ACK, the served user is randomly selected. The feedback BER is denoted by q.

Maximum SNR strategy. In practical systems we can use 2^k-state quantization with k-bit scheduling feedback. Then we divide active users into 2^k sets and select the user from the set with the highest received SNR. When the number of quantization sets grows to infinity, we obtain an ideal case where the SNR of each user is perfectly known at the BS. This asymptotic case is known as Maximum SNR scheduling.

The scheduling strategies explained above represent the three main cases of the usage of channel state information. Namely, the Round Robin is independent from channel state, the Maximum SNR assumes perfect channel knowledge at the transmitter and the On-Off strategy uses the scarcest possible channel feedback: one bit. Thus, by comparing these three strategies, the impact of feedback on system performance can be properly assessed.

The practical justification for reduced feedback approaches like the On-Off strategy is based on the fact that control channels carrying short-term channel information have very strict time delays and overhead constraints, as well as high robustness requirements. Using strong error-correction codes on the feedback channel is not a favorable solution, because it may critically increase signaling delay and overhead.

Generalizations of the one-bit quantization scheme and corresponding capacity expressions have been derived in [60] assuming that only the users, whose CQIs exceed a given threshold, report their unquantized CQIs to the base station. Quantization examples of CQI, together with max. SNR scheduling were studied in [85], but no clear explanation was given of how to select the quantization levels. Furthermore, capacity or block error rate maximizing quantizations within max. SNR scheduling, which depend on the SNR, were considered in [55] to show that two-bit quantization obtains 90% of the unquantized CQI capacity in the example scenarios. The quantization regions were found using numerical optimization. Furthermore, Bit-Error Probability (BEP) expressions for general CQI quantization were derived in [68] showing that when outage is allowed, two-bit quantization already provides most of the gains of maximum SNR scheduling, even when the quantization is not optimized with respect to BEP. Such a quantization simplifies system design, because it does not depend on the SNR operation point of different users.

In case of MISO and MIMO systems, CQI feedback consisting only of received SNR is not able to achieve the boundary of the multiuser capacity region. A natural way of extending the maximum SNR scheduler to work with MIMO systems is to report a received SNR from each transmit antenna, and transmit M parallel data streams so that each user receives a single data stream at a time [140], where M refers to the number of transmit antennas. This requires M times more feedback than the MISO case. However, signaling of eigenvectors is avoided, because the basis is provided by the transmit antennas. Another way of extending this scheduler is to take into account the type of the receiver [73, 77] when generating CQI feedback. [73] observes that the scheduler's maximizing capacities chooses the user with "the most invertible" instantaneous channel covariance matrix, when there is a large number of users and their channel statistics are the same. However, this is not necessarily true if the statistics are different. A similar approach is used by [77], whereby space-time coding and a corresponding equivalent channel matrix are used to calculate CQI feedback instead of the conventional channel matrix.

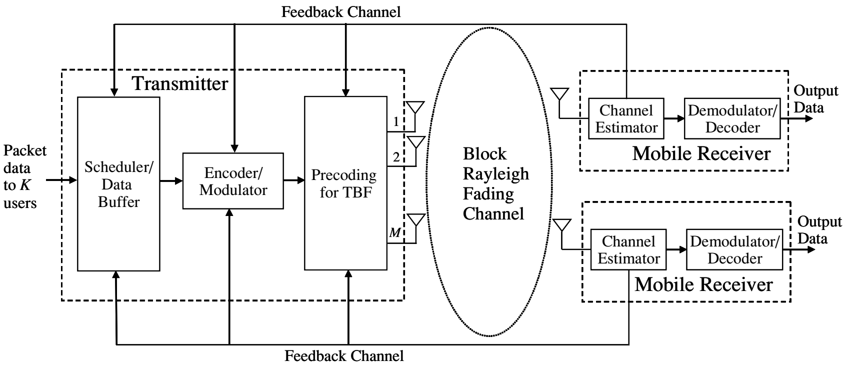


Figure 12.5: Model of the transmission system.

13 Performance of On-off scheduling strategy in presence of transmit beamforming

Practical signaling techniques to reduce feedback overhead were developed and simulated during 3GPP standardization. Yet in most cases it is difficult to analytically describe the performance of practical transmit beamforming and scheduling methods. Reference [109] provides a thorough survey of these limited feedback issues in wireless systems. Transmit beamforming has previously been analyzed in [38,64,65,123], but the effects of different scheduling strategies were not considered in these studies. It should be noted that transmit beamforming can be also used for implementing spatial multiple access [84] instead of scheduling, i.e. time division. On the other hand, limited threshold-based feedback is proposed in [71] for reducing overhead with any scheduling scheme, but the concept is not combined with transmit beamforming.

A significant step towards combining transmit beamforming and user selection is taken in [155]. We analyze the performance of joint transmit beamforming and scheduling by considering the lower and upper bounds for the scheduling gain [80]. As a reference, we use the performance of Round Robin scheduling, where radio resource usage is decided independently from the channel state. On the other hand, the upper bound for performance is defined by Maximum SNR scheduling, where the transmitter admits a perfect CQI for all users. We investigate the performance of the limited feedback scheduler by assuming that the On-off strategy, where scheduling decisions are based on one-bit CQI, is implemented. Thus, users report usage of one bit per channel and indicate to the transmitter whether their received SNRs are below or above a predefined threshold. If scheduling is successful, the scheduler finds a nonempty group of users whose SNRs are above the threshold, before randomly picking one of the users from the group and transmitting to it. The corresponding bit error probability (BEP) expressions were derived from [66], [67], while closed-form expressions for average capacity can be found in [69].

A one-bit scheduler has a lower performance bound than can be achieved with limited feedback.

It is not completely new result that joint beamforming and scheduling may result in significant gain. However, in previous work this problem has not been analyzed thoroughly when different types of beamforming methods are applied. For example, in case of OFDMA frameworks, the on-off type scheduling strategy was recently examined in [103] where feedback is defined based on resource blocks. The analysis by [103] focuses on the case where the quantization threshold is fixed but, on the other hand, the only transmit beamforming method they consider is antenna selection. We examine the performance of on-off scheduling with more advanced transmit beamforming methods, in which the decision threshold is optimized. We show that feedback errors may become a problem in this case and, furthermore, we provide performance results for the use of the HSDPA transmit beamforming method. Furthermore, we analyze the effect of feedback errors for scheduling. The applied performance analysis methodology was recently developed in [69] and [48].

13.1 Analysis

The primary intention of the analysis is not to provide absolute performance results, but rather our goal is to investigate the scheduling benefit of scarce relative SNR feedback, and track the impact of feedback errors on On-off scheduling.

We assume a feedback channel structure where uncoded control information is transmitted to the BS at each TTI in the dedicated control channel. For example, with UTRA FDD the fast PC is applied to the uplink control channel so that the control information is received at an approximately constant SNR, and it can be assumed that feedback bit errors are uniformly distributed over time. The effect of feedback latency is neglected in this study, but it is approximately valid within low mobility environments.

Finally, we note that the delay experienced by the user may be fixed in the first scheduling strategy to K TTIs, where K is the number of scheduled users. In second and third strategies the delay varies, the expectation being the same K TTIs. In practice, the variation of the delay depends on the queuing principles of the schedulers and type of services and is, therefore, outside the scope of this study.

We begin the analysis by stating the SNR distributions for applied transmit beamforming methods. Then we deduce the SNR distributions of the scheduling strategies we are investigating and, in last part of this section, we compute outage rates and discuss the results.

13.1.1 SNR distributions for Round Robin and Maximum SNR strategies

Since Round Robin strategy does not apply channel state information, the performance is defined by the underlying transmit beamforming method. Therefore, the first thing we consider for antenna selection are the distributions, Mode 1 and Mode 2. The SNR distribution for antenna selection should be kept in mind, see [17]. It is assumed that M are independent, exponentially distributed variables. Then, for the antenna selection case, the cumulative distribution function (CDF) and probability distribution function (PDF) admit the expressions

$$\begin{aligned} f_{\text{as}}(\gamma) &= \frac{Me^{-\gamma/\bar{\gamma}}}{\bar{\gamma}} \left(1 - e^{-\gamma/\bar{\gamma}}\right)^{M-1}, \\ F_{\text{as}}(\gamma) &= \left(1 - e^{-\gamma/\bar{\gamma}}\right)^{M}. \end{aligned} \tag{13.1}$$

When various performance measures are computed analytically, the binomial series expansion in (13.1) can be used for both the PDF and CDF.

For Mode 1 and 2 the distribution of $|\mathbf{h} \cdot \widehat{\mathbf{w}}|^2$ is difficult to deduce and, therefore, we use the approximation deduced in [70]. We start by noting that for the SNR there holds:

$$\gamma = |\mathbf{h} \cdot \widehat{\mathbf{w}}|^2 = \Big| \sum_{m=1}^{M} \widehat{u}_m \sqrt{\gamma_m}\, e^{j\phi_m} \Big|^2, \quad \gamma_m = |h_m|^2, \tag{13.2}$$

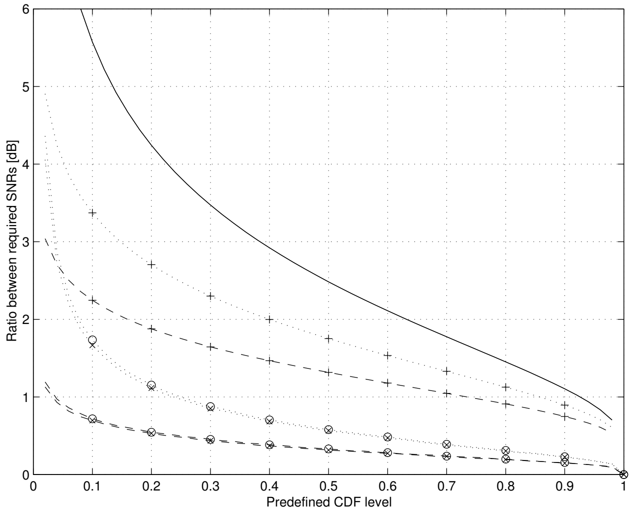


Figure 13.1: The ratio $\hat{\gamma}/\tilde{\gamma}$ as a function of predefined CDF level when $K = 2$ and single antenna transmission (solid curve), antenna selection (+), Mode 1 (x) and Mode 2 (o) are applied. Dashed curves correspond to case $M = 2$ and dotted curves correspond to case $M = 4$. The mean SNR is 10dB.

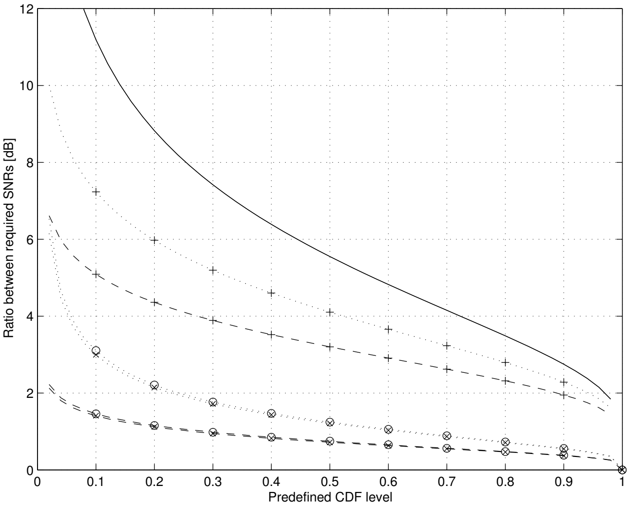


Figure 13.2: The ratio $\hat{\gamma}/\check{\gamma}$ as a function of the predefined CDF level, where $K = 8$ and single antenna transmission (solid curve), antenna selection (+), Mode 1 (x) and Mode 2 (o) are applied. Dashed curves correspond to case $M = 2$ and dotted curves correspond to case $M = 4$. The mean SNR is 10dB.

where $\phi_1 = 0$ and adjusted phases ϕ_m $(m > 1)$ are uniformly distributed over the interval $(-\pi/2^{N_{\text{rp}}}, \pi/2^{N_{\text{rp}}})$. Moreover, $\widehat{u}_m$ refers to the amplitude weight of antenna m, and the component SNRs γ_m is scaled such that $\mathrm{E}\{\gamma_m\} = \bar{\gamma}$. For SNR distribution, we need the so-called SNR gain G that is defined by:

$$G = \mathrm{E}\{|\mathbf{h} \cdot \widehat{\mathbf{w}}|^2\}. \tag{13.3}$$

The SNR gain provides information on the coherent combining gain caused by transmit weights. If there is no feedback, transmit weights are selected independently of channel state, resulting in:

$$\begin{aligned}\mathrm{E}\{|\mathbf{h} \cdot \widehat{\mathbf{w}}|^2\} &= \mathrm{E}\left\{\Big|\sum_{m=1}^{M} w_m \sqrt{\gamma_m}\Big|^2\right\}\\ &= \sum_{m=1}^{M} |w_m|^2 \mathrm{E}\{\gamma_m\} = \bar{\gamma}|\mathbf{w}|^2 = \bar{\gamma}.\end{aligned}$$

Thus without the channel information $G = 1$, there is no gain in the SNR. The SNR gain for Mode 1 and Mode 2 can be computed analytically and the resulting values can be found in [70] and [48], where it is shown that a good fit for the SNR distribution of Mode 1 and 2 is provided by a scaled χ^2-distribution with $2M$ degrees of freedom:

$$f_{\text{Mode n}}(\gamma) = \frac{1}{\Gamma(M)}\Big(\frac{M}{G_n\bar{\gamma}}\Big)^M \gamma^{M-1} e^{-M\gamma/G_n\bar{\gamma}}. \tag{13.4}$$

Here subscript $n \in \{1, 2\}$ refers to mode number. The CDF for SNR is obtained after integrating (13.4). The result is given by:

$$F_{\text{Mode n}}(\gamma) = \frac{1}{\Gamma(M)}\Upsilon\left(M, M\gamma/G_n\bar{\gamma}\right), \tag{13.5}$$

where Υ is the incomplete gamma function defined by:

$$\Upsilon(a, z) = \int_0^z t^{a-1} e^{-t} dt, \tag{13.6}$$

see [13], (6.5.1).

For the Round Robin strategy, the PDF and CDF formulaes given in Eqs. 13.1, 13.4 and 13.5 apply as such.

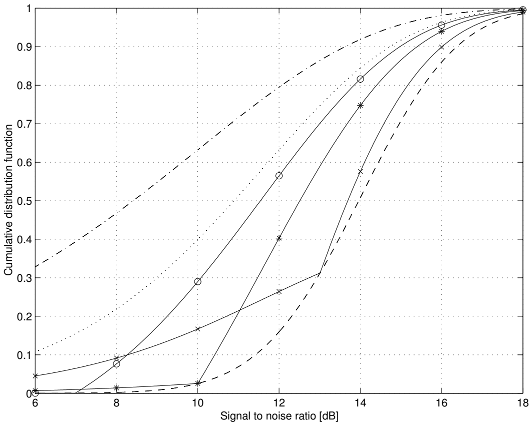


Figure 13.3: Cumulative distribution function for SNR when $\bar{\gamma} = 10$dB and antenna selection with $M = 2$ is applied. Dotted curve: Round Robin strategy. Solid curves: On-off strategy with 4 users and threshold $\xi = -3$dB (o), $\xi = 0$dB (*) and $\xi = 3$dB (x). Dashed curve: Max SNR strategy. The dash-dot curve refers to the case of a continuous single antenna transmission.

In the case of the Maximum SNR strategy, the BS is perfectly aware of the relative SNR of all K users when the served user is selected. Then the PDF and CDF of the received SNR on the shared channel are given by:

$$\begin{aligned} f_{\text{Max}}(\gamma) &= K \cdot f_{\text{TBF}}(\gamma) \cdot F_{\text{TBF}}(\gamma)^{K-1}, \\ F_{\text{Max}}(\gamma) &= F_{\text{TBF}}(\gamma)^{K}, \end{aligned} \tag{13.7}$$

where the subscript on the right refers to transmit beamforming.

We recall that the Round Robin strategy provides the lower bound for the performance of the On-off strategy, while the Maximum SNR strategy provides the upper bound. Furthermore, it is acknowledged that the benefit of channel-aware scheduling increases when number of scheduled users increases. On the other hand, if the number of transmit antennas increases, the deviation of the received signal becomes smaller and the available gain from channel-aware scheduling decreases. In order to illustrate the available gain from channel-aware scheduling we set:

$$F_0 = F_{\text{RR}}(\check{\gamma}),\ F_0 = F_{\text{Max}}(\hat{\gamma}),$$

where F_0 is a predefined CDF level, and $\check{\gamma}$ and $\hat{\gamma}$ are the SNRs that are needed in the Round Robin and Max SNR strategies respectively to reach the predefined CDF level. In Fig.13.1 ($K = 2$) and Fig.13.2 ($K = 8$) we have plotted the ratio $\hat{\gamma}/\check{\gamma}$ as a function of the predefined CDF level for single antenna transmission, Mode 1 and Mode 2 where $M = 2$ and $M = 4$.

Figure 13.1 and Figure 13.2 also provide an insight into the performance of the On-off scheduling strategy, since the gain from the On-off strategy should be less than the gain from the Maximum SNR strategy. Thus, according to Figure 13.1, in a two-user system we can obtain at most a scheduling gain of around 5.5dB, when considering 10%-tile of CDF and $M = 1$. If there are two transmit antennas, then the 10%-tile gains are around 2.3dB at most for antenna selection, and around 1.7dB for Mode 1 and 2. When using transmit beamforming, Maximum SNR scheduling does not bring much benefit in a two-user case, but gain increases rapidly with additional users. In eight-user system shown in

Figure 13.2, a gain of around 11.0dB can be achieved at 10%-tile of CDF with $M = 1$. With two-antenna transmission, gains of around 5dB can be reached for antenna selection, and around 3dB for Mode 1 and 2.

It should be emphasized that, although transmit beamforming is reducing the gain from scheduling, it improves the overall system performance and is, therefore, a justified technique from the system perspective. However, the results shown in Figure 13.1 and Figure 13.2 indicate that using accurate feedback for channel-aware scheduling is not necessarily reasonable if transmit beamforming has already been applied in the system. Instead, it is more effective to use very scarce scheduling feedback and to design the scheduler such that feedback is used to the maximum extent possible.

13.1.2 SNR distribution for the On-off strategy

Considering the On-off strategy without feedback errors, since the threshold in On-off scheduling is denoted by γ_0, the probability of the user sending a NACK is:

$$P(\text{NACK}) = F_{\text{TBF}}(\gamma_0),$$

where F_{TBF} refers to the CDF of the applied transmit beamforming method. Then, the probability of ACK is $1 - F_{\text{TBF}}(\gamma_0)$ and, according to Bayes' theorem, the PDF for the SNR of the on-off strategy attains the form:

$$\begin{aligned} f_{\text{on-off}}(\gamma) = (1{-}P(\text{NACK})^K) \cdot f_{\text{TBF}}(\gamma|\text{ACK}) \\ + P(\text{NACK})^K \cdot f_{\text{TBF}}(\gamma|\text{NACK}). \end{aligned} \tag{13.8}$$

Here the conditional PDFs in Eq. 13.8 are of the form:

$$\begin{aligned} f_{\text{TBF}}(\gamma|\text{ACK}) &= u_{\gamma_0}(\gamma) f_{\text{TBF}}(\gamma)/P(\text{ACK}), \\ f_{\text{TBF}}(\gamma|\text{NACK}) &= (1 - u_{\gamma_0}(\gamma)) f_{\text{TBF}}(\gamma)/P(\text{NACK}), \end{aligned} \tag{13.9}$$

where u_{γ_0} is a step function, such that:

$$u_{\gamma_0}(\gamma) = \begin{cases} 1, \ \gamma > \gamma_0 \\ 0, \ \gamma \leq \gamma_0 \end{cases}. \tag{13.10}$$

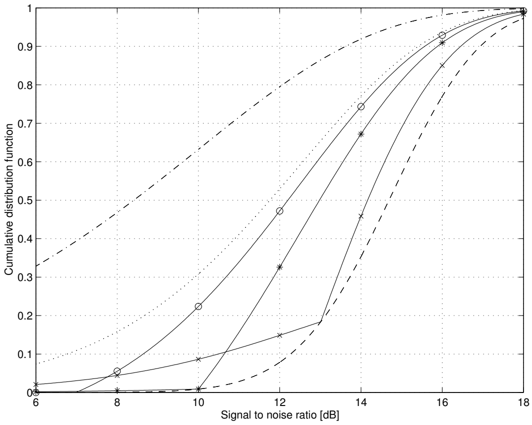


Figure 13.4: Cumulative distribution function for SNR when $\bar{\gamma} = 10$dB and Mode 1 with $M = 2$ is applied. Dotted curve: Round Robin strategy. Solid curves: On-off strategy with 4 users and threshold $\xi = -3$dB (o), $\xi = 0$dB (*) and $\xi = 3$dB (x). Dashed curve: Max SNR strategy. The dash-dot curve refers to the case of a continuous single antenna transmission.

Thus, the SNR distribution in Eq. 13.8 depends on the probability $P(\text{ACK})$ of erroneous feedback, the number of users K and the SNR threshold γ_0, which is also visible in the conditional SNR distributions in Eq. 13.9 of individual users.

After integrating Eq. 13.8, we obtain the CDF the for On-off strategy. If $\gamma \leq \gamma_0$, then:

$$F_{\text{on-off}}(\gamma) = P(\text{NACK})^{K-1} F_{\text{TBF}}(\gamma). \tag{13.11}$$

Otherwise:

$$\begin{aligned} F_{\text{on-off}}(\gamma) = & \frac{(1 - P(\text{NACK})^K)}{1 - P(\text{NACK})} \left(F_{\text{TBF}}(\gamma) - P(\text{NACK})\right) \\ & + P(\text{NACK})^K. \end{aligned}$$

In case of antenna selection and Mode 1, the SNR CDFs for the examined scheduling strategies are presented in Figure 13.3 and Figure 13.4 respectively. The results of Mode 1 and Mode 2 are very close to each other and, therefore, results have only been shown for Mode 1. It is found that Mode 1 outperforms antenna selection and the difference is between 0.5dB and 1dB for both the Round Robin and Max SNR strategies. If the On-off strategy is applied, then the gain from Mode 1 varies up to 2dB when $\xi = 3$dB, and low CDF percentiles are considered.

Note that the following holds true for the CDF On-off strategy:

$$F_{\text{on-off}}(\gamma_0) = F_{\text{Max}}(\gamma_0). \tag{13.12}$$

Thus, a one-bit On-off strategy is able to provide the same performance as a Max SNR strategy when $\gamma = \gamma_0$. This property is also visible in Figure 13.3 and Figure 13.4 and it can be used to optimize the performance of the On-off strategy for the selected operation point. Yet, if feedback errors occur, then Eq. 13.12 is no longer valid. This will be discussed in the below.

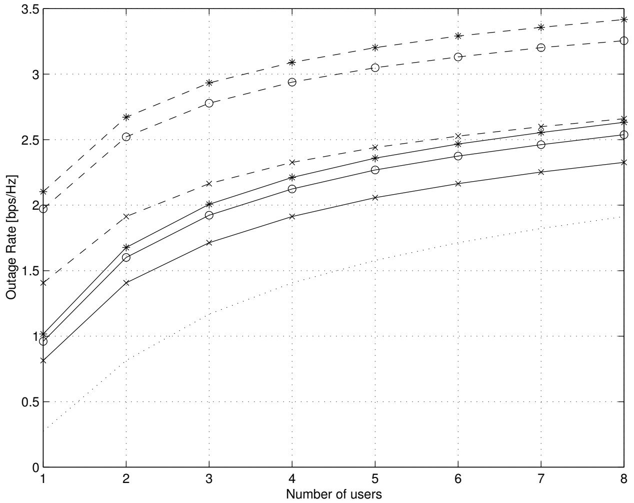


Figure 13.5: Outage rate for the Max SNR scheduling strategy as a function of the number of users, when $P^{\text{out}} = 0.1$ and the mean received SNR is 3dB. The numbers of transmit antennas are $M = 1$ (dotted curve), $M = 2$ (solid curves) and $M = 4$ (dashed curves) and antenna selection (x), Mode 1 (o) and Mode 2 (*) are applied in cases $M = 2, 4$.

13.1.3 On-off strategy in the presence of feedback errors

When ACK/NACK feedback errors corrupt the decisions in the BS we have:

$$f^{\text{err}}_{\text{TBF}}(\gamma|\text{ACK}) = \frac{((1-q)u_{\gamma_0}(\gamma) + q(1-u_{\gamma_0}(\gamma)))\, f_{\text{TBF}}(\gamma)}{P^{\text{err}}(\text{ACK})},$$
$$f^{\text{err}}_{\text{TBF}}(\gamma|\text{NACK}) = \frac{((1-q)(1-u_{\gamma_0}(\gamma)) + qu_{\gamma_0}(\gamma))\, f_{\text{TBF}}(\gamma)}{P^{\text{err}}(\text{NACK})},$$

where q is the ACK/NACK error probability and the denominator gives the ACK/NACK probability in BS in the presence of feedback errors. There are two possible events that lead to a positive ACK decision in the BS: the SNR received by the MS is above the given threshold and received feedback in the BS is correct, or the received SNR is below the threshold, but the feedback is corrupted and the BS receives ACK. Analogously, a NACK decision in the BS precedes two similar events. Thus:

$$P^{\text{err}}(\text{ACK}) = (1-q)P(\text{ACK}) + qP(\text{NACK}),$$
$$P^{\text{err}}(\text{NACK}) = (1-q)P(\text{NACK}) + qP(\text{ACK}).$$

The final PDF for On-off scheduling with corrupted feedback is given by:

$$\begin{aligned} f^{\text{err}}_{\text{on-off}}(\gamma) = (1 - &P^{\text{err}}(\text{NACK})^K) \cdot f^{\text{err}}_{\text{TBF}}(\gamma|\text{ACK}) \\ &+ P^{\text{err}}(\text{NACK})^K \cdot f^{\text{err}}_{\text{TBF}}(\gamma|\text{NACK}). \end{aligned} \tag{13.13}$$

The CDF is obtained by integrating over Eq. 13.13. If $\gamma \leq \gamma_0$, we have:

$$\begin{aligned} F^{\text{err}}_{\text{on-off}}(\gamma) = q \cdot &\frac{1 - P^{\text{err}}(\text{NACK})^K}{P^{\text{err}}(\text{ACK})} F_{\text{TBF}}(\gamma) \\ &+ (1-q) \cdot P^{\text{err}}(\text{NACK})^{K-1} F_{\text{TBF}}(\gamma). \end{aligned} \tag{13.14}$$

On the other hand, if $\gamma > \gamma_0$ there holds:

$$\begin{aligned} F^{\text{err}}_{\text{on-off}}(\gamma) = &\frac{(1 - P^{\text{err}}(\text{NACK})^K)}{P^{\text{err}}(\text{ACK})} \Big((1-q)(F_{\text{TBF}}(\gamma) \\ &- P(\text{NACK})) + qP(\text{NACK}) \Big) \\ &+ P^{\text{err}}(\text{NACK})^{K-1} \Big((1-q)P(\text{NACK}) \\ &+ q(F_{\text{TBF}}(\gamma) - P(\text{NACK}) \Big). \end{aligned} \tag{13.15}$$

Due to feedback errors the equality (13.12) is no longer valid. Instead, by Eq. 13.14 we have:

$$\begin{aligned} F^{\text{err}}_{\text{on-off}}(\gamma_0) =& F_{\text{Max}}(\gamma_0)\Big(q \cdot \frac{1 - P^{\text{err}}(\text{NACK})^K}{P^{\text{err}}(\text{ACK})P(\text{NACK})^{K-1}} \\ &+ (1-q) \cdot \frac{P^{\text{err}}(\text{NACK})^{K-1}}{P(\text{NACK})^{K-1}}\Big), \end{aligned}$$

where the factor in the brackets defines the impact of feedback errors.

13.1.4 Outage rate

It is also worthwhile to investigate system performance in terms of outage rate. Suppose we define the outage rate by the expression:

$$R^{\text{out}}(P^{\text{out}}) = A \cdot \log_2\left(1 + B \cdot \gamma(P^{\text{out}})\right), \tag{13.16}$$

where $\gamma(P^{\text{out}})$ is the SINR needed to achieve a given outage probability P^{out}, and the parameters A and B are the bandwidth and SNR efficiency factors used to fit the rate of the system with the set of adaptive modulation and coding curves obtained through system simulations. For example, it has been shown that values $A = 0.83$ and $B = 1/1.25$ provide a good fit with the set of LTE adaptive modulation and coding curves [114]. In this paper, we shall set the values $A = B = 1$ in order to provide the upper bound for the system transmission rate.

The value of $\gamma(P^{\text{out}})$, can be obtained as the solution to the following equation:

$$\begin{aligned} P^{\text{out}} &= P\left(\log_2(1+\gamma) < R_0\right) \\ &= \int_0^{\hat{\gamma}} f(\gamma)d\gamma = F(\hat{\gamma}). \end{aligned} \tag{13.17}$$

In this case, $\hat{\gamma} = 2^{R_0} - 1$ is the SINR related to the limit rate R_0, and the CDF is F_{TBF}, F_{Max} or $F_{\text{on-off}}$ depending on the applied scheduling strategy. Note that for a given P^{out}, the solution of Eq. 13.17 can be computed either numerically or analytically by using the

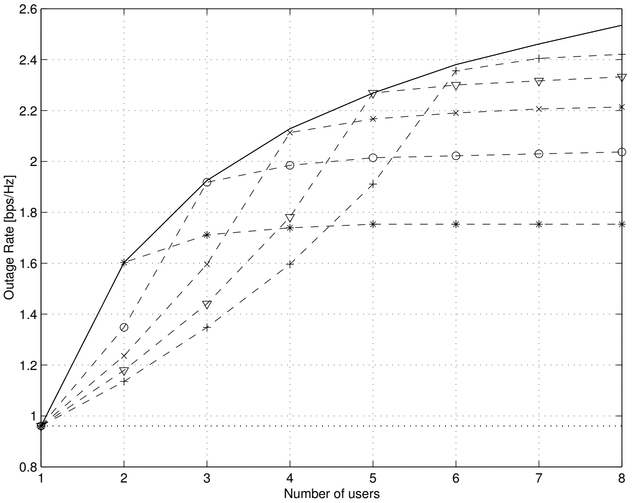


Figure 13.6: Outage rates for an error free on-off strategy (dashed curves), Max SNR strategy (solid curve) and Round Robin strategy (dotted line) as a function of the number of users when $P^{\text{out}} = 0.1$ and the mean received SNR is 3dB. The On-off strategy has been optimized for 2 (*), 3 (o), 4 (x), 5 (∇) and 6 (+) users. The underlying transmit beamforming method is the two-antenna Mode 1.

deduced formulae for CDFs. The analytical solution can be obtained if Round Robin or Maximum SNR scheduling strategies with antenna selection are assumed.

Figure 13.5 shows an outage rate for the Maximum SNR scheduling strategy, assuming a 10% outage and a mean received SNR of 3dB. In systems like HSDPA and LTE, the assumed 10% outage is a feasible criteria for a best effort data service on the cell edge, where SINR is typically a few decibels. Results show that Mode 1 and 2 clearly perform better than antenna selection, especially when four transmit antennas are applied.

It is worth of noticing that the maximum outage rate for the On-off strategy is obtained when the threshold is selected such that $\gamma_0 = \hat{\gamma}$ and $\xi = \hat{\gamma}/\bar{\gamma}$. Then, according to Eq. 13.11 and Eq. 13.12 we have:

$$\gamma_0 = F_{\text{TBF}}^{-1}\left((P^{\text{out}})^{\frac{1}{K}}\right). \tag{13.18}$$

Thus, we find that there is an optimal threshold for the On-off scheduling strategy for a given outage probability and number of users. The number of active users varies in a practical system and, therefore, it is not possible to fix the threshold. Instead, it is more feasible to tabulate a finite set of thresholds beforehand and broadcast the indices of new threshold values when load variations are large.

Outage rates for the On-off scheduling strategy are shown in Figure 13.6 and Figure 13.7, assuming a 10% outage and a mean received SNR of 3dB. Results are only given for the two-antenna Mode 1, since it is the most relevant method from the HSDPA perspective and, on the other hand, because results for antenna selection and Mode 2 provide the same conclusions regarding the On-off scheduling strategy.

In Figure 13.6 the On-off strategy has been optimized for different numbers of users according to Eq. 13.18. It is found that outage rate of the On-off strategy increases until the strategy reaches its optimal performance. Then performance increase saturates and only a very small gain is obtained, although there is an increase in the number of users in the system.

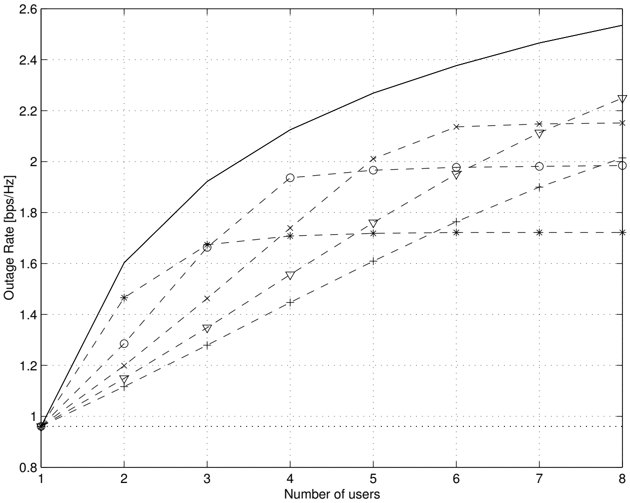


Figure 13.7: Outage rates for the on-off strategy (dashed curves), the Max SNR strategy (solid curve) and the Round Robin strategy (dotted line) as a function of number of users when $P^{\text{out}} = 0.1$, $q = 0.05$ and the mean received SNR is 3dB. The On-off strategy has been optimized for 2 (*), 3 (o), 4 (x), 5 (∇) and 6 (+) users. The underlying transmit beamforming method is the two-antenna Mode 1.

The feedback errors may have a strong negative impact to the performance of the On-off strategy. According to Figure 13.7 even a 5% error rate in feedback channel clearly reduces performance, especially when the threshold is optimized for large numbers of active users.

13.2 Conclusions

Results for the cumulative distribution of received SNR show that the use of transmit beamforming reduces the difference between upper and lower scheduling performance limits. Nevertheless, the joint usage of transmit beamforming and scheduling leads to a remarkable increase in performance. The variation in system performance due to different transmit beamforming methods was largest with low CDF percentiles, where Mode 1 and Mode 2 clearly outperformed antenna selection. At high CDF percentiles the differences were small.

It has been shown that the On-off scheduling strategy reaches its optimal performance when the received SNR and the decision threshold are equal. Thus, for this particular SNR value, the one-bit scheduling method is able to provide the same gain as the Maximum SNR scheduling strategy, which assumes perfect channel knowledge. In further work, the decision threshold could be optimized by taking the effect of feedback errors into account. When focusing on outage rates it was also found that system performance can be optimized for a certain number users when outage probability is fixed. This is interesting from a wireless systems design perspective since with LTE-A, for example, the performance requirements on the cell edge are given in terms of outage probabilities and rates. However, when the decision threshold is optimized for a certain number of users, additional users do not provide any noticeable increase in gain. Conversely, with the Maximum SNR scheduling strategy, an increasing number of users leads to a logarithmic increase in outage rate. Finally, results also show that even a small feedback error rate may seriously degrade the on-off scheduling performance. The feedback errors decrease the outage rate saturation level as a function of the number of users. This can be seen especially when outage probability is low.

14 Energy-adaptive scheduling in infrastructureless 4G networks

In hybrid 4G networking scenarios, the network architecture consists of both infrastructure-based and infrastructureless parts. Infrastructureless networks can be any type of relay, adhoc or mesh network. Since MTs are small battery powered devices with limited energy capacity, energy consumption is one of the most critical technical problems that need to be solved in this context. This chapter describes relevant energy consumption, energy models and energy saving mechanisms designed for infrastructureless networks and presents our novel cross-layer energy-adaptive scheduling and queue management framework EAED (Energy Aware Early Detection) [121] for minimizing energy consumption in WLAN mesh networks.

14.1 Energy consumption in wireless devices

In wireless devices energy is consumed in a Power Amplifier (PA) on the transmitter side, as well as in Application-specific Integated Circuit (ASIC), the Host Central Processing Unit (CPU) and the display etc. ASIC is used to run algorithms and MAC protocols while the Host CPU is responsible for running applications and other protocols. Figure 14.1 depicts the interfaces between these components.

14.2 Energy saving mechanisms

In energy constraint mesh points, the scheduling and transmission strategies used should be selected to be as energy efficient as allowed by quality of service constraints. Several different approaches have been suggested, including Transmission Power Control (TPC),

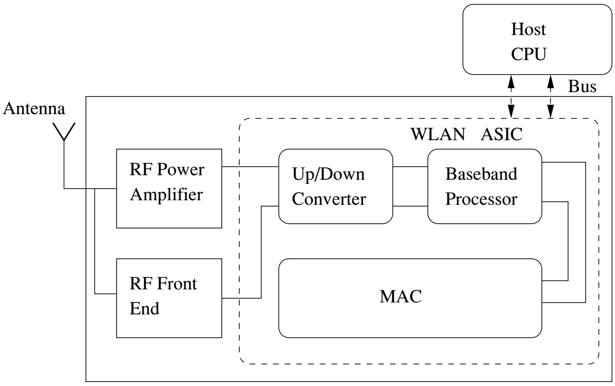


Figure 14.1: Wireless Network Interface (WNI)

energy-efficient routing, Adaptive modulation and coding (AMC), lazy scheduling and sleeping-based mechanisms. This section reviews the most important mechanisms, along with our own approach.

14.2.1 Transmission Power Control

TPC, combined with scheduling and the selection of routes, can be utilized to minimize energy consumption caused by packet transmissions. The basic idea behind transmission power control is to select the smallest possible transmission power, such that the received power is just above the receiver sensitivity level. [88] proposes a very simple distributed scheme where MTs are allocated a minimum transmission power to keep the network connected. However, in [126] the authors argue that a minimum transmission power strategy might not always result in maximum throughput. Instead, they believe that optimal transmission power should be determined based on network load, the number of stations and the network area. The authors propose two adaptive power control schemes, Common Power Control (CPC), where all MTs use the same transmission power, and Independent Power Control (IPC), where MTs can independently choose the transmission power. IPC requires a mechanism for synchronizing the transmission powers between MTs. This kind of a mechanism is available, for example, in the 802.11h standard. Both CPC and IPC operate to achieve the lowest possible contention time in the network. Contention time is observed through periodic measurements during an epoch T and transmission power is either increased or decreased when a certain contention time threshold is reached. A significant amount of other work has been done on the use of transmission power control and the joint definition of routing and power levels in ad hoc networks, see for example [50].

In multi-hop networks, the amount of tranmission power utilized will have an impact on network topology, which then affects routing [44], [83], [54]. In [83] it is assumed that paths between sources and destinations are given, and the problem is how to divide the data flow among the paths and how to set transmission times, rates and powers of all the links along the path to maximize network lifetime. This is equivalent to maximizing the operating time of the worst network node. The authors investigate the relationships between different metrics and conclude that, if there is no power threshold in the link, using a low transmission power does not lead to optimal energy consumption due to increased transmission times and, as a result, increased interference. Overall, it is most beneficial to use higher transmission power, so that less energy is needed to support the required average data rates in the network.

It should be noted that routing in adhoc networks is a challenging task due to mobility, time-dependent wireless channels and interference coming from other nodes. Thus, other solutions for minimizing energy consumption should be considered, to be used either independently or alongside routing.

14.2.2 Adaptive modulation and coding

Most transmission power literature assumes that the data rate is fixed. Some methods have utilized multiple modulation and coding schemes supported, for example, by IEEE

802.11g, a, h and n air interfaces. [131] proposes a MiSer energy-saving scheme, which uses an optimal, adaptive rate-power combination table in order to maximize energy efficiency for each data transmission attempt. Energy efficiency is defined as the ratio between the expected delivered data payload and the expected total energy consumption. The rate-power table is computed offline for different data transmission status quaduplets, consisting of data payload length, path loss condition and frame retry counts. The assumption is that simply transmitting at a lower power or using a higher PHY layer rate does not necessarily decrease energy consumption, since transmission would be more likely to fail and re-transmissions would eventually lead to a higher overall energy consumption. In addition, transmitting at different power levels could possibly aggravate the well-known hidden node problem. In MiSer this problem is solved by exchanging RTS/CTS frames to reserve the medium for data transmission. One disadvantage of MiSer is that it requires a knowledge of network configuration in terms of the number of contending MTs and the RTS collision probability, as well as a wireless channel model. In their analysis, the authors assume that the required information is available a priori. Furthermore, BER is the only QoS criteria considered.

When considering real-time traffic, such as VoIP or video streaming, modulation scaling [132] or lazy scheduling [59], [148], [91], [158] can be used to select the modulation and coding method, such that energy consumption is minimized while packet delay bounds are enforced. Typically, the lower the order modulation used, the less power is needed to transmit the packet with a given packet error rate. [59] defines the lazy scheduling problem as follows:

Supposing M packet arrivals at the transmitter at random times t_i during the interval $[0, T]$ destined for n receivers and given:

1. *a vector of packet arrival times $\{t_i, i = 1, ..., M\}$, where $t_1 = 0, t_i < t\{i+1\}$ and $t_M < T$, and*

2. *energy functions $w_i(\tau)$ which, for each $i \in \{1, ..., M\}$ are non-negative, monotonically decreasing and strictly convex functions of τ;*

find a feasible schedule so as to minimize the total transmission energy: $\sum_{i=1}^{M} w_i(\tau_i)$.

The authors find that there is no explicit solution for this problem and propose a practical MoveRight Algorithm to solve it iteratively. The idea of the MoveRight algorithm is to move the starting times of packet transmissions iteratively to the right, packet by packet, so that each move optimizes the energy function locally. It is shown that iterative local optimization also eventually leads to a global optimum.

Only theoretical work has been done so far on modulation scaling and lazy scheduling. [130], [148] and [91] propose optimal packet transmission schedules for simplified channel models, under strict assumptions about traffic arrival process. [132] and [158] do not make such strict assumptions about incoming traffic, but they also fail to consider the relevant MAC and physical layer effects of specific radio technology. These effects can considerably increase the total delay experienced by packets and, therefore, scheduling decisions should not only be based on expected transmission times.

14.2.3 Sleeping

Besides transmission power control, the best way to save energy is by switching off the radio. There are several different sleeping options. With MAC layers, a state of the art energy saving scheme is the X-MAC [33], in which the nodes are set to sleep in a random fashion. A node wishing to transmit a packet sends multiple RTS type packets until the receiver wakes up and replies with CTS. Another approach is to control sleeping using the transmit buffer [51]. [122] provides a theoretical framework for combining both dynamic modulation scaling and sleeping. [86] and [142] propose simple policies to adaptively force a WLAN device into doze mode at selected moments. However, sleeping mechanisms might not be feasible for real-time traffic that requires strict delay bounds, since they would require a very frequent synchronization procedure to wake up the nodes and initiate communication.

14.2.4 Queue management

Another method for saving energy is dropping packets early, well before the battery empties. As far as the authors are aware, the idea of energy-aware early dropping has not been proposed before. The only related solution can be found in queuing theory: a concept called "impatient customers" that refers to the case where customers leave the system after their patience runs out, even if they have not received any service. In a wireless context, this is equivalent to mesh points dropping out any packet that cannot meet its deadline. The idea of early dropping does not just involve dropping those packets that cannot meet their deadline in the given link, but also the dropping of packets to prevent congestion, as in RED.

14.2.5 Our approach: an Energy Aware Early Detection (EAED) framework for 802.11s

Our goal was to design a scheduling and queue management framework for a use case where real-time VoIP and video conferencing applications are used in a 802.11s mesh network. These applications set maximum packet delay and packet loss limits, while small, battery-powered MTs set requirements for high energy efficiency. This kind of use case could become a reality in developing countries or when coping with emergency. Our approach uses the energy models presented in the following section as a basis for designing a new, distributed energy-adaptive cross-layer scheduling and queue management framework (EAED) to enable this use case.

The aim of EAED is to save energy by selecting an appropriate modulation and coding scheme under the constraints on packet delay. Transmission power is chosen based on the modulation scheme, code rate and channel conditions. In addition to adapting the modulation, code rate and transmission power, the EAED framework applies early packet dropping to ensure availability of energy resources in the future. EAED does not make any assumptions about traffic arrival processes or channel conditions, contrary to many previously proposed approaches. Most importantly, EAED realistically considers the effects of the physical and MAC layers of the 802.11s on total delay, whereas other schemes have only based scheduling decisions on the expected transmission time. Since our scheduling scheme is based on real-time estimations it can dynamically adapt to the effects of radio technology as well as different traffic arrival patterns.

Used energy model

In Section 14.1, energy consumption in a wireless device was shown to depend on different components, such as the Power Amplifier (transmitter side), the radio electronics (ASIC), the Host CPU and the display. All of these components should be modeled in order to fully analyze the energy efficiency of protocols and algorithms. However, most of the existing work concerns the measurement of total energy consumption in wireless network interfaces [146], [49], without considering the effect of components independently. In addition, the effects of different radio parameters such as modulation, coding and propagation models have not been analyzed sufficiently accurately. Furthermore, these measurements have been performed with Laptops or PDAs, not with mobile devices or Internet tablets. As no feasible experimental energy model exists, we will use the theory of dynamic modulation scaling (dms) [132] to model energy consumption.

The scheduler part of our framework aims to save energy by delaying packets so that transmission power consumption can be decreased correspondingly. Theoretically the operation of our algorithm is based on the principles of dynamic modulation scaling (dms).

The modulation level (number of bits per symbol) is denoted by b. The average time to transmit one bit (T_{bit}) with a selected modulation level and symbol rate R_S can be expressed as:

$$T_{bit} = \frac{1}{b \times R_S}. \tag{14.1}$$

The energy consumed for transmitting one bit (E_{bit}) is then given by [132]

$$E_{bit} = (P_{tx} + P_E) \times T_{bit}, \tag{14.2}$$

where P_{tx} denotes the power consumed by the Power Amplifier (PA) for transmission and P_E denotes the power consumed in electronic circuitry.

From Eq. 14.2 it becomes obvious that transmitting at maximum modulation level is an optimal strategy in terms of energy if a constant power is used. However, if P_{tx} can be controlled, delaying packets becomes more beneficial.

Firstly, we will derive expressions for P_{tx} and P_E, assuming for simplicity that the QAM modulation method is used. BER, signal to noise ratio (SNR) and noise power P_N are given by [132].

$$BER = \frac{4}{b}\left(1 - \frac{1}{2^{\frac{b}{2}}}\right) \cdot Q\left(\sqrt{3 \cdot \frac{SNR}{2^b - 1}}\right) \tag{14.3}$$

$$SNR = \frac{P_{tx}}{P_N} \cdot A \tag{14.4}$$

$$P_N = N_0 \cdot \beta \cdot R_S, \tag{14.5}$$

where A symbolizes all transmission loss components, N_0 denotes the noise power spectral density and β is a factor that takes into account other elements such as filter non-idealities. Manipulating these equations yields:

$$P_{tx} = C_S \cdot R_S \cdot (2^b - 1) \tag{14.6}$$

$$C_S = \frac{N_0 \cdot \beta}{A} \cdot \Gamma \tag{14.7}$$

$$\Gamma = \frac{1}{3}\left[Q^{-1}\left(\frac{1}{4} \cdot \left(1 - \frac{1}{2^{b/2}}\right)^{-1} \cdot b \cdot BER\right)\right]^2. \tag{14.8}$$

Assuming a fixed symbol rate, the expression for the power consumed in the electronic circuitry is given by

$$P_E = (C_E + C_R) \cdot R_S, \tag{14.9}$$

where factors (C_E,C_R) $\propto V^2$ (operation voltage). Combining the expressions derived above, the amount of energy consumed for transmitting one bit is given by

$$E_{bit} = C_S \times \frac{2^b - 1}{b} + (C_E + C_R) \times \frac{1}{b}, \tag{14.10}$$

where the first part describes the energy required to generate electro-magnetic waves that carry information and the second part describes the rest of the radio's energy consumption.

The total transmission energy consumption as a function of transmission time is a monotonically decreasing, convex function, as depicted in Figure 14.2 (R_S = 250kHz, C_S = 100 nJ and $C_E + C_R$ = 180 nJ, corresponding to the implementation of an adaptive QAM system). Thus, the more time used for transmitting packets, the more energy can be saved, assuming that transmission power is adapted according to Eq.14.6. Figure 14.2 also shows total energy consumption (transmitter and receiver) with arbitrarily selected receiver parameters. Depending on the receiver parameters, total energy consumption may not decrease monotonically for the smallest values of b. However, this thesis focuses on energy consumption on the transmitter side when the device is operating in an active 802.11s mode.

14.2.6 EAED packet scheduler

802.11s specification physical and MAC layer features set boundary conditions for the operation of EAED. We have assumed a physical layer that corresponds to 802.11a/802.11g standards and a MAC layer that corresponds to 802.11e. Supported modulations and code rates are depicted in Table 14.1. The modulation and code rate must be selected

Table 14.1: Resulting data rates with different modulation and coding schemes

Modulation	**Code-Rate**	**Data Rate (Mbps)**	**Coded bits/ symbol**	**Coded data bits/ symbol**
BPSK	1/2	6	48	24
BPSK	3/4	9	48	36
QPSK	1/2	12	96	48
QPSK	3/4	18	96	72
16-QAM	1/2	24	192	96
16-QAM	3/4	36	192	144
64-QAM	2/3	48	288	192
64-QAM	3/4	54	288	216

from these methods and MAC layer effects must be considered when determining how much extra delay can be allowed by the EAED algorithm for energy saving purposes.

The EAED scheduler aims to find an optimal modulation and coding scheme (MCS) m. Energy consumption of packet k at hop h is denoted by $\tilde{E}_{k,h}$ and the total packet delay is denoted by $D_{k,h}$. In addition, the delay bound of a packet is denoted by D_{max}. The operation of EAED can be formalized as the following optimization problem:

$$\min_{m} E\{\sum_{h=1}^{H} \tilde{E}_{k,h}\} \tag{14.11}$$

s.t.

$$\sum_{h=1}^{H} D_{k,h} \leq D_{max}. \tag{14.12}$$

We used a heuristic approach to solve this problem, based on the theory presented in Section 14.2.5. We assume that whenever the MP is ready to transmit a packet, the optimal solution in terms of energy is to select the lowest possible modulation and code rate combination with respect to the constraint on packet delays.

It should be noted that all packets currently in the node's buffers must be transmitted within the delay bound. The estimated waiting time of all data packets in the node's buffers due to MAC layer mechanisms is denoted by $D_{mac,h}$, and the actual time for transmitting all the packets with a certain MCS by $T_{m,h}$. Additionally, $Pdus_h$ is used to denote the total number of data pdus in the node's buffers at hop h, $SIFS_h$ denotes the short inter-frame space used before sending an ACK, and $D_{ack,h}$ denotes the duration of ACK. It is assumed that a constant modulation scheme and code rate is used for acknowledgements. D_h can be calculated as follows:

$$D_h = T_{m,h} + D_{mac,h} + Pdus_h * (SIFS_h + D_{ack,h}). \tag{14.13}$$

$D_{mac,h}$ is calculated by weighting the estimated MAC delays $D_{mac(i),h}$ for each access category i with the number of data pdus in the access category. The number of access categories is denoted by AC.

$$D_{mac,h} = \sum_{i=0}^{i=AC} Pdus(i)_h * D_{mac(i),h}. \tag{14.14}$$

$D_{mac(i),h}$ is estimated in the MAC layer by the EWMA estimator. α denotes the filtering coefficient of the estimator, the next inter frame space in access category i is denoted by $IFS(i)_h$, and the value of the previous backoff timer by $B(i)_h$.

$$D_{mac(i),h} = \alpha * (IFS(i)_h + B(i)_h) + (1 - \alpha) * D_{mac(i),h}. \tag{14.15}$$

Delaying packets without violating the delay bound might not be possible during periods of heavy congestion. In this case the scheduler has to propose the highest possible MCS, even though it temporarily maximizes energy consumption. We also use the MP's own medium utilization estimate U (fraction of time when the medium senses that the physical layer is virtually busy or has ongoing transmission) as an indication of a decreased ability to delay packets. If $U > U_{max}$, a fixed MCS (16-QAM and 3 / 4) is used to avoid unnecessary collisions that might occur due to additional delaying.

In the physical layer, the MCS m proposed by the MAC layer must be checked against current channel conditions. The required transmit power for link h, with a path loss of L_h and using MCS m is $P_{tx,h,m} = \delta L_h P_{rx,m}$, where $\delta > 1$ is a power margin needed to compensate for time varying fading, changes in the noise power, and inaccuracy of the path loss measurement. The set $M_h = \{m : P_{tx,m} \leq P_{max}\}$ defines MCSs that can be utilized within the power budget P_{max} (20dBm).

An open loop transmit power control scheme is utilized by our scheduler. The required transmission power $P_{tx,m,h}$ is computed iteratively in set M_h as follows:

$$P_{tx,m,h} = \left(\frac{1.0}{L_h}\right) * \delta, \tag{14.16}$$

where:

$$\delta = RS_h * RSCoefficient(m,h) * RbFactor_h. \tag{14.17}$$

Modulation and/or code rate are decreased until the power constraint is met or the lowest m has been reached. The MP estimates the path loss L_h from the received frame. Other values required for computing δ are directly related to the 802.11 a/g radio: ReceiverSensitivity (RS_h) and the shadow facing margin $RbFactor_h$ are set when the MP is configured. The value used for $RSCoefficient(m,h)$ is selected by table lookup, based on the modulation and coding scheme proposed by the scheduler, as indicated in Table 14.2. The minimum transmission power was set to 2 dBm due to possible inaccuracies in the power adaptation algorithm.

The energy model used by our scheduler assumed a power adaptation logic corresponding to Eq. 14.6 in section 2. Our power control strategy is similar, but we only measure the value of the path loss, use coefficients and factors derived from commercial WLAN cards for other parameters.

Transmitting at different power levels could possibly aggravate the well-known hidden-node problem in wireless networks. If our scheme were to be used in large mesh networks, it should be combined with the mechanism of sending RTS/CTS packets with constant,

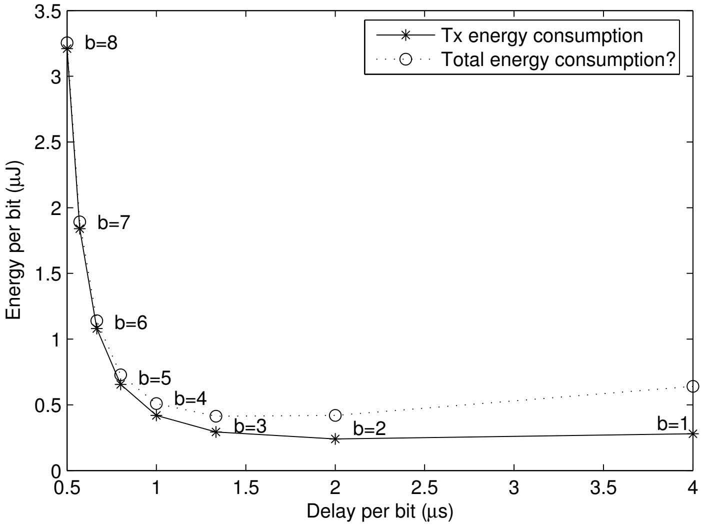


Figure 14.2: Energy versus transmission time

Table 14.2: RSCoefficients for different MCS for 802.11a/g radio

Modulation	**CodeRate**	**RSCoefficient**
BPSK	1/2	10.0
BPSK	3/4	10.0
QPSK	1/2	25.12
QPSK	3/4	100.0
16-QAM	1/2	158.49
16-QAM	3/4	300.0
64-QAM	2/3	795.0
64-QAM	3/4	800.0

sufficient power as proposed by authors in [131]. The authors thoroughly analyzed this kind of scheme and verified that adaptive transmission power does not exacerbate the hidden-node problem if combined with intelligent RTS/CTS transmission.

When analyzing the performance of EAED in the experimental part of the thesis, it was found that operations without an EAED scheduler were equivalent to the water-filling approach. With water-filling, energy consumption is minimized by transmitting packets as fast as possible while keeping a constant transmission power (20dBm). In practice, the highest possible MCS that can be supported by this power in current channel conditions is chosen.

EAED packet dropper

Several active queue management mechanisms have been devised for congestion avoidance in wireline networks. RED [56] and Weighted Random Early Detection (WRED) proactively drop packets based on filtered queue length to prevent congestion in the first place. Our EAED dropper follows the general RED paradigm but aims to avoid both the congestion and exhaustion of energy resources. The dropper uses the EWMA filtered modulation and code rate combination, proposed by the EAED scheduler in the MAC layer, as a measure of energy consumption: the higher the modulation and code rate, the higher the energy consumption. The EWMA filtered MCS m is denoted by $\tilde{m}$ s.t. (m, $\tilde{m}$) $\in$ (0,7). The $\tilde{m}$ value is updated in the MAC layer as follows:

$$\tilde{m} = \gamma * m + (1 - \gamma) * \tilde{m}, \tag{14.18}$$

where γ is a filtering coefficient. The early dropping probability $p(i)$ in access category i is an increasing function of $\tilde{m}$. It is assumed that there are two access categories, one for a streaming traffic type and the other for a VoIP traffic type. If $\tilde{m} < 2$, early packet dropping is not necessary, since the low modulation and code rate level alone guarantee low energy consumption. Otherwise $p(i)$ for streaming type traffic is given by:

$$p(i) = 1.5 * (\tilde{m} - 1), \tag{14.19}$$

and for VoIP type of traffic by:

$$p(i) = 0.5 * (\tilde{m} - 1). \tag{14.20}$$

An ACK is sent back in order to prevent MAC layer retransmissions of early dropped packets, even though the packet has not actually been received.

The EAED dropper has been designed specifically for real-time traffic. It is especially feasible for streaming type media, since the most advanced audio/video codecs can tolerate a packet loss as high as 15-20$\%$, depending on which frame types are dropped. The dropping process is even more important than the loss rate; randomized packet drops have a less severe effect on quality than consecutive drops due to deadline violation.

15 Simulations of EAED framework

In the previous chapters we presented several mechanisms for saving energy in wireless mesh networks and performed qualitative analyses on these mechanisms. We introduced our approach, a distributed and measurement-based energy-adaptive cross-layer scheduling and queue management framework (EAED) for supporting low energy consumption VoIP and Video traffic in 802.11s mesh networks. In this chapter the performance of the EAED is evaluated using simulations, and the topology, traffic types and other simulation parameters are described and the results are presented.

15.1 Performance evaluation of the EAED algorithm

15.1.1 Goals of the simulation study

The objective of this simulation analysis was to evaluate the energy saving potential and application performance of the EAED algorithm. The simulation tool used was WLANSim, a dedicated WISE library-based WLAN simulator developed at the Nokia Research Center. Compared with other WLAN simulation tools that are available, such as ns2, WLANSim provides support for more realistic physical layer models. For the purposes of this research work, WLANSim was upgraded to include mesh networking capabilities and EAED functionality.

The energy saving potential of our algorithm was tested in several scenarios to evaluate the effect of important parameters, such as the distance between nodes, the pathloss exponent and traffic load, on algorithm operation. The following three alternative packet handling principles were compared for all scenarios:

- No EAED, packets sent with maximum power limited water-filling approach
- EAED without Early Dropping (TailDrop)
- EAED with Early Dropping

Due to the lack of other schemes that are sufficiently similar, we did not compare EAED with other energy saving methods. Other methods are either based on sleeping, which is not suitable for real-time traffic, or lazy scheduling type approaches, which do not consider the dynamics of traffic arrival processes or the WLAN MAC and PHY layers.

15.1.2 Simulation topology and parameters

A network consisting of three MPs was used for the simulations. A small number of nodes can be justified by the fact that packet level scheduling and queue management mechanisms should be first evaluated in a setup that allows different parameters to be analyzed in detail. The small number of hops is also motivated by service quality constraints. See [139]. It was assumed that all MPs utilize the same frequency (802.11g radio with 2.4GHz) and compete for the same channel. The modulation and code rate adaptation logic of the EAED algorithm was used in all MPs, whereas the dropper was only implemented in the intermediate MP that acted as a router.

Two distance scenarios were defined, as depicted in Figure 15.1.

Propagation was supposed to be line-of-sight in the first distance scenario, whereas it was assumed that some obstacles were present in the transmission path in the second scenario. The type of propagation model used took into account the effects of pathloss, as well as fast and slow fading. Propagation conditions were simply modified by varying the MaxLOSRange and PathlossExponent parameters. The PathLoss was either 2.0 or 2.4. When the pathLoss was at 2.4, the MaxLOSRange value was set to 15.0 in distance scenario 1, and to 40.0 in scenario 2. ReceiverSensitivity was -73dBm and the RbFactor was set to 2.0 or 4.0 depending on the scenario.

Since EAED is primarily designed for real-time traffic, VoIP and Video were used as traffic types in the simulations. Random number generators were used to produce traffic patterns resembling the behavior of these applications. The VoIP packet interarrival time and packet size were constant, with means of 20 ms and 178 bytes, whereas Video used a normal distribution with parameters (20 ms, 6.7 ms) and (750, 250) bytes.

The simulation time was 37.5 million OFDM symbols to guarantee the reliability of results. It was verified that a longer simulation time would not have significantly increased the accuracy of results. The MAC layer parameters (AIFS, minCW, maxCW) for both access categories were set to reasonably small values to guarantee delay bounds without EAED. Traffic load was varied by changing the number of connections so that there were eight VoIP and Video connections in the first load scenario, and eight VoIPs and ten videos in the second scenario. Half of the VoIP and Videos were transmitted in an upstream direction and the other half transmitted downstream.

15.2 EAED algorithm simulation results

This section presents the results from the simulation scenarios. Extensive simulations have been performed to analyze the operation of EAED framework in terms of their end-to-end application performance and energy consumption. Conventionally, energy consumption has been modeled in a single node as a function of a selected modulation scheme, see [132]. We used a more holistic approach and studied energy consumption at the level of the network and of individual MPs. Our choice is justified by the fact that traffic processes and channel quality are highly variable in time in real networking scenarios. As a result, modulation, code rate and transmission power are interrelated and have to be adapted frequently, depending on traffic characteristics and channel conditions.

Table 15.1: Scenarios

	Distance Scenario	Load Scenario	Pathloss Exponent
Scen1	1	1	2.0
Scen2	1	2	2.0
Scen3	1	1	2.4
Scen4	2	1	2.0

Table 15.2: Energy saving potential

	Energy Saving			
	EAED NoDrop AC2	EAED Drop AC2	EAED NoDrop AC3	EAED Drop AC3
Scen1	50.38 %	52.47 %	59.26 %	57.99 %
Scen2	49.86 %	51.86 %	59.77 %	58.65 %
Scen3	43.72 %	45.87 %	49.80 %	50.37 %
Scen4	3.43 %	2.89 %	1.82 %	3.1 %

15.2.1 EAED algorithm performance

Firstly, we examined the energy saving potential of the EAED algorithm with and without early dropping. Table 15.2 shows the total energy saving in the whole network for both access categories (AC2=Video and AC3=VoIP). The scenarios in Table 15.2 are defined in Table 15.1.

The energy saving percentages in Table 15.2 were obtained by weighting the energy consumptions of individual MPs with the relative amount of traffic transmitted by these MPs.

Table 15.3: AC2 Energy consumption in Scenario1

	NoEAED	**EAED NoDrop**	**EAED Drop**
Energy, MP 3	0.86 J	0.46 J	0.46 J
Energy, MP 2	2.29 J	1.18 J	1.12 J
Energy, MP 1	1.53 J	0.64 J	0.64 J
Weighted energy	1.74 J	0.87 J	0.89 J
Energy saving		50.38 %	52.47 %

As an example, Table 15.3 shows detailed energy consumption results for each MP in scenario1 for Video traffic. It should be noted that the energy consumption of the EAED algorithm with a dropper only differs from the pure EAED algorithm due to MP2 acting as a router, since the dropper is not implemented in source and destination MPs.

It can be concluded that considerable energy savings of up to 40-60% can be obtained when the load level and distance are low or moderate. When the results are extrapolated it becomes evident that smaller loads and distances would allow even more energy to be saved. On the other hand, when distance is increased the energy saving potential drops to just a few percentage points, as can be observed from the results of scenario4. This is due to the fact that a sufficiently high transmission power must be used to guarantee the delay bound of real-time applications when attenuation increases. The same effect was observed when the algorithm was tested with increased load levels.

A small energy saving percentage is obtained by early dropping, compared to operation with pure EAED scheduling. There is only a slightly less amount of energy saved in scenario1 and scenario2, for VoIP traffic, when the dropper is used. It should be noted, however, that if EAED were also implemented in the source and destination MP, it is most likely that a greater amount of energy would be saved. Furthermore, in a larger network topology early dropping could result in more considerable energy savings, since a packet that might otherwise be dropped after having consumed resources in many hops could be dropped early by the first router along the path of the connection. Early dropping is feasible even with higher loads, if the packet loss caused by congestion is below the maximum packet loss requirement. This is not the case with delaying.

The energy saving depicted in Table 15.2 is possible due to adaptive transmission power, modulation and coding. Figure 15.2 (a) depicts uplink and downlink transmission power distributions for the EAED algorithm without early dropping in scenario1 and scenario2. Power distribution with an early dropper is almost the same as with a pure EAED, whereas the transmission power distribution without the EAED algorithm is a constant 20 dBm. The transmission power distribution for the EAED reveals that varying modulation and code rate levels are used by the algorithm, depending on MAC layer delays and channel quality. As an example, Figures 15.2 (b), (c) and (d) depict modulation and code rate distributions in each MP for both access categories, with and without the EAED (1=BPSK 1/2, 8=64-QAM 3/4), in scenario1. The corresponding transmission rate distribution could be derived from the values given in Table 14.1. It is evident that the EAED uses a smaller modulation and code rate whenever possible. Lower modulations and code rates consume more time for the actual transmission, but save energy due to smaller required transmit power.

Figure 15.3 depicts end-to-end delay distributions for both access categories, with and without the EAED algorithm, in all scenarios. The EAED algorithm increases end-to-end delays, but delays remain small enough to guarantee an acceptable QoS for applications in scenario 1 and scenario 2. Conversely, end-to-end delay becomes almost unacceptable, especially for VoIP traffic, in scenario 4 with longer distances.

Table 15.4, Table 15.5 and Table 15.6 depict link statistics as well as average packet losses and goodputs of connections in scenarios 1, 2 and 3. It can be observed that, in all scenar-

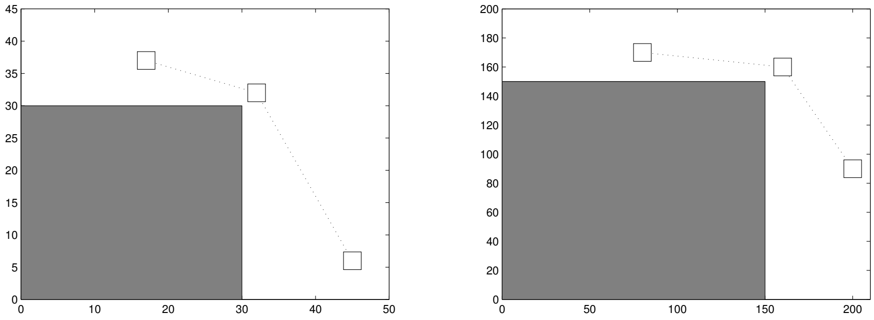
Table 15.4: Physical layer, packet loss and goodput statistics in scenario1

	NoEAED	**EAED NoDrop**	**EAED Drop**
T_{idle}	54.97 %	33.0 %	33.79 %
$T_{transmission}$	44.3 %	64.74 %	63.92
$T_{collision}$	0.73 %	2.26 %	2.29 %
AC2, $L_{EarlyDrop}$			4 %
AC2, L_{Drop}	2 %	2 %	7 %
AC3, $L_{EarlyDrop}$			1.13 %
AC3, L_{Drop}	3 %	3 %	5 %
AC2, goodput (Mbps)	2.36	2.35	2.24
AC3, goodput (Mbps)	0.55	0.55	0.54

ios, the EAED algorithm spends more time in a state where there is one ongoing transmission in the network. Correspondingly, more time is also spent in a collision state. Early drop probabilities and total packet losses (including early dropped packets) remained at a reasonable level, especially for Video traffic that could have allowed even more dropping assuming advanced codecs. Goodput was slightly better without the EAED algorithm.

Table 15.5: Physical layer, packet loss and goodput statistics in scenario2

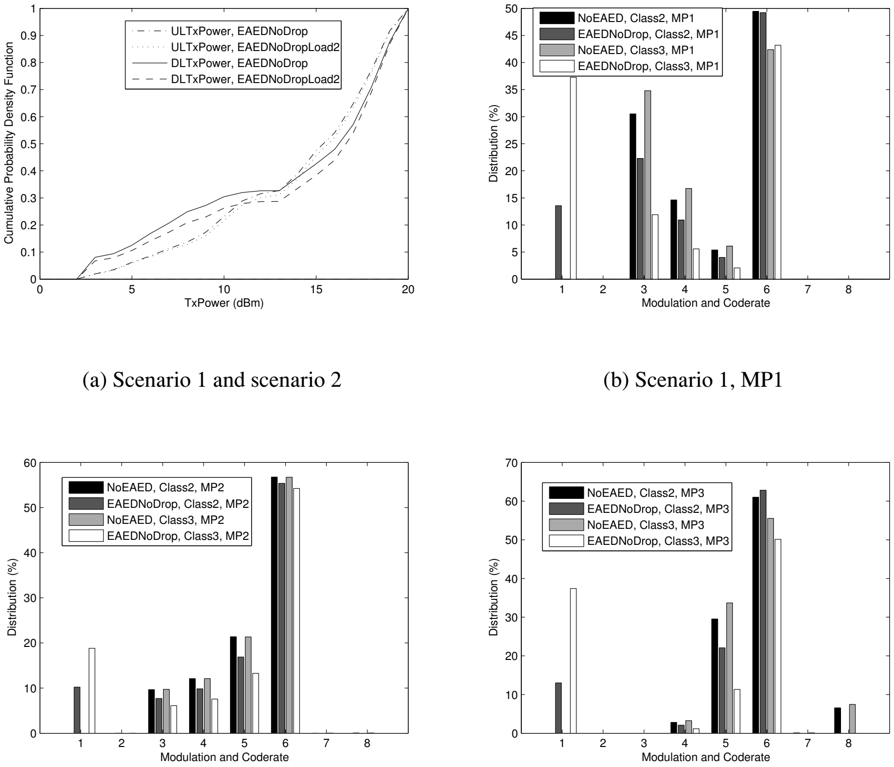
	NoEAED	**EAED NoDrop**	**EAED Drop**
T_{idle}	47.06 %	22.99 %	24.24 %
$T_{transmission}$	52.22 %	74.54 %	73.30
$T_{collision}$	0.73 %	2.47 %	2.46 %
AC2, $L_{EarlyDrop}$			4.3 %
AC2, L_{Drop}	2 %	3 %	7 %
AC3, $L_{EarlyDrop}$			1 %
AC3, L_{Drop}	3 %	3 %	5 %
AC2, goodput (Mbps)	2.95	2.93	2.79
AC3, goodput (Mbps)	0.55	0.55	0.55



(a) Distance scenario 1

(b) Distance scenario 2

Figure 15.1: Distance scenarios



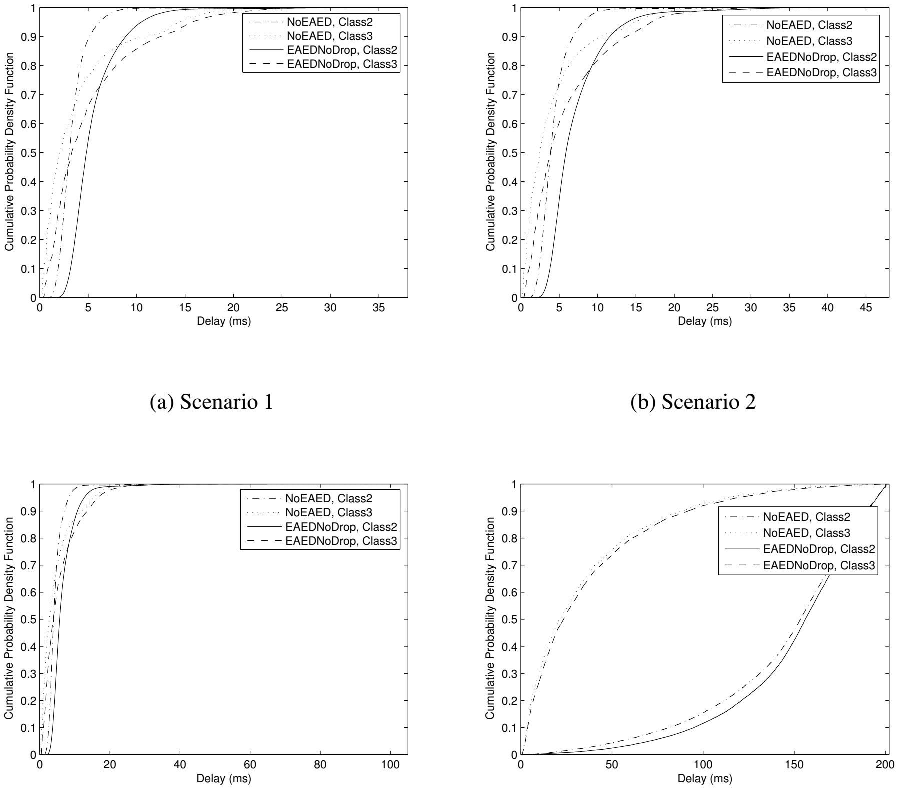
(a) Scenario 1 and scenario 2

(b) Scenario 1, MP1

(c) Scenario 1, MP2

(d) Scenario 1, MP3

Figure 15.2: Powers, modulations and coderates



(a) Scenario 1

(b) Scenario 2

(c) Scenario 3

(d) Scenario 4

Figure 15.3: Delay distributions in all scenarios

Table 15.6: Physical layer, packet loss and goodput statistics in scenario3

	NoEAED	**EAED NoDrop**	**EAED Drop**
T_{idle}	43.71 %	24.62 %	25.68 %
$T_{transmission}$	55.33 %	73.02 %	71.94
$T_{collision}$	0.95 %	2.37 %	2.38 %
AC2, $L_{EarlyDrop}$			4 %
AC2, L_{Drop}	3 %	3 %	7 %
AC3, $L_{EarlyDrop}$			1 %
AC3, L_{Drop}	3 %	3 %	5 %
AC2, goodput (Mbps)	2.35	2.34	2.24
AC3, goodput (Mbps)	0.55	0.55	0.54

16 Conclusions

Networking scenarios in the future will be complex and will include fixed networks and hybrid Fourth Generation (4G) networks consisting of both infrastructure-based and infrastructure-less, wireless parts. In such scenarios, adaptive provisioning and management of network resources becomes of critical importance. Adaptive mechanisms are desirable since they enable a self-configurable network that is able to adjust itself to varying traffic and channel conditions. The operation of adaptive mechanisms is heavily based on measurements: the state of the network is monitored to produce an estimate of a desired quantity, which may then be used on different network control time scales.

The goal of this thesis was to focus on the packet-level time scale and investigate how measurement based, adaptive packet scheduling algorithms can be utilized in different networking environments. Adaptive scheduling algorithms were designed and analyzed separately for fixed Internet (IP-based backbone infrastructure) and hybrid 4G networks, consisting of wireless infrastructure-based and infrastructure-less networks. In fixed IP networks it was assumed that DiffServ architecture is used and that the scheduler either dynamically adjusts the class resources periodically or on a packet per packet basis, so that the policy chosen by the operator will be fulfilled regardless of traffic conditions. In infrastructure-based 4G networks, adaptivity was utilized by choosing the scheduled user based on regular CQI feedback, therefore guaranteeing temporal fairness between users. In infrastructure-less WLAN mesh networks, online measurements were used to predict future delays and to select modulation and coding schemes to preserve the delay bounds of traffic classes while minimizing energy consumption.

The first part of the thesis began with a review of relevant networking technologies and the desirable properties of conventional and adaptive scheduling algorithms in fixed and wireless environments. This was followed by a more detailed presentation of the state of the art of fixed IP scheduling algorithms, and a new delay-based scheduling algorithm, the

Delay-Bounded Hybrid Proportional Delay (DBHPD) for adaptive provisioning was introduced. Next, the general theory behind delay estimators was described and a practical Exponential Weighted Moving Average (EWMA) estimator for the DBHPD algorithm's delay estimation problem was presented. Finally, the DBHPD algorithm was thoroughly evaluated by ns2-simulations and measurements in a prototype router network.

The first part of the evaluation process was to use ns2 simulations to compare the performance of static and adaptive provisioning methods, in order to see what kind of performance advantage can be achieved by adaptivity. For the static provisioning case we used the capacity-based Deficit Round Robin (DRR) algorithm. Next, we developed the first working implementation of the DBHPD algorithm in a FreeBSD-based ALTQ prototype router and compared it to the Class Based Queueing (CBQ) hierarchical bandwidth sharing algorithm, with FTP, HTTP, Video Streaming and VoIP traffic in underload, overload and heavy overload conditions. Finally, we performed a comparative study of the simulation and measurement results for both the DBHPD and CBQ algorithms in order to see how well the real implementations correspond to the theoretical models of these algorithms.

According to the ns2-simulations performed in a network setup, the DBHPD algorithm achieved the targeted provisioning goal in a better manner than the static DRR algorithm, regardless of the load level, application mix or queue management method used. The most distinctive difference could be observed in the way the algorithms served HTTP-traffic, as DRR resulted in intolerable packet losses. We examined the adaptability of both algorithms by testing them with different load levels, application mixes and queue management methods. We observed that the delay-bounded DBHPD algorithm was considerably more robust in all of the cases. For both algorithms, we also observed that provisioning was more challenging when the traffic mix was strongly dominated by a single traffic type. This implies that load balancing or intelligent routing methods should be used to avoid large deviations in the loads of different traffic classes.

The implementation results proved that the DBHPD algorithm operates well and in line with its differentiation model. It preserves both the desired delay-bound and the delay ratios between the classes. Comparisons with CBQ showed that DBHPD is able to utilze links at least as well as CBQ and, in addition, results in more controllable packet losses and much better and predictable differentiation in terms of delays. Besides the ability to adapt to the packet delays that it experiences, DBHPD also controls packet losses caused by buffer overflows (e.g. excessive queuing times). CBQ, on the other hand, is only a semi-adaptive algorithm and is not able to provide as good tracking ofto track the offered traffic in contrast to available resources as well as DBHPD does.

We also proposed new delay estimators for the DBHPD algorithm, namely a simple Exponential Weighted Moving Average (EWMA) estimator, an EWMA estimator with restart (EWMA-r) and an EWMA based on the proportional error of the estimate (EWMA-pe). We used ns2 simulations to compare these estimators with the original, simple sum estimator. We used three traffic mixes in the simulations: pure CBR-traffic, pure Pareto-ON-OFF traffic and mixed traffic from several real applications.

According to the evaluation the simple sum and EWMA estimators often lead to false scheduling decisions. Therefore, they are not appropriate solutions to the delay estimation problem. On the other hand, both the EWMA-r, and especially the EWMA-pe estimator proved to be promising alternatives for all traffic mixes. Furthermore, the EWMA-r and EWMA-pe only require small changes to be made to the original EWMA estimator, which is extremely simple to implement in practice. However, in order to judge which one of the EWMA-r and the EWMA-pe estimators is better, network level performance evaluations with real traffic should be conducted with both estimators.

The comparative study of measurements and simulation results revealed that real implementations of the algorithms produce results that are reasonably close to the differentiation models in terms of quality. However, clear differences were observed in the exact shape of the delay distributions. These deviations can be explained partly by the differences in the offered load processes between measurements and simulations. However, most of the differences are likely due to the simplifications used in real implementations, as well as overhead caused by the estimation procedures. With CBQ, for example, demanding borrowing operations can cause extra packet delays in a real router. Initial kernel profiling experiments also confirmed that CBQ is a significantly more complex and resource-intensive algorithm than DBHPD.

Based on the simulation and implementation results it can be stated that DBHPD satisfies the requirements for algorithms aiming at relative differentiation. In addition, it is robust, flexible and simple to implement. Robustness could be increased further with more advanced delay estimation algorithms. Stability has not been analytically proved but according to the simulation and measurement results differentiation was controllable and predictable even in the presence of TCPtraffic.

The second part of the thesis reviewed the state of the art of wireless scheduling algorithms and discussed specific requirements set by hybrid 4G networking scenarios. Methods for joint scheduling and transmit beamforming in 3.9G or 4G networks were described and quantitatively analyzed using statistical methods. Finally, a novel cross-layer energy-adaptive scheduling and queue management framework, EAED (Energy Aware Early Detection), for minimizing energy consumption in WLAN mesh networks, was proposed and evaluated with simulations.

When studying the performance of joint transmit beamforming and channel-aware scheduling, we considered antenna selection as a reference transmit beamforming method for the so-called Mode 1 and Mode 2. A two-antenna version of latter methods is currently used in the UTRA FDD and its HSDPA extension. Out of all the scheduling strategies, the focus was on the one-bit On-off strategy, where all users send acknowledgement (ACK) or negative acknowledgement (NACK) messages after each transport time interval (TTI), based on received relative SNR. If possible, the served user is then selected randomly from the set of users with positive ACKs. The reference scheduling strategies were the

Round Robin, where channel state information is not available in the transmitter and, on the other hand, the Maximum SNR strategy, where the transmitter admits perfect channel information.

Results for the cumulative distribution of received SNR showed that the use of transmit beamforming reduces the difference between upper and lower scheduling performance limits. Nevertheless, the joint usage of transmit beamforming and scheduling leads to a remarkable increase in performance. The variation in system performance due to different transmit beamforming methods was largest with low CDF percentiles, where Mode 1 and Mode 2 clearly outperformed antenna selection. The differences were small with high CDF percentiles. It was shown that the On-off scheduling strategy reaches its optimal performance when the received SNR and decision threshold are equal. Thus, for this certain SNR value, the one-bit scheduling method is able to provide the same gain as the Maximum SNR scheduling strategy that assumes perfect channel knowledge. When focusing on outage rates, it was found that system performance can also be optimized for a certain number of users when the outage probability is fixed. This is interesting from a wireless systems design perspective, since in networks like LTE-A, the performance requirements on the cell edge are given in terms of outage probabilities and rates. However, when the decision threshold is optimized for a certain number of users, additional users do not provide any noticeable increase in gain. Conversely, with a Maximum SNR scheduling strategy, increasing the number of users leads to a logarithmic outage rate increase. Finally, the results also show that even a small feedback error rate may seriously degrade the on-off scheduling performance. The feedback errors decrease the outage rate saturation level as a function of the number of users. This can especially be seen when the outage probability is low.

In the simulation analysis of the EAED algorithm, operation with and without a packet dropper was compared to the so-called waterfilling approach, where the maximum transmit power allowed by current channel conditions is used for transmitting the packets. Simulations revealed that our scheme can save considerable amounts of transmission energy without violating application level QoS requirements with reasonable traffic loads and distances. Even with higher loads and larger distances moderate savings can be obtained. Maximal energy saving was achieved by using the joint adaptation of transmission power, modulation and code rate as well as the early dropping of packets. Since transmission rate is determined through modulation and coding schemes, both of which are adapted frequently, no absolute throughput guarantees can be provided. This makes our scheme suboptimal for very bandwidth intensive applications. However, using 802.11 as a basic radio technology, EAED would be beneficial in any use case where distances are expected to stay within the limits given by our simulation scenarios and connectivity is preferred over high bitrate. Public Safety Communications and basic connectivity in energy scarce locations are the most evident examples of such cases. Delay and disruption tolerant networks (DTNs) would also appear to be a prominent area of application. An example DTN type application is a walkie-talkie like device for providing voice messaging when hiking outdoors or coping with emergencies.

In summary, this thesis first presented a new adaptive scheduling algorithm, Delay-Bounded Hybrid Proportional Delay (DBHPD), with a practical implementation in a real FreeBSD prototype router. The evaluation process that was conducted proved that DBHPD results in differentiation that is considerably more controllable than that provided by basic static bandwidth sharing algorithms. The prototype router measurements showed that the DBHPD algorithm can be easily implemented in practice. DBHPD results in less processing overheads than a well known Class Based Queueing (CBQ) algorithm and provides more predictable quality differentiation, even though CBQ is a pseudo-adaptive algorithm designed for intelligent bandwidth redistribution. The performance of DBHPD could be further improved with adaptive delay estimation algorithms, such as EWMA-r and EWMA-pe. Following on from this, the thesis presented and analyzed joint scheduling and transmit beamforming methods for 3.9G and 4G networks. The analysis revealed that the combined gain of channel aware scheduling and transmit beamforming is substantial, even though the use of transmit beamforming with an increased number of antennas reduces the scheduling gain. It was shown that an On-off strategy can achieve the performance of an ideal Max SNR strategy if the feedback threshold is optimized. However, even a low feedback bit error rate causes the performance to degrade. Finally, this thesis presented a novel cross-layer energy-adaptive scheduling and queue management framework EAED (Energy Aware Early Detection) for preserving delay bounds and minimizing energy consumption in WLAN mesh networks. The simulations showed that our scheme can save considerable amounts of transmission energy without violating application level QoS requirements with reasonable traffic loads and distances. When considering the general requirements set for wireless scheduling algorithms, EAED satisfies a subset of them: It is robust, simple to implement and energy efficient. Other requirements are not relevant in the context that EAED is primarily designed for.

16.1 Further work

Regarding fixed IP scheduling, one major goal for the future could be to conduct both performance measurements and larger scale simulations on the DBHPD algorithm with the new estimators in order to see if the scheduling algorithm can be further improved with more sophisticated estimation procedures. Load profiles produced by the simulations and measurements could also be examined more carefully and profiles could be created to allow a more accurate comparison to be made between simulation and measurement results. Depending on the estimator results, the DBHPD algorithm itself could also be developed, for example by using a bandwidth sharing scheduler as a basis and adapting the class service order in a real congestion situation with the help of measured queuing delays.

In the joint scheduling and transmit beamforming analysis, MS sent separate feedback for transmit beamforming and scheduling. In future work, transmit beamforming and scheduling actions could be determined jointly based on combined feedback and the decision threshold could be optimized by taking the effect of feedback errors into account. It would also be interesting to analyze the effect of heterogeneous users on performance.

Heterogeneity could arise, for example, in terms of varying QoS requirements or due to the primary/secondary use of spectrum resources in cognitive radio.

Regarding the EAED algorithm simulations covered in this thesis, parallel measurements with real wireless devices are under way. The aim is to accurately model the energy consumption of a WLAN enabled mobile phone and to optimize the algorithm according to the model. Another important area of future work is the development and comparison of different types of power saving schemes, including sleeping based mechanisms, for identifying optimal strategies for a variety of applications and scenarios. For TCP-like non-real-time traffic, early dropping could be replaced with Explicit Energy Notification (EEN) where the endpoints and intermediate routers indicate, by marking packets, that the source's congestion window should be decreased due to increasing energy consumption.

In this thesis, the problem of adaptive scheduling was solved separately for different types of networking environments. An ambitious goal for the future could be to evaluate the adaptive scheduling approach in a truly heterogeneous environment, in which both the wireless access networks and the fixed IP core network deploy adaptive scheduling and traffic handling mechanisms, and performance is evaluated end-to-end.

References

[1] 3GPP. URL `http://www.3gpp.org`.

[2] 3GPP. 2005. 3GPP Technical Specification, TS 04.60, Ver. 8.27.0.

[3] 3GPP. 2005. Inter-cell Interference Handling for E-UTRA. 3GPP TSG-RAN WG1 R1-050764.

[4] 3GPP. 2006. Downlink Inter-Cell Interference Coordination/Avoidance-Evaluation of Frequency Reuse. 3GPP TSG-RAN WG1 R1-061374.

[5] 3GPP. 2007. On ICIC Schemes Without/With Traffic Load Indication, 3GPP TSG-RAN WG1 R1-074444.

[6] 3GPP. 2008. Requirements for Further Advancements for E-UTRA (LTE-Advanced), (Release 8). 3GPP TSG-RAN Technical Report, TR 36.913, Ver. 8.0.0.

[7] 3GPP. 2009. 3GPP Technical Report, TR 25.913, Ver. 9.0.0.

[8] 3GPP. 2009. Physical Channels and Mapping of Transport Channels onto Physical Channels (FDD) (Release 7). 3GPP Technical Specification, TS 25.211, Ver. 7.8.0.

[9] 3GPP. 2009. Physical Channels and Modulation (Release 8). 3GPP Technical Specification, TS 36.211, Ver. 8.8.0.

[10] 3GPP. 2009. Physical Layer Procedures (FDD) (Release 7). 3GPP Technical Specification, TS 25.214, Ver. 7.13.0.

[11] 3GPP. 2009. Physical Layer Procedures (Release 8). 3GPP Technical Specification, TS 36.213, Ver. 8.8.0.

[12] S. Aalto and P. Lassila. 2007. Impact of Size-Based Scheduling on Flow-Level Performance in Wireless Downlink Data Channels. In: Proceedings of the 20th International Teletraffic Congress, pages 1096–1107. Ottawa, Canada.

[13] M. Abramowitz and I. Stegun. 1972. Handbook of Mathematical Functions. Washington DC: National Bureau of Standards.

[14] A. Abrardo, A. Alessio, P. Detti, and M. Moretti. 2007. Centralized Radio Resource Allocation for OFDMA Cellular Systems. In: Proceedings of IEEE International Conference on Communications (ICC), pages 269–274. Glasgow, Ireland.

[15] A. Agnetis, G. Brogi, G. Ciaschetti, P. Detti, and G. Giambene. 2003. Optimal Packet Scheduling in UTRA-TDD. IEEE Communications Letters 7, no. 3, pages 112–114.

[16] M. Al-Rawi, R. Jäntti, J. Torsner, and M. Sagfors. 2007. Opportunistic Uplink Scheduling for 3G LTE Systems. In: Proceedings of IEEE International Conference on Innovations in Information Technology, pages 705–709. Dubai, United Arab Emirates.

[17] M.-S. Alouini and A. Goldsmith. 1999. Capacity of Rayleigh Fading Channels under Different Adaptive Transmission and Diversity-Combining Techniques. IEEE Transactions on Vehicular Technology 48, pages 1165–1181.

[18] P. Ameigeiras, J. Wigard, and P. Mogensen. 2004. Performance of Packet Scheduling Methods with Different Degree of Fairness in HSDPA. In: Proceedings of IEEE Vehicular Technology Conference (VTC), volume 2, pages 860–864. Los Angeles, California, USA.

[19] T. Anjali, C. Scoglio, and G. Uhi. 2003. A New Scheme for Traffic Estimation and Resource Allocation for Bandwidth Brokers. Computer Networks 41, no. 6, pages 761–777.

[20] J. Antila and M. Luoma. 2004. Adaptive Scheduling for Improved Quality Differentiation. In: Proceedings of Multimedia Interactive Protocols and Systems (MIPS), volume 3311, pages 143–152. Grenoble, France.

[21] J. Antila and M. Luoma. 2005. Robust Delay Estimation of an Adaptive Scheduling Algorithm. In: Proceedings of Quality of Service in Multiservice IP Networks (QoSIP), volume 3375, pages 626–642. Catania, Italy.

[22] T. Auld and A. W. Moore. 2007. Bayesian Neural Networks for Internet Traffic Classification. IEEE Transactions on Neural Networks 18, no. 1, pages 223–239.

[23] F. Aune. 2004. Cross-Layer Design Tutorial.

[24] K.C. Beh, A. Doufexi, and S. Armour. 2007. Performance Evaluation of Hybrid ARQ Schemes of 3GPP LTE OFDMA System. In: Proceedings of Personal, Indoor and Mobile Radio Communications (PIMRC), pages 1–5. Athens, Greece.

[25] O. Benali, K. El-Khazen, D. Garrec, M. Guiraudou, and G. Martinez. 2004. A Framework for an Evolutionary Path Toward 4G by Means of Cooperation of Networks. IEEE Communications Magazine 42, no. 5, pages 82–89.

[26] J.C.R. Bennett and H. Zhang. 1996. WF^{2}Q: Worst-case Fair Weighted Fair Queueing. In: Proceedings of IEEE Infocom, pages 120–127. San Francisco, California, USA.

[27] Hendrik Berndt. 2003. Advanced Service Provisioning in 4G Mobile Systems. Lecture material. URL http://netlab.hut.fi/opetus/s38001/s03/schedule03.shtml.

[28] V. Bharghavan and T. Nandagopal. 1999. Fair Queueing in Wireless Networks: Issues and Approaches. IEEE Personal Communications Magazine 6, no. 1, pages 44–53.

[29] G. Bianchi and I. Tinnirello. 2003. Kalman Filter Estimation of the Number of Competing Terminals in an IEEE 802.11 network. In: Proceedings of IEEE Infocom, volume 2, pages 844–852. San Francisco, California, USA.

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

[31] D. Borah, A. Daga, G. Lovelace, and P. De Leon. 2005. Performance Evaluation of the IEEE 802.11a and b WLAN Physical Layer on the Martian Surface. In: Proceedings of IEEE AeroConf, pages 1429–1437. Big Sky, Montana, USA.

[32] R. Braden, D. Clark, and S. Shenker. 1994. Integrated Services in the Internet Architecture: An overview. RFC 1633.

[33] M. Buettner, G. V. Yee, E. Anderson, and R. Han. 2006. X-MAC: a Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks. In: Proceedings of 4th international Conference on Embedded Networked Sensor Systems, Sensys'06, pages 307–320. Boulder, Colorado, USA.

[34] F. D. Calabrese, P. H. Michaelsen, C. Rosa, M. Anas, C. U. Castellanos, D. L. Villa, K. I. Pedersen, and P. E. Mogensen. 2008. Search-Tree Based Uplink Channel Aware Packet Scheduling for UTRAN LTE. In: Proceedings of IEEE Vehicular Technology Conference (VTC), pages 1949–1953. Singapore, Malaysia.

[35] F. D. Calabrese, C. Rosa, M. Anas, P. H. Michaelsen, K. I. Pedersen, and P. E. Mogensen. 2008. Adaptive Transmission Bandwidth Based Packet Scheduling for LTE Uplink. In: Proceedings of IEEE Vehicular Technology Conference (VTC), pages 1–5. Calgary, Alberta, Canada.

[36] K. Cho. 1999. Managing Traffic with ALTQ. In: Proceedings of USENIX 1999 Annual Technical Conference, pages 121–128. Monterey, California, USA.

[37] K. Cho. 2004. Fitting Theory into Reality in the ALTQ Case. In: Proceedings of ASIA BSD conference. Taipei, Taiwan.

[38] J. Choi. 2002. Performance Analysis for Transmit Antenna Diversity With/Without Channel Information. IEEE Transactions on Vehicular Technology 51, no. 1, pages 101–113.

[39] N. Christin, J. Liebeherr, and T. Abdelzaher. 2002. A Quantitative Assured Forwarding Service. In: Proceedings of IEEE Infocom, volume 2, pages 864–873. New York, New York, USA.

[40] K. Claffy, G. Polyzos, and H. Braun. 1993. Application of Sampling Methodologies to Network Traffic Characterization. In: Proceedings of USENIX SIGCOMM, pages 194–203. San Francisco, California, USA.

[41] M. Conti and E. Gregori. 2004. Traffic and Interference Adaptive Scheduling for Internet Traffic in UMTS. Mobile Networks and Applications 9, pages 265–277.

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

[43] J. Dingde and H. Guangmin. 2008. Large-Scale IP Traffic Matrix Estimation Based on the Recurrent Multilayer Perceptron Network. In: Proceedings of IEEE International Conference on Communications (ICC), pages 366–370. Beijing, China.

[44] S. Doshi, S. Bhandare, and T. X. Brown. 2002. An On-Demand Minimum Energy Routing Protocol for a Wireless Ad Hoc Network. Mobile Computing and Communications Review 6, no. 2, pages 50–66.

[45] A. Doulamis, N. Doulamis, and S. D. Kollias. 2003. An Adaptable Neural-Network Model for Recursive Nonlinear Traffic Prediction and Modeling of MPEG Video Sources. IEEE Transactions on Neural Networks 14, no. 1, pages 150–166.

[46] C. Dovrolis, D. Stiliadis, and P. Ramanathan. 1999. Proportional Differentiated Services: Delay Differentiation and Packet Scheduling. In: Proceedings of ACM SIGCOMM, pages 109–119. Cambridge, Massachusetts, USA.

[47] C. Dovrolis, D. Stiliadis, and P. Ramanathan. 2002. Proportional Differentiated Services: Delay Differentiation and Packet Scheduling. IEEE/ACM Transactions on Networking 10, no. 2, pages 12–26.

[48] A. A. Dowhuszko, G. Corral-Briones, J. Hämäläinen, and R. Wichman. 2009. On Throughput-Fairness Trade-Off in Virtual MIMO Systems with Limited Feedback. to appear in EURASIP Journal on Wireless Communications and Networking, Special Issue "Fairness in Radio Resource Management for Wireless Networks" .

[49] J-P. Ebert, S. Aier, G. Kofahl, A. Becker, B. Burns, and A. Wolisz. 2002. Measurement and Simulation of the Energy Consumption of an WLAN Interface. Technical Report, University of Berlin.

[50] P. Edmond and B. Li. 2003. SmartNode: Achieving 802.11 MAC Interoperability in Power-Efficient Ad hoc Networks with Dynamic Range Adjustments. In: Proceedings of 23rd International Conference on Distributed Computing Systems, pages 650–657. Providence, Rhode Island, USA.

[51] A. Fallahi, E. Hossain, and A. S. Alfa. 2006. QoS and Energy Tradeoff in Distributed Energy-Limited Mesh/Relay/Networks: A Queuing analysis. IEEE Transactions on Parallel and Distributed Systems 17, no. 6, pages 576–592.

[52] X. Fan, S. Chen, and X. Zhang. 2007. An Inter-Cell Interference Coordination Technique Based on Users Ratio and Multi-Level Frequency Allocations. In: Proceedings of IEEE Wireless Communications, Networking and Mobile Computing (WiCom), pages 799–802. Shanghai, China.

[53] H. Fattah and C. Leung. 2002. An Overview of Scheduling Algorithms in Wireless Multimedia Networks. IEEE Wireless Communications 9, no. 5, pages 76–83.

[54] P. Floreen, P. Kaski, J. Kohonen, and P. Orponen. 2005. Lifetime Maximization for Multicasting in Energy-Constrained Wireless Networks. IEEE Journal on Selected Areas in Communications 23, no. 1, pages 117–126.

[55] F. Florén, O. Edfors, and B.-A. Molin. 2003. The Effect of Feedback Quantization on the Throughput of a Multiuser Diversity Scheme. In: Proceedings of IEEE

Global Telecommunications Conference (Globecom), volume 1, pages 497–501. San Francisco, California, USA.

[56] S. Floyd and V. Jacobson. 1993. Random Early Detection Gateways for Congestion Avoidance. IEEE/ACM Transactions on Networking 1, no. 4, pages 397–413.

[57] G. J. Foschini and M. J. Gans. 1998. On Limits of Wireless Communications in a Fading Environment when Using Multiple Antennas. Wireless Personal Communications 6, pages 311–335.

[58] M. Frodigh, S. Parkvall, C. Roobol, P. Joihansson, and P. Larsson. 2001. Future Generation Wireless Networks. IEEE Personal Communications Magazine 8, no. 5, pages 10–17.

[59] A. E. Gamal, C. Nair, B. Prabhakar, E. Biyikoglu, and S. Zahedi. 2002. Energy-Efficient Scheduling of Packet Transmissions over Wireless Networks. In: Proceedings of IEEE Infocom, volume 3, pages 1773–1782. New York, New York, USA.

[60] D. Gesbert and S. Alouini. 2004. How much Feedback is Multi-User Diversity Really Worth? In: Proceedings of IEEE International Conference on Communications (ICC), pages 234–238. Paris, France.

[61] M. Gidlund and J-C. Laneri. 2008. Scheduling Algorithms for 3GPP Long-Term Evolution Systems: From a Quality of Service Perspective. In: Proceedings of the 10th IEEE International Symposium on Spread Spectrum Techniques and Applications (ISSSTA), pages 114–117. Bologna, Italy.

[62] S.J. Golestani. 1994. A Self-Clocked Fair Queueing Scheme for High Speed Applications. In: Proceedings of IEEE Infocom, volume 2, pages 636–646. Toronto, Ontario, Canada.

[63] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. 2010. 802.11 with Multiple Antennas for Dummies. Editorial note. Computer Communications Review (CCR) 40, no. 1, pages 19–25.

[64] J. Hämäläinen and R. Wichman. 2000. Closed-Loop Transmit Diversity for FDD WCDMA Systems. In: Proceedings of Asilomar Conference on Signals, Systems and Computers, volume 1, pages 111–115. Pacific Grove, California, USA.

[65] J. Hämäläinen and R. Wichman. 2002. Performance Analysis of Closed-Loop Transmit Diversity in the Presence of Feedback Errors. In: Proceedings of IEEE PIMRC, volume 5, pages 2297–2301. Lisbon, Portugal.

[66] J. Hämäläinen and R. Wichman. 2003. Bit Error Probabilities in a Two-Rate Communication System. In: Proceedings of IEEE International Conference on Communications, volume 4, pages 2693–2697. Anchorage, Alaska, USA.

[67] J. Hämäläinen and R. Wichman. 2003. Performance of Transmit Time Selection Diversity in Multipath Fading Channels. In: Proceedings of International Symposium on Wireless Personal Multimedia Communications. Yokosuka, Japan.

[68] J. Hämäläinen and R. Wichman. 2004. Quantization of Channel Quality Indicator in Maximum SNR Scheduling. In: Proceedings of Nordic Radio Symposium. Oulu, Finland.

[69] J. Hämäläinen and R. Wichman. 2006. Capacities of Physical Layer Scheduling Strategies on a Shared Link. Wireless Personal Communications 39, no. 1, pages 115–134.

[70] J. Hämäläinen, R. Wichman, A. A. Dowhuszko, and G. Corral-Briones. 2010. Capacity of Generalized UTRA FDD Closed-Loop Transmit Diversity Modes. Wireless Personal Communications 54, no. 3, pages 467–484.

[71] V. Hassel, D. Gesbert, M.-S. Alouini, and G.E. Oien. 2007. A Threshold-Based Channel State Feedback Algorithm for Modern Cellular Systems. IEEE Transactions on Wireless Communications 6, no. 7, pages 2422–2426.

[72] D. He and C. Shen. 2003. Simulation Study of IEEE 802.11e EDCF. In: Proceedings of IEEE Vehicular Technology Conference (VTC), volume 1, pages 685–689. Jeju, Korea.

[73] J.R.W. Heath, M. Airy, and A. Paulraj. 2001. Multiuser Diversity for MIMO Wireless Systems with Linear Receivers. In: Proceedings of Asilomar Conference on Signals, Systems and Computers, volume 2, pages 1194–1199. Pacific Grove, California, USA.

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

[75] H. Holma and A. Toskala. 2007. WCDMA for UMTS: HSPA Evolution and LTE. John Wiley, Fourth Edition.

[76] J. Holtzman. 2001. Asymptotic Analysis of Proportional Fair Algorithm. In: Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), volume 2, pages 33–37. San Diego, California, USA.

[77] A. Hottinen, O. Tirkkonen, and R. Wichman. 2003. Multi-antenna Transceiver Techniques for 3G and Beyond. John Wiley & sons .

[78] H.Paloheimo, J.Manner, J.Nieminen, and A.Ylä-Jääski. 2006. Challenges in Packet Scheduling in 4G Wireless Networks. In: Proceedings of Annual Symposium of Personal, Indoor and Mobile Radio Communications (PIMRC), pages 1–6. Helsinki, Finland.

[79] V. Huang and W. Zhuang. 2002. QoS-Oriented Access Control for 4G Mobile Multimedia CDMA Communications. IEEE Communications Magazine 40, no. 3, pages 118–125.

[80] M. Husso, J. Hämäläinen, R. Jäntti, J. Nieminen, T. Riihonen, and R. Wichman. 2011. Performance of On-Off Scheduling Strategy in Presence of Transmit Beamforming. Elsevier Journal of Physical Communications 4, no. 1, pages 3–12.

[81] IEEE. 2007. LAN/MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications. In: Amendment: ESS Mesh Networking, IEEE Unapproved draft P802.11s/D1.02.

[82] IEEE802. URL http://www.ieee802.org.

[83] R. Jantti and S. Kim. 2006. Joint Data Rate and Power Allocation for Lifetime Maximization in Interference Limited Ad Hoc Networks. IEEE Transactions on Wireless Communications 5, no. 5, pages 1086–1094.

[84] N. Jindal. 2008. Antenna Combining for the MIMO Downlink Channel. IEEE Transactions on Wireless Communications 7, no. 10, pages 3834–3844.

[85] M. Johansson. 2003. Benefits of Multiuser Diversity with Limited Feedback. In: Proceedings of IEEE International Workshop on Signal Processing Advances in Wireless Communications, pages 155–159. Rome, Italy.

[86] E.-S. Jung and N. H. Vaidya. 2002. An Energy Efficient MAC Protocol for Wireless LANs. In: Proceedings of IEEE INFOCOM, volume 3, pages 1756–1764. New York, New York, USA.

[87] V. Kawadia and P. Kumar. 2005. A Cautionary Perspective on Cross-Layer Design. IEEE Wireless Communications 12, no. 1, pages 3–11.

[88] V. Kawadia, S. Narayanaswamy, R. S. Sreenivas, R. Rozovsky, and P. R. Kumar. 2001. Protocols for Media Access Control and Power Control in wireless Networks. In: Proceedings of the 40th IEEE Conference on Decision and Control, pages 1935–1940. Orlando, Florida, USA.

[89] P. Kela, J. Puttonen, N. Kolehmainen, T. Ristaniemi, T. Henttonen, and M. Moisio. 2008. Dynamic Packet Scheduling Performance in UTRA Long Term Evolution Downlink. In: Proceedings of IEEE International Symposium on Wireless Pervasive Computing, pages 308–313. Santorini, Greece.

[90] F. Khan. 2009. LTE for 4G Mobile Broadband: Air Interface Technologies and Performance. Cambridge University Press.

[91] M. A. Khojastepour and A. Sabharwal. 2004. Delay-Constrained Scheduling: Power Efficiency, Filter Design, and Bounds. In: Proceedings of IEEE Infocom, volume 3, pages 1938–1949. Hong Kong, China.

[92] E. Knightly. 1996. Traffic Models and Admission Control for Integrated Services Networks, Ph.D Thesis. Department of EECS, University of California, Berkeley, California .

[93] R. Knopp. 1997. Coding and Multiple-Access over Fading Channels. Ph.D. thesis, Ecole Polytehnique Fédérale de Lausanne, Switzerland.

[94] I. Koutsopoulos and L. Tassiulas. 2006. Cross-Layer Adaptive Techniques for Throughput Enhancement in Wireless OFDM-Based Networks. IEEE Transactions on Networking 14, no. 5, pages 1056–1066.

[95] B. Krasniqi, M. Wrulich, and C. F. Mecklenbrauker. 2009. Network-Load Dependent Partial Frequency Reuse for LTE. In: Proceedings of 9th International Symposium on Communication and Information Technology (ISCIT 2009), pages 672–676. Incheon, Korea.

[96] R. Kwan and C. Leung. 2010. A Survey of Scheduling and Interference Mitigation in LTE. Hindawi Journal of Electrical and Computer Engineering. Article ID 273486 URL `http://www.hindawi.com/journals/jece/2010/273486/`.

[97] N. Laneman, D. Tse, and G. Wornell. 2004. Cooperative Diversity in Wireless Networks: Efficient Protocols and Outage Behavior. IEEE Transactions on Information Theory 50, no. 12, pages 3062–3080.

[98] A. Larmo, M. Lindström, M. Meyer, G. Pelletier, J. Torsner, and H. Wiemann. 2009. The LTE Link-Layer Design. IEEE Communications Magazine pages 52–59.

[99] R. Laroia, S. Uppala, and J. Li. 2004. Designing a Mobile Broadband Wireless Access Networks. IEEE Signal Processing Magazine pages 20–28.

[100] P. Lassila and S. Aalto. 2008. Combining Opportunistic and Size-Based Scheduling in Wireless Systems. In: Proceedings of the 11th ACM International Con-

ference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), pages 323–332. Vancouver, British Columbia, Canada.

[101] S. Lee, S. Choudhury, A. Khoshnevis, S. Xu, and S. Lu. 2009. Downlink MIMO with Frequency-Domain Packet Scheduling for 3GPP LTE. In: Proceedings of IEEE Infocom, pages 1269–1277. Rio de Janeiro, Brazil.

[102] S. Lee, I. Pefkianakis, A. Meyerson, S. Xu, and S. Lu. 2009. Proportional Fair Frequency-Domain Packet Scheduling for 3GPP LTE Uplink. In: Proceedings of IEEE Infocom, pages 2611–2615. Rio de Janeiro, Brazil.

[103] J. Leinonen, J. Hämäläinen, and M. Juntti. 2009. Performance Analysis of Downlink OFDMA Resource Allocation with Limited Feedback. IEEE Transactions on Wireless Communications 8, no. 6.

[104] M. Leung, J. Lui, and D. Yau. 2001. Adaptive Proportional Delay Differentiated Services: Characterization and Performance Evaluation. IEEE/ACM Transactions on Networking 9, no. 6, pages 801–817.

[105] G. Li and H. Liu. 2006. Downlink Radio Resource Allocation for Multi-cell OFDMA System. IEEE Transactions on Wireless Communications 5, no. 12, pages 3451–3459.

[106] R. Liao and A. Campbell. 2001. Dynamic Core Provisioning for Quantitative Differentiated Service. In: Proceedings of IWQoS, pages 9–26. Karlsruhe, Germany.

[107] Y. Lin and G. Yue. 2008. Channel-Adapted and Buffer-Aware Packet Scheduling in LTE Wireless Communication System. In: Proceedings of IEEE International Conference on Wireless Communications, Networking and Mobile Computing, pages 1–4. Dalian, China.

[108] C. Liu and J. Layland. 1973. Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment. Journal of ACM 20, no. 1, pages 46–61.

[109] D.J. Love, R.W. Heath Jr, V.K.N. Lau, D. Gesbert, B.D. Rao, and M. Andrews. 2008. An Overview of Limited Feedback in Wireless Communication Systems. IEEE Journal on Selected Areas in Communications 26, no. 8, pages 1341–1365.

[110] S. Lu, T. Nandagopal, and V. Bharghavan. 1998. A Wireless Fair Service Algorithm for Packet Cellular Networks. In: Proceedings of International Conference on Mobile Computing and Networking (Mobicom), pages 10–20. Dallas, Texas.

[111] S. Malik and D. Zeghlache. 2002. Improving Throughput and Fairness on the Downlink Shared Channel in UMTS WCDMA Networks. In: Proceedings of European Wireless Conference. Florence, Italy.

[112] M. Malkowski, A. Kemper, and W. Xiaohua. 2007. Performance of Scheduling Algorithms for HSDPA. In: Proceedings of CHINACOM'07, pages 1052–1056. Shanghai, China.

[113] P. Mogensen. 2001. High Speed Downlink Packet Access (HSDPA) -the Path Towards 3.5G. In: Proceedings of IEEE Workshop on Signal Processing Systems (SiPS), page 344. Antwerp , Belgium.

[114] P. Mogensen, W. Na, I. Z. Kovacs, F. Frederiksen, A. Pokhariyal, K. I. Pedersen, T. Kolding, K. Hugl, and M. Kuusela. 2007. LTE Capacity Compared to the Shannon Bound. In: Proceedings of IEEE Vehicular Technology Conference (VTC), pages 1234–1238. Dublin, Ireland.

[115] A. Moore. 2002. Measurement-based Management of Network Resources, Ph.D Thesis, University of Cambridge, Computer Laboratory.

[116] L. Munoz, R. Agero, J. Choque, J.A. Lrastorza, L. Sanchez, M. Petrova, and P. Mahonen. 2004. Empowering Next-Generation Wireless Personal Communication Networks. IEEE Communications Magazine 42, no. 5, pages 64–70.

[117] A. Naguib, N. Seshadri, , and A. R. Calderbank. 2000. Increasing Data Rate over Wireless Channels. IEEE Signal Processing Magazine 17, no. 3, pages 77–92.

[118] Q. Ni. 2005. Performance Analysis and Enhancements for IEEE 802.11e Wireless Networks. IEEE Network 19, no. 4, pages 21–27.

[119] J. Nieminen, M. Luoma, O-P.Lamminen, and A. Paju. 2007. Implementation and Simulation of DBHPD and CBQ Scheduling - A Comparative Study. In: Proceedings of IEEE International Conference on Communications (ICC), pages 524–529. Glasgow, Ireland.

[120] J. Nieminen, M. Luoma, and A. Paju. 2005. Implementation and Performance Measurements of a delay-bounded HPD Algorithm in an ALTQ-based Router. In: Proceedings of Conference on Future Networking Technologies (CoNEXT), pages 61–70. Toulouse, France.

[121] J. Nieminen, H. Paloheimo, and R. Jäntti. 2010. Energy-Adaptive Scheduling and Queue Management in Wireless LAN Mesh Networks. In: Proceedings of Wireless Internet Conference (WICON), pages 1–9. Singapore, Malaysia.

[122] P. Nuggehalli, V. Srinivasan, and R. Rao. 2006. Energy Efficient Transmission Scheduling for Delay Constrained Wireless Networks. IEEE Transactions on Wireless Communications 5, no. 3, pages 531–539.

[123] E.N. Onggosanusi, A. Gatherer, A.G. Dabak, and S. Hosur. 2001. Performance Analysis of Closed-Loop Transmit Diversity in the Presence of Feedback Delay. IEEE Transactions on Communications 49, no. 9, pages 1618–1630.

[124] H. Paloheimo and A. Ylä-Jääski. 2005. Interoperability of MANET and 4G RAN Routing in Terms of Energy Conservation. In: Proceedings of IEEE Advanced Information Networking and Applications (AINA) 2005, volume 1, pages 447–452. Singapore, Malaysia.

[125] A.K. Parekh and R.G. Gallager. 1993. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case. IEEE/ACM Transactions on Networking 1, no. 3, pages 344–357.

[126] S-J. Park and R. Sivakumar. 2002. Load Sensitive Transmission Power Control in Wireless Ad-hoc Networks. In: Proceedings of IEEE Global Telecommunications Conference (Globecom), volume 1, pages 42–46. Taipei, Taiwan.

[127] W. Pattara-atikom, P. Krishnamurthy, and S. Banerjee. 2003. Distributed Mechanisms for Quality of Service in Wireless LANs. IEEE Wireless Communications Magazine 10, no. 3, pages 26–34.

[128] A. Pokhariyal, T. E. Kolding, and P. E. Mogensen. 2006. Performance of Downlink Frequency Domain Packet Scheduling For the UTRAN Long Term Evolution. In: Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), pages 1–5. Helsinki, Finland.

[129] A. Pokhariyal, K.I. Pederson, G. Monghal, I.Z. Kovavs, C. Rosa, T.E. Kolding, and P.E. Mogensen. 2007. HARQ Aware Frequency Domain Packet Scheduler with Different Degrees of Fairness for the UTRAN Long Term Evolution. In: Proceedings of IEEE Vehicular Technology Conference (VTC), pages 2761–2765. Dublin, Ireland.

[130] B. Prabhakar, E. Biyikoglu, and A. El. Gamal. 2001. Energy-Efficient Transmission over a Wireless Link via Lazy Packet Scheduling. In: Proceedings of IEEE Infocom, volume 1, pages 386–394. Anchorage, Alaska, USA.

[131] D. Qiao, S. Choi, A. Jain, and K. Shin. 2003. MiSer: an Optimal Low-Energy Transmission Strategy for IEEE 802.11a/h. In: Proceeding of International Conference on Mobile Computing and Networking (Mobicom), pages 161–175. San Diego, California, USA.

[132] V. Raghunathan, S. Ganeriwal, M. Srivastava, and C. Schurgers. 2004. Energy Efficient Wireless Packet Scheduling and Fair Queuing. ACM Transactions on Embedded Computing Systems 3, no. 1, pages 3–23.

[133] V. Raisinghani and S. Iyer. 2004. Cross-Layer Design Optimizations in Wireless Protocol Stacks. Computer Communications 27, no. 8, pages 720–724.

[134] M. Rinne, P. Pasanen, P. Seppinen, and K. Leppänen. 2004. Dual Bandwidth Approach to New Air Interface. WWRF.

[135] B. Sadiq, R. Madan, and A. Sampath. 2009. Downlink Scheduling for Multiclass Traffic in LTE. EURASIP Journal on Wireless Communications and Networking Article ID 510617.

[136] M. Sagfors, R. Ludwig, M. Meyer, and J. Peisa. 2003. Queue Management for TCP Traffic over 3G Links. In: Proceedings of IEEE Wireless Communications and Networking (WCNC), volume 3, pages 1663–1668. New Orleans, Louisiana, USA.

[137] M. Schwartz. 2005. Mobile Wireless Communications. Cambridge University Press.

[138] S. Shakkottai and A. Stolyar. 2001. Scheduling Algorithms for a Mixture of Real-Time and Non-Real-Time Data in HDR. In: Proceedings of the 17th International Teletraffic Congress (ITC-17). Salvador da Bahia, Brazil.

[139] H-P. Shiang and M. van der Schaar. 2007. Informationally Decentralized Video Streaming Over Multihop Wireless Networks. Multimedia, IEEE Transactions 9, no. 6, pages 1299–1313.

[140] O.-S. Shin and K. Bok. 2003. Antenna-Assisted Round Robin Scheduling for MIMO Cellular Systems. IEEE Communication Letters 7, no. 3, pages 109–111.

[141] M. Shreedhar and G. Varghese. 1995. Efficient Fair Queueing using Deficit Round Robin. In: Proceedings of ACM SIGCOMM, volume 4, pages 375–385. Cambridge, Massachusetts, USA.

[142] T. Simunic, L. Benini, P. Glynn, and G. D. Micheli. 2000. Dynamic Power Management for Portable Systems. In: Proceedings of International Conference on Mobile Computing and Networking (MobiCom), pages 11–19. Boston, Massachusetts, USA.

[143] V. Sivaraman. 2000. End-to-End Delay Service in High Speed Packet Networks using Earliest Deadline First Scheduling, Ph.D Thesis, University of California, Los Angeles, USA.

[144] D. Skoutas and A. Rouskas. 2004. A Dynamic Traffic Scheduling Algorithm for the Downlink Shared Channel in 3G WCDMA. In: Proceedings of IEEE International Conference on Communications (ICC), volume 5, pages 2975–2979. Paris, France.

[145] A. Soule, A. Lakhina, N. Taft, K. Papagiannaki, K. Salamatian, A. Nucci, M. Crovella, and C. Diot. 2005. Traffic Matrices: Balancing Measurements, Inference and Modeling. In: Proceedings of SIGMETRICS, pages 362–373. Banff, Alberta, Canada.

[146] M. Stemm and R. H. Katz. 1997. Measuring and Reducing Energy Consumption of Network Interfaces in Hand-Held devices. IEICE Transactions on Communications E80-, no. B(8), pages 1125–1131.

[147] I. Stoica, H. Zhang, and T. Ng. 2000. A Hierarchical Fair Service Curve Algorithm for Link-Sharing, Real-Time and Priority Service. IEEE/ACM Transactions on Networking 8, no. 2, pages 185–199.

[148] A. Tarello, J. Sun, and M. Zafer. 2008. Minimum Energy Transmission Scheduling Subject to Deadline Constraints. Wireless Networks 14, no. 5, pages 633–645.

[149] P. Viswanath, D. Tse, and R. Laroia. 2002. Opportunistic Beam-Forming Using Dumb Antennas. IEEE Transactions of Information Theory 48, no. 6, pages 1277–1294.

[150] Jörn von Häfen et al. 2004. WINNER D7.1 System requirements.

[151] L. C. Wang and W. J. Lin. 2004. Throughput and Fairness Enhancement for OFDMA Broadband Wireless Access Systems Using the Maximum C/I Scheduling. In: Proceedings of IEEE Vehicular Technology Conference (VTC), volume 7, pages 4696–4700.

[152] Z. Wang. 2001. Internet QoS: Architecture and Mechanisms for Quality of Service. Morgan Kaufmann Publishers.

[153] G. Welch and G. Bishop. 2004. An Introduction to the Kalman Filter.

[154] WWI. 2004. Wireless World Initiative. Press release, N:o.1. URL http://www.wireless-world-initiative.org/press/WWI_press1.pdf.

[155] T. Yoo, N. Jindal, and A. Goldsmith. 2007. Multi-Antenna Downlink Channels with Limited Feedback and User Selection. IEEE Journal on Selected Areas in Communications 25, no. 7, pages 1478–1491.

[156] P. Young. 1984. Recursive Estimation and Time-Series Analysis. Springer-Verlag.

[157] H. Zhang. 1995. Service Disciplines for Guaranteed Performance Service in Packet-Switching Networks. Proceedings of the IEEE 83, no. 10, pages 1374–1396.

[158] X. Zhong and C-Z. Xu. 2005. Delay-Constrained Energy-Efficient Wireless Packet Scheduling with QoS Guarantees. In: Proceedings of IEEE Global Telecommunications Conference (Globecom), pages 3335–3340. St. Louis, Missouri, USA.

Networking scenarios in the future will be complex and will include fixed networks and hybrid Fourth Generation (4G) networks, consisting of both infrastructure-based and infrastructure-less, wireless parts. In such scenarios, adaptive provisioning and management of network resources becomes of critical importance. Adaptive mechanisms are desirable since they enable a self-configurable network that is able to adjust itself to varying traffic and channel conditions. The operation of adaptive mechanisms is heavily based on measurements. The aim of this thesis is to investigate how measurement based, adaptive packet scheduling algorithms can be utilized in different networking environments.

Aalto-DD 39/2011

9 789526 041148

ISBN: 978-952-60-4115-5 (pdf)
ISBN: 978-952-60-4114-8
ISSN-L: 1799-4934
ISSN: 1799-4942 (pdf)
ISSN: 1799-4934

Aalto University
School of Electrical Engineering
Department of Communications and Networking
www.aalto.fi