

**Universität Stuttgart**

Institut für Kommunikationsnetze und Rechnersysteme

Prof. Dr.-Ing. habil. Dr. h. c. mult. P. J. Kühn

**88. Bericht über verkehrstheoretische Arbeiten**

**Verfahren zur relativen  
Dienstgütedifferenzierung in IP-Netzknöten**

von

**Stefan Bodamer**

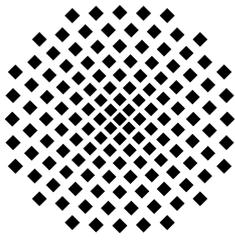
2004

D 93

© 2004 Institut für Kommunikationsnetze und Rechnersysteme, Universität Stuttgart

Druck: E. Kurz & Co., Druck und Medientechnik GmbH, Stuttgart

ISBS 3-922403-98-0



**University of Stuttgart**

Institute of Communication Networks and Computer Engineering

Prof. Dr.-Ing. habil. Dr. h. c. mult. P. J. Kühn

**88th Report on Studies in Congestion Theory**

**Mechanisms for Relative Quality of Service  
Differentiation in IP Network Nodes**

by

**Stefan Bodamer**

2004



# **Verfahren zur relativen Dienstgütedifferenzierung in IP-Netzknoten**

Von der Fakultät Informatik, Elektrotechnik und Informationstechnik  
der Universität Stuttgart zur Erlangung der Würde  
eines Doktor-Ingenieurs (Dr.-Ing.) genehmigte Abhandlung

vorgelegt von

**Stefan Bodamer**

geb. in Pforzheim

Hauptberichter:	Prof. Dr.-Ing. Dr. h. c. mult. Paul J. Kühn
Mitberichter:	Prof. Dr.-Ing. Ralf Lehnert (TU Dresden)
Tag der Einreichung:	15. Juli 2002
Tag der mündlichen Prüfung:	19. Juli 2004

Institut für Kommunikationsnetze und Rechnersysteme  
der Universität Stuttgart  
Juli 2004



# Verfahren zur relativen Dienstgütedifferenzierung in IP-Netzknotten

## Kurzfassung

Die herausragende Bedeutung des Internet und seiner Dienste und Anwendungen für die Entwicklung der Kommunikationstechnik in den letzten Jahren ist unbestritten. Eine zentrale Rolle kommt dabei dem *Internet Protocol* (IP) zu, das sich immer mehr als gemeinsamer Nenner für die Konvergenz von Netzen darstellt. Ein Problem in IP-basierten Netzen besteht allerdings darin, dass dort zunächst nur der so genannte *Best Effort*-Dienst bekannt ist, bei dem alle Verkehrsströme gleich behandelt werden. Falls nicht eine meist teure Überdimensionierung von Netzressourcen vorgenommen wird, können daher die unterschiedlichen Anforderungen der verschiedenen Anwendungen oft nicht gleichzeitig erfüllt werden.

Eine Lösung dieses Problems bringt die Einführung eines Verkehrsmanagements, das eine differenzierte Behandlung von Verkehrsströmen vornimmt. Im Rahmen dieser Arbeit wird dazu nach einer Aufarbeitung der Grundlagen von IP-Netzen sowie einer Klassifikation der verschiedenen Verkehrsmanagement-Funktionen ein Verfahren vorgeschlagen, das Pakete innerhalb eines IP-Netzknottes entsprechend ihrer Zugehörigkeit zu einer Dienstgütekategorie differenziert behandelt. Das Verfahren, das als *Weighted Earliest Due Date* (WEDD) bezeichnet wird, basiert auf einer Abarbeitung nach kürzester Fristen. Diese ergeben sich aus der so genannten Maximalverzögerung, für die in den einzelnen Klassen unterschiedliche Werte festgelegt werden können. Gleichzeitig arbeitet WEDD nach dem Grundsatz der relativen Differenzierung, indem ein weiterer klassenspezifischer Differenzierungsparameter eingeführt wird, der eine Gewichtung der Überschreitungswahrscheinlichkeiten der jeweiligen Maximalverzögerung in den unterschiedlichen Klassen ermöglicht. Durch die darüber hinaus mögliche Verwerfung von Paketen, die ihre Maximalverzögerung überschritten haben, kann somit eine gleichzeitige Verzögerungs- und Verlustdifferenzierung vorgenommen werden.

Ein Schwerpunkt der Arbeit stellt die Leistungsuntersuchung des entwickelten Verfahrens dar. Dabei liegt die Schwierigkeit unter anderem in der Modellierung von IP-Verkehr, der durch ein burstartiges Verhalten und Effekte wie Langzeitabhängigkeit und Selbstähnlichkeit charakterisiert ist. Es werden verschiedene Modelle für echtzeitkritischen sowie für elastischen IP-Verkehr verwendet sowie entsprechende aussagekräftige Leistungsmetriken definiert. Auf dieser Grundlage werden in umfangreichen Simulationsstudien – teilweise unter Verwendung analytisch abgeleiteter Bezugsgrößen – der Nachweis für die Funktionsfähigkeit des entwickelten Verfahrens und insbesondere seine Robustheit in Bezug auf den ankommenden Verkehr erbracht, die Auswirkungen verschiedener Systemparameter gezeigt sowie Vergleiche mit anderen Verfahren vorgenommen. Darüber hinaus werden systematisch einige grundlegende Effekte herausgearbeitet, die – über das entwickelte Verfahren hinaus – einen tieferen Einblick in das Verkehrsverhalten in IP-Netzen geben.

# Mechanisms for Relative Quality of Service Differentiation in IP Network Nodes

## Summary

Recent development of communication networks has been heavily influenced by the Internet and its services and applications. The central building block in this environment is the Internet Protocol (IP) which more and more serves as a common denominator for network integration. One of the problems in IP-based networks, however, is caused by IP's best effort service providing equal treatment of all traffic streams. This often results in poor quality of service for many applications unless an expensive over-dimensioning of network resources is performed.

The problem can be solved by the introduction of traffic management mechanisms enabling differentiation of traffic streams in the network. After an overview on IP network basics and a classification of traffic management functions a new mechanisms for differentiated packet handling within an IP network node is proposed in this thesis. The mechanism called weighted earliest due date (WEDD) basically serves packets according to an earliest deadlines. A deadline is derived from the delay bound parameter for which different values can be defined in each class. Moreover, WEDD operation is based on the concept of relative differentiation. Therefore, an additional differentiation parameter is introduced which enables weighting of delay bound violation probabilities. If in addition packets exceeding their deadline are discarded, the mechanism can provide delay and loss differentiation in parallel.

A major part of this thesis is on performance evaluation of the proposed mechanism. One of the issues that has to be solved in this context is traffic modelling especially since IP traffic is characterised by bursty behaviour and effects like long-range dependence and self-similarity. Various models for real-time and elastic IP traffic are used and relevant performance metrics are defined. Based on that, comprehensive simulation studies – partially using analytically derived relation quantities – are carried out in order to prove the operation capability of the developed mechanism including its robustness with respect to traffic variability, to show the impact of various system parameters, and to compare it to other mechanisms. Furthermore, some basic effects are highlighted beyond the scope of the proposed mechanism providing deeper insight into traffic behaviour in IP networks.

# Inhaltsverzeichnis

<b>Kurzfassung</b> .....	<b>i</b>
<b>Summary</b> .....	<b>ii</b>
<b>Inhaltsverzeichnis</b> .....	<b>iii</b>
<b>Abkürzungen</b> .....	<b>viii</b>
<b>Formelzeichen</b> .....	<b>xiii</b>
<b>1 Einleitung</b> .....	<b>1</b>
<b>2 Grundlagen von IP-Netzen</b> .....	<b>5</b>
2.1 Geschichte des Internets .....	5
2.2 Netzarchitektur .....	6
2.3 Basisprotokolle .....	8
2.3.1 Internet Protocol .....	9
2.3.2 Transmission Control Protocol .....	10
2.3.3 User Datagram Protocol .....	13
2.4 Anwendungen .....	13
2.4.1 Anwendungen mit gesichertem Datentransfer .....	14
2.4.1.1 Dateitransfer .....	14
2.4.1.2 World Wide Web .....	15
2.4.1.3 Rechnerfernzugriff .....	16
2.4.1.4 Nachrichtenaustausch .....	17
2.4.2 Echtzeitanwendungen .....	18
<b>3 Dienstgüteunterstützung in IP-Netzen</b> .....	<b>20</b>
3.1 Dienstgüte: Definitionen und Merkmale .....	20
3.1.1 Begriffsdefinitionen .....	20
3.1.2 Dienstgüteparameter .....	21
3.1.3 Differenzierung oder Überdimensionierung .....	22
3.1.4 Dienstgütegarantien .....	23
3.1.4.1 Absolute Garantien .....	23
3.1.4.2 Relative Garantien .....	24
3.1.5 Aggregation .....	25
3.1.6 Wirkungsbereiche von Dienstgütemechanismen .....	27
3.2 Verkehrsmanagement-Funktionen .....	27

3.2.1	Verwaltung von Ressourcen in Netzknoten	28
3.2.1.1	Bedienstrategien	28
3.2.1.2	Puffermanagement	34
3.2.2	Quellflusskontrolle und Verkehrsformung	39
3.2.2.1	Quellflusskontrolle	39
3.2.2.2	Verkehrsformung	40
3.2.3	Verbindungsannahmesteuerung	41
3.2.4	Verwaltung von Netzressourcen	43
3.2.5	Verkehrslenkung	44
3.2.5.1	Klassische Verkehrslenkungsverfahren in IP-Netzen	44
3.2.5.2	Dienstgüteorientierte Verkehrslenkung	46
3.2.5.3	QoSR ohne lokale Dienstgütemechanismen in Netzknoten	46
3.2.6	Entgelterhebung	47
3.2.6.1	Begriffe	48
3.2.6.2	Anforderungen	49
3.2.6.3	Pauschales Nutzungsgsentgelt	50
3.2.6.4	Nutzungsabhängige Entgelterhebung	51
3.2.6.5	Lösungsansätze für nutzungsabhängige Entgelterhebung	52
3.3	Ansätze und Architekturen zur Dienstgüteunterstützung in IP-Netzen	55
3.3.1	Dienstgüteunterstützung unterhalb der IP-Schicht	55
3.3.1.1	ATM-Dienstklassen	55
3.3.1.2	Verkehrsmanagement-Funktionen in ATM	56
3.3.1.3	Ansätze und Probleme einer Nutzung für IP-Verkehr	57
3.3.1.4	Multi-Protocol Label Switching	58
3.3.2	Die Integrated-Services-Architektur	59
3.3.2.1	Dienstmodell	60
3.3.2.2	Verkehrsmanagementfunktionen	61
3.3.2.3	Reservierungsprotokoll	63
3.3.2.4	Bewertung	65
3.3.3	Die Differentiated-Services-Architektur	66
3.3.3.1	Kerngedanken	66
3.3.3.2	Elemente der DiffServ-Architektur	67
3.3.3.3	Standardisierte PHBs	69
3.3.3.4	Relative Dienstgüte innerhalb von DiffServ	71
3.3.4	Hybride Ansätze	73
<b>4</b>	<b>Modellierung und Leistungsbewertung von IP-Netzen</b>	<b>74</b>
4.1	Ziele und Methoden der Leistungsbewertung	74
4.1.1	Ziele	74
4.1.2	Methoden	75
4.1.2.1	Untersuchungen am realen System	75
4.1.2.2	Untersuchungen mit Hilfe eines physikalischen Modells	76
4.1.2.3	Untersuchungen auf der Basis eines formalen Modells	77
4.2	Verkehrsmodellierung	79
4.2.1	Modellierung auf Paket- und Büschelebene	80

4.2.1.1	Erneuerungsprozesse . . . . .	80
4.2.1.2	Modelle für büschelförmigen Verkehr mit Kurzzeitabhängigkeit . .	81
4.2.1.3	Modelle für selbstähnlichen Verkehr . . . . .	85
4.2.2	Modelle für elastischen IP-Verkehr. . . . .	88
4.2.2.1	Modellierung von TCP-Eigenschaften . . . . .	89
4.2.2.2	Ungesättigte TCP-Quellen. . . . .	97
4.2.2.3	Modelle für dynamischen TCP-Verkehr . . . . .	98
4.2.2.4	Anwendungs- und nutzerbezogene WWW-Verkehrsmodelle . . . .	102
4.2.3	Modelle für echtzeitkritischen IP-Verkehr . . . . .	103
4.3	Leistungskenngrößen . . . . .	104
4.3.1	Grundlegende Leistungsmaße . . . . .	105
4.3.2	Kenngrößen für elastischen Verkehr . . . . .	106
4.3.2.1	Ungesättigte Quellen . . . . .	107
4.3.2.2	Dynamischer TCP-Verkehr . . . . .	108
4.3.2.3	Anwendungs- und Nutzermodelle . . . . .	111
4.3.3	Kenngrößen für Echtzeitverkehr . . . . .	112
4.3.4	Subjektive Qualität . . . . .	114
4.3.5	Leistungsmaße zur Beschreibung von transientem Verhalten . . . . .	116
<b>5</b>	<b>Ein neues Verfahren zur relativen Dienstgütedifferenzierung in IP-Netzknoten . .</b>	<b>118</b>
5.1	Randbedingungen und Anforderungen . . . . .	118
5.2	Prinzipielle Design-Alternativen . . . . .	121
5.3	Weighted Earliest Due Date. . . . .	122
5.3.1	Grundsätzliches Verhalten. . . . .	122
5.3.2	Realisierung . . . . .	124
5.3.3	Abschätzung der Überschreitungshäufigkeit. . . . .	125
5.3.4	Verhinderung des Aushungerns von Klassen . . . . .	128
5.3.5	Implementierungsaspekte . . . . .	129
5.4	Qualitativer Vergleich mit anderen Verfahren. . . . .	131
5.4.1	Scheduling . . . . .	131
5.4.1.1	Statische Prioritäten. . . . .	131
5.4.1.2	FQ-Verfahren . . . . .	132
5.4.1.3	Scheduler mit proportionaler Differenzierung der mittleren Verzögerung . . . . .	133
5.4.2	Puffermanagement. . . . .	135
5.4.2.1	Klassische Verfahren. . . . .	135
5.4.2.2	Verfahren auf der Basis von RED. . . . .	136
5.4.2.3	Verfahren zur proportionalen Verlustdifferenzierung. . . . .	137
5.4.3	Integrierte Verfahren . . . . .	138
5.5	Ende-zu-Ende-Betrachtung . . . . .	139

<b>6</b>	<b>Leistungsuntersuchung eines IP-Netzknotts mit Dienstgütedifferenzierung . . . . .</b>	<b>142</b>
6.1	Differenzierung von Echtzeitverkehr . . . . .	143
6.1.1	Modell . . . . .	143
6.1.2	Untersuchung der Grundeigenschaften von WEDD . . . . .	146
6.1.2.1	WEDD-Differenzierung bei unterschiedlichen Verkehrsrandbedingungen . . . . .	147
6.1.2.2	Auswirkungen der klassenspezifischen WEDD-Parameter . . . . .	149
6.1.2.3	Kurzzeitverhalten . . . . .	151
6.1.3	Einfluss von WEDD-Optionen und Systemparametern . . . . .	152
6.1.3.1	Verwerfen von verspäteten Paketen . . . . .	152
6.1.3.2	Sicherheitsabstand . . . . .	155
6.1.3.3	Realisierungsformen für die Schätzung von Überschreitungshäufigkeiten . . . . .	156
6.1.3.4	Auswirkungen der CSA-Erweiterung . . . . .	160
6.1.3.5	Einfluss der Puffergröße . . . . .	161
6.2	Differenzierung von elastischem Verkehr . . . . .	165
6.2.1	Modell . . . . .	165
6.2.2	Untersuchungen mit ungesättigten Quellen . . . . .	167
6.2.2.1	Einfluss der Quellenanzahl . . . . .	168
6.2.2.2	Auswirkung von CSA . . . . .	171
6.2.2.3	Einfluss der eingestellten maximalen Verzögerung . . . . .	172
6.2.2.4	Vergleich mit anderen Verfahren . . . . .	174
6.2.3	Untersuchungen mit dynamischem TCP-Verkehr . . . . .	178
6.2.3.1	Detaillierte Betrachtung eines einzelnen Lastpunktes . . . . .	180
6.2.3.2	Betrachtung unter variablem Angebot . . . . .	183
6.2.3.3	Einfluss der Maximalverzögerung . . . . .	185
6.2.3.4	Vergleich mit anderen Verfahren . . . . .	188
6.3	Integration von elastischem und unelastischem Verkehr . . . . .	191
6.3.1	Szenario . . . . .	191
6.3.2	Ergebnisse . . . . .	192
6.3.2.1	Untersuchungen ohne ungesättigte TCP-Quellen . . . . .	192
6.3.2.2	Untersuchungen mit ungesättigten TCP-Quellen . . . . .	194
<b>7</b>	<b>Zusammenfassung und Ausblick . . . . .</b>	<b>198</b>
	<b>Literaturverzeichnis . . . . .</b>	<b>202</b>
<b>A</b>	<b>Analyse eines TCP-Einzeltransfers . . . . .</b>	<b>221</b>
A.1	Modell . . . . .	221
A.2	Minimale Transferzeit ohne Delayed-ACK-Option . . . . .	223
A.3	Minimale Transferzeit mit Delayed-ACK-Option . . . . .	225
A.3.1	Approximative Analyse . . . . .	226
A.3.2	Exakte Analyse . . . . .	228
A.3.3	Berücksichtigung der Zeitüberwachung beim Empfänger . . . . .	230

A.4	Stochastische Betrachtung mit zufälliger Burstgröße . . . . .	232
A.4.1	Erwartungswert . . . . .	232
A.4.2	Verteilungsfunktion . . . . .	233
A.5	Numerische Auswertung . . . . .	234
<b>B</b>	<b>TCP-Simulationen für einen IP-Netzknoten ohne Dienstgütedifferenzierung . . . . .</b>	<b>237</b>
B.1	Untersuchungen mit ungesättigten Quellen . . . . .	238
B.1.1	Einfluss von Verkehrs- und Systemparametern . . . . .	238
B.1.1.1	Einfluss der Anzahl von Quellen . . . . .	238
B.1.1.2	Einfluss der Puffergröße . . . . .	239
B.1.1.3	Einfluss der absoluten Verzögerung . . . . .	241
B.1.2	Vergleich mit analytischen Ergebnissen . . . . .	243
B.1.3	Betrachtung der Warteschlangenlänge . . . . .	245
B.1.4	Vergleich von DT und RED . . . . .	247
B.2	Untersuchungen mit dynamischem TCP-Verkehr . . . . .	249
B.2.1	Detaillierte Betrachtung in einzelnen Lastpunkten . . . . .	250
B.2.1.1	Bedingte Mittelwerte . . . . .	250
B.2.1.2	Verteilungen . . . . .	254
B.2.1.3	Scatter Plots . . . . .	255
B.2.2	Betrachtung unter variablem Angebot . . . . .	257
B.2.3	Einfluss der Puffergröße . . . . .	259

# Abkürzungen

AAL	ATM Adaptation Layer
ABE	Alternative Best Effort
ABR	Available Bit Rate
ACK	Acknowledgement
ACR	Available Cell Rate
AdSpec	Additional Specification
AF	Assured Forwarding
AQM	Active Queue Management
ARPA	Advanced Research Project Agency
AS	Autonomous System
ATM	Asynchronous Transfer Mode
BA	Behavior Aggregate
BB	Bandwidth Broker
BE	Best Effort
BPR	Backlog-Proportional Rate (Scheduler)
BT	Burst Tolerance
CAC	Connection Admission Control
CBQ	Class-Based Queueing
CBR	Constant Bit Rate
CCDF	Complementary Cumulative Distribution Function
CDV	Cell Delay Variation
CL	Controlled-Load Service
CLIP	Classical IP over ATM
CLP	Cell Loss Priority
CLR	Cell Loss Ratio
COPS	Common Open Policy Service
CORBA	Common Object Request Broker Architecture
CoS	Classes of Service
CSA	Class Starvation Avoidance
CWnd	Congestion Window
DiffServ	Differentiated Services
DNS	Domain Name System
DRR	Deficit Round Robin

DSCP	Differentiated Services Code Point
DupACK	Duplicate Acknowledgement
DT	Drop Tail
E-Business	Electronic Business
E-Commerce	Electronic Commerce
E-Mail	Electronic Mail
ECN	Explicit Congestion Notification
EDD	Earliest Due Date
EDF	Earliest Deadline First
EF	Expedited Forwarding
EGP	Exterior Gateway Protocol
EPD	Early Packet Discard
EWMA	Exponentially Weighted Moving Average
FBM	Fractional Brownian Motion
FCFS	First Come First Serve
FEC	Forwarding Equivalency Class (im Zusammenhang mit MPLS)
FEC	Forward Error Correction (im Zusammenhang mit Echtzeitverkehr)
FGN	Fractional Gaussian Noise
FIFO	First In First Out
FlowSpec	Flow Specification
FQ	Fair Queueing
FTP	File Transfer Protocol
GCRA	Generic Cell Rate Algorithm
GFR	Guaranteed Frame Rate
GMDP	Generally Modulated Deterministic Process
GMPP	Generally Modulated Poisson Process
GMPLS	Generalized Multi-Protocol Label Switching
GoS	Grade of Service
GPS	Generalized Processor Sharing
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
IDP	Interrupted Deterministic Process
IGP	Interior Gateway Protocol
IntServ	Integrated Services
IP	Internet Protocol
IPP	Interrupted Poisson Process
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6

IRC	Internet Relay Chat
ISDN	Integrated Service Digital Network
ISP	Internet Service Provider
ITU	International Telecommunication Union
JoBS	Joint Buffer Management and Scheduling
LAN	Local Area Network
LANE	LAN Emulation
LB	Leaky Bucket
LDP	Label Distribution Protocol
LIFO	Last In First Out
LPD	Late Packet Discard
LRD	Long-Range Dependence
LSP	Label Switched Path
LSR	Label Switch Router
MAP	Markovian Arrival Process
MaxCTD	Maximum Cell Transfer Delay
MBS	Maximum Burst Size
MCR	Minimum Cell Rate
MDP	Mean Delay Proportional (Scheduler)
MMDP	Markov Modulated Deterministic Process
MMPP	Markov Modulated Poisson Process
MOS	Mean Opinion Score
MPLS	Multi-Protocol Label Switching
MPOA	Multi-Protocol over ATM
MSS	Maximum Segment Size
MTU	Maximum Transfer Unit
NAT	Network Address Translation
NFS	Network File System
NNTP	Network News Transfer Protocol
nrt-VBR	Non-Real-Time Variable Bit Rate
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
PBP	Poisson Burst Process
PBS	Partial Buffer Sharing
PCR	Peak Cell Rate
PDD	Proportional Delay Differentiation
PDU	Protocol Data Unit
PHB	Per-Hop Behavior
PHop	Previous Hop

PLQ	Probabilistic Longest Queue First
PLR	Proportional Loss Rate (Mechanism)
PMP	Paris Metro Pricing
PNNI	Private Network-Network Interface
POP	Post Office Protocol
PPD	Partial Packet Discard
PPP	Point to Point Protocol
pPPD	Preemptive Partial Packet Discard
PQCM	Proportional Queue Control Mechanism
PS	Processor Sharing
PSP	Progressive Second Price
PVC	Permanent Virtual Channel Connection
QoS	Quality of Service
QoSR	Quality of Service Routing
RCSP	Rate-Controlled Static Priority
RED	Random Early Detection
REM	Rate Envelope Multiplexing
RFC	Request for Comments
RIO	RED with In/Out Bit
RIP	Routing Information Protocol
RPQ	Rotating Priority Queues
RSpec	Reservation Specification
RSVP	Resource Reservation Protocol
RTO	Retransmission Timeout
RTP	Real-Time Transport Protocol
RTT	Round Trip Time
rt-VBR	Real-Time Variable Bit Rate
RWnd	Receiver Advertised Window
SACK	Selective Acknowledgement
SAR	Segmentation and Reassembly
SCFQ	Self-Clocked Fair Queueing
SCR	Sustainable Cell Rate
SDH	Synchronous Digital Hierarchy
SDP	Switched Deterministic Process
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SMTP	Simple Mail Transfer Protocol
SPP	Switched Poisson Process
SRD	Short-Range Dependence

SSR	Slow Start Restart
SSThresh	Slow Start Threshold
STE	Shortest Time to Extinction
SVC	Switched Virtual Channel Connection
SWSA	Silly Window Syndrome Avoidance
TB	Token Bucket
TC	Traffic Conditioning
TCA	Traffic Conditioning Agreement
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
ToS	Type of Service
TSpec	Traffic Specification
UBR	Unspecified Bit Rate
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
USD	User Share Differentiation
VC	Virtual Clock (Scheduling-Verfahren)
VC	Virtual Channel (im Zusammenhang mit ATM)
VCI	Virtual Channel Identifier
VLL	Virtual Leased Line
VoIP	Voice over IP
VP	Virtual Path
VPI	Virtual Path Identifier
VPN	Virtual Private Network
WAN	Wide Area Network
WDM	Wavelength Division Multiplexing
WEDD	Weighted Earliest Due Date
WFQ	Weighted Fair Queueing
WF <sup>2</sup> Q	Worst Case Fair Weighted Fair Queueing
WRED	Weighted Random Early Detection
WRR	Weighted Round Robin
WTP	Waiting Time Priority
WWW	World Wide Web
xDSL	Digital Subscriber Line (Sammelbegriff)
ZV	Zufallsvariable

# Formelzeichen

$a$	Spitzigkeit von büschelförmigem Verkehr (Abschnitt 4.2.1.3)
$A$	(Gesamt-)Angebot
$A_i$	Angebot in Klasse $i$
$A(i)$	Index des letzten Pakets aus Runde $i$ eines TCP-Bursttransfers, das sofort quittiert wurde (Anhang A)
$A_s$	Angebot, bei dem im integrierten Szenario nach Abschnitt 6.3 in Klasse 0 die Sättigung erreicht ist
$A_t$	ZV für die in einem Intervall der Länge $t$ ankommende Flüssigkeits-/Datenmenge (Abschnitte 3.2.2.1 und 4.2.1.3)
$b$	mittlere Burstgröße (in Byte)
$B$	ZV für die Burstgröße (in Byte)
$\tilde{B}$	ZV für fiktive Burstgröße bei Modell mit konstanter Paketlänge (Abschnitt 6.2.1)
$B(t)$	Menge der Warteschlangen bei FQ-Scheduling, die zum Zeitpunkt $t$ nicht leer sind (Abschnitt 3.2.1.1)
$\hat{B}_i$	Wert der Burstgröße am unteren Rand von Bereich $i$ beim Verfahren zur Bestimmung eines bedingten Mittelwerts (Abschnitt 4.3.2.2)
$c$	Faktor bei der Berechnung von $\text{VAR}[A_t]$ bei selbstähnlichem Verkehr (Abschnitt 4.2.1.3)
$c_i$	Überlaststempel ( <i>congestion tag</i> ) in Klasse $i$ (Abschnitt 5.3.2)
$C$	Linkrate
$d_i$	mittlere Verzögerung in Klasse $i$ (Abschnitt 5.4.1.3)
$d_p$	Ausspielverzögerung bei Multimedieverkehr (Abschnitt 4.3.3)
$E_i(t)$	Abschätzung für die Fristüberschreitungshäufigkeit in Klasse $i$ zum Zeitpunkt $t$ bei WEDD-Abarbeitung (Abschnitt 5.3.2)
$f(x)$	Verteilungsdichtefunktion (allgemein)
$f_B$	Büschelfaktor (Abschnitt 4.2.1.2)

$f_F(\Delta t)$	Fairness-Index, bezogen auf Intervalllänge $\Delta t$ (Abschnitt 4.3.2.1)
$f_\Delta$	Verzögerungsfaktor ( <i>delay factor</i> ) im PS-Modell (Anhang B.2)
$F(x)$	Verteilungsfunktion (allgemein)
$F_B(s)$	Verteilungsfunktion der Burstgröße (Anhang A)
$F_T(t)$	Verteilungsfunktion der Dauer eines TCP-Bursttransfers (Abschnitt 6.2.3.1)
$F_{T_{min}}(t)$	Verteilungsfunktion der minimalen Dauer eines TCP-Bursttransfers bei gegebener Verteilung der Burstgröße (Anhang A.4.2)
$F_\Phi(x)$	Verteilungsfunktion des <i>Fun Factor</i> bei dynamischem TCP-Verkehr (Abschnitt 6.2.3.1, Anhang B.2.1.2)
$g$	mittlerer Nutzdurchsatz einer TCP-Verbindung (Anhang B)
$g(s)$	bedingter mittlerer Nutzdurchsatz bei gegebener Burstgröße $s$ (Abschnitt 4.3.2.2)
$g_i$	mittlerer Nutzdurchsatz pro TCP-Verbindung in Klasse $i$
$G$	ZV für den Nutzdurchsatz innerhalb eines TCP-Bursttransfers (Abschnitt 4.3.2.2)
$G_{max}$	maximaler Durchsatz innerhalb eines TCP-Bursttransfers (Abschnitt 4.3.2.2)
$G_i(\Delta t)$	ZV für Nutzdurchsatz von Verbindung $i$ in einem Zeitintervall der Länge $\Delta t$ (Abschnitt 4.3.2.1)
$h$	Bedienzeit für ein TCP-Paket maximaler Länge (Anhang A)
$H$	Hurst-Parameter
$I(s)$	Index der „letzten Runde“ eines TCP-Bursttransfers bei gegebener Burstgröße $s$ (Anhang A)
$\tilde{I}(s)$	Näherung für den Index der „letzten Runde“ eines TCP-Bursttransfers bei gegebener Burstgröße $s$ im Fall von <i>Delayed ACK</i> (Anhang A)
$J(i)$	Index des ersten Pakets in Runde $i$ eines TCP-Bursttransfers (Anhang A)
$\tilde{J}(i)$	Näherung für den Index des ersten Pakets in Runde $i$ eines TCP-Bursttransfers im Fall von <i>Delayed ACK</i> (Anhang A)
$j_{max}$	Index des letzten Pakets eines TCP-Bursttransfers (Anhang A)
$k$	Minimalwert bei der Pareto-Verteilung (Abschnitt 4.2.1.3)
$\hat{k}$	Anzahl der Bits, um die ein Rechtsschieben erfolgt, im Zusammenhang mit der Realisierung der Häufigkeitsabschätzung bei WEDD (Abschnitt 5.3.5)
$K$	Anzahl von Klassen/Warteschlangen

$L$	Paketlänge
$L_{max}$	maximale Paketlänge (Abschnitt 4.2.1.2)
$L_{MSS}$	maximale Segmentgröße bei TCP-Verkehr
$L_0$	erwartete Paketlänge bei RED (Abschnitt 3.2.1.2)
$m$	mittlere (Gesamt-)Ankunftsrate (in bit/s)
$m_i$	Ankunftsrate in Klasse $i$ (Kapitel 6)
$m_E$	mittlere TCP-Empfangsrate (Bruttodurchsatz), bezogen auf eine (persistente) TCP-Verbindung (Anhang B)
$m_S$	mittlere TCP-Senderate, bezogen auf eine (persistente) TCP-Verbindung (Abschnitt 6.2.2, Anhang B)
$m_t$	ZV für die in einem Intervall der Länge $t$ auftretende mittlere Rate (Abschnitt 4.2.1.3)
$max_p$	Maximalwert bei der Bestimmung der Verwerfungswahrscheinlichkeit bei RED (Abschnitt 3.2.1.2)
$max_{p,i}$	Maximalwert in Klasse $i$ bei der Bestimmung der Verwerfungswahrscheinlichkeit bei WRED (Abschnitt 3.2.1.2)
$max_{th}$	oberer Schwellwert zur Bestimmung der Verwerfungswahrscheinlichkeit bei RED (Abschnitt 3.2.1.2)
$max_{th,i}$	oberer Schwellwert in Klasse $i$ bei der Bestimmung der Verwerfungswahrscheinlichkeit bei WRED (Abschnitt 3.2.1.2)
$min_{th}$	unterer Schwellwert zur Bestimmung der Verwerfungswahrscheinlichkeit bei RED (Abschnitt 3.2.1.2)
$min_{th,i}$	unterer Schwellwert in Klasse $i$ bei der Bestimmung der Verwerfungswahrscheinlichkeit bei RED (Abschnitt 3.2.1.2)
$M_i$	Zähler für die verworfenen Pakete in Klasse $i$ bei WEDD (Abschnitt 5.3)
$\Delta M$	mittlere Anzahl von Paketen zwischen zwei Reduktionen der $M_i$ -Zähler bei WEDD-Zählerreduktion mit kooperativem Schwellwertverfahren (Abschnitt 6.1.3.3)
$n$	Anzahl von Verbindungen/Quellen/Verkehrsflüssen (allgemein)
$n_i$	Anzahl von Verbindungen/Quellen/Verkehrsflüssen in Klasse $i$ (allgemein)
$n_B$	Anzahl der Bereiche bei der Bestimmung des bedingten Mittelwerts (Abschnitt 4.3.2.2)
$n_{max}$	maximale Anzahl von TCP-Verbindungen beim dynamischen TCP-Modell (Abschnitt 4.2.2.3)

$N(i)$	Anzahl der in Runde $i$ eines TCP-Transfers gesendeten Pakete (Anhang A)
$\tilde{N}(i)$	Näherung für die Anzahl der in Runde $i$ eines TCP-Transfers gesendeten Pakete im Fall von <i>Delayed ACK</i> (Anhang A)
$\hat{N}$	Schwellwert, bei dem eine Reduktion der für die Häufigkeitsabschätzung bei WEDD benötigten Zähler erfolgt (Abschnitt 5.3)
$N_i$	Zähler für bearbeitete und verworfene Pakete in Klasse $i$ bei WEDD (Abschnitt 5.3)
$N_A(i)$	Gesamtzahl der in den Runden 1 bis $i$ gesendeten Pakete eines TCP-Bursttransfers (Anhang A)
$N_P$	ZV für die Anzahl von Paketen innerhalb eines Bursts
$\Delta N_i$	mittlere Anzahl von Paketen in Klasse $i$ zwischen zwei Zählerreduktionen bei nichtkooperativem Schwellwertverfahren (Abschnitt 6.1.3.3)
$p$	Gesamtverwerfungswahrscheinlichkeit über alle Klassen (Kapitel 6)
$p(t_1, t_2)$	Verlustwahrscheinlichkeit im Intervall $[t_1, t_2]$ (Abschnitt 4.3.5)
$p_i$	Verwerfungs-/Fristüberschreitungswahrscheinlichkeit in Klasse $i$ (Kapitel 5)
$p_i^{(v)}$	Verwerfungswahrscheinlichkeit in Klasse $i$ in Knoten $v$ (Abschnitt 5.5)
$p_{i, g}$	Ende-zu-Ende-Verwerfungswahrscheinlichkeit in Klasse $i$ (Abschnitt 5.5)
$p_g$	Gesamtverlustwahrscheinlichkeit (Verluste im Netz sowie im Endgerät) bei Multimedieverkehr (Abschnitt 4.3.3)
$p_N$	Wahrscheinlichkeit, dass ein (Multimedia-)Paket im Netz verloren geht (Abschnitt 4.3.3)
$p_V$	Verwerfungswahrscheinlichkeit bei RED (Abschnitt 3.2.1.2)
$q$	Anzahl der Pakete, nach deren Erhalt ein TCP-Empfänger im Fall von <i>Delayed ACK</i> eine Bestätigung sendet (Anhang A)
$q_i$	Wert eines (allgemeinen) Leistungsmaßes in Klasse $i$ (Abschnitt 3.1.4)
$\bar{q}_i(t, t + \Delta t)$	Mittelwert eines (allgemeinen) Leistungsmaßes in Klasse $i$ im Intervall $(t, t + \Delta t)$ (Abschnitt 3.1.4)
$Q(t)$	Warteschlangenlänge zum Zeitpunkt $t$ (Abschnitt 3.2.1.2)
$Q_i(t)$	Warteschlangenlänge in Klasse $i$ zum Zeitpunkt $t$ (Abschnitt 5.4.1.2)
$\bar{Q}$	gemittelte Warteschlangenlänge bei RED (Abschnitt 3.2.1.2)
$r$	Verhältnis der Raten auf dem Engpass- und dem Zugangslink (Anhang B)
$r_g$	ganzzahliger Teil des Ratenverhältnisses $r$ (Anhang B.2)

$R_i(t)$	Bedienrate für Warteschlange $i$ zum Zeitpunkt $t$ bei FQ-Scheduling (Abschnitte 3.2.1.1 und 5.4.1.2)
$R_{LB}$	Abflussrate (in bit/s) eines <i>Leaky Bucket</i> (Abschnitte 3.2.2.1 und 3.3.2)
$R_{max}$	maximale Rate einer TCP-Verbindung/Bedienrate des Zugangslinks (Kapitel 6, Anhang A, Anhang B)
$R_P$	Paketrate (Spitzenrate) innerhalb eines Bursts (Abschnitte 4.2.1.2 und 6.1)
$R_{Peak}$	Spitzenrate, die in der TSPEC von RSVP angegeben wird (Abschnitt 3.3.2)
$R_{Res}$	reservierte Rate laut RSpec von RSVP Abschnitt 3.3.2)
$R_{Res, i}$	für Fluss $i$ reservierte Rate laut RSpec von RSVP Abschnitt 3.3.2)
$s$	Laufvariable für die Burstgröße (allgemein)
$s'$	verbleibende Datenmenge eines TCP-Bursts nach Ende der Timeout-Runden (Anhang A.3.3)
$s_i$	(allgemeiner) Differenzierungsparameter in Klasse $i$ bei proportionaler Dienstgütedifferenzierung (Abschnitte 3.1.4 und 5.4)
$s_{ij}$	Verhältnis der Differenzierungsparameter in Klasse $i$ und $j$ (Abschnitt 5.4.1.3)
$S$	Pufferkapazität allgemein sowie im Bediensystem, das den Engpasslink modelliert
$S_E$	Puffergröße im Bediensystem, das den empfangsseitigen Zugangslink modelliert
$S_{LB}$	Kapazität (in Byte) eines <i>Leaky Bucket</i> (Abschnitt 3.2.2.1 und 3.3.2)
$S_S$	Puffergröße im Bediensystem, das den sendeseitigen Zugangslink modelliert
$S_{TCP}$	Größe des TCP-Empfangspuffers
$t$	Zeit (allgemein)
$\Delta t$	Länge eines Zeitintervalls (allgemein)
$\Delta t_0$	Zeitintervall seit dem letzten Bedienende bei RED (Abschnitt 3.2.1.2)
$t(s)$	bedingte mittlere Dauer eines TCP-Bursttransfers (Abschnitt 4.3.2.2)
$t_A$	Ankunftszeitpunkt eines Pakets (Abschnitt 5.3.2)
$t_P(i)$	Ausspielzeitpunkt für Paket mit Index $i$ innerhalb eines Multimediaverkehrsstroms (Abschnitt 4.3.3)
$t_{PS}(s)$	bedingte mittlere Bearbeitungsdauer einer Anforderung im PS-Modell (Anhang B.2)

$t_R(i)$	Ankunftszeit der ersten Daten aus Runde $i$ (Anhang A)
$t_S(i)$	Sendezeitpunkt für Paket mit Index $i$ innerhalb eines Multimedieverkehrstroms (Abschnitt 4.3.3)
$t_0$	Sendezeitpunkt für das erste Paket eines Multimedieverkehrstroms (Abschnitt 4.3.3)
$T$	ZV für die Dauer eines TCP-Bursttransfers (Abschnitt 4.3.2.2)
$T_0$	Schwellwert bei modifiziertem <i>Fun Factor</i> (Abschnitt 4.3.2.2)
$T_B$	Ankunftsabstand von Bursts im M/G/ $\infty$ -Modell
$T_D$	ZV für die Paketdurchlaufzeit in einem Knoten (Kapitel 6)
$\overline{T_D}$	mittlere Paketdurchlaufzeit in einem Knoten (Anhang B)
$T_D^{(v)}$	ZV für die Paketdurchlaufzeit in Knoten $v$ (Abschnitt 5.5)
$T_{D,i}$	ZV für die Paketdurchlaufzeit in einem Knoten in Klasse $i$ (Kapitel 6)
$T_{DelACK}$	Zeitdauer, nach der der <i>Delayed ACK</i> -Timer beim TCP-Empfänger abläuft (Anhang A)
$T_{min}$	minimale Dauer eines TCP-Bursttransfers (allgemein, Abschnitt 4.3.2.2)
$T_{min}(s)$	minimale Dauer eines TCP-Bursttransfers bei gegebener Burstgröße $s$ (Abschnitt 4.3.2.2, Anhang A)
$T_{min}'(s)$	minimale Dauer eines TCP-Bursttransfers bei gegebener Burstgröße $s$ einschließlich eventueller Timeout-Runden (Anhang A.3.3)
$\overline{T_{min}}$	mittlere Minimaldauer eines TCP-Bursttransfers bei gegebener Verteilung der Burstgröße (Anhang A.4.1)
$T_{Off}$	ZV für die Dauer der „Off“-Phase beim On-Off-Modell
$T_{Offset}(s)$	Anteil an der minimalen Dauer eines TCP-Bursttransfers bei gegebener Burstgröße $s$ , der durch die TCP-Mechanismen verursacht wird (Anhang A)
$T_P$	Ankunftsabstand von Paketen innerhalb eines Bursts (Abschnitte 4.2.1 und 6.1)
$T_Q$	Wartezeit im IntServ-Modell (Abschnitt 3.3.2)
$T_{Q,i}(t)$	Wartezeit des nächsten zum Zeitpunkt $t$ in Klasse $i$ anstehenden Pakets bei WTP-Scheduling (Abschnitt 5.4.1.3)
$T_{RTO}$	minimale Zeitspanne, nach der ein Wiederholungstimeout bei TCP eintritt (Anhang B)
$\overline{T_{RTT}}$	mittlere RTT bei TCP-Verkehr (Kapitel 6, Anhang B)
$\overline{T_{RTT,i}}$	mittlere RTT in Klasse $i$ bei TCP-Verkehr (Kapitel 6)

$T_{RTT, max}$	maximale RTT bei TCP-Verkehr (Anhang B)
$w$	initiale Größe des CWnd (Anhang A)
$w'$	Größe des CWnd nach Ende der Timeout-Runden (Anhang A.3.3)
$w_i$	WEDD-Gewichtungsfaktor für Klasse $i$ (Kapitel 5)
$w_q$	Gewichtungsfaktor bei der Mittelung der Warteschlangenlänge bei RED (Abschnitt 3.2.1.2)
$W$	Sendefenster (Abschnitt 4.2.2.1)
$W(i)$	Größe des CWnd/Sendefensters am Beginn von Runde $i$ (Anhang A.3.2)
$W_C$	Größe des CWnd (Abschnitt 4.2.2.1)
$W_R$	Größe des dem Sender mitgeteilten Empfangsfensters (Abschnitt 4.2.2.1)
$W_F$	<i>Flight Size</i> beim TCP-Sender (Abschnitt 4.2.2.1)
$\hat{z}$	Anzahl der Bits für die Zähler zur Häufigkeitsabschätzung bei WEDD (Abschnitt 5.3.5)
$Z(i)$	Index des letzten in Runde $i$ eines TCP-Bursttransfers gesendeten Pakets (Anhang A)
$Z'(i)$	Index des letzten in der Timeout-Runde $i$ eines TCP-Bursttransfers gesendeten Pakets (Anhang A.3.3)
$Z_t$	ZV für einen normierten FBM-Ankunftsprozess (Abschnitt 4.2.1.3)
$\alpha$	Formparameter der Pareto-Verteilung (Abschnitt 4.2.1.3)
$\beta$	Parameter für die Reduktion der für die Häufigkeitsabschätzung bei WEDD benötigten Zähler (Abschnitt 5.3)
$\hat{\beta}$	Parameter für die Reduktion der für die Häufigkeitsabschätzung bei WEDD benötigten Zähler bei Verwendung eines Zählerschwellwerts (Abschnitt 5.3)
$\gamma$	Exponent bei der Realisierung der CSA-Option von WEDD (Abschnitt 5.3)
$\delta_i$	Maximalverzögerung in Klasse $i$ allgemein (Abschnitt 3.2.1.1) sowie bei WEDD im Speziellen (Kapitel 5)
$\delta_i^{(v)}$	maximale Verzögerung in Klasse $i$ in Knoten $v$ (Abschnitt 5.5)
$\delta_{i, g}$	maximale Gesamtverzögerung in Klasse $i$ (Abschnitt 5.5)
$\Delta$	ZV für den <i>Delay Factor</i> (Abschnitt 4.3.2.2)
$\Delta'$	ZV für den modifizierten <i>Delay Factor</i> mit Schwellwert (Abschnitt 4.3.2.2)
$\varepsilon_i$	Sicherheitsabstand in Klasse $i$ bei WEDD-Abarbeitung (Abschnitt 5.3.2)
$\theta_S$	Wert der <i>Slow-Start-Schwelle</i> (SSThresh) bei TCP (Abschnitt 4.2.2.1)

$\Theta_i$	Pufferschwellwert für Klasse $i$ bei PBS (Abschnitt 3.2.1.2)
$\kappa$	Index der „letzten Runde“ eines TCP-Transfers (Anhang A)
$\lambda$	Ankunftsrate allgemein (4.2.1.1) sowie von Bursts im M/G/ $\infty$ -Modell (Abschnitt 4.2.1.2)
$\Lambda(i)$	Hilfsfunktion des Rundenindex $i$ zur Berechnung von Mittelwert und Verteilungsfunktion der Minimaldauer eines TCP-Bursttransfers (Anhang A.4)
$\mu$	approximative Maßzahl für die Erhöhung des CWnd bei TCP mit <i>Delayed ACK</i> (Anhang A.3.1)
$\nu$	Anzahl der Timeout-Runden bei einem TCP-Bursttransfer (Anhang A.3.3)
$\tau$	konstanter Anteil an der RTT (z. B. Signallaufzeit) bei TCP-Verkehr (Abschnitt 5.4.1.3, Kapitel 6, Anhang A, Anhang B)
$\hat{\tau}$	modifizierte konstante Verzögerung zur Berechnung der maximalen Transferdauer von TCP-Übertragungen ohne Paketverlust (Anhang B.2)
$\phi_i$	Gewichtungsfaktor für Klasse/Warteschlange $i$ bei FQ-Scheduling (Abschnitte 3.2.1.1 und 3.3.2.2)
$\phi_{CL}$	Gewichtungsfaktor für aggregierten CL-Verkehr (Abschnitt 3.3.2.2)
$\varphi_{\Delta}$	mittlerer <i>Fun Factor</i> bei Mittelung über $\Delta$ (Abschnitt 4.3.2.2)
$\varphi_{\Delta}(s)$	bedingter mittlerer <i>Fun Factor</i> bei gegebener Burstgröße $s$ und Mittelung über $\Delta$ (Abschnitt 4.3.2.2)
$\varphi_{\Delta, PS}$	mittlerer <i>Fun Factor</i> bei Mittelung über $\Delta$ im PS-Modell (Abschnitt B.2)
$\varphi_{\Phi}$	mittlerer <i>Fun Factor</i> bei Mittelung über $\Phi$ (Abschnitt 4.3.2.2)
$\varphi_{\Phi}(s)$	bedingter mittlerer <i>Fun Factor</i> bei gegebener Burstgröße $s$ und Mittelung über $\Phi$ (Abschnitt 4.3.2.2)
$\Phi$	ZV für den <i>Fun Factor</i> (Abschnitt 4.3.2.2)
$\Phi'$	ZV für den modifizierten <i>Fun Factor</i> mit Schwellwert (Abschnitt 4.3.2.2)
$\Psi_i(t)$	Hilfsfunktion in Abhängigkeit vom Rundenindex $i$ zur abschnittswisen Beschreibung der Verteilungsfunktion der Minimaldauer eines TCP-Bursttransfers (Anhang A.4.2)
$\hat{\Psi}(t)$	Hilfsfunktion zur Beschreibung des letzten Abschnitts der Verteilungsfunktion der Minimaldauer eines TCP-Bursttransfers (Anhang A.4.2)
$\Omega(i)$	Hilfsfunktion des Rundenindex $i$ zur Berechnung des Mittelwerts der Minimaldauer eines TCP-Bursttransfers (Anhang A.4.1)

# Kapitel 1

## Einleitung

In den letzten Jahren wurde erkannt, dass der Kommunikationstechnik in Verbindung mit der Informationstechnik eine immer wichtiger werdende Rolle innerhalb unserer Gesellschaft zukommt, und zwar sowohl im privaten wie im geschäftlichen Bereich. Beigetragen zu dieser Erkenntnis hat vor allem die Entwicklung des Internets, das als Netz zur Verbindung von Forschungseinrichtungen entstand und sich mittlerweile zu einem weltumspannenden Verbund von Rechnernetzen mit Hunderten Millionen von Nutzern entwickelt hat.

Als Erfolgsgarant für das Internet gilt der verteilte Ansatz auf der Basis des *Internet Protocol* (IP), der eine vergleichsweise einfache Anbindung neuer Endgeräte und Teilnetze an ein bestehendes IP-Netz ermöglicht. Auf dieser Basis konnten sich im Lauf der Zeit zahlreiche Dienste etablieren, deren Datenaufkommen mittlerweile das von Sprachtelefonie als klassischer Kommunikationsanwendung deutlich übersteigt. Herausragende Bedeutung unter den Diensten im Internet hat vor allem das *World Wide Web* (WWW) erlangt, das zunächst einen Abruf von Hypertext-basierten Dateien ermöglicht, inzwischen aber auch als Plattform für elektronischen Handel (E-Commerce) und viele weitere Anwendungen dient. Darüber hinaus spielt der Austausch von Nachrichten über IP-Netze eine große Rolle, der sich immer mehr hin zum *Unified Messaging* entwickelt mit dem Ziel, ein Versenden von Nachrichten jeder Art über ein gemeinsames System zu ermöglichen. Schließlich gewinnt auch die Multimediakommunikation mit ihren Echtzeitanforderungen im Internet an Bedeutung. Dies reicht bis hin zu der teilweise bereits Realität gewordenen Vorstellung, auch Sprachtelefonie über IP abzuwickeln und damit eine Konvergenz von klassischen Telekommunikationsnetzen und Datennetzen zu einem gemeinsamen Netz auf der Basis von IP herbeizuführen.

Die Entwicklung des Internets und seiner Dienste und Anwendungen ist begleitet von einem enormen Fortschritt bei den Technologien, die als Grundlage für IP-Netze dienen. Hier ist in erster Linie die Zunahme an Übertragungskapazität in Weitverkehrsnetzen zu nennen, die vor allem durch die Einführung optischer Netze auf der Basis der WDM-Technologie (*wavelength*

*division multiplexing*) ermöglicht wurde. Andererseits haben Technologien wie xDSL (*digital subscriber line*) im Zugangsnetz oder Gbit/s-Ethernet im Bereich lokaler Netze auch dort zu einer Erhöhung der Bandbreite geführt. Schließlich sind auch neue Generationen von Mobilfunknetzen zu sehen, die ebenfalls höhere Übertragungsraten erlauben.

Trotz des Ausbaus von Kapazitäten kommt es in IP-Netzen häufig zu Überlastsituationen, die sich für den Anwender in Form einer reduzierten Dienstgüte äußern. Grund dafür ist der durch zunehmende Nutzerzahlen, bandbreiteintensive Anwendungen und erhöhte Anschlussbandbreiten verursachte stetige Anstieg des Verkehrsaufkommens. Das Problem verschärft sich dadurch, dass einzelne Anwendungen (vor allem aus dem Bereich der Multimedia-Kommunikation) erhöhte Anforderungen im Hinblick auf Verzögerungen haben. Das nach dem Prinzip des *Best Effort* arbeitende IP sieht jedoch zunächst keine Möglichkeiten zum Schutz von Verkehrsflüssen im Falle einer Überlast vor. Daher gibt es seit einiger Zeit Bestrebungen, durch zusätzliche Mechanismen des Verkehrsmanagements eine Dienstgüteunterstützung in IP-Netzen zu bieten und damit den unterschiedlichen Anforderungen von Anwendern und Anwendungen gerecht zu werden.

Unter den Ansätzen für eine Dienstgüteunterstützung werden in dieser Arbeit schwerpunktmäßig solche betrachtet, die eine relative Dienstgütedifferenzierung vornehmen. Bei diesem Ansatz werden verschiedene Dienstgüteklassen unterschieden, wobei lediglich eine Besserstellung einzelner Klassen gewährleistet wird, ohne jedoch absolute Garantien bzgl. der Einhaltung konkreter Werte von Dienstgüteparametern zu geben. Von den verschiedenen Verkehrsmanagementfunktionen kommt dabei den Bedienstrategien und Puffermanagementmechanismen, die innerhalb der einzelnen IP-Netzknoten wirken und für eine differenzierte Behandlung von Verkehrsströmen sorgen, die größte Bedeutung zu. Die Untersuchung derartiger Mechanismen steht daher auch im Mittelpunkt der vorliegenden Arbeit. Ziel ist dabei die Entwicklung eines integrierten Scheduling- und Puffermanagementverfahrens, bei dem eine proportionale Differenzierung vorgenommen wird, sodass der Grad der Besserstellung einzelner Klassen quantifiziert werden kann. Gleichzeitig stellt die Leistungsuntersuchung eines mit diesem Verfahren arbeitenden Netzknotens auf der Basis einer geeigneten Verkehrs- und Systemmodellierung einen Schwerpunkt dieser Arbeit dar.

Im folgenden Kapitel werden zunächst einige Grundlagen zu IP-Netzen aufgearbeitet und in kompakter Form präsentiert. Dazu gehören die Bestandteile einer IP-Netzarchitektur ebenso wie die grundlegenden Protokolle der Vermittlungs- und Transportschicht. Darüber hinaus erfolgt eine Zusammenstellung, Klassifizierung und Charakterisierung wichtiger Anwendungen in IP-Netzen. Dabei findet eine grundsätzliche Unterteilung in Anwendungen, bei denen die gesicherte Übermittlung von Daten im Vordergrund steht, und solche mit Echtzeitanforderungen an die Datenübertragung statt.

Kapitel 3 beschäftigt sich allgemein mit Mechanismen und Architekturen zur Dienstgüteunterstützung und ermöglicht so eine Einordnung des im Rahmen dieser Arbeit entwickelten Verfahrens. Zunächst werden dabei einige grundlegende Merkmale im Zusammenhang mit dem Begriff der Dienstgüte identifiziert. Anschließend erfolgt eine Vorstellung der verschiedenen Verkehrsmanagement-Funktionen, die innerhalb von Dienstgüte unterstützenden Architekturen eine Rolle spielen. Ein Schwerpunkt liegt dabei auf der Klassifikation und Beschreibung von Scheduling- und Puffermanagementmechanismen. Darauf aufbauend werden die wichtigsten Ansätze und Architekturen zur Bereitstellung von Dienstgüte in IP-Netzen präsentiert. Insbesondere wird gezeigt, wie die einzelnen Verkehrsmanagement-Funktionen darin zur Anwendung kommen.

Im vierten Kapitel geht es um die Modellierung von IP-Netzen vor dem Hintergrund einer Leistungsbewertung von Systemen und Verfahren. Es erfolgt zuerst eine Gegenüberstellung verschiedener Untersuchungsmethoden und der damit verbundenen Anforderungen und Schwierigkeiten im Kontext von IP-Netzen. Eine große Herausforderung bei der Untersuchung stellt die in Messungen beobachtete Langzeitabhängigkeit von IP-Verkehr und das damit verbundene Phänomen der Selbstähnlichkeit dar. Dies muss auch bei der Verkehrsmodellierung berücksichtigt werden. Ein weiterer wichtiger Punkt bei der Modellierung von IP-Verkehr ist die Adaptivität des erzeugten Verkehrs an den aktuellen Netzzustand. Diese ergibt sich aus der Verwendung des in IP-Netzen überwiegend als Transportprotokoll eingesetzten *Transmission Control Protocol* (TCP). Im Hinblick auf die Modellierung erfordert dies eine Berücksichtigung der TCP-Mechanismen zur Fluss- und Überlaststeuerung und weiterer Eigenschaften dieses Protokolls. Darüber hinaus gilt es aber auch, ebenso einfache wie repräsentative Modelle für den Verkehr zu finden, der von den auf TCP aufsetzenden Anwendungen erzeugt wird. Gleiches gilt für den von Multimedia-Anwendungen generierten Echtzeitverkehr. Schließlich werden in Kapitel 4 eine Reihe von Leistungsmetriken vorgestellt, die für eine Beurteilung in Frage kommen und an denen sich auch Verfahren des Verkehrsmanagement orientieren müssen. Dabei findet wie im Fall der Verkehrsmodellierung eine Betrachtung auf unterschiedlichen Ebenen statt.

Aufbauend auf der Klassifikation von Scheduling- und Puffermanagementverfahren in Kapitel 3 wird in Kapitel 5 ein Verfahren vorgeschlagen, das eine relative Differenzierung von Verkehrsströmen innerhalb eines IP-Netzknotens ermöglicht. Das Verfahren erlaubt zum einen die Formulierung unterschiedlicher Verzögerungsanforderungen. Zum anderen wird eine proportionale Differenzierung hinsichtlich der Wahrscheinlichkeiten für die Überschreitung vorgegebener Maximalverzögerungen vorgenommen, indem für die einzelnen Klassen unterschiedliche Gewichte vergeben werden. Damit ist es neben einer Unterscheidung von echtzeitkritischem und TCP-basiertem Verkehr auch möglich, innerhalb dieser beiden Verkehrsgruppen mehrere Klassen zu unterscheiden. Auf Fragen im Zusammenhang mit der Realisierung des Verfahrens geht Kapitel 5 ebenso ein wie auf eine Reihe von Optionen, die

bei der Anwendung des Verfahrens zur Verfügung stehen. Außerdem wird ein qualitativer Vergleich mit aus der Literatur bekannten Mechanismen zur relativen Differenzierung vorgenommen. Schließlich werden auch Aspekte behandelt, die in einem Netzscenario anstelle der Betrachtung eines einzelnen Knotens zu berücksichtigen sind, insbesondere die sich daraus ergebenden prinzipiellen Auswirkungen auf die Ende-zu-Ende-Dienstgüte.

Kapitel 6 enthält die Ergebnisse von Leistungsuntersuchungen des in Kapitel 5 vorgestellten Verfahrens unter verschiedenen Verkehrsrandbedingungen sowie von Vergleichsstudien mit anderen Verfahren. Untersuchungen, die abgesehen von analytischen Randbetrachtungen simulativ erfolgen, werden zunächst für ein Szenario durchgeführt, das nur Echtzeitverkehr auf der Basis eines einfachen Modells für büschelförmigen Verkehr enthält. Dabei werden neben den Parametern des Verkehrsmodells auch verschiedene Systemparameter variiert sowie der Einfluss bestimmter Optionen des Verfahrens untersucht. Anschließend folgen Simulationsstudien, in denen die Eigenschaften von TCP-Verkehr berücksichtigt werden. Zunächst wird dabei von persistenten TCP-Verbindungen ausgegangen, die immer Daten zu senden haben. Darüber hinaus werden Untersuchungen mit dynamischem TCP-Verkehr durchgeführt, bei dem Verbindungen nur von kurzer Dauer sind und daher das Startverhalten innerhalb einer TCP-Übertragung eine große Rolle spielt. In beiden Fällen erfolgt eine Bezugnahme auf Ergebnisse im Fall ohne differenzierte Behandlung von Verkehrsströmen, die im Anhang präsentiert werden. Diesen Ergebnissen liegen teilweise analytische Betrachtungen (Anhang A), teilweise Simulationsstudien (Anhang B) zugrunde. Abschließend erfolgt in Kapitel 6 die Untersuchung eines integrierten Szenarios, in dem sowohl Echtzeitverkehr als auch TCP-Verkehr zur Anwendung kommt.

# Kapitel 2

## Grundlagen von IP-Netzen

Um die in den folgenden Kapiteln behandelten Dienstgütemechanismen einordnen zu können, werden in diesem Kapitel zunächst die elementaren Grundlagen von IP-Netzen vermittelt. Dazu gehören Netzarchitektur und Basisprotokolle ebenso wie die in IP-Netzen vorzufindenden Anwendungen, die insbesondere im Zusammenhang mit der Modellierung von IP-Verkehr von Interesse sind. Einen umfangreicheren Einblick in Protokolle und Architektur IP-basierter Netze geben Übersichtswerke wie [65, 76, 237, 243, 259] sowie die zahlreichen RFCs (*requests for comments*) der Internet-Standardisierungsorganisation IETF (*Internet Engineering Task Force*).

### 2.1 Geschichte des Internets

Das Internet ist ein auf der Paketvermittlungstechnik basierendes Netz, an das heute Millionen von Endgeräten angeschlossen sind. Seine Entstehung verdankt es einer Initiative der ARPA (*Advanced Research Project Agency*) Ende der 1960er Jahre mit dem Ziel, Rechner an verschiedenen US-amerikanischen Forschungs- und Militäreinrichtungen miteinander zu verbinden. Als Ergebnis entstand das ARPANET, in dem die einzelnen Rechner zunächst direkt über Punkt-zu-Punkt-Verbindungen auf Basis unterschiedlicher Technologien kommunizierten.

Mit zunehmender Größe sah man sich in folgenden Jahren zur Einführung einer Architektur gezwungen, die durch Verwendung von IP als einheitliches Kommunikationsprotokoll der Vermittlungsschicht eine Unabhängigkeit von den verschiedenen Netzwerk-Technologien ermöglichte. IP wurde zusammen mit dem Transportprotokoll TCP standardisiert [226, 227] und als integraler Bestandteil zunächst des UNIX-Betriebssystems, später auch in anderen Betriebssystemen implementiert und bereitgestellt. Im weiteren Verlauf entwickelte sich aus dem ARPANET das Internet als globales Netz, das aber zunächst weitgehend auf Forschungseinrichtungen beschränkt blieb.

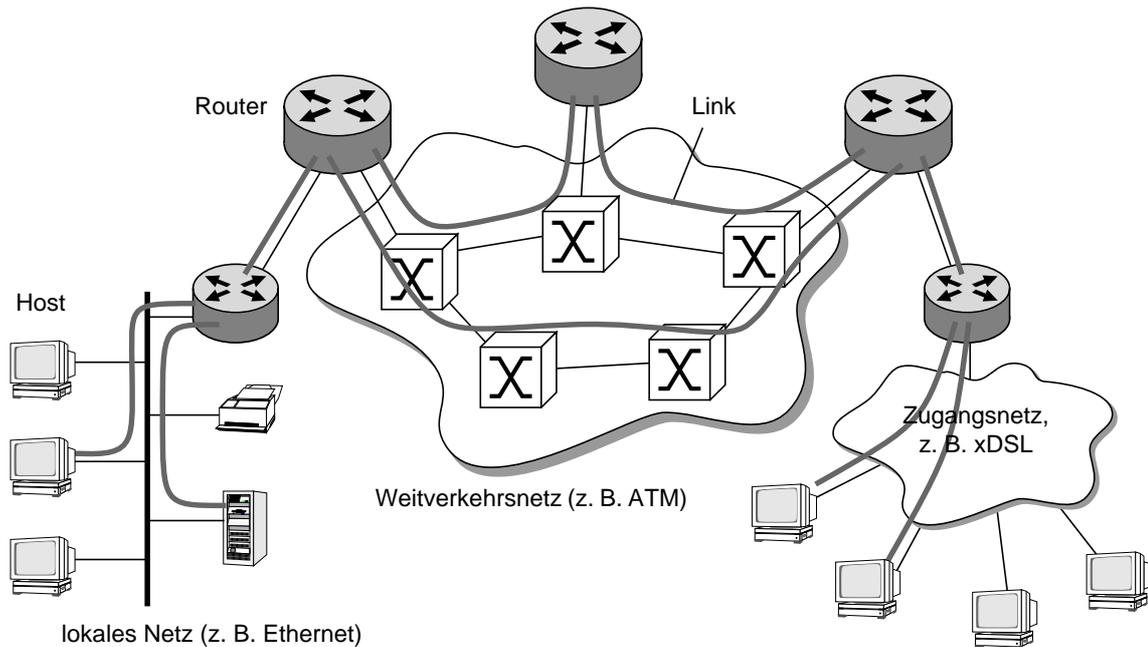
Ein wichtiger Schritt in der weiteren Entwicklung war die Einführung der WWW-Technologie (*world wide web*), die es ermöglicht, miteinander verknüpfte Hypertext-Dokumente von WWW-Server-Rechnern abzurufen und mit Hilfe eines Client-Anwendungsprogramms (WWW-Browser) zu betrachten. Bis dahin waren Rechnerfernzugriff (z. B. Telnet), elektronische Post (*electronic mail*, E-Mail) und Dateitransfer die wichtigsten Anwendungen im Internet. Neben WWW wurden im Laufe der Zeit noch eine Vielzahl weiterer Anwendungen, z. B. zur Multimedia-Kommunikation, entwickelt. Insgesamt fand damit das Internet auch den Weg hin zu Firmen und Privathaushalten, sodass sich aus dem einstigen Forschungsnetz das kommerzielle Internet entwickelte, wie wir es heute kennen. Die Grundlage bilden dabei weiterhin die Basisprotokolle, die in Abschnitt 2.3 genauer beschreiben werden, insbesondere das Internet-Protokoll IP.

Darüber hinaus gibt es aber auch immer mehr Netze, die auf IP basieren, aber nicht zum globalen Internet gehören, unter dem im engeren Sinne nur der öffentlich zugängliche Teil der weltweit miteinander verbundenen IP-Netze verstanden wird. Unter diesen nicht öffentlichen IP-Netzen sind v. a. Firmennetze zu nennen, aber z. B. auch Teile von Mobilfunknetzen, die auf der Basis von IP realisiert sind. Die im Rahmen dieser Arbeit entwickelten und untersuchten Mechanismen beziehen sich allgemein auf IP-basierte Netze, weshalb in der Folge dieser Begriff anstelle des uneinheitlich eingesetzten Terms „Internet“ verwendet wird.

## 2.2 Netzarchitektur

Innerhalb von IP-Netzen können – wie in vielen anderen Netzen auch – im Wesentlichen drei Arten von Komponenten unterschieden werden (siehe auch Bild 2.1):

- **Endgeräte** (*hosts*): Damit handelt es sich i. d. R. um Rechner, die bzgl. der IP-Schicht die Kommunikationsendpunkte darstellen. In Abhängigkeit von den Anwendungen, die über IP realisiert werden, ist häufig eine Unterteilung in Client- und Server-Rechner möglich, entsprechend der Rolle als Dienstanutzer bzw. -erbringer. Endgeräte können aber auch Peripheriegeräte (z. B. Drucker) oder Mobiltelefone sein.
- **Vermittlungsknoten** (*routers*): In einem Router wird anhand der in IP-Paketen eingetragenen Zieladresse entschieden, auf welchem Verbindungsabschnitt das Paket weitergeschickt wird. Dabei findet die eigentliche Verkehrslenkung (*routing*) losgelöst von der Paketweiterleitung (*forwarding*) statt, indem in vergleichsweise großen Zeitabständen Informationen zwischen Routern ausgetauscht werden und Algorithmen ablaufen, die zu jedem Ziel den nächsten Wegabschnitt ermitteln (siehe dazu auch Abschnitt 3.2.5). Das Ergebnis wird in jedem Knoten in einer Tabelle (*routing table*) eingetragen, sodass für einzelne Pakete nur noch ein Nachschlagen in dieser Tabelle (*table lookup*) und ein Weiterleiten zum entsprechenden Ausgang des Knotens stattfinden muss.



**Bild 2.1:** Beispielhafte Struktur eines IP-basierten Netzes

- **Verbindungsabschnitte (links):** Die Abschnitte, die Vermittlungsknoten und Endgeräte untereinander sowie miteinander verbinden, können in Abhängigkeit von der darunter liegenden Technologie auf unterschiedliche Weise realisiert sein. So findet die Kommunikation in lokalen Netzen meist über Stern- oder Busanordnungen statt, die teilweise als gemeinsames Medium (*shared medium*) genutzt werden. Andererseits bestehen im Weitverkehrsnetz Links zwischen IP-Routern häufig aus mehreren Abschnitten mit dazwischen liegenden ATM- (*asynchronous transfer mode*), SDH- (*synchronous digital hierarchy*) oder optischen Vermittlungsknoten. Für Heimanwender wiederum ist der Anschluss an das IP-Netz über ein Zugangsnetz von Bedeutung, das beispielsweise auf einer xDSL-Technologie (*digital subscriber line*) basiert. Einer der Vorteile von IP liegt darin, dass es von der darunter liegenden Technologie abstrahiert und damit eine einfache Möglichkeit bietet, Endgeräte und Teilnetze an ein bestehendes IP-Netz anzubinden.

Neben der Unterteilung in unterschiedliche Komponenten existiert in IP-Netzen auch eine strukturelle Gliederung in verschiedener Hinsicht. Eine Strukturierung ist zunächst durch das mit IP definierte Adressierungsschema gegeben. Eine IP-Adresse, die in der momentan verwendeten IP-Version 4 (IPv4) 32 bit umfasst (siehe Abschnitt 2.3.1), ist unterteilt in eine Netzadresse sowie einen Adressteil mit lokaler Gültigkeit, der sich i. d. R. wiederum aus einer Subnetzadresse und der eigentlichen Endgeräteadresse zusammensetzt.

Da aus Nutzersicht IP-Adressen, die in der Regel mit Hilfe von Dezimalzahlen für die einzelnen Oktetts notiert werden (z. B. 129.69.170.156), wenig handhabbar sind, wurde im Internet das DNS-Konzept (*domain name system*) eingeführt, das eine hierarchische Unterteilung in

Namensbereiche ermöglicht (z. B. „netsrv3.ind.uni-stuttgart.de“). Die Umsetzung eines Domainnamens in die IP-Adresse des entsprechenden Endgerätes (*DNS lookup*) erfolgt mit Hilfe spezieller, ebenfalls hierarchisch organisierter DNS-Server.

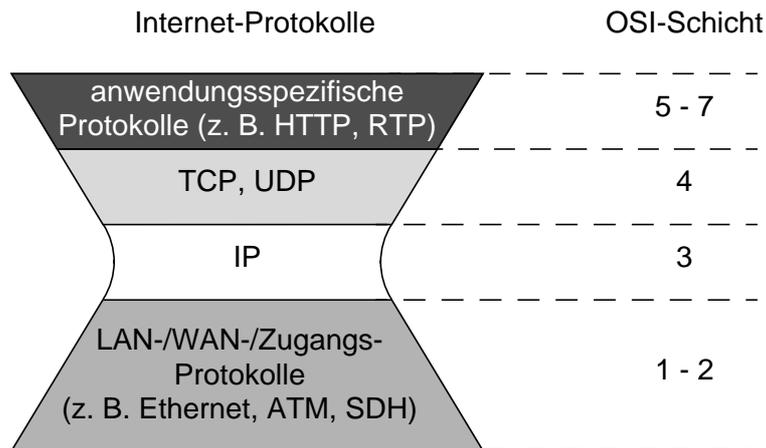
Neben der durch IP-Adressierung und DNS gegebenen Struktur ergibt sich eine Gliederung eines IP-Netzes häufig implizit aus den in den unterschiedlichen Bereichen verwendeten Netzwerk-Technologien oder der Zugehörigkeit zu unterschiedlichen Organisationen. So können lokale Netze als eigene Teilnetze angesehen werden, ebenso wie das Kernnetz (*core network, backbone*), in dem z. B. die Hochgeschwindigkeits-Router eines Internet-Diensteanbieters (*Internet service provider, ISP*) miteinander verbunden sind, eine eigene Struktureinheit bildet (Bild 2.1). Meist spiegelt sich diese Gliederung auch in den verwendeten IP-Adressen wieder, wobei zunehmend eingesetzte Technologien wie NAT (*network address translation*) und *IP Tunnelling* bewirken, dass die Zusammenhänge hier undeutlicher werden.

Schließlich ist für die Verkehrslenkung die Unterteilung in Bereiche (*routing domains*), in denen ein Informationsaustausch zwischen den Routern mit dem gleichen Verkehrslenkungsprotokoll stattfindet, von großer Bedeutung. Dabei findet eine Gliederung in so genannte autonome Systeme (*autonomous systems, AS*) statt, die häufig einer Organisation (z. B. einem Netzbetreiber/ISP) zuzuordnen sind. Während innerhalb eines AS alle Router Verkehrslenkungsinformationen austauschen, erfolgt eine Kommunikation zum Zwecke der Verkehrslenkung über AS-Grenzen hinweg nur zwischen ausgezeichneten Routern innerhalb der verschiedenen AS-Bereiche (siehe auch Abschnitt 3.2.5).

## 2.3 Basisprotokolle

Kennzeichen der hier betrachteten Netze ist die einheitliche Verwendung von IP als Protokoll der Vermittlungsschicht. Sowohl in den darüber liegenden als auch in den darunter befindlichen Schichten ist eine große Vielfalt verwendeter Protokolle vorzufinden (Bild 2.2). Oberhalb von IP werden auf der Transportschicht vor allem das verbindungsorientierte TCP sowie das verbindungslose UDP (*user datagram protocol*) eingesetzt. Diese können zusammen mit IP als Basisprotokolle in IP-Netzen angesehen werden. In den nachfolgenden Abschnitten werden die wichtigsten Eigenschaften dieser Basisprotokolle vorgestellt.

Die oberhalb der Transportschicht verwendeten Protokolle, die nach dem OSI-Referenzmodell (*open systems interconnection*) [133] den Schichten 5 -7 zuzuordnen sind, spiegeln die Erfordernisse der jeweiligen Anwendung wieder. Als Beispiele seien das zur Kommunikation zwischen WWW-Client und -Server eingesetzte, auf TCP aufsetzende *Hypertext Transfer Protocol* (HTTP) sowie das im Zusammenhang mit Echtzeitkommunikation angewandte *Real-Time Transport Protocol* (RTP), das auf UDP basiert, erwähnt (siehe Abschnitt 2.4).



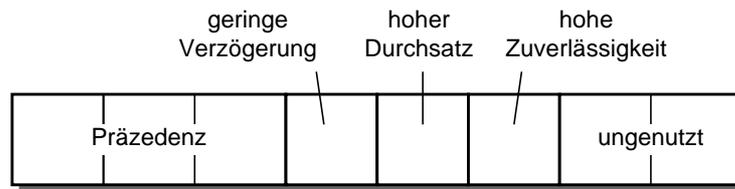
**Bild 2.2:** Protokollschichten in IP-Netzen

Die unterhalb von IP vorzufindenden Protokolle sind abhängig von der jeweiligen Technologie. In lokalen Netzen (*local area networks*, LANs) dominieren die verschiedenen Versionen von Ethernet, während in Weitverkehrsnetzen (*wide area networks*, WANs) IP häufig auf ATM oder SDH aufsetzt. Aus Sicht von Heimanwendern spielen außerdem Protokolle wie PPP (*point to point protocol*), die eine Punkt-zu-Punkt-Verbindung zu einem Einwahlknoten bereitstellen, eine große Rolle.

### 2.3.1 Internet Protocol

Das auf OSI-Schicht 3 anzusiedelnde IP definiert im Wesentlichen das Format von Paketen und ein Schema zur Adressierung von Netzen und Endgeräten [226]. Auf Funktionalität verzichtet IP fast vollständig; lediglich die Möglichkeit zur Fragmentierung ist gegeben, die aber aufgrund ihrer negativen Auswirkungen auf die Effizienz kaum genutzt wird. Insbesondere findet keine Flusssteuerung und Reihenfolgesicherung statt, d. h. IP arbeitet nach dem *Best-Effort*-Prinzip, bei dem nur versucht wird, Pakete korrekt auszuliefern, ohne dies jedoch zu garantieren. Ein weiteres Merkmal von IP ist die verbindungslose Kommunikation. Bei einem Protokoll der Vermittlungsschicht bedeutet dies, dass in jedem Knoten auf dem Weg anhand der Zielinformation der Link ermittelt werden muss, auf dem das Paket weitergesendet wird.

Das Format von IP-Paketen erlaubt Nutzdaten mit variabler Länge, denen ein Paketkopf mit Steuerinformation vorangestellt ist. Insgesamt darf die Länge eines Pakets  $2^{16} - 1$  Byte nicht überschreiten. In der Praxis findet jedoch eine Begrenzung entsprechend der maximalen Rahmengröße der darunter liegenden Protokolle (*maximum transfer unit*, MTU) statt. Häufig ist die Länge von IP-Paketen daher auf die Ethernet-MTU von 1500 Byte begrenzt. Der IP-Paketkopf umfasst in der IP-Version 4 (IPv4) i. d. R. 20 Byte und enthält u. a. folgende Felder:



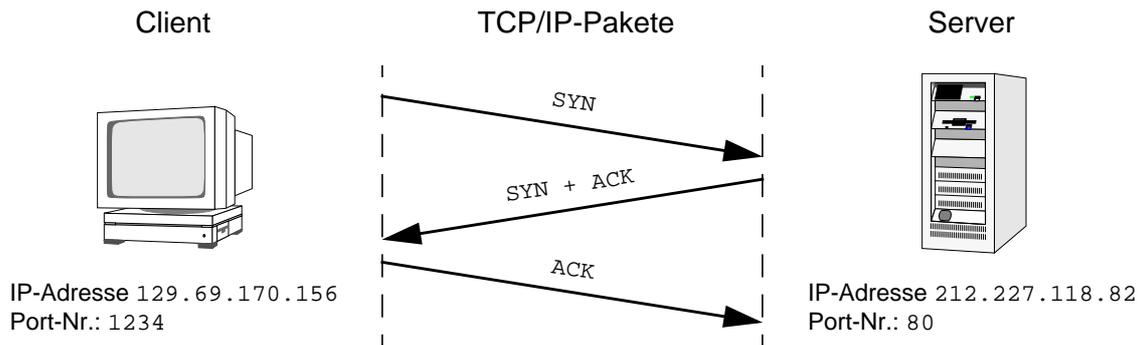
**Bild 2.3:** ToS-Feld im IP-Paketkopf in seiner ursprünglichen Interpretation

- **Sender-/Empfängeradresse:** Die IP-Adressen von Sender- und Empfänger-Host belegen jeweils 32 bit. Die bereits erwähnte Aufteilung in Netzadresse und lokale Adresse hängt von der Zuordnung zu einer Adressklasse ab. Darüber hinaus gibt es IP-Adressen, die speziell für Punkt-zu-Mehrpunkt-Kommunikation vorgesehen sind.
- **Transportprotokoll:** Dieses Feld enthält eine eindeutige Identifikationsnummer des in den Nutzdaten verwendeten Transportprotokolls. Spezielle Nummern existieren z. B. für TCP, UDP und die Verkehrslenkungsprotokolle.
- **Diensttyp** (*type of service*, ToS): Das acht bit breite ToS-Feld ist in seiner ursprünglichen Bedeutung unterteilt in einen drei bit breiten Präzedenzteil, mit dem acht verschiedene Prioritätsstufen angegeben werden können, und weitere Bits, die eine Anzeige erhöhter Anforderungen bzgl. Verzögerung, Durchsatz und Zuverlässigkeit ermöglichen (Bild 2.3). Während das ToS-Feld in dieser Interpretation wenig genutzt wurde, hat es im Rahmen der in Abschnitt 3.3.3 beschriebenen Dienstgütearchitektur DiffServ (*differentiated services*) eine wichtige Rolle eingenommen.
- **Prüfsumme:** Mit Hilfe einer 16-bit-Prüfsumme können Bitfehler innerhalb des Paketkopfs erkannt werden. Eine Fehlererkennung bzgl. der Nutzdaten muss ggf. von höheren Protokollschichten übernommen werden.

In der für zukünftige Netze definierten Version 6 von IP (IPv6) [83] wurden einige der Felder aus IPv4 in ihrer Größe oder Bedeutung geändert. Insbesondere stehen für Sender- und Empfängeradresse je 128 bit zur Verfügung. Das ToS-Feld wurde in IPv6 lediglich umbenannt (*traffic class*), sodass sich in Bezug auf die Verwendung zur Dienstgüteunterstützung keine Änderung ergibt.

### 2.3.2 Transmission Control Protocol

Bei TCP handelt es sich um ein auf OSI-Schicht 4 anzusiedelndes Transportprotokoll, das eine sichere Ende-zu-Ende-Kommunikation über ein IP-Netz ermöglicht [227]. Die TCP-Protokollschicht ist also nur in Endgeräten implementiert – meist als ein von IP nicht zu trennender Bestandteil des Betriebssystems. Für Anwendungsprogramme erfolgt der Zugang zu TCP über vordefinierte Schnittstellenfunktionen (*socket interface*).



**Bild 2.4:** Beispiel eines TCP-Verbindungsaufbaus

Während das Internet-Protokoll nur einen verbindungslosen Datagrammdienst anbietet, arbeitet TCP verbindungsorientiert. Das bedeutet, dass ein Verbindungsauf- und -abbau mit Hilfe von Steuerpaketen – gekennzeichnet durch Kombinationen der Steuerbits SYN, FIN und ACK im Kopf der TCP-PDU (*protocol data unit*) – erfolgt. Die Kommunikationsendpunkte sind dabei neben den Feldern für IP-Sender- und Empfängeradresse sowie Transportprotokoll durch 16-bit-Portnummern auf Sender- und Empfängerseite festgelegt. Über die Portnummern wird gleichzeitig der Zugang zu einem bestimmten Dienst, der auf TCP aufsetzt, ermöglicht. Dazu werden hauptsächlich vordefinierte Portnummern (*well-known ports*) für die verschiedenen Dienste verwendet. Es können aber auch freie Portnummern für proprietäre Dienste eingesetzt werden.

Der Verbindungsaufbau erfolgt in drei Schritten (*three-way handshake*), was am Beispiel einer Client-Server-Beziehung wie im Fall von HTTP gezeigt werden soll (Bild 2.4). Zunächst gibt der Client in einem Verbindungsaufbaupaket (mit SYN = 1 und ACK = 0) die spezielle Portnummer für diesen Dienst (z. B. 80 für HTTP) als Empfängerportnummer im TCP-PDU-Kopf sowie die IP-Adresse des Servers im IP-Paketkopf an. Als Senderportnummer kann prinzipiell eine beliebige beim Client noch freie Portnummer gewählt werden; diese dient lediglich der Identifikation der TCP-Verbindung. Trifft das Paket beim Server ein, wird anhand der im Verbindungsaufbaupaket angegebenen Empfängerportnummer überprüft, ob der angesprochene Dienst unterstützt wird. In diesem Fall wird das Paket an den für diesen Dienst zuständigen Prozess weitergeleitet, der ggf. den Verbindungswunsch akzeptiert und das Senden eines Bestätigungspakets (mit SYN = 1 und ACK = 1) auslöst. Nachdem der Client wiederum eine Gegenbestätigung (mit SYN = 0 und ACK = 1) gesendet hat, gilt die Verbindung als aufgebaut und es kann eine Datenübertragung stattfinden. Der Verbindungsaufbau wird ebenso wie der Abbau durch einen endlichen Automaten auf beiden Seiten der Verbindung und mit Hilfe von Zeitüberwachungsmechanismen gesteuert.

Ist eine Verbindung aufgebaut, kann darüber bidirektional kommuniziert werden, indem an beiden Endpunkten ein Datenstrom (*stream*) in Segmente aufgeteilt wird, die jeweils in einem TCP/IP-Paket gesendet werden. Auf der Empfängerseite werden die Segmente wieder zu

einem Datenstrom zusammengefügt, der über eine entsprechende Schnittstelle von der zugehörigen Anwendung entgegengenommen wird. In beiden Kommunikationsendpunkten sind dabei jeweils zwei Puffer für die zu sendenden Daten sowie für die empfangenen Pakete erforderlich, die – zusammen mit anderen verbindungspezifischen Informationen – mittels separater Datenstrukturen für jede Verbindung vom Betriebssystem verwaltet werden.

Eine der wichtigsten Funktionen von TCP ist die Gewährleistung einer gesicherten Übertragung. Dies geschieht einerseits mit Hilfe einer Prüfsumme im PDU-Kopf, die sich sowohl auf die TCP-Steuerdaten als auch auf die Nutzdaten sowie Teile des IP-Paketkopfes erstreckt. Darüber hinaus findet eine Reihenfolgesicherung statt, die mit Hilfe von Sequenznummern, Bestätigungen (*acknowledgements*) und einer senderseitigen Zeitüberwachung realisiert wird. Bestätigungspakete enthalten jeweils die nächste vom Empfänger erwartete Sequenznummer, womit gleichzeitig eine Empfangsbestätigung für alle kleineren Sequenznummern verbunden ist (kumulative Quittierung). Im Fall einer bidirektionalen Kommunikation können Quittungen auch zusammen mit Nutzdaten in der Gegenrichtung gesendet werden. Die Zeitüberwachung sorgt dafür, dass Segmente spätestens dann wiederholt werden, wenn nach einer gewissen Zeitspanne noch keine Bestätigung eingetroffen ist. Die Zeitspanne wird dabei mit Hilfe von Messungen der Umlaufzeit (*round trip time*, RTT) zwischen dem Senden von Daten und dem Empfang der entsprechenden Quittung laufend angepasst.

Neben der Reihenfolgesicherung gehört die Flusssteuerung zu den Aufgaben von TCP. Dies bedeutet zunächst, dass ein Überlaufen des Empfangspuffers (z. B. verursacht durch eine Anwendung auf der Empfangsseite, die im Puffer anstehende Daten nur mit einer geringen Rate auslesen und verarbeiten kann) verhindert werden muss. Dazu wird ein Fenstermechanismus (*sliding window*) eingesetzt, bei dem der Kommunikationspartner zusammen mit den Bestätigungen die aktuelle Größe des Empfangsfensters (*receiver advertised window*, RWnd) mitteilt. Diese gibt die Differenz von Puffergröße und der Menge von Daten an, die zwar bestätigt, aber noch nicht aus dem Empfangspuffer ausgelesen worden sind. Für die gegenüber liegende Senderseite bedeutet dies, dass das Volumen von gesendeten, aber noch nicht bestätigten Paketen den zuletzt mitgeteilten Wert des RWnd nicht überschreiten darf, d. h. das Sendefenster wird durch die Größe des RWnd begrenzt. Zu beachten ist, dass dies im Fall einer großen RTT auch bei verzögerungsfreiem Auslesen des Empfangspuffers zu einer Begrenzung der Senderate führen kann.

Die Flusssteuerung verhindert jedoch nicht nur das Überlaufen des Empfangspuffers, sondern umfasst auch Mechanismen, die auf eine Vermeidung von Überlast im Netz abzielen. Diese Mechanismen, die auch als Überlaststeuerung (*congestion control*) bezeichnet werden, sorgen insbesondere dafür, dass die Senderate an die momentane Lastsituation im Netz angepasst wird [5, 107, 142]. Realisiert wird dies dadurch, dass das Sendefenster nicht nur durch die Größe des RWnd, sondern auch durch den aktuellen Wert des Überlastfensters (*congestion window*,

CWnd) beschränkt wird. Durch Veränderung der CWnd-Größe als Reaktion auf bestimmte Ereignisse, die auf eine Überlastsituation schließen lassen (z. B. Paketverlust), kann damit auf die Senderate Einfluss genommen werden. Eine genauere Beschreibung der Überlaststeuerungs-Mechanismen sowie weiterer TCP-Eigenschaften, die im Zusammenhang mit dieser Arbeit relevant sind, ist in Abschnitt 4.2.2 zu finden.

### 2.3.3 User Datagram Protocol

Das alternativ zu TCP verwendete UDP [225] kann ebenfalls der OSI-Schicht 4 zugeordnet werden. Zwar wird es ebenso wie TCP als Ende-zu-Ende-Protokoll eingesetzt, jedoch verzichtet UDP auf typische Funktionalität eines Transportprotokolls wie Reihenfolgesicherung und Flusststeuerung. Es findet lediglich eine Fehlersicherung mit Hilfe einer Prüfsumme im PDU-Kopf statt. Daher benötigt UDP auch keine Verbindung, sondern bietet einen reinen Datagrammdienst an. Genau wie bei TCP erfolgt die Anbindung an anwendungsspezifische Protokolle mit Hilfe von Portnummern auf Sender- und Empfängerseite.

UDP wird vor allem dort eingesetzt, wo bewusst auf Paket-Wiederholungen verzichtet werden soll. Dies ist z. B. bei Anwendungen mit Echtzeitanforderungen wie Multimedia-Applikationen gegeben. In diesem Fall übernimmt das in die Anwendungsschicht integrierte RTP die hier notwendigen Aufgaben eines Transportprotokolls (siehe Abschnitt 2.4). Darüber hinaus wird UDP häufig dann eingesetzt, wenn die Anwendung selbst eine einfache, aber ihren Ansprüchen genügende Reihenfolgesicherung durchführt, sodass der Verzicht auf das wesentlich komplexere TCP die insgesamt effizientere Lösung darstellt.

Bezogen auf den Verkehrsanteil in heutigen IP-Netzen spielt UDP gegenüber TCP eine vergleichsweise geringe Rolle [191, 266, 269]. Mit zunehmender Bedeutung von Multimedia-Verkehr ist aber auch ein Wachstum des UDP-Anteils am Verkehrsaufkommen verbunden. Außerdem muss UDP-Verkehr aufgrund seiner – im Gegensatz zu TCP – fehlenden Adaptivität im Fall von Überlastsituationen Beachtung finden, da dies ohne besondere Maßnahmen zu dem bekannten Effekt der Verdrängung des TCP-Verkehrs führt [106].

## 2.4 Anwendungen

IP-Netze sind geprägt durch eine große Vielfalt von Diensten und Anwendungen, die auf der Basis von IP sowie der Transportprotokolle TCP und UDP realisiert werden. Im Hinblick auf eine Klassifikation kann zunächst zwischen Anwendungen unterschieden werden, die hohe Ansprüche an die Integrität der übertragenen Daten stellen, und solchen mit hohen Anforderungen bzgl. der zeitlichen Relationen innerhalb des übertragenen Datenstroms. Im einen Fall handelt es sich um Anwendungen, die elastischen Verkehr (*elastic traffic*) auf der Basis von TCP produzieren, während im anderen Fall vereinfachend von Echtzeitanwendungen bzw. -ver-

**Tabelle 2.1:** Übersicht über wichtige Anwendungen in IP-Netzen

Kategorie	Anwendung/Protokoll	Transportprotokoll
Dateitransfer	FTP	TCP
	TFTP	UDP
	NFS	UDP
WWW	HTTP	TCP
Rechnerfernzugriff	Telnet	TCP
	X Window System Protocol	TCP
	verteilte Anwendungen (CORBA, Java)	TCP
Nachrichtenaustausch	E-Mail (SMTP, POP)	TCP
	News (NNTP)	TCP
	Chat (IRC)	TCP
Multimedia-Kommunikation	interaktive Videokommunikation	UDP/RTP
	interaktive Sprachkommunikation	UDP/RTP
	Multimedia Streaming	UDP/RTP, TCP

kehr (*real-time traffic, stream traffic*) gesprochen werden kann [235]. Darüber hinaus versucht Tabelle 2.1 die wichtigsten Anwendungen bestimmten Anwendungsgruppen zuzuordnen, deren Grenzen allerdings als fließend betrachtet werden müssen.

## 2.4.1 Anwendungen mit gesichertem Datentransfer

Aus Tabelle 2.1 wird deutlich, dass der überwiegende Teil der Anwendungen auf TCP basiert. In den meisten Fällen bedeutet dies, dass die gesicherte Übertragung im Vordergrund steht, während die Einhaltung zeitlicher Randbedingungen eine geringere Rolle spielt. Die Verwendung von TCP bringt, wie bereits in Abschnitt 2.3.2 erläutert, eine Anpassung der Senderate an die Lastsituation im Netz und damit eine Elastizität des erzeugten Verkehrs mit sich.

### 2.4.1.1 Dateitransfer

Zu den Anwendungen, die eine gesicherte Übertragung erfordern, gehören zunächst Dateitransferanwendungen. Das im Internet seit langer Zeit etablierte *File Transfer Protocol* (FTP) stellt Funktionalität bereit, um einen Dateitransfer über beliebige IP-Netze zu realisieren [229]. FTP basiert dabei auf TCP, wobei zwei getrennte TCP-Verbindungen für die Steuerbefehle und die eigentlichen Nutzdaten verwendet werden. FTP wird auch heute noch bevorzugt verwendet, wenn es um die Übertragung großer Dateien geht, wie sie z. B. beim Herunterladen kompletter Softwarepakete auftreten.

Neben FTP werden aber auch noch andere Protokolle eingesetzt, die vorwiegend für den Einsatz in LANs konzipiert sind. Hier ist zum einen TFTP (*trivial file transfer protocol*) zu nennen, das z. B. beim Hochfahren von Arbeitsplatzrechnern für das Laden des Systemprogramms vom Server verwendet wird [256]. Im Gegensatz zu FTP basiert TFTP auf UDP, dessen geringe Funktionalität es um einen einfachen Handshake-Mechanismus zur Reihenfolgesicherung ergänzt. Damit ist TFTP deutlich schlanker als eine TCP-basierte Lösung, sodass es z. B. auch auf einem plattenlosen System betrieben werden kann.

Ein weiteres Beispiel für Dienste und Anwendungen, die zwar eine gesicherte Übertragung benötigen, aber dennoch auf UDP aufsetzen, ist das *Network File System* (NFS)<sup>1</sup>, das ein virtuelles Dateisystem bereitstellt, bei dem die Dateien auf unterschiedlichen Rechnern innerhalb eines (lokalen) Netzes liegen können [251]. Auch hier wird aus Gründen der Einfachheit und Robustheit bei Server-Ausfällen ein Handshake-Mechanismus in der Anwendungsschicht implementiert.

#### 2.4.1.2 World Wide Web

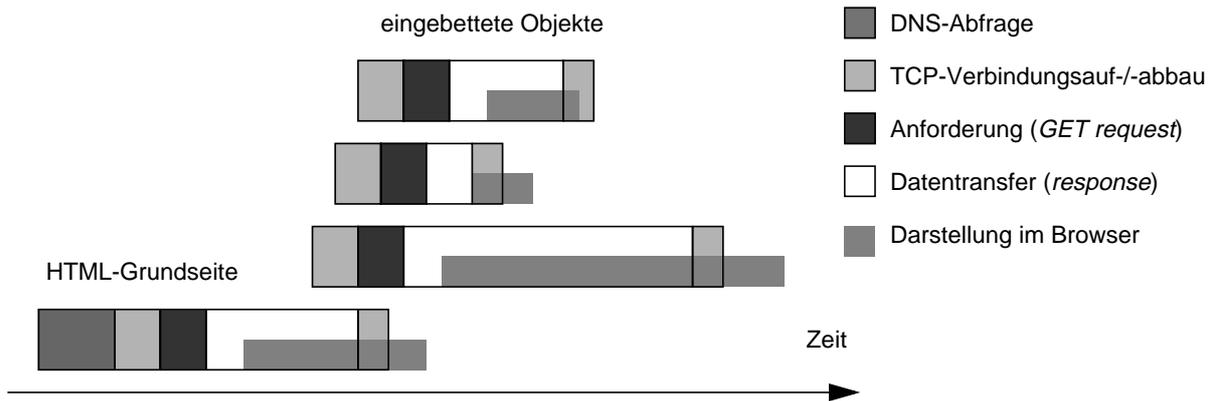
Der überwiegende Anteil des Verkehrs in heutigen IP-Netzen ist dem WWW mit seinen vielfältigen Ausprägungen zuzuordnen [191, 266, 269]. Der WWW-Dienst, der wie die meisten der in diesem Abschnitt behandelten Dienste und Anwendungen auf dem Client-Server-Prinzip beruht, stellt zunächst nur eine erweiterte Form eines Dateiabrufs dar.

Die Kommunikation erfolgt dabei über das auf TCP aufsetzende HTTP [104]. Allerdings ist die von HTTP selbst bereitgestellte Funktionalität eher gering. Im Wesentlichen wird für jedes zu übertragende Objekt (i. d. R. eine Datei) vom Client eine Anforderung (*GET request*) mit Angabe der Adresse des Objekts in Gestalt eines URI (*uniform resource identifier*) an den Server geschickt. Die Antwortnachricht (*response*) enthält Informationen über den Erfolg des Transfers und ggf. das abgerufene Objekt selbst.

Die Funktionalität ist vielmehr in den Anwendungen (WWW-Browser und -Server) enthalten. Die Komplexität der auszuführenden Operationen hängt dabei stark von den abgerufenen Inhalten ab. Im einfachsten Fall handelt es sich um Hypertext-Dateien, die mit Hilfe von HTML (*hypertext markup language*) codiert sind. Ein Abruf umfasst dabei die HTML-Datei selbst sowie alle darin eingebetteten Objekte wie z. B. Bild- oder Audiodateien. Wie diese Abrufe genau erfolgen, hängt stark von der Implementierung vor allem des WWW-Clients ab. Eine Vorgehensweise besteht darin, für jeden HTTP-Transfer eine eigene TCP-Verbindung auf- und nach vollendetem Transfer wieder abzubauen (Bild 2.5). Neuere Versionen von HTTP und Anwendungssoftware hingegen ermöglichen die „Wiederverwendung“ von bereits aufgebauten TCP-Verbindungen für weitere Transfers, sodass im Extremfall der Abruf einer WWW-Seite inklusive aller eingebetteten Objekte über eine TCP-Verbindung erfolgt.

---

<sup>1</sup> Neuere Versionen von NFS unterstützen auch TCP als Transportprotokoll.



**Bild 2.5:** Abruf einer WWW-Seite für den Fall separater TCP-Verbindungen

Eine erhöhte Komplexität auf Server-Seite ergibt sich, wenn HTML-Seiten dynamisch zum Zeitpunkt des Abrufs generiert werden. In diesem Fall ist der bereitgestellte Dienst mehr als ein Dateitransfer, da zusätzlich Operationen (z. B. Datenbankabfragen) auf dem Server-Rechner ausgeführt werden müssen, deren Ergebnis schließlich übertragen wird. Auf die Abläufe beim Client sowie auf den erzeugten Verkehr hat dies allerdings kaum Auswirkungen. Umgekehrt entsteht ein deutlicher Mehraufwand beim Client, wenn das abgerufene Objekt auszuführende Programme oder Programmteile, z. B. in Form von Java-Applets, enthält. Die Kombination dieser Mechanismen auf Server- und Client-Seite spielt besonders bei Anwendungen zur elektronischen Geschäftsabwicklung (E-Commerce, E-Business) eine große Rolle.

### 2.4.1.3 Rechnerfernzugriff

Eine weitere Gruppe von Anwendungen, die bereits in der Anfangszeit des Internet von Bedeutung war, sind solche, die den Zugriff auf entfernte – d. h. über ein IP-Netz erreichbare – Rechnerressourcen ermöglichen. Klassischerweise sind hier Terminalanwendungen wie Telnet [228] zu nennen, die eine Einwahl auf einem über ein IP-Netz zugänglichen Server-Rechner ermöglichen und hierfür eine TCP-Verbindung zu diesem aufbauen. Über diese Verbindung werden dann bei Telnet die über die Tastatur eingegebenen Zeichen an das auf dem Server gestartete Programm übertragen. In der Gegenrichtung wird die Textausgabe des Programms an den Client gesendet. Obwohl durch das Versenden einzelner Zeichen in separaten Paketen zu den Nutzdaten ein Vielfaches an TCP/IP-Steuerdaten hinzukommt, ist das durch Telnet verursachte Verkehrsaufkommen eher gering. Allerdings existieren im Gegensatz zu anderen Anwendungen mit gesicherter Übertragung erhöhte zeitliche Anforderungen.

Ebenfalls in diese Kategorie einzuordnen sind Protokolle wie das unter Unix verwendete *X Window System Protocol*, das die Kommunikation zwischen einem graphischen Anwendungsprogramm und dem Endgerät inklusive der dort angeschlossenen Ein- und Ausgabebezüge regelt [245]. Dabei greift das Anwendungsprogramm, das z. B. per Telnet auf einem Server-Rechner gestartet wurde, als *X Client* über eine TCP-Verbindung auf die Ressourcen des

Endgeräts zu, das in diesem Fall die Rolle des *X Server* übernimmt. Der Einsatz erfolgt hauptsächlich in lokalen Netzen, ist aber z. B. im Zusammenhang mit Telearbeit nicht auf diesen Bereich beschränkt. Während die Anforderungen an die Antwortzeiten mit denen von Telnet vergleichbar sind, ist das pro Zeiteinheit übertragene Datenvolumen im Gegensatz zu Telnet je nach Applikation nicht unerheblich.

Eine zunehmende Bedeutung haben in dieser Kategorie verteilte Anwendungen erlangt, bei denen die Kommunikation zwischen den auf verschiedenen Plattformen lokalisierten Teilen der Applikation durch Prozedur-Fernaufrufe (*remote procedure calls*) über Middleware-Technologien wie CORBA (*common object request broker architecture*) [210] oder entsprechende Java-Mechanismen realisiert wird. Dabei ergibt sich einerseits ein fließender Übergang zu den oben erwähnten WWW-Technologien, die eine Ausführung von Operationen auf Client- bzw. Server-Seite ermöglichen. Andererseits sind hier auch so genannte *Peer-to-Peer*-Anwendungen zu nennen, die z. B. zur Realisierung von Tauschbörsen für Musikdateien eingesetzt werden und dabei für ein nicht zu vernachlässigendes Verkehrsaufkommen sorgen [191], oder Spezialanwendungen wie Netzspiele, die allerdings aufgrund erhöhter Echtzeitanforderungen teilweise auf UDP aufsetzen und die Sicherung der Übertragung ggf. in der Anwendungsschicht realisieren.

#### **2.4.1.4 Nachrichtenaustausch**

Zu den klassischen Internet-Anwendungen gehört auch der Austausch von Nachrichten zwischen einzelnen Nutzern. Insbesondere E-Mail ist nach wie vor eines der am meisten genutzten Kommunikationsmedien. Für die Realisierung wurden im Laufe der Jahre zahlreiche Protokolle entwickelt, von denen SMTP (*simple mail transfer protocol*) die größte Bedeutung hat [156]. Dabei wird eine TCP-Verbindung zwischen Mail-Client (Ursprung) und Mail-Server (Ziel) aufgebaut, über die zunächst die Zieladresse der E-Mail mitgeteilt und überprüft wird. Anschließend wird darüber die eigentliche Nachricht übertragen, die aus wenigen Zeichen bestehen, aber auch durch angehängte Dateien mehrere MByte an Daten umfassen kann. Wie die eingehenden Nachrichten schließlich dem Nutzer angezeigt werden, hängt von der Implementierung der Anwendungsprogramme ab. Häufig werden die Nachrichten vom Mail-Server nur in einer Datei („Mailbox“) zwischengespeichert und müssen von dort z. B. mit Hilfe des *Post Office Protocol* (POP) abgeholt werden [239].

Um Nachrichten innerhalb einer Gruppe von Nutzern zu verbreiten, steht mit News-Diensten eine weitere Möglichkeit zur Verfügung. Im Gegensatz zu Mailing-Listen, über die ebenfalls mehrere Adressaten gleichzeitig erreichbar sind, werden hier Nachrichten nicht automatisch an alle Nutzer zugestellt, sondern können bei Bedarf von einem zentralen, i. d. R. allgemein zugänglichen News-Server selektiv abgerufen werden. Anstelle von SMTP im Fall von E-Mail wird bei News als Kommunikationsprotokoll NNTP (*network news transfer protocol*) eingesetzt, das ebenso wie SMTP auf TCP als Transportprotokoll aufsetzt [149].

Eine wesentliche Eigenschaft von E-Mail und News besteht darin, dass sehr geringe Anforderungen bzgl. Antwortzeiten bestehen. Es ist also durchaus akzeptabel, wenn Mails erst einige Minuten nach dem Abschicken beim Adressaten eintreffen. Allerdings gibt es auch andere Formen des Nachrichtenaustauschs wie z. B. das zur Spontankommunikation eingesetzte IRC (*Internet relay chat*) [208], die engere zeitliche Anforderungen an das System stellen.

## 2.4.2 Echtzeitanwendungen

Anwendungen mit hohen zeitlichen Anforderungen, bei denen aber auf eine gesicherte Übertragung verzichtet werden kann, haben in den letzten Jahren im Kontext von IP-Netzen an Bedeutung gewonnen und stellen gerade auch im Hinblick auf die Einführung von Dienstgüte-Architekturen einen wichtigen Treiber dar. Obwohl auch einige der in Abschnitt 2.4.1 vorgestellten Applikationen gewisse Anforderungen bzgl. der Antwortzeiten haben, werden im Zusammenhang mit IP-Netzen unter Echtzeitanwendungen vorwiegend Multimedia-Anwendungen verstanden. Diese sind dadurch charakterisiert, dass Audio- oder Videosignale in digitalisierter Form in Pakete verpackt und über ein IP-Netz zu einem Endgerät übertragen werden, wo eine Resynchronisation und ein anschließendes Ausspielen (*playout*) der Signale erfolgt.

Grundsätzlich kann bei Echtzeitkommunikation zwischen Anwendungen mit harten und solchen mit weichen Echtzeitanforderungen unterschieden werden [10]. Während Anwendungen mit harten Echtzeitbedingungen nur korrekt arbeiten, wenn alle Pakete eines Stroms jeweils innerhalb einer bestimmten Frist beim Empfänger vorliegen, kommen Anwendungen mit weichen Bedingungen auch bis zu einem gewissen Maß mit verspätet ankommenden oder unterwegs verloren gegangenen Paketen zurecht. Allerdings geht dies i. d. R. zu Lasten der Audio- oder Bildqualität. Durch die Entwicklungen bei Audio- und Videocodierung stellen heute nur noch wenige Anwendungen harte Echtzeitanforderungen. Statt dessen ist eine andere Klassifizierung wichtiger geworden, nämlich die Unterscheidung zwischen interaktiven Multimedia-Anwendungen (*conversational media*) und den so genannten *Streaming-Media*-Anwendungen. Bei letzteren erfolgt eine Übertragung nur in einer Richtung, sodass der Beginn einer Audio- oder Videosequenz beim Empfänger mitunter stark verzögert sein kann, ohne eine Beeinträchtigung für den Nutzer entstehen zu lassen. Lediglich die zeitlichen Abstände der Signale müssen beim Empfänger wiederhergestellt werden. Bei interaktiven Anwendungen wie IP-Telefonie (*voice over IP*, VoIP), die vor dem Hintergrund der Konvergenz von Sprach- und Datennetzen zu einem gemeinsamen Netz auf IP-Basis eine besonders große Rolle spielt, oder Videokonferenzsystemen hingegen würde eine Verzögerung im Sekundenbereich eine Konversation unmöglich machen.

Die große Toleranz gegenüber Verzögerungen im Fall von *Streaming Media* führt dazu, dass für diese Anwendungen TCP als Transportprotokoll genutzt werden kann [159]. Dies hat in erster Linie praktische Vorteile z. B. im Hinblick auf ein leichteres Passieren von Sicherheits-

barrieren (*firewalls*) an Netzgrenzen (besonders bei Verwendung von HTTP oberhalb von TCP). Für interaktive Anwendungen scheidet TCP mit seinen Wiederholungsmechanismen aus, sodass auf der Transportschicht UDP als Alternative bleibt. Um allerdings die Synchronisation auf der Empfängerseite zu ermöglichen, muss oberhalb von UDP ein weiteres Protokoll eingesetzt werden. Als Standard hat sich hierbei RTP etabliert [247], das direkt in das Anwendungsprogramm integriert wird. Bei RTP werden die einzelnen Pakete mit einer Sequenznummer und einem Zeitstempel versehen. Dadurch ist es möglich, die Reihenfolge der Pakete, die beim Empfänger in einem Puffer zwischengespeichert werden, wiederherzustellen und zudem den Ausspielzeitpunkt festzulegen, sodass die im Netz auftretenden Verzögerungsschwankungen ausgeglichen werden. Wenn ein Paket bis zu seinem Ausspielzeitpunkt noch nicht eingetroffen ist, wird allerdings nicht wie im Fall von TCP das Ausliefern der nachfolgenden Daten an die nächste Verarbeitungsstufe verzögert, um ggf. auf eine Wiederholung zu warten. Statt dessen wird auf die Information in dem fehlenden Paket verzichtet. Kommt das Paket zu spät an, muss es i. d. R. verworfen werden (siehe auch Abschnitt 4.3.3).

Neben dem Transportprotokoll sind vor allem bei interaktiven Multimedia-Anwendungen noch weitere Protokolle erforderlich, z. B. zur Steuerung von Medienströmen oder zur Signalisierung und Verbindungssteuerung. Entsprechende Protokolle wurden von der ITU (*International Telecommunication Union*) standardisiert und im Rahmenstandard H.323 zusammengefasst [136]. Alternativ dazu existieren von der IETF vorgeschlagene Lösungen wie z. B. das *Session Initiation Protocol* (SIP) zum Aufbau von Multimedia-Sitzungen [121]. Darüber hinaus spielen Verfahren zur Codierung und Decodierung von Multimediadaten (*codec*) eine bedeutende Rolle. Diese haben vor allem einen großen Einfluss auf die Wahrnehmbarkeit von Paketverlusten und -verwerfungen.

# Kapitel 3

## Dienstgüteunterstützung in IP-Netzen

Der Erfolg des IP-Netzkonzepts in den vergangenen Jahren liegt zu weiten Teilen darin begründet, dass es aufgrund des verteilten Ansatzes relativ einfach ist, eine Anbindung (*connectivity*) von Teilnetzen oder Kommunikationsendpunkten an ein bestehendes Netz zu realisieren. Mit der zunehmenden Bedeutung dieses Netzkonzepts sind aber darüber hinaus neue Anforderungen entstanden. Neben Fragen der Verfügbarkeit (*availability*) und Sicherheit (*security*) ist hier vor allem die Forderung nach differenzierter Dienstgüte (*quality of service*, QoS) zu nennen, die aus der generellen Problematik knapper Ressourcen erwächst. Insbesondere ist das derzeit existierende, nach dem Best-Effort-Prinzip arbeitende Internet nicht in der Lage, unterschiedlichen Anforderungen von Anwendungen und Nutzern gerecht zu werden.

In diesem Kapitel wird ein Überblick über die vielfältigen Vorschläge für IP-basierte Netzarchitekturen gegeben, die eine Dienstgüteunterstützung bieten. Abschnitt 3.1 enthält zunächst einige grundsätzliche Erklärungen zum Begriff der Dienstgüte. Der darauf folgende Abschnitt stellt die Funktionen des Verkehrsmanagements vor, die dazu eingesetzt werden können, um Dienstgüte bereitzustellen. Schließlich werden im Abschnitt 3.3 konkrete Ansätze und Architekturen vorgestellt, die eine Dienstgüteunterstützung in IP-Netzen bieten.

### 3.1 Dienstgüte: Definitionen und Merkmale

#### 3.1.1 Begriffsdefinitionen

Dienstgüte wird in [134] unter Bezugnahme auf [137] definiert als die „Gesamtheit aller Qualitätsmerkmale eines Kommunikationsnetzes aus der Sicht des Benutzers eines betrachteten Dienstes“. Die Betonung der Nutzersicht macht hierbei den Hauptunterschied zwischen dieser Definition und der eines ansonsten eng verwandten Begriffes, der Netzgüte (*network performance*), aus. Während Dienst- und Netzgüte als übergeordnete Begriffe zu sehen sind,

beschreibt die Verkehrsgüte (*grade of service*, GoS) laut [134] den quantifizierbaren Teil von Dienst- und Netzgüte, der von der Bemessung der Ressourcen im Netz abhängt.

Eine Unterscheidung der genannten Begriffe ist in Literaturbeiträgen, die sich mit Dienstgüte in IP-Netzen beschäftigen, selten zu finden. Vor allem in der englischsprachigen Literatur wird nahezu ausschließlich der Begriff *Quality of Service* verwendet, und zwar sowohl als Überbegriff als auch in den Fällen, in denen eine ganz bestimmte Form der Bereitstellung von Dienstgüte zugrunde gelegt wird. Zu diesem Zusammenhang ist außerdem der Begriff CoS (*classes of service*) zu nennen, der auch recht häufig in der Literatur zu finden ist. Von CoS wird gesprochen, wenn eine Differenzierung verschiedener Dienstklassen als Mittel, um Dienstgüte bereitzustellen, im Vordergrund steht [132].

Obwohl häufig je nach Schwerpunkt der Aussage auch die Begriffe Netzgüte oder CoS zutreffend wären, wird im Folgenden aufgrund seiner umfassenden Bedeutung und seiner Dominanz in der Literatur ausschließlich der Begriff Dienstgüte oder QoS verwendet und auf eine weitere Begriffsdifferenzierung verzichtet.

### **3.1.2 Dienstgüteparameter**

Trotz des allgemeinen Charakters der in Abschnitt 3.1.1 erwähnten Definition der Dienstgüte werden Dienstgüteanforderungen meist mit Bezug auf konkrete Leistungsmaße formuliert. In [140] sind dazu vor dem Hintergrund klassischer Telekommunikationsnetze eine ganze Reihe von Parametern genannt, die zur Beschreibung der Dienstgüte herangezogen werden können.

Im einfachsten Fall werden Maße verwendet, die der Paketebene zuzuordnen sind, wie z. B. Paketverlustwahrscheinlichkeit sowie Mittelwert, Verteilung oder Quantil der Verzögerung. Darüber hinaus sind auch Blockierungswahrscheinlichkeiten bei Reservierungswünschen oder der in höheren Protokollschichten wahrnehmbare Durchsatz wichtige Parameter in Bezug auf die Dienstgüte. Allerdings spiegeln die genannten Leistungsmaße die vom Benutzer empfundene Qualität, die letztendlich für eine Bewertung entscheidend ist, oft nur unzureichend wieder (siehe auch Abschnitt 4.3).

Andererseits ist es auch wünschenswert, Dienstgüteanforderungen so zu formulieren, dass sie möglichst einfach und direkt von den Verkehrssteuerungsmechanismen umgesetzt werden können, z. B. bei der Reservierung von Ressourcen. Aus diesem Grund wird, gerade in IP-Netzen, häufig die benötigte Bandbreite als Dienstgüteparameter verwendet, obwohl der Zusammenhang zwischen reservierter Bandbreite und der sich ergebenden Leistung aus Nutzer- oder Anwendungssicht nicht immer unmittelbar klar ist.

### 3.1.3 Differenzierung oder Überdimensionierung

Ausgangspunkt der Dienstgüteproblematik sind unterschiedliche Anforderungen in Bezug auf die Dienstgüteparameter. Diese lassen sich zum einen begründen durch die Verschiedenheit der Anwendungen und Dienste, wie sie für den Fall von IP-Netzen in Abschnitt 2.4 vorgestellt wurden. Insbesondere ist hier die Unterteilung bzgl. der Echtzeitanforderungen entscheidend. Darüber hinaus haben aber auch die Nutzer selbst unterschiedliche Profile, z. B. im Hinblick auf die zur Verfügung stehende Zeit, Budget oder Arbeitsweise. Daraus ergeben sich weitere Unterschiede in den Anforderungen, die im Vergleich zu denen bei den Anwendungen noch wesentlich schwerer zu beschreiben, vorherzusehen und zu behandeln sind.

Um den angeführten unterschiedlichen Anforderungen von Anwendungen und Nutzern gerecht zu werden, stehen zunächst zwei prinzipielle Möglichkeiten zur Auswahl:

- die Bereitstellung einer differenzierten Dienstgüte, also eine unterschiedliche Behandlung von Verkehrsströmen
- eine Überdimensionierung (*overprovisioning*) von Ressourcen, d. h. eine Dimensionierung, die sich an den höchsten auftretenden Ansprüchen orientiert.

Die zweite der genannten Alternativen hat durchaus Vorteile, z. B. den Verzicht auf teilweise sehr komplexe Mechanismen zum Verkehrsmanagement (siehe Abschnitt 3.2) und die Vermeidung der mit der Einführung solcher Mechanismen verbundenen Interoperabilitäts- und Migrationsprobleme. Allerdings setzt dies voraus, dass Ressourcen, insbesondere Bandbreite, im Überfluss vorhanden sind, oder, genauer ausgedrückt, dass Bandbreitkosten deutlich geringer sind als Kosten, die ein komplexeres Verkehrs- und Netzmanagement inklusive einer daran angepassten Entgelterhebung mit sich bringen würde. Das rasante Wachstum im Bereich neuer Übertragungstechnologien wie der optischen WDM-Technologie (*wavelength division multiplexing*) im Kernbereich oder den xDSL-Technologien im Zugangsbereich scheint eine Erfüllung dieser Voraussetzung tatsächlich in nahe Zukunft zu rücken. Betrachtet man jedoch die Entwicklung der letzten Jahrzehnte in der Kommunikationstechnik, aber auch in der Rechner-technik, die jeweils von einer exponentiellen Zunahme der Kapazitäten getragen war [93], stellt man fest, dass der Bedarf und die Anforderungen etwa im gleichen Maße gestiegen sind.<sup>1</sup> Darüber hinaus gibt es physikalische Randbedingungen, die ein exponentielles Ansteigen der zur Verfügung stehenden Ressourcen in der Zukunft behindern könnten, z. B. die Begrenzung der Signalausbreitung durch die Lichtgeschwindigkeit oder die Endlichkeit der Ressourcen im Zugangsbereich, insbesondere auf der Luftschnittstelle sowie auf existierenden Zugangsleitungen. Schließlich laufen Kapazitätserweiterungen in verschiedenen Netzbereichen weder kontinuierlich noch synchron ab, sodass man folgern kann, dass Ressourcenengpässe nicht dauerhaft verhindert, sondern nur von einem Netzbereich zum anderen verlagert werden.

---

<sup>1</sup> In Bezug auf die Kommunikationstechnik gilt dies zumindest für Teilbereiche, z. B. für Zugangsnetze.

Wenn man also von einer Knappheit der Ressourcen ausgeht, gilt es, diese Ressourcen möglichst effizient zu nutzen, sodass die Anwender von Netzdiensten ein möglichst hohes Maß an Zufriedenheit erfahren. Dies macht dann den Einsatz von Mechanismen zur differenzierten Dienstgüteunterstützung unentbehrlich. Daher wird im Rahmen dieser Arbeit angenommen, dass mit der Bereitstellung von Dienstgüte stets eine Differenzierung einhergeht.

### **3.1.4 Dienstgütegarantien**

Die Bereitstellung von Dienstgüte erfolgt grundsätzlich, indem mehr oder weniger umfangreiche Garantien im Hinblick auf bestimmte Dienstgüteparameter gegeben werden. Hierbei können zwei elementare Formen unterschieden werden: absolute und relative Garantien [119].

#### **3.1.4.1 Absolute Garantien**

Erfolgt die Bereitstellung von Dienstgüte in Form von absoluten Garantien, so werden für einen Dienst Leistungskriterien (z. B. bzgl. Verzögerung oder Durchsatz) definiert, die das Netz erfüllen muss. Dabei kann eine weitere Unterteilung erfolgen, je nach dem, ob diese Garantien deterministischer oder statistischer Natur sind. Ein Beispiel für eine deterministische Garantie ist die Festlegung, dass kein Paket im Netz eine Verzögerung erfahren darf, die einen vorgegebenen Grenzwert überschreitet. Eine statistische Garantie hingegen ist z. B. schon gegeben, wenn gewährleistet ist, dass eine Paketübertragung mit gewisser Wahrscheinlichkeit dieser Bedingung genügt. Eine noch schwächere Form absoluter Garantien liegt vor, wenn die Zielwahrscheinlichkeit der Überschreitung der absoluten Leistungswerte nicht durch den Nutzer vorgegeben werden kann, sondern lediglich durch geeignete Bereitstellung (*provisioning*) von Ressourcen versucht wird, einen für den Nutzer zufriedenstellenden Wert zu erreichen.

Mit der Gewährung absoluter Garantien geht i. d. R. eine Beschränkung des vom Nutzer erzeugten Verkehrs (z. B. bzgl. der mittleren Rate und Spitzenrate) einher. Eine entsprechende Verkehrsbeschreibung sowie die Angaben über die Zielwerte der Dienstgüteparameter bilden die wesentlichen Bestandteile eines Verkehrsvertrags, wie er bei verschiedenen Dienstgütearchitekturen (siehe Abschnitt 3.3) zum Tragen kommt. Signalisierung ist erforderlich, um die Vertragsbestandteile auszuhandeln und die entsprechenden Mechanismen im Netz (z. B. zur Reservierung von Ressourcen) anzustoßen, die eine Einhaltung der Garantien ermöglichen.

Der große Vorteil absoluter Garantien liegt in der Vorhersehbarkeit der Dienstgüte, sowohl für den Benutzer selbst als auch für die Anwendung, die die Schnittstelle zwischen Nutzer und Netz darstellt. Andererseits besteht für Nutzer und Anwendung die Schwierigkeit, im Voraus bereits Angaben über das Profil des erzeugten Verkehrs zu machen. Ein weiterer Nachteil absoluter Garantien liegt in der Notwendigkeit komplexer Mechanismen zur Überwachung des Verkehrsvertrages und zur Verbindungsannahmesteuerung. Dazu kommt, dass insbesondere

die Verbindungsannahmesteuerung Dienstgüte nicht „kostenlos“ liefert. Der „Preis“, der für die Garantie eines bestimmten Leistungsmaßes bezahlt werden muss, liegt darin, dass Verbindungen mit gewisser Wahrscheinlichkeit abgewiesen werden, sodass unter Umständen überhaupt keine Kommunikation zustande kommt.

### 3.1.4.2 Relative Garantien

Eine wesentlich einfachere Form der Bereitstellung von Dienstgüte beruht auf einer reinen Priorisierung von Verkehrsströmen. Man spricht auch von relativen Garantien, relativer Dienstgüte oder relativer Differenzierung (*relative differentiation*) [88]. Voraussetzung ist die Zuordnung eines Verkehrsstromes zu einer bestimmten Dienstgüteklasse. An die Stelle der absoluten Garantie tritt dann die Zusage, dass die Dienstgüte mit steigender Priorität „besser“ wird.

Wie bei absoluten Garantien ist auch beim relativen Ansatz eine Abstufung hinsichtlich der Tragweite einer solchen Zusage möglich. Gilt die o. g. Aussage für jedes Paket, kann man von deterministischer relativer Garantie sprechen. Dies trifft z. B. bei einem System mit statischen Prioritäten (siehe Abschnitt 3.2.1.1) zu, wenn als Beurteilungskriterium die Verzögerung herangezogen wird. Oft wird es jedoch nur möglich sein, zu garantieren, dass eine Bevorzugung mit gewisser Wahrscheinlichkeit stattfindet (relative Garantie im statistischen Sinn). Denkbar sind sogar Architekturen, bei denen die Gültigkeit einer Zusage allein vom Verhalten der Masse von Benutzern abhängt, auf das lediglich durch eine entsprechende Preisgestaltung Einfluss genommen wird [207].

Eine weitere Unterteilung bezieht sich auf die Grundlage eines Vergleichs. Nimmt man als Vergleichskriterium wieder die Verzögerung in einem Einknotensystem, so ergeben sich zwei mögliche Verfeinerungen der o. g. Aussage:

- Die Verzögerung eines Pakets einer höheren Klasse ist (in deterministischem oder statistischem Sinn) nicht größer als die eines beliebigen anderen Paketes in einer Dienstgüteklasse mit niedrigerer Priorität.
- Die Verzögerung eines Pakets wäre (in deterministischem oder statistischem Sinn) nicht größer, wenn das gleiche Paket einer höher priorisierten Dienstgüteklasse zugewiesen würde.

Betrachtet man z. B. ein Prioritätensystem ohne Unterbrechung der Bedienung, trifft die zweite Aussage in jedem Fall zu, während die erste nur in statistischem Sinn gilt, da eine Abhängigkeit von der aktuellen Belegung von Warteschlange und Bedieneinheit besteht.

Architekturen, die relative Dienstgüte bereitstellen, können ohne Quellflusskontrolle und Verbindungsannahmesteuerung auskommen, bieten dann allerdings bereits aktiven Verkehrsflüssen auch keinen Schutz vor plötzlichen Änderungen in der Lastsituation und einem damit verbundenen Absinken der absoluten Dienstgüte.

Während relative Dienstgüte im einfachsten Fall lediglich qualitativer Natur ist und somit als Erweiterung des existierenden Best-Effort-Prinzips angesehen werden kann (*better than best effort*), geht das von Dovrolis et al. in [88] vorgestellte Modell für eine proportionale Differenzierung (*proportional differentiation*) einen Schritt weiter. Es fordert, dass ein bestimmtes für die Dienstgüte relevantes Leistungsmaß  $q_i$  in verschiedenen Dienstklassen proportional zu den jeweiligen Differenzierungsparametern  $s_i$  ist:

$$\frac{\bar{q}_i(t, t + \Delta t)}{\bar{q}_j(t, t + \Delta t)} = \frac{s_i}{s_j} \quad (3.1)$$

Durch die Verwendung der Notation  $\bar{q}_i(t, t + \Delta t)$ , die den Mittelwert des Leistungsmaßes im Intervall  $(t, t + \Delta t)$  repräsentiert, soll dabei ausgedrückt werden, dass diese Beziehung auch in kurzen Zeitausschnitten gelten soll. Als mögliche Leistungsmaße werden in [88] die mittlere Verzögerung sowie die Verlustrate vorgeschlagen, prinzipiell sind jedoch auch andere Maße denkbar, wie auch in Kapitel 5 bei der Vorstellung eines Scheduling- und Puffermanagement-Verfahrens gezeigt wird, das auf dem Modell der proportionalen Differenzierung beruht.

### 3.1.5 Aggregation

Einen wesentlichen Unterschied macht es insbesondere im Fall von absoluten Garantien, ob sich Dienstgüte auf einzelne Verkehrsflüsse (*flows*) oder auf aggregierte Verkehrsströme bezieht. Bei der Anwendung von relativer Dienstgüte hingegen ist die Aggregation in Form einer Bildung von Dienstgüteklassen bereits inhärenter Bestandteil des Konzepts.

Unter einem Verkehrsfluss ist dabei ein Strom von Paketen zu verstehen, der von einer einzelnen Benutzeraktivität herrührt [47]. Die Zuordnung eines IP-Pakets zu einem Verkehrsfluss erfolgt in der Realität anhand eines Fünftupels bestehend aus den Paketkopffeldern Quell-IP-Adresse, Ziel-IP-Adresse, Protokoll-Kennzeichnung (z. B. TCP, UDP), Quell-Portnummer und Ziel-Portnummer [103, 272] (vgl. Abschnitt 2.3).

Eine Zusammenfassung von IP-Verkehrsflüssen zu einem aggregierten Verkehrsstrom kann nach verschiedenen Kriterien erfolgen:

- **IP-Adresse:** Eine Zusammenfassung von Paketen, bei denen führende Bits (*prefix*) von entweder Quelladresse, Zieladresse oder Quell- und Zieladresse übereinstimmen, wird auch als topologische Aggregation bezeichnet.
- **Transportprotokoll:** Eine grobe Unterteilung in elastischen und echtzeitkritischen Verkehr kann einfach über das Transportprotokollfeld im IP-Paketkopf erfolgen (TCP bzw. UDP). Die Zusammenfassung elastischer bzw. echtzeitkritischer Verkehrsflüsse ist daher eine nahe liegende und auch Gewinn bringende Form der Aggregation [86].

- **Anwendung:** Eine Aggregation von Verkehrsflüssen, die zur gleichen Anwendung (z. B. WWW, FTP oder telnet) gehören, stellt eine gegenüber der Aggregation gemäß Transportprotokoll verfeinerte Variante dar. Eine Identifikation von Anwendungen ist dabei in vielen Fällen über bekannte Portnummern im Kopf von PDUs des Transportprotokolls möglich. Diese Aggregationsform wird z. B. im Zusammenhang mit *Class-Based Queueing* (CBQ) vorgeschlagen [110] (siehe Abschnitt 3.2.1.1).
- **Dienstgüteklasse:** In diesem Fall wird die Zugehörigkeit zu einer Klasse direkt vom Benutzer eingestellt oder am Netzrand aus der Verkehrscharakteristik abgeleitet (z. B. mit Hilfe eines *Leaky Bucket*, siehe Abschnitt 3.2.2) und in einem speziell dafür verwendeten Feld des IP-Protokollkopfes eingetragen. Diese Form der Aggregation, die allein zur differenzierten Behandlung im Hinblick auf die Dienstgüte durchgeführt wird, ist die in DiffServ-Netzen vorherrschende (siehe Abschnitt 3.3.3). Im Gegensatz zu den o. g. Aggregationskriterien ist es hierbei prinzipiell möglich, dass einzelne Pakete eines Verkehrsflusses unterschiedlichen Aggregaten zugeordnet werden.

Aggregation bringt in verschiedener Hinsicht eine Vereinfachung mit sich. Wenn für einen Verkehrsstrom Reservierungen vorgenommen werden, ist dazu jeweils ein gewisser Aufwand für Signalisierung und Speicherung der Zustandsinformationen in den Netzknoten erforderlich. Verzichtet man auf Aggregation, fällt dieser Aufwand für jeden Verkehrsfluss an, was im Kernnetz zu Skalierungsproblemen führt [86]. Eine weitere Vereinfachung ergibt sich, wenn – je nach Ausprägung der Architektur – auf Angaben des Endnutzers zu seinem Verkehrsprofil ganz oder teilweise verzichtet wird und statt dessen die zur Bereitstellung von Dienstgüte erforderlichen Kenntnisse über die Verkehrscharakteristik des aggregierten Stroms aus Messungen abgeleitet werden.

Aggregation bedeutet – ungeachtet des für die Zusammenfassung gewählten Kriteriums –, dass alle Pakete, die zu einem aggregierten Verkehrsstrom gehören, prinzipiell gleich behandelt werden bis zu dem Punkt, an dem wieder eine Deaggregation stattfindet [55]. Gleichbehandlung heißt im Hinblick auf die Dienstgüte vor allem auch, dass zugrunde liegende Leistungsmaße auf den aggregierten Strom bezogen werden. Dies kann sich für einzelne Verkehrsflüsse nachteilig auswirken, wie die beiden folgenden Beispiele verdeutlichen:

- Wird garantiert, dass die Verlustwahrscheinlichkeit eine bestimmte Größe nicht überschreitet, so kann es trotzdem vorkommen, dass einzelne Verkehrsflüsse z. B. aufgrund einer ungünstigeren Verkehrscharakteristik eine höhere Verlustwahrscheinlichkeit erfahren [160].
- Liegen Bandbreitereservierungen für die einzelnen Verkehrsflüsse vor, die im Zuge der Aggregation zu einer Reservierung der Summenbandbreite zusammengefasst werden, tritt u. U. eine starke Benachteiligung von Verkehrsflüssen, deren Sender im Überlastfall die Senderate reduziert, gegenüber nicht adaptiven Flüssen im gleichen aggregierten Strom auf.

Insgesamt bedeutet dies, dass bei Aggregation Gesamtanforderungen nach wie vor eingehalten werden, eine Isolation einzelner Flüsse aber nicht mehr ohne weiteres gegeben ist.

### 3.1.6 Wirkungsbereiche von Dienstgütemechanismen

Die Definition der Dienstgüte mit Bezug auf die vom Nutzer wahrgenommene Qualität erfordert das Zusammenwirken aller an einer Kommunikationsbeziehung beteiligten Komponenten auf dem Weg zwischen zwei Kommunikationsendpunkten. Dies umfasst die verschiedenen Bereiche des Netzes (Zugangs-/Kernnetz) ebenso wie die an der Kommunikation beteiligten Endgeräte. Andererseits stellt die Bereitstellung einer Ende-zu-Ende-Dienstgüte eine sehr komplexe Aufgabe dar, was eine dekompositorische Betrachtung der verschiedenen Bereiche nahe legt. Diese Dekomposition des Problems findet meist in den Netzbereichen ihre Fortsetzung, indem einzelne Verbindungsabschnitte oder verschiedene Netzebenen (z. B. optische Ebene, elektrische Transportebene, Paketebene) separat betrachtet werden.

Während die Einflussmöglichkeiten im Bereich des Kommunikationsnetzes durch die in Abschnitt 3.2 behandelten Verkehrsmanagement-Funktionen beschrieben werden, existieren auch innerhalb des Endgeräts Möglichkeiten, um zur Erfüllung von differenzierten Dienstgüterequirements beizutragen. Exemplarisch können hier die Unterscheidung von Prioritätsklassen beim Zugriff auf Rechnerressourcen in WWW-Servern [6] oder Mechanismen wie unterschiedliche Priorisierung von Bildern in einem Videostrom (*layered coding*) im Fall von Multimedia-Anwendungen [20] genannt werden.

## 3.2 Verkehrsmanagement-Funktionen

Als Verkehrsmanagement (*traffic management*) kann die Summe aller Funktionen innerhalb eines Netzes bezeichnet werden, die zur Bereitstellung von Dienstgüte beitragen. Die in IP-Netzen eingesetzten Mechanismen weisen teilweise große Ähnlichkeit mit den aus ATM-Netzen bekannten Funktionen auf [16, 53, 138, 160], ohne dass jedoch vergleichbare Standards definiert wurden. Die in relevanten IETF-Dokumenten gemachten Aussagen zu Verkehrsmanagement orientieren sich häufig an einer konkreten Implementierung in Routern [46, 47].

Beim Verkehrsmanagement in ATM-Netzen findet man häufig eine Unterteilung in Verkehrssteuerung (*traffic control*) und Überlastbehandlung (*congestion control*) [16, 138, 160]. Damit soll der Unterschied zwischen präventiven und reaktiven Maßnahmen deutlich gemacht werden. Eine explizite Zuweisung der nachfolgend beschriebenen Verkehrsmanagement-Funktionen in IP-Netzen zu den genannten Kategorien gestaltet sich jedoch als schwierig. Ein Grund dafür ist, dass die meisten Verkehrsmanagement-Funktionen, insbesondere Ressourcenmanagement in Netzknoten (Abschnitt 3.2.1) und Verkehrslenkung (Abschnitt 3.2.5), sowohl präventive als auch reaktive Komponenten enthalten können. Zudem findet parallel zu den

Mechanismen im Netz ein wesentlicher Teil der Überlastbehandlung in den Endgeräten statt, nämlich in Form der TCP-Fenstermechanismen. Gleichzeitig zielen diese Mechanismen aber auch auf eine möglichst hohe Auslastung ab und tragen damit selbst zu Überlast bei. Dadurch stellt, vereinfacht ausgedrückt, Überlast in IP-Netzen nicht wie bei ATM eine Ausnahmesituation, sondern eher den Normalfall dar.

Die im Folgenden genannten Verkehrsmanagement-Funktionen arbeiten auf unterschiedlichen Zeitebenen. Zwar kann die in [160] für ATM-Netze festgelegte feingranulare Zuordnung nicht ohne weiteres übernommen werden, da einzelne Funktionen (z. B. die Verbindungsannahmesteuerung) auf ganz verschiedenen Zeitebenen zum Einsatz kommen können. Doch gilt weitgehend die Aussage, dass die in den Abschnitten 3.2.1 und 3.2.2 beschriebenen Verfahren eher im Zeitbereich von Paketankünften arbeiten, während die in den Abschnitten 3.2.3 bis 3.2.6 geschilderten Mechanismen auf eine längerfristige Wirkung ausgelegt sind.

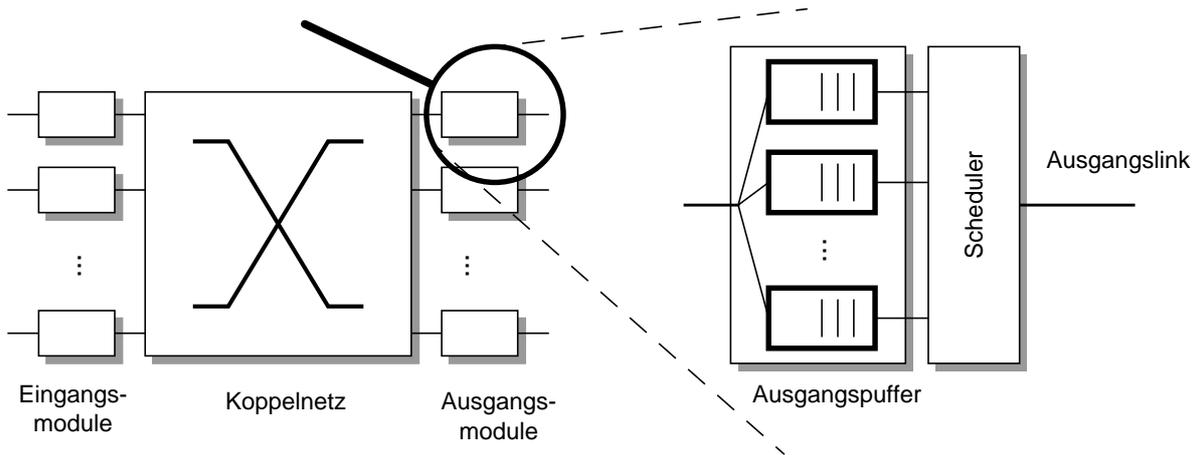
### **3.2.1 Verwaltung von Ressourcen in Netzknoten**

Innerhalb eines Netzknotens gilt es vornehmlich zwei Arten von Ressourcen zu verwalten: Bandbreite auf den Ausgangs- und Zwischenleitungen sowie Speicherplatz in verschiedenen Puffern. In diesem Abschnitt wird allerdings nur die Ressourcenverwaltung auf der Paketebene betrachtet, während bzgl. einer Betrachtung auf höheren Ebenen auf Abschnitt 3.2.4 verwiesen wird. Dies gilt insbesondere, wenn es um die Bandbreite als Ressource geht. Da Verfahren zur Bedienung von Paketen und zur Pufferverwaltung Hauptgegenstand dieser Arbeit sind, ist die in diesem Abschnitt betrachtete Form des Verkehrsmanagements von besonderer Bedeutung.

#### **3.2.1.1 Bedienstrategien**

Die zeitliche Organisation der Zuteilung von Betriebsmitteln wird allgemein als *Scheduling* bezeichnet. Die Aufgabe des Scheduling und die Entwicklung von Reihenfolgmodellen stellen eine spezielle Problemklasse innerhalb des Operations Research dar [265], für die schon seit Langem eine eigene Theorie existiert [77]. Zu den klassischen Anwendungsgebieten gehören Projekt- und Produktionsplanung sowie Logistik. Besondere Bedeutung haben Scheduling-Verfahren außerdem im Bereich der Rechnertechnik erlangt. Hier gehört es zu den zentralen Aufgaben des Betriebssystems, die Zuteilung von Prozessorressourcen an Prozesse zu organisieren [58, 163, 262].

Im Rahmen dieser Arbeit wird Scheduling im Zusammenhang mit der Bestimmung der Bedienreihenfolge von Datenpaketen in Vermittlungsknoten verwendet. Dabei wird davon ausgegangen, dass es sich um einen Netzknoten mit Ausgangspuffern handelt [264] und dass Engpässe nur in den Ausgangsmodulen auftreten (Bild 3.1), d. h. beim Zugang von den Ausgangspuffern (*output buffers*) auf die Ausgangsleitungen (*output links*). Weiter wird angenommen, dass jeder Ausgangspuffer physikalisch oder virtuell in mehrere Warteschlangen unterteilt ist,



**Bild 3.1:** Betrachtete Ressourcen in einem Vermittlungsknoten

in denen jeweils Pakete der gleichen Dienstgütekategorie gespeichert werden. Der Scheduler muss dann entscheiden, aus welcher dieser Warteschlangen er das nächste Paket entnimmt. Ist das Koppelnetz des Knotens nicht blockierungsfrei, können auch – je nach Architektur – innerhalb des Koppelnetzes oder in den Eingangsmodulen Engpässe auftreten. In diesem Zusammenhang kommen ebenfalls Scheduling-Verfahren zum Einsatz [192], deren Funktionalität aber über die der hier vorgestellten hinausgeht und die im Weiteren nicht behandelt werden.

Manche der aus anderen Anwendungsfeldern des Scheduling bekannten Bedienstrategien (*service disciplines*) kommen auch im hier betrachteten Fall der Zuteilung von Bandbreite zum Einsatz. Da es hier jedoch um die Zuteilung von Verarbeitungskapazität an einzelne Pakete geht, deren Ankunftszeitpunkte im System im Allgemeinen vorher unbekannt sind und meist nur mit Hilfe stochastischer Modelle beschrieben werden können, fallen etliche der klassischen Modelle wie etwa statische Rechenpläne bei der Betrachtung weg. Eine weitere Einschränkung besteht darin, dass Pakete in der Regel nur als Ganzes auf der Ausgangsleitung gesendet werden können, sodass Systeme mit unterbrechender Bedienung (*preemptive service*) nicht eingesetzt werden können.

Scheduling-Verfahren können grundsätzlich in arbeitserhaltende (*work-conserving*) und nicht arbeitserhaltende (*non-work-conserving*) unterteilt werden. Die arbeitserhaltenden Mechanismen, zu denen die überwiegende Mehrheit der nachfolgend vorstellten Verfahren gehört, garantieren eine Belegung, solange Pakete zur Bedienung anstehen. Bei nicht arbeitserhaltenden Systemen kann es vorkommen, dass mit der Belegung gewartet wird, bis bestimmte Bedingungen erfüllt sind, auch wenn dann die Linkkapazität zeitweise ungenutzt bleibt. Im verkehrstheoretischen Sinn bedeutet das eine Erhöhung der zu verrichtenden Arbeit. Dies kann für spezielle Anwendungen sinnvoll sein, führt jedoch grundsätzlich zu einer geringeren Effizienz.

Die einfachste und am weitesten verbreitete Strategie ist die Abarbeitung der Pakete entsprechend ihrer Ankunftsreihenfolge (*first in first out*, FIFO, auch *first come first serve*, FCFS).

Kommt ausschließlich diese Strategie zum Einsatz, ist keine Differenzierung möglich. Allerdings wird FIFO auch in Systemen mit Differenzierung in Kombination mit anderen Verfahren verwendet, und zwar zur Festlegung der Abarbeitungsreihenfolge für Pakete der gleichen Dienstgüteklasse, d. h. innerhalb einer logischen Warteschlange. Weitere Bedienstrategien ohne Differenzierung, die aber in der Praxis kaum eine Rolle spielen, sind LIFO (*last in first out*) und die zufällige Auswahl eines wartenden Pakets (*random service*).

Die arbeitserhaltenden Scheduling-Verfahren, die zur Dienstgütendifferenzierung in Vermittlungsknoten eingesetzt werden, lassen sich – teilweise in Anlehnung an [119, 279] – weitgehend einer der nachfolgend beschriebenen Gruppen zuordnen. Ein alternatives Modell zur Charakterisierung und Klassifizierung von Scheduling-Mechanismen ist in [260] beschrieben.

### Statische Prioritäten

Bei dieser sehr einfachen Strategie ist jeder logischen Warteschlange und somit jeder Dienstgüteklasse eine feste Verzögerungspriorität zugeordnet. Dies bedeutet, dass Pakete aus einer Klasse mit niedriger Priorität nur dann bedient werden, wenn sich in keiner der Warteschlangen mit höherer Priorität noch Pakete befinden. Somit erfahren Pakete der höchsten Priorität keinerlei Verzögerung durch niederpriorie Pakete – abgesehen von einer Restbedienzeit, die aufgrund der nicht unterbrechbaren Bedienung (*non-preemptive service*) unvermeidbar ist.

### Verfahren zur fairen Aufteilung der Linkbandbreite

Vorbild der *Fair Queueing*-Verfahren (FQ) ist ein Modell, das in [217] als *Generalized Processor Sharing* (GPS)<sup>2</sup> vorgestellt wurde und eine Erweiterung des aus der Modellierung von Rechnersystemen bekannten *Processor Sharing* (PS) [155] darstellt. Grundgedanke des auf einem Flüssigkeitsflussansatz (*fluid flow model*) beruhenden PS-Modells ist eine gleichmäßige Aufteilung der Rechenkapazität unter den aktiven Prozessen. Bei GPS treten an die Stelle der aktiven Prozesse diejenigen Warteschlangen, die nicht leer sind (*backlogged queues*). Im Gegensatz zu PS wird hier eine Gewichtung einzelner Warteschlangen vorgenommen, indem Gewichtungparameter  $\phi_i$  vergeben werden. Für die Rate  $R_i(t)$ , mit der die Warteschlange  $i$  zum Zeitpunkt  $t$  bedient wird, gilt dann [119]:

$$R_i(t) = \begin{cases} \frac{\phi_i}{\sum_{j \in B(t)} \phi_j} \cdot C & i \in B(t) \\ 0 & \text{sonst} \end{cases} \quad (3.2)$$

wobei  $B(t)$  die Menge der zum Zeitpunkt  $t$  nicht leeren Warteschlangen und  $C$  die Linkrate bezeichnen.

<sup>2</sup> Eine andere häufig hierfür verwendete Bezeichnung ist FFQ (*fluid flow fair queueing*).

Damit garantiert GPS eine vom jeweiligen Gewichtungsfaktor abhängige Mindestbedienrate für jede der  $K$  Warteschlangen:

$$R_i(t) \geq \frac{\phi_i}{\sum_{j=0}^{K-1} \phi_j} \cdot C \quad (3.3)$$

Aufgrund dieser Eigenschaft eignet sich GPS besonders zur Differenzierung hinsichtlich des Durchsatzes bei der Anwendung auf einzelne Verkehrsflüsse.

Wie im Falle von PS handelt es sich allerdings auch bei GPS um ein theoretisches Modell, das sich angesichts der Tatsache, dass zu jedem Zeitpunkt nur eine Warteschlange bedient werden kann und die Bedienung von Paketen nicht unterbrochen werden kann, nicht exakt implementieren lässt. Jedoch wurde im Laufe der Zeit eine Vielzahl von Verfahren entwickelt, die eine mehr oder weniger genaue Nachbildung der Funktionsweise von GPS unter realen Bedingungen erlauben. Als bekanntester Vertreter ist dabei *Weighted Fair Queueing* (WFQ)<sup>3</sup> zu nennen [84]. Bei WFQ wird unter den wartenden Paketen dasjenige als nächstes ausgewählt, dessen Bedienung im entsprechenden GPS-System zuerst abgeschlossen wäre, falls nach dem Beginn der Bedienung keine anderen Pakete mehr ankommen würden [279]. Andere bekannte Verfahren, die sich entweder durch eine genauere Nachbildung des GPS-Modells oder durch einfache Implementierbarkeit auszeichnen, sind z. B. WF<sup>2</sup>Q (*worst case fair weighted fair queueing*) [25], SCFQ (*self-clocked fair queueing*) [115], VC (*virtual clock*) [280], WRR (*weighted round robin*) [152] und DRR (*deficit round robin*) [253].

### **Fristenbasierte Verfahren**

Im Gegensatz zu FQ-Verfahren zielen fristenbasierte Mechanismen nicht auf eine garantierte Mindestbandbreite, sondern auf eine Begrenzung der Paketverzögerung ab und eignen sich dadurch besonders für die differenzierte Behandlung von echtzeitkritischem Verkehr. Bei der in [141, 179] vorgeschlagenen und in vielen Anwendungsbereichen eingesetzten Strategie *Earliest Deadline First* (EDF), auch als *Earliest Due Date* (EDD) bezeichnet, können für jede Klasse unterschiedliche maximale Verzögerungen (*delay bounds*)  $\delta_i$  angegeben werden. Bei Ankunft wird jedem Paket ein Zeitstempel (*tag*) zugeordnet, der eine Frist (*deadline, due date*) beinhaltet. Diese entspricht der Summe von Ankunftszeit und maximaler Verzögerung. Der Scheduling-Mechanismus versucht nun, eine Überschreitung der maximalen Verzögerung zu verhindern, indem die Pakete in der Reihenfolge steigender Fristen abgearbeitet werden.

In [113] wird gezeigt, dass EDD die optimale Strategie darstellt, wenn es um das Scheduling einzelner Verkehrsflüsse geht, deren Verkehrsparameter durch einen Quellflusskontrollmecha-

---

<sup>3</sup> Weitere in der Literatur vorzufindende Bezeichnungen für WFQ sind PGPS (*packet-by-packet generalized processor sharing*) [217] sowie PFQ (*packet-by-packet fair queueing*) [115].

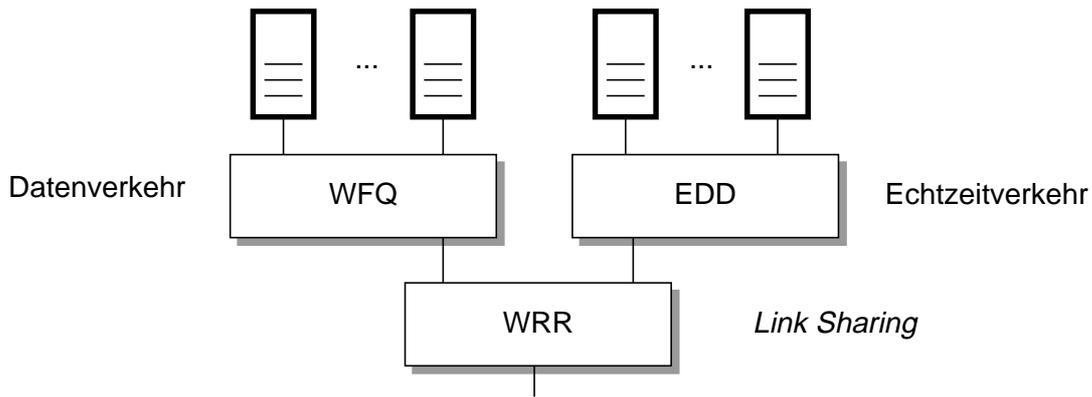
nismus (siehe Abschnitt 3.2.2) überwacht werden und für die eine deterministische Garantie bzgl. der Verzögerung gegeben werden soll. Optimal bedeutet hier zunächst, dass bei gegebener Anzahl  $n_i$  von Verkehrsflüssen für jede Klasse  $i$  (mit jeweils gleichen Verkehrsparametern und gleicher Maximalverzögerung  $\delta_i$ ) die maximal auftretende Verspätung (*maximum latency*) bei Verwendung von EDD am kleinsten ist. Daraus folgt dann, dass mit EDD die Ausdehnung der so genannten *Schedulable Region* am größten ist. Darunter kann die Menge aller  $K$ -Tupel  $(n_0, n_1, \dots, n_{K-1})$  verstanden werden, für die eine Einhaltung der maximalen Verzögerungen für alle Pakete garantiert werden kann. Anschaulich betrachtet heißt das, dass die Verbindungsannahmesteuerung (siehe Abschnitt 3.2.3) weniger Reservierungsanforderungen ablehnen muss, wenn die Bedienung von Paketen entsprechend der EDD-Strategie erfolgt.

Wird im Netz keine Verbindungsannahmesteuerung vorgenommen, besteht die Gefahr, dass einzelne Pakete erst nach Ablauf ihrer Frist bearbeitet werden. Dabei zeigt sich jedoch eine andere Eigenschaft von EDD, die besonders im Zusammenhang mit der hier betrachteten relativen Differenzierung interessant ist: Im Fall eines „unkontrolliertem“ Zugangs (keine Verbindungsannahmesteuerung) ist die Wahrscheinlichkeit einer Überschreitung der Bearbeitungsfrist in allen Klassen gleich [77].

Der Nachteil von EDD liegt in der komplexeren Implementierung. Zum einen muss für jedes Paket ein Zeitstempel mit der zugehörigen Frist berechnet und zusammen mit dem Paket in der Warteschlange gespeichert werden – bei FQ-Verfahren ist häufig nur ein *Tag* pro Warteschlange erforderlich. Zum anderen bringt die Bestimmung des Pakets mit der kleinsten Frist bei vielen Klassen einen nicht unerheblichen Sortieraufwand mit sich. Um letztgenanntem Effekt entgegenzuwirken, wurden Verfahren wie RPQ (*rotating priority queues*) entwickelt, das mit einem effizienteren Sortieralgorithmus arbeitet und dabei eine gute Approximation der EDD-Strategie liefert [275].

### **Ratengesteuerte Bedienstrategien**

Die ratengesteuerten Bedienstrategien (*rate-controlled service disciplines*), die der Klasse der nicht arbeitserhaltenden Scheduling-Mechanismen zuzuordnen sind, wurden vor dem Hintergrund der isolierten Behandlung einzelner Verkehrsflüsse entwickelt [282]. Sie zeichnen sich dadurch aus, dass vor dem eigentlichen Scheduling eine Verkehrsformung (siehe Abschnitt 3.2.2.2) stattfindet, die u. a. eine Ratenbegrenzung vornimmt. Bei dem in [281] vorgestellten RCSP (*rate-controlled static priority*) erfolgt nach der Verkehrsformung eine Abarbeitung auf der Basis statischer Prioritäten. RCSP ermöglicht gegenüber FQ und EDD eine getrennte Umsetzung von Bandbreite- und Verzögerungsgarantien.



**Bild 3.2:** Beispiel für hierarchisches Scheduling

### Weitere Verfahren

Neben den genannten Gruppen von Verfahren gibt es zahlreiche weitere Mechanismen, die eine Differenzierung bei der Bedienung von Paketen zulassen, indem sie sich an der Warteschlangenlänge, der Wartezeit von Paketen oder laufend gemessenen Größen orientieren. Als Beispiel sei hier die auf [155] zurückgehende WTP-Disziplin (*waiting time priority*) genannt, bei der Pakete in der Reihenfolge ihrer gewichteten Wartezeit bedient werden. Als Gewichtungsfaktoren werden dabei klassenspezifische Differenzierungsparameter  $s_i$  verwendet. WTP wird aufgrund der Eigenschaft, dass die mittleren Wartezeiten proportional zu den Gewichtsparametern sind, auch im Zusammenhang mit relativer Differenzierung verwendet [88].

### Hierarchisches Scheduling

Die bisher angesprochenen Scheduling-Verfahren ermöglichen eine Differenzierung nach bestimmten Kriterien (z. B. Durchsatz oder maximale Verzögerung). Allerdings ist es nicht unbedingt sinnvoll, für alle Arten von Verkehr die gleichen Kriterien anzuwenden. So ist eine Differenzierung hinsichtlich der zugeteilten Bandbreite, wie sie z. B. die FQ-Verfahren anbieten, vor allem für (elastischen) Datenverkehr geeignet, während zur Unterscheidung verschiedener Echtzeitverkehrsklassen eine fristenbasierte Strategie adäquat ist. In diesem Fall bietet hierarchisches Scheduling, wie exemplarisch in Bild 3.2 gezeigt, einen möglichen Ausweg. Im dargestellten Beispiel sorgt eine erste Scheduling-Stufe für die Aufteilung der Link-Bandbreite (*link sharing*) auf die beiden Verkehrsarten, während zur Differenzierung innerhalb des Daten- bzw. Echtzeitverkehrs die jeweils dafür geeigneten Scheduler zum Einsatz kommen. Weitere Beispiele für die Anwendung von hierarchischem Scheduling sind:

- die Aufteilung der Link-Bandbreite an verschiedene Organisationen (z. B. Dienstanbieter), die wiederum ihren Anteil unter einzelnen Nutzern aufteilen [110].
- die Verteilung des Scheduling auf verschiedene Systemteile (Basisstation, Mobilstation) bei der Realisierung drahtloser Hochgeschwindigkeits-Zugangsnetze [254].

Hierarchisches Scheduling kann prinzipiell durch Kaskadierung einzelner Scheduler erreicht werden, indem der auf einer höheren Hierarchie-Ebene befindliche Scheduler alle Warteschlangen, die von jeweils einem Scheduler der nächstniederen-Ebene verwaltet werden, als eine virtuelle Warteschlange betrachtet. Am Kopf der virtuellen Warteschlange befindet sich das Paket, das von dem Scheduler der niederen Hierarchieebene als nächstes ausgewählt wird. Wenn allerdings die grundlegenden Mechanismen auf den einzelnen Ebenen bekannt sind (z. B. möglichst gute GPS-Approximation auf allen Ebenen), so können für das hierarchische Scheduling häufig spezielle Lösungen gefunden werden, die effizienter arbeiten als die schematische Kaskadierung [26].

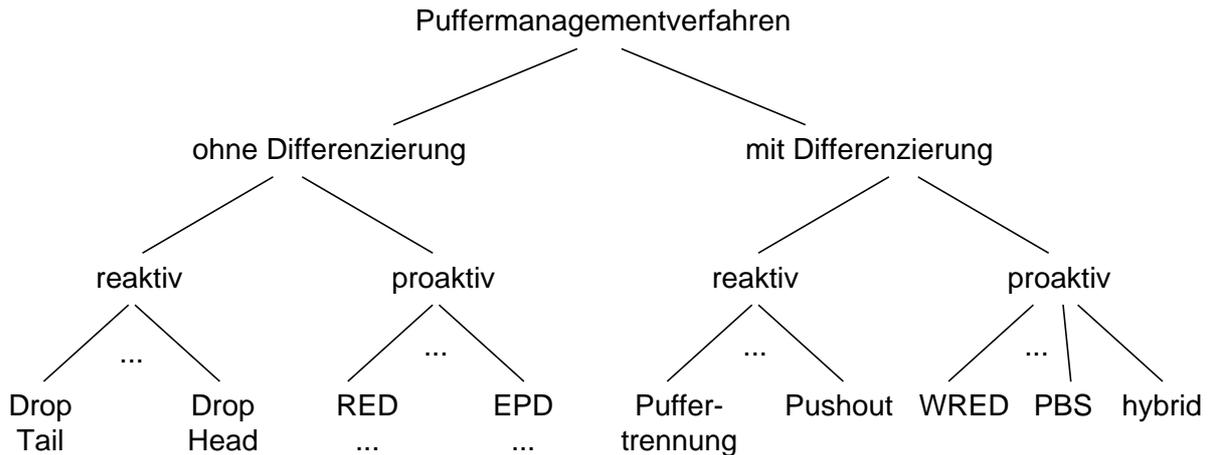
Eine weitere Realisierungsmöglichkeit für hierarchisches Scheduling, die hier aufgrund ihrer weiten Verbreitung nicht unerwähnt bleiben soll, ist das in [110] vorgeschlagene *Class-Based Queueing* (CBQ). Bei diesem Verfahren sorgen die Scheduler auf den oberen Hierarchie-Ebenen (*link sharing scheduler*) für eine gewichtete Aufteilung der Linkbandbreite und bedienen sich dazu eines auf Messungen der mittleren Rate der jeweiligen aggregierten Verkehrsströme beruhenden Regelungsmechanismus. Auf der untersten Hierarchie-Ebene können verschiedene bekannte Scheduling-Verfahren eingesetzt werden. Untersuchungen in [187] haben allerdings gezeigt, dass CBQ in manchen Szenarien zu (unerwünschter) Unfairness führt.

### 3.2.1.2 Puffermanagement

Wie im vorigen Abschnitt gezeigt, werden Pakete, die auf die Übertragung warten, in einem Puffer zwischengespeichert, der bei Anwendung eines differenzierenden Scheduling-Mechanismus physikalisch oder logisch in einzelne Warteschlangen unterteilt ist. Da es sich beim Speicherplatz in diesem Puffer grundsätzlich um eine begrenzte Ressource handelt, kann es in Überlastsituationen zu Paketverlusten kommen. Die Aufgabe des Puffermanagements besteht nun darin, einerseits insgesamt für eine effiziente Nutzung der Speicherressourcen zu sorgen und andererseits in Analogie zum Scheduling eine Differenzierung der Verkehrsklassen hinsichtlich der Verluste zu ermöglichen. Darüber hinaus ist es – besonders im Hinblick auf Echtzeitanforderungen – häufig wünschenswert, die Warteschlangenlänge und damit die Paketverzögerung zu begrenzen und zu diesem Zweck Pakete zu verwerfen, obwohl noch genügend Speicherplatz vorhanden wäre.

Die Entwicklung geeigneter Pufferverwaltungsstrategien ist schon seit längerer Zeit Gegenstand der Forschung. In Ergänzung zur nachfolgenden Klassifikation geben [87, 119, 160] eine Übersicht über entsprechende Verfahren.

Wie in Bild 3.3 dargestellt, kann wie bei den Scheduling-Mechanismen zunächst zwischen Verfahren mit und ohne Verlustdifferenzierung unterschieden werden. In beiden Gruppen existieren einerseits reaktive Verfahren, bei denen Pakete nur verworfen werden, wenn der Speicherplatz ausgeschöpft ist. Andererseits haben besonders im Kontext von IP-Netzen proaktive



**Bild 3.3:** Klassifikation von Puffermanagementverfahren

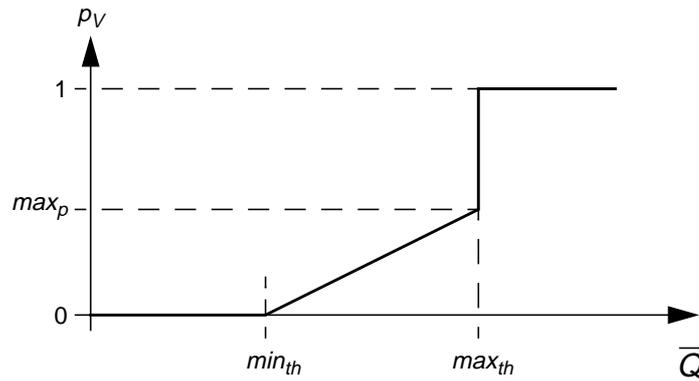
Verfahren (*active queue management*, AQM) eine besondere Bedeutung. Diese Mechanismen sind dadurch gekennzeichnet, dass Pakete verworfen werden können, obwohl der Puffer noch nicht voll ist, z. B. um einer drohenden Überlast frühzeitig entgegenzuwirken.

### Verfahren ohne Differenzierung

Die einfachste und gleichzeitig am weitesten verbreitete Pufferverwaltungs-„Strategie“ ist *Drop Tail* (DT). Dabei werden ankommende Pakete verworfen, wenn sie nicht mehr im Puffer gespeichert werden können. Es handelt sich damit bei DT um ein reaktives Verfahren. Alternativ dazu können Pakete am Kopf der Warteschlange (*drop head*, *drop from front*) [167] verworfen werden, damit das ankommende Paket Platz findet. Dies kann z. B. bei TCP-Verkehr vorteilhaft sein, um ein frühzeitigeres Ansprechen der TCP-Überlaststeuerungsmechanismen zu bewirken. Eine weitere Möglichkeit, die im Zusammenhang mit der Vermeidung von Synchronisationseffekten bei TCP-Verkehr vorgeschlagen wurde, ist die Verwerfung eines zufällig ausgewählten Pakets im Puffer (*random drop*) [108].

Eine große Bedeutung unter den proaktiven Verfahren hat in IP-Netzen vor allem RED (*random early detection*) [109] erlangt, das von der IETF als Standardverfahren in Routern vorgeschlagen wird [46]. Die Grundidee von RED ist, durch zufälliges Verwerfen ankommender Pakete mit einer vom Pufferfüllstand abhängigen Wahrscheinlichkeit die mittlere Warteschlangenlänge und damit die mittlere Paketverzögerung im Vergleich zu DT zu reduzieren. Gleichzeitig soll RED Synchronisationseffekte, wie sie durch die Rückkopplungsmechanismen von TCP prinzipiell entstehen können [108], verhindern und damit für eine bessere Auslastung des Links sowie eine erhöhte Fairness sorgen.

Die Abhängigkeit der Verwerfungswahrscheinlichkeit  $p_V$  von der Warteschlangenlänge ist in Bild 3.4 dargestellt und lässt sich folgendermaßen formulieren:



**Bild 3.4:** Funktionsverlauf der Verwerfungswahrscheinlichkeit bei RED

$$p_V = \begin{cases} 0 & \bar{Q} \leq \min_{th} \\ \max_p \cdot \frac{\bar{Q} - \min_{th}}{\max_{th} - \min_{th}} & \min_{th} < \bar{Q} < \max_{th} \\ 1 & \bar{Q} \geq \max_{th} \end{cases} \quad (3.4)$$

Dabei wird anstelle des aktuellen Pufferfüllstandes  $Q(t)$  die über einen längeren Zeitraum gemittelte Warteschlangenlänge  $\bar{Q}$  herangezogen, um eine bessere Stabilität des Verfahrens zu erreichen. Eine Verwerfung findet nur statt, wenn  $\bar{Q}$  größer ist als eine untere Schwelle  $\min_{th}$ . Die Verwerfungswahrscheinlichkeit steigt von 0 linear an bis zu einem Wert  $\max_p$ , der erreicht ist, wenn  $\bar{Q}$  der oberen Schwelle  $\max_{th}$  entspricht. Ist  $\bar{Q}$  größer als  $\max_{th}$ , werden ankommende Pakete sicher verworfen. Dabei ist zu beachten, dass die tatsächliche Warteschlangenlänge  $Q(t)$  zeitweise deutlich größer sein kann als  $\bar{Q}$  und insbesondere auch als  $\max_{th}$ . Daraus kann abgeleitet werden, dass für  $\max_{th}$  ein Wert gewählt werden sollte, der kleiner ist als die Puffergröße, um häufige Verluste aufgrund vollständig gefüllten Puffers zu vermeiden.

Neben den drei Parametern  $\min_{th}$ ,  $\max_{th}$  und  $\max_p$  gibt es noch weitere Größen, die das Verhalten von RED beeinflussen. Dazu gehören vor allem die bei der EWMA-Mittelung (*exponentially weighted moving average*) der Warteschlangenlänge auftretenden Parameter:

$$\bar{Q} = \begin{cases} (1 - w_q) \cdot \bar{Q} + w_q \cdot Q(t) & Q(t) > 0 \\ (1 - w_q)^{\Delta t_0 \cdot C / L_0} \cdot \bar{Q} & Q(t) = 0 \end{cases} \quad (3.5)$$

Dies ist zum einen der Gewichtungsfaktor  $w_q$ , dann aber auch die erwartete mittlere Paketlänge  $L_0$ , die im Fall einer bei Ankunft leeren Warteschlange zusammen mit der Bedienrate  $C$  und dem Zeitintervall  $\Delta t_0$  seit dem Bedienende des letzten Pakets in die Berechnung von  $\bar{Q}$

eingeht. Schließlich muss noch die selten erwähnte Option genannt werden, aufeinander folgende Pakete zu zählen, die eine mittlere Warteschlangenlänge im Bereich  $\min_{th} \leq \bar{Q} < \max_{th}$  vorgefunden haben, jedoch nicht verworfen wurden, und dann die Verwerfungswahrscheinlichkeit in Abhängigkeit von diesem Zähler zu erhöhen. Für Details sei auf [109] verwiesen.

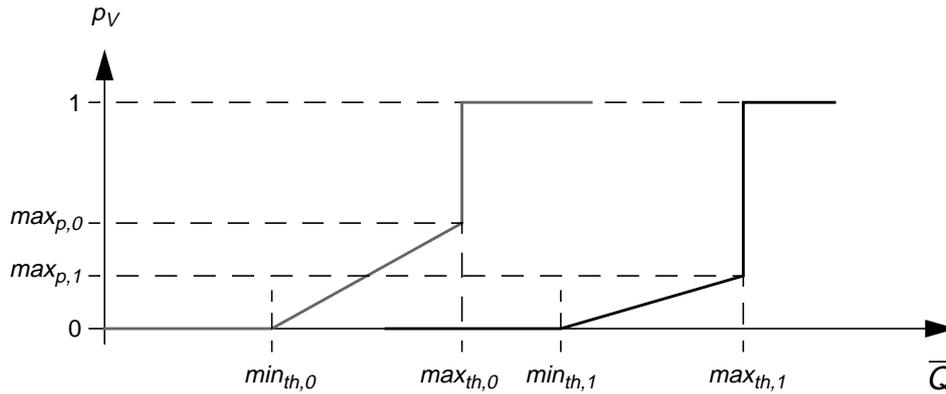
Seit der Vorstellung von RED hat es eine Vielzahl von Vorschlägen zur Erweiterung und Modifikation gegeben, z. B. im Hinblick auf eine verbesserte Fairness in Gegenwart nicht adaptiver Verkehrsflüsse [176, 211, 214], eine automatische Anpassung der Parameter [102] oder einen anderen Verlauf der Verlustwahrscheinlichkeit über der gemittelten Warteschlangenlänge [13]. Darüber hinaus gibt es eine Reihe von Verfahren wie z. B. das in [101] vorgeschlagene BLUE, die im Wesentlichen auf dem gleichen Prinzip beruhen wie RED, aber die Verwerfungswahrscheinlichkeit aus anderen Größen ableiten als der gemittelten Warteschlangenlänge.

Zu den proaktiven Verfahren sind schließlich auch selektive Verwerfungsmechanismen zu zählen, wie sie eingesetzt werden, wenn IP-Pakete über ein ATM-Netz übertragen werden (siehe Abschnitt 3.3.1). Mechanismen wie *Early Packet Discard* (EPD) [238], *Partial Packet Discard* (PPD) [11] oder *preemptive Partial Packet Discard* (pPPD) [168] haben zum Ziel, bereits möglichst frühzeitig – z. B. bei Ankunft der ersten ATM-Zelle, die zu einem IP-Paket gehört – eine Entscheidung über die Verwerfung zu treffen und dann nach Möglichkeit von den Zellen, die zu diesem Paket gehören, entweder alle oder keine zu verwerfen. Die Entscheidung erfolgt dabei in der Regel auf der Basis von Schwellwerten bzgl. der Pufferbelegung.

### **Verfahren mit Differenzierung**

Im Zusammenhang mit Verkehrsmanagement sind in erster Linie diejenigen Verfahren interessant, die eine Verlustdifferenzierung ermöglichen. Hierbei sind zunächst die drei klassischen, aus der Untersuchung von ATM-Netzen bekannten Prioritätsmechanismen zu nennen [160, 161]:

- **Puffertrennung:** Jede Warteschlange wird durch einen physikalisch separaten Puffer oder einen fest zugeordneten logischen Teilpuffer realisiert. Dies macht vor allem in Kombination mit einem differenzierenden Scheduling-Mechanismus Sinn. Die Verlustdifferenzierung kann durch unterschiedliche Puffergrößen erreicht werden, hängt aber darüber hinaus auch vom Angebot und den Scheduling-Parametern der jeweiligen Klasse ab.
- **Verdrängung:** Dieser Mechanismus kann als Analogon zu den statischen Verzögerungsprioritäten beim Scheduling gesehen werden, d. h. jeder Klasse ist eine statische Verlustpriorität zugeordnet. Findet ein Paket bei seiner Ankunft keinen Platz im Puffer, wird versucht, durch Verdrängung (*pushout*) bereits im Puffer befindlicher Pakete mit niedriger Priorität, für das höherprioritäre Paket Platz zu schaffen. Nur wenn dies nicht gelingt, wird das ankommende Paket verworfen. Die Suche nach niederprioritären Paketen beginnt jeweils bei der niedrigsten Prioritätsstufe entweder am Pufferein- oder -ausgang.



**Bild 3.5:** Funktionsverlauf der Verwerfungswahrscheinlichkeit bei WRED

- **Schwellwertverfahren:** Hierbei steht den einzelnen Klassen nur ein (je nach Klasse unterschiedlich großer) Teilbereich des Puffers zur Verfügung (*partial buffer sharing*, PBS). Die Abgrenzung der Teilbereiche wird über Belegungsschwellwerte erreicht, d. h. ein ankommendes Paket wird abgewiesen, wenn der Pufferfüllstand den für diese Klasse definierten Schwellwert  $\Theta_i$  überschritten hat. Wie beim Verdrängungsmechanismus unterliegen auch hier die Klassen einer eindeutigen Ordnungsrelation, d. h.  $\Theta_i > \Theta_j$  bedeutet eine Besserstellung von Klasse  $i$  gegenüber Klasse  $j$ . Es findet allerdings keine strikte Priorisierung statt. Eine Variante des Verfahrens verwendet anstatt auf den Gesamtfüllstand bezogener Schwellwerte klassenspezifische Schwellwerte, d. h. ein Paket der Klasse  $i$  wird verworfen, wenn die Anzahl an Paketen der Klasse  $i$  im Puffer größer ist als  $\Theta_i$ .

Während Puffertrennung und Verdrängung als reaktive Maßnahmen zu sehen sind, kann das Schwellwertverfahren den proaktiven Mechanismen zugeordnet werden, da Pakete niedriger Priorität auch verworfen werden können, obwohl der Puffer nicht komplett belegt ist.

Neben diesen klassischen Puffermanagementstrategien gibt es auch Verfahren, die als Basismechanismus RED verwenden, gleichzeitig jedoch eine differenzierte Behandlung erlauben. Unter diesen Verfahren ist in erster Linie das auf [71] zurückgehende WRED (*weighted random early detection*) zu nennen.<sup>4</sup> Dieses realisiert eine Differenzierung durch die Einführung klassenspezifischer Parameter  $min_{th,i}$ ,  $max_{th,i}$  und  $max_{p,i}$  anstelle von  $min_{th}$ ,  $max_{th}$  und  $max_p$  (Bild 3.5). Als Basisgröße für die Bestimmung der Verwerfungswahrscheinlichkeit kann weiterhin die gemittelte Gesamtwarteschlangenlänge verwendet werden. Eine Alternative besteht darin, in Analogie zu der o. g. Variante beim Schwellwertverfahren nur die wartenden Pakete derjenigen Klasse zu berücksichtigen, zu der das ankommende Paket gehört. Dies ist weitgehend identisch mit einem Puffermanagement, das die Puffer für die einzelnen Klassen in der oben geschilderten Weise trennt und in jedem Puffer einen RED-Mechanismus mit unterschiedlichen Parameterwerten implementiert.

<sup>4</sup> In [71] wird anstelle von WRED mit Bezug auf die in Abschnitt 3.3.3 beschriebene Architektur die Bezeichnung RIO (*RED with In/Out bit*) verwendet.

Wie in [38] gezeigt, kann WRED auch zur proportionalen Differenzierung eingesetzt werden, insbesondere wenn nur  $\max_{p,i}$  für die einzelnen Klassen variiert wird und für die Schwellenparameter identische Werte in allen Klassen gewählt werden. Daneben gibt es eine Reihe weiterer Verfahren, die speziell für eine proportionale Verlustdifferenzierung konzipiert wurden [37, 89, 124, 174, 175, 198]. Auf diese wird in Kapitel 5 im Zusammenhang mit dem dort vorgeschlagenen Mechanismus noch genauer eingegangen werden.

Schließlich gibt es auch eine Reihe von Mechanismen, die eine Kombination aus Scheduling und Puffermanagement darstellen (in Bild 3.3 als „hybride Verfahren“ bezeichnet). Dazu gehört z. B. STE (*shortest time to extinction*), ein Scheduling-Mechanismus, der wie EDD mit Fristen arbeitet, im Gegensatz zum ursprünglichen EDD aber Pakete verwirft, die ihre Frist überschritten haben [216]. Als hybrides Verfahren kann auch der für das in [131] vorgeschlagene ABE (*alternative best effort*) eingesetzte Mechanismus bezeichnet werden. Dieses Schema dient zur Trennung von Echtzeitverkehr und elastischem Verkehr. Dabei wird für Echtzeitverkehr eine begrenzte Paketverzögerung garantiert, die aber komplett zu Lasten höherer Verluste in der Echtzeitklasse geht. Elastischer Verkehr erfährt also die gleiche Verlustwahrscheinlichkeit wie im Fall eines FIFO-Schedulings.

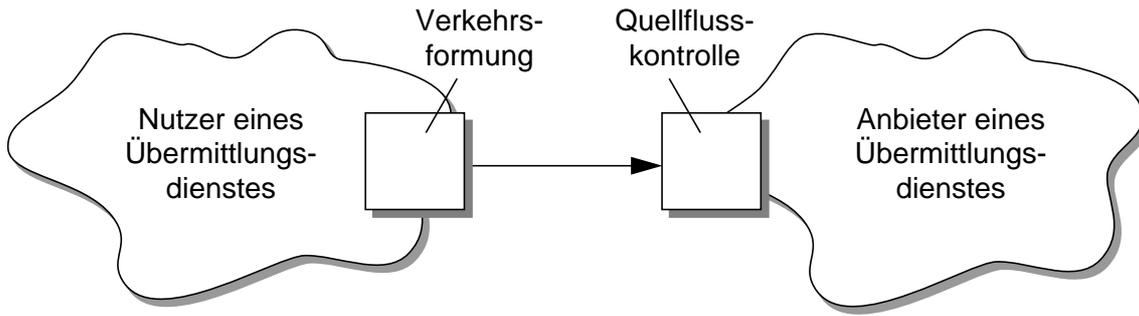
### 3.2.2 Quellflusskontrolle und Verkehrsformung

Quellflusskontrolle und Verkehrsformung sind zwei grundsätzlich zu trennende Verkehrsmanagement-Funktionen mit unterschiedlichen Zielsetzungen, die jedoch auf der Anwendung ähnlicher Mechanismen beruhen. Beide wirken auf der Paketebene und können sich auf einzelne Verkehrsflüsse oder auf aggregierte Verkehrsströme beziehen.

#### 3.2.2.1 Quellflusskontrolle

Ziel der Quellflusskontrolle (*source policing*) ist die Überwachung der im Verkehrsvertrag angegebenen Verkehrsparameter. In Netzarchitekturen, die Dienstgüte in Form von absoluten Garantien (siehe Abschnitt 3.1.4.1) bereitstellen, stellt sie eine unverzichtbare Komponente dar. Klassisch wird zwischen einer Parameterüberwachung am Netzzugang (*usage parameter control*, UPC) und einer zwischen angrenzenden Netzen stattfindenden Kontrolle (*network parameter control*, NPC) unterschieden [16, 138].

Parameter, die es zu überwachen gilt, sind in erster Linie Spitzenrate und mittlere Rate von Verkehrsströmen. Für beide eignen sich Verfahren, die auf dem aus ATM-Netzen bekannten Konzept des *Leaky Bucket* (LB) oder *Token Bucket* (TB) beruhen [233]. Diesem liegt die Vorstellung eines Flüssigkeitsspeichers mit Kapazität (*bucket depth*)  $S_{LB}$  und Abflussrate (*bucket rate*)  $R_{LB}$  zugrunde. Konformität liegt in diesem Modell dann vor, wenn der Speicher nicht überläuft, d. h. wenn für die in einem beliebigen Intervall der Länge  $t$  ankommende Flüssigkeitsmenge  $A_t$  gilt [236]:



**Bild 3.6:** Zusammenwirken von Quellflusskontrolle und Verkehrsformung

$$A_t \leq R_{LB} \cdot t + S_{LB} \quad (3.6)$$

Die für ATM-Netze vorgeschlagenen Implementierungen dieses Modells verwenden einen Zähler, der bei der Ankunft einer ATM-Zelle inkrementiert und parallel dazu mit Rate  $R_{LB}$  dekrementiert wird [16]. Eine Verletzung liegt vor, wenn der Zähler den Wert  $S_{LB}$  überschreitet. Im Fall von IP-Paketen muss außerdem die Paketlänge berücksichtigt werden. Wird der LB-Mechanismus zur Überwachung der mittleren Rate herangezogen, ist ein größerer Wert für  $S_{LB}$  zu wählen, um dem Auftreten von Bursts Rechnung zu tragen, während bei der Anwendung zur Kontrolle der Spitzenrate ein geringer Toleranzwert gewählt werden kann.

Wird durch die Quellflusskontrolle festgestellt, dass die Charakteristik des gemessenen Verkehrs nicht mit den vereinbarten Parametern übereinstimmt, werden Pakete entweder verworfen oder einer Dienstgüteklasse mit geringerer Priorität zugeordnet. In IP-Netzen findet meist eine Herabstufung in die niedrigste Prioritätsklasse (Best-Effort-Verkehr) statt.

### 3.2.2.2 Verkehrsformung

Die Verkehrsformung (*traffic shaping*) tritt meistens in Kombination mit der Quellflusskontrolle auf, allerdings auf der Seite des Vertragspartners, der sich zur Einhaltung der Verkehrsparameter verpflichtet hat (Bild 3.6). Es wird versucht, durch Pufferung von Paketen die Charakteristik des abgehenden Verkehrs so zu beeinflussen, dass die Quellflusskontrolle auf der Gegenseite keine Pakete verwerfen oder markieren muss. Das einfachste Beispiel ist in diesem Zusammenhang die (Wieder-) Herstellung des Mindestabstandes zwischen Paketen gemäß einer vereinbarten Spitzenrate (*spacing*). Wenn es um die Einhaltung einer vereinbarten mittleren Rate geht, kann wie bei der Quellflusskontrolle ein LB-Mechanismus verwendet werden, um die Zeitpunkte zu bestimmen, an denen gepufferte Pakete weitergesendet werden.

Ein weiterer Anwendungsfall ist die Verkehrsformung innerhalb des Netzes mit dem Ziel, ein möglichst günstiges Verkehrsprofil zu erzeugen, was in der Regel bedeutet, dass der Verkehr durch Pufferung von dicht hintereinander ankommenden Paketen und verzögertes Weitersenden geglättet wird.

### 3.2.3 Verbindungsannahmesteuerung

Soll Dienstgüte in Form von absoluten Garantien bereitgestellt werden, ist eine Verbindungsannahmesteuerung (*connection admission control*<sup>5</sup>, CAC) unerlässlich. Entsprechende Mechanismen entscheiden, ob ein Verbindungswunsch erfüllt werden kann, ohne dass Dienstgütereorderungen bestehender Verbindungen oder der neuen Verbindung verletzt werden. Die Entscheidung ist somit in der Regel vom aktuellen Netzzustand abhängig.

Dabei ist der Begriff der Verbindung – gerade im Zusammenhang mit IP-Netzen – in einer erweiterten Bedeutung zu sehen. Das heißt, CAC bezieht sich nicht nur auf Ende-zu-Ende-Nutzerverbindungen, für die ein vom Nutzer definiertes Verkehrsprofil vorliegt. Statt dessen haben Annahmemechanismen in zunehmendem Maße ihre Bedeutung im Kontext der Aggregation von Verkehrsströmen (siehe Abschnitt 3.1.5) und der Bildung virtueller Verbindungsleitungen (*trunks*). Dort sind es meist Verkehrsmessungen, die als Grundlage für die Annahmementscheidung dienen.

Die Verbindungsannahmesteuerung ist außerdem eng verknüpft mit der Verkehrslenkung (siehe Abschnitt 3.2.5). Je nach Sichtweise kann die Verbindungsannahmesteuerung als ein Bestandteil der Verkehrslenkung [257] oder die Verkehrslenkung als eine Teilaufgabe der Verbindungsannahmesteuerung [16] interpretiert werden. Allerdings findet in der Praxis häufig keine integrierte Betrachtung dieser beiden Funktionen statt, sondern es kommt der in Abschnitt 3.1.6 angesprochene Dekompositionsansatz zum Tragen. Das bedeutet, dass CAC-Algorithmen aufgrund ihrer Komplexität in aller Regel jeweils nur den Zustand in einem Knoten – mitunter sogar nur an einer Stelle eines Knotens (z. B. dem Ausgangsmodul) – berücksichtigen und sich die Gesamtentscheidung dann als logische Kombination der einzelnen Entscheidungsergebnisse ergibt.

Der Klassifizierung von Verbindungsannahme-Mechanismen in ATM-Netzen in [160] folgend, kann grob eine Unterteilung in direkte und indirekte Verfahren vorgenommen werden. Bei den direkten Verfahren werden die für die Dienstgüte relevanten Leistungsparameter (z. B. die Paketverlustwahrscheinlichkeit) für den Fall ermittelt, dass die neue Verbindung angenommen wird. Wenn sich ergibt, dass garantierte Größen verletzt werden würden, muss die Verbindung abgelehnt werden. Die notwendigen Berechnungen werden zum Zeitpunkt des Verbindungsaufbauwunsches, also in Echtzeit, durchgeführt, was erhebliche Anforderungen an die Kapazität von Steuerungsrechnern stellen kann, auf denen die entsprechenden Algorithmen ausgeführt werden.

Indirekte Verfahren hingegen beruhen auf dem Konzept der effektiven Bandbreite [114, 117, 153]. Die Idee dieses Konzepts ist es, die Annahmementscheidung auf ähnlich einfache Weise

---

<sup>5</sup> Gleichbedeutende Begriffe im Englischen sind *Call Admission Control* oder *Call Acceptance Control*. Im IP-Umfeld wird meist nur von *Admission Control* gesprochen.

durchzuführen wie in Netzen mit Mehrkanal-Durchschaltvermittlung, nämlich durch Aufsummieren der Bandbreite einzelner Verbindungen und Vergleich mit der zur Verfügung stehenden Kapazität. An die Stelle der Kanalbandbreiten in durchschaltvermittelnden Netzen tritt im Fall von paketvermittelnden Netzen und Verkehrsströmen mit variabler Bitrate die effektive Bandbreite als Maß für den Ressourcenbedarf. Diese hängt von verschiedenen Faktoren ab. Dazu zählen in erster Linie die Charakteristik des Verkehrsstroms (z. B. mittlere Rate, Spitzenrate), aber auch die einzuhaltenden Dienstgüteparameter (z. B. Zielverlustwahrscheinlichkeit) sowie die Eigenschaften des zugrunde liegenden Systems (z. B. Abarbeitungsdisziplin). Die Schwierigkeit liegt nun allerdings in der Bestimmung der effektiven Bandbreite, die in der Praxis meist nur approximativ erfolgen kann. Außerdem tritt das Problem auf, dass die effektive Bandbreite, wie z. B. in [36] gezeigt, von der aktuellen Verkehrszusammensetzung (*traffic mix*) abhängt und sich somit während der Dauer einer Verbindung ändern kann.

Neben der Unterteilung in direkte und indirekte Verfahren ist es sinnvoll, zwischen Mechanismen zu unterscheiden, die allein die im Verkehrsvertrag festgelegten Parameter bzgl. der Charakteristik eines Verkehrsstroms berücksichtigen, und solchen, bei denen auch die Ergebnisse von aktuellen Messungen von Verkehrs- oder Leistungsparametern einfließen. Gemessen werden dabei häufig Mittelwert, Varianz oder auch Verteilung der Ankunftsrate von Paketen in bestimmten Intervallen bzw. Verlustraten oder Warteschlangenlängen. Zur einzigen Möglichkeit werden auf Messungen beruhende Verfahren dann, wenn beim Verbindungsaufbau keine Parameter festgelegt werden oder der betreffende Knoten gar nicht explizit in die zum Verbindungsaufbau erforderliche Signalisierung einbezogen wird. Dies ist z. B. der Fall, wenn es um eine Annahmesteuerung für TCP-Verbindungen geht [188]. Einen ausführlichen Überblick über messbasierte CAC-Verfahren geben [51, 252].

Das Sekundärziel von CAC-Verfahren neben der Einhaltung von vereinbarten Dienstgüteparametern besteht in einer möglichst effizienten Auslastung der Ressourcen. Wenn lediglich eine Dienstklasse vorliegt, also alle Verbindungen die gleichen Anforderungen haben, bedeutet dies, durch Ausnutzung von statistischem Multiplexen möglichst viele Verbindungen anzunehmen, sodass die geforderten Werte für die Dienstgüteparameter gerade noch eingehalten werden. Im Fall von differenzierter Dienstgüte mit mehreren QoS-Klassen hingegen bietet sich ein breites Spektrum von möglichen Optimierungszielen, da in den verschiedenen Klassen nicht nur unterschiedliche Anforderungen bzgl. Leistungsmaßen wie Verlustwahrscheinlichkeiten existieren, sondern auch die resultierenden Blockierungswahrscheinlichkeiten auf Verbindungsebene potenziell unterschiedlich sind, vielleicht sogar unterschiedlich sein sollen. Die sich daraus ergebenden Strategien sind Teil des im nächsten Abschnitt behandelten Netzressourcenmanagements.

### 3.2.4 Verwaltung von Netzressourcen

Bei der in diesem Abschnitt behandelten Netzressourcenverwaltung (*network resource management*) geht es hauptsächlich um die Verwaltung von Bandbreite auf Verbindungsabschnitten. Sie unterscheidet sich von der in Abschnitt 3.2.1.1 vorgestellten Art der Bandbreitenverwaltung durch die Betrachtung der Verbindungsebene<sup>6</sup> anstelle der Paketebene. Gleichzeitig sind in der Regel mehrere Netzknoten betroffen, wodurch die Verbindung zur Verkehrslenkung hergestellt wird.

Netzressourcenmanagement ist eng verknüpft mit der Verbindungsannahmesteuerung (siehe Abschnitt 3.2.3) im Fall von mehreren Klassen, wobei es sich dabei um Dienstgüteklassen oder um andere Formen von Aggregaten handeln kann (siehe Abschnitt 3.1.5). Im Wesentlichen geht es darum, die auf den Verbindungsleitungen zur Verfügung stehende Bandbreite so aufzuteilen, dass einerseits die Dienstgüteanforderungen auf der Paketebene (z. B. Paketverlustwahrscheinlichkeiten) eingehalten werden und andererseits die Leistung auf der Verbindungsebene, ausgedrückt durch die Blockierwahrscheinlichkeit, bestimmten Kriterien genügt. Bei der Formulierung dieser Kriterien steht wieder das gesamte Spektrum von Garantien (siehe Abschnitt 3.1.4) zur Auswahl, von der Gleichberechtigung aller Klassen auf der Verbindungsebene als Ziel über eine proportionale Abstufung von Blockierungswahrscheinlichkeiten bis hin zu festen Obergrenzen für die Blockierung in einzelnen Klassen. Die Umsetzung der Kriterien erfolgt jedoch nicht auf direktem Weg, sondern durch eine – möglicherweise dynamische – Zuweisung von Bandbreiteanteilen (*bandwidth allocation*), wofür verschiedene Verfahren existieren.

Eine einfache Strategie stellt die gemeinsame Nutzung (*complete sharing*) der gesamten Bandbreite dar, die dann allen Klassen gleichberechtigt zur Verfügung steht. Auf die resultierenden Blockierungswahrscheinlichkeiten, die in diesem Fall in Klassen mit strengeren Dienstgüteanforderungen grundsätzlich höher ausfallen, kann kein Einfluss genommen werden [148].

Dem gegenüber steht die komplette Partitionierung (*complete partitioning*) der zur Verfügung stehenden Bandbreite. Dabei erfolgt eine feste Zuweisung von Anteilen an der Gesamtbandbreite an die einzelnen Klassen. Hat eine Klasse ihren Bandbreiteanteil ausgeschöpft, müssen weitere Verbindungen abgewiesen werden; der Effekt des statistisches Multiplexens zwischen Klassen bleibt also ungenutzt. Derartige Verfahren sind eng an Strategien zur Reservierung von Verbindungsleitungen (*trunk reservation*) in Netzen mit Durchschaltvermittlung angelehnt.

Eine hybride Form des Bandbreitenmanagements ist die teilweise gemeinsame Nutzung (*partial sharing*), bei der wie im Fall der kompletten Partitionierung gewisse Bandbreiteanteile exklusiv für die einzelnen Klassen reserviert sind, daneben aber auch noch ein Bandbreitean-

---

<sup>6</sup>Zum Verbindungsbegriff in IP-Netzen siehe die entsprechenden Anmerkungen in Abschnitt 3.2.3.

teil existiert, der von allen Klassen gemeinsam genutzt werden kann. Eine Hybridlösung stellt auch die virtuelle Partitionierung (*virtual partitioning*) dar [42, 195], bei der die Schwelle für die Annahme einer neuen Verbindung vom aktuell genutzten Bandbreiteanteil in der entsprechenden Klasse abhängt. Bei derartigen Strategien spricht man auch von dynamischem Bandbreitenmanagement. Diese Verfahren ermöglichen sogar die Formulierung konkreter Anforderungen an die Blockierwahrscheinlichkeiten im Sinne relativer Garantien [148]. Weitere Beispiele für dynamische Bandbreitenzuteilung sind in [42, 160, 180] genannt.

### 3.2.5 Verkehrslenkung

Die Verkehrslenkung (*routing*) ist eine der zentralen Aufgaben der IP-Schicht. Neben der Erfüllung ihrer elementaren Funktion, nämlich für IP-Pakete den Weg zu der angegebenen Zieladresse zu finden, hat sie auch einen großen Einfluss auf die Dienstgüte und kann damit als Teil des Verkehrsmanagements gesehen werden.

Obwohl aufgrund der verbindungslosen Kommunikation bei IP die Verkehrslenkung prinzipiell für jedes Paket durchgeführt werden muss, laufen die Verkehrslenkungsalgorithmen eigentlich auf einer höheren Ebene und in weitaus größeren Zeitabständen ab, indem für (in der Regel aggregierte) Verkehrsströme ein Weg durch das Netz oder zumindest der nächste Knoten festgelegt und in den Verkehrslenkungstabellen (*routing tables*) der Netzknoten eingetragen wird. Auf der Paketebene selbst findet dann nur noch ein Weiterleiten (*forwarding*) von Paketen gemäß den Einträgen in den Verkehrslenkungstabellen der einzelnen Knoten statt. Weiterhin ist im Fall von IP-Netzen zu beachten, dass auch die Verkehrslenkung in darunter liegenden Transportnetzen (z. B. ATM- oder optisches Transportnetz) von großer Bedeutung ist und idealerweise auch bei der Verkehrslenkung in der IP-Schicht berücksichtigt werden sollte.

#### 3.2.5.1 Klassische Verkehrslenkungsverfahren in IP-Netzen

Die Verkehrslenkung in IP-Netzen ist hierarchisch organisiert und beruht auf einer Aufteilung des Netzes in so genannte autonome Systeme (*autonomous systems*, AS). Es wird grundsätzlich zwischen Verfahren unterschieden, die innerhalb eines autonomen Systems arbeiten (*interior gateway protocols*, IGP)<sup>7</sup>, und solchen, die autonome Systeme wie abgeschlossene Einheiten betrachten und nur die Verkehrslenkung zwischen diesen behandeln (*exterior gateway protocols*, EGP)<sup>8</sup> [76].<sup>8</sup>

---

<sup>7</sup> Im IP-Umfeld, wo traditionell protokolltechnische Aspekte im Vordergrund gesehen werden, wird bei der Verkehrslenkung stets von einem Verkehrslenkungsprotokoll (*routing protocol*) gesprochen, auch wenn damit nicht nur das Verfahren zur Verteilung von Verkehrslenkungsinformationen gemeint ist.

<sup>8</sup> Da als Synonym für AS auch der Begriff der Domäne (*domain*) verwendet wird, spricht man auch von *Intra-domain Routing* bzw. *Interdomain Routing* [103].

Die meisten Verfahren innerhalb beider Gruppen berücksichtigen nur die Zieladresse von IP-Paketen (*destination-based routing*) als Grundlage für eine abschnittsweise Festlegung des Weges (*hop-by-hop routing*). Im einfachsten Fall orientiert sich das Verfahren bei dieser Entscheidung ausschließlich an der Netztopologie, indem der kürzeste Weg (*shortest path*) auf Basis der Anzahl zwischen Quelle und Ziel liegender Abschnitte (*hop count*) bestimmt wird. Darüber hinaus gibt es Verfahren, die bei der Bestimmung des „kürzesten“ Weges eine Gewichtung der einzelnen Abschnitte vornehmen, z. B. mit der auf diesem Abschnitt gemessenen Verzögerung. In diesem Fall ist das Verkehrslenkungsverfahren in der Lage, sich der gegebenen Lastsituation anzupassen und Pakete über weniger belastete Abschnitte zu leiten (adaptive Verkehrslenkung). Andererseits neigen solche Verfahren zur Erzeugung von Oszillationen.

Für die Ermittlung des kürzesten Weges sowie die dazu erforderliche Verteilung von Informationen zwischen Routern gibt es wiederum verschiedene Möglichkeiten. Ältere Verfahren wie das *Routing Information Protocol* (RIP) [123, 184] aus der Gruppe der IGP-Verfahren beruhen auf der Verwendung von Distanzvektoren (*distance vectors*) in den einzelnen Routern, die als Elemente jeweils die Entfernungen (als *Hop Count* oder in gewichteter Form) zu allen anderen Routern enthalten. Der Distanzvektor eines Routers „A“ wird in periodischen Abständen an alle benachbarten Router von „A“ weitergereicht. Die Nachbarn überprüfen dann für jeden Weg, der über „A“ führt, ob sich die Distanz zum Zielknoten geändert hat, und passen ggf. die Entfernungswerte in ihrem Distanzvektor an. Wird außerdem festgestellt, dass sich die Distanz verringert, wenn der Weg zu einem bestimmten Ziel über „A“ anstatt über einen anderen Router geführt wird, erfolgt eine entsprechende Aktualisierung der Verkehrslenkungstabelle sowie des Distanzvektors, der wiederum an alle Nachbarn verschickt wird. Diese Vorgehensweise ist jedoch mit einigen Problemen verbunden, z. B. der Gefahr der Entstehung von Schleifen. Diese Probleme verschärfen sich mit zunehmender Größe des Netzes, sodass diese Klasse von Verfahren als kritisch im Hinblick auf Skalierbarkeit gilt.

Eine Alternative zur Verwendung von Distanzvektoren bieten die so genannten *Link-State*-Protokolle. Dabei sendet jeder Router Informationen über den Status der von ihm ausgehenden Abschnitte (*links*) an die anderen Router innerhalb der Domäne, die dadurch eine vollständige Sicht der Netztopologie erhalten. Auf dieser Basis kann dann jeder Router den kürzesten Weg zu jedem beliebigen Ziel innerhalb des Zuständigkeitsbereichs des Verkehrslenkungsprotokolls bestimmen, z. B. mit Hilfe des Algorithmus von Dijkstra. Der bekannteste Vertreter dieser Klasse von Verkehrslenkungsverfahren in IP-Netzen ist das OSPF-Protokoll (*open shortest path first*) aus der Gruppe der IGP-Verfahren [199]. Es bietet neben der Gewichtung von Abschnitten bei der Wegberechnung mit einer aus der Verzögerung abgeleiteten Größe auch Einsatzmöglichkeiten im Zusammenhang mit der Verkehrsdifferenzierung.

Die genannten Protokolle sowie weitere Verkehrslenkungsverfahren in IP-Netzen sind z. B. in [76, 122] beschrieben. Eine umfassende Betrachtung von Verkehrslenkungsverfahren für verschiedene Netztypen inklusive einer Klassifikation liefern [18, 257].

### 3.2.5.2 Dienstgüteorientierte Verkehrslenkung

Viele klassische Verkehrslenkungsverfahren in IP-Netzen bieten durch ihre Fähigkeit zur Anpassung an den aktuellen Netzzustand bereits eine Möglichkeit, Dienstgüte aktiv zu verbessern. Von dienstgüteorientierter Verkehrslenkung (*QoS routing*, QoSR) kann jedoch erst gesprochen werden, wenn eine Bereitstellung differenzierter Dienstgüte unterstützt wird. Einfache Formen, die eine relative Differenzierung ermöglichen, finden sich teilweise als Option in etablierten Verfahren. So bietet OSPF die Möglichkeit, den Wert des ToS-Feldes im IP-Paketkopf bei der Verkehrslenkung mit einzubeziehen und somit unterschiedliche Wege für Pakete mit gleichem Ziel, aber unterschiedlichem ToS-Wert festzulegen (*ToS routing*). Meist wird der Begriff QoSR allerdings erst im Zusammenhang mit der Gewährung absoluter Garantien verwendet, d. h. wenn verlangt wird, dass sich die Verkehrslenkung an harten Randbedingungen in Form von konkreten Dienstgüteanforderungen orientiert [80]. In diesem Fall ist QoSR als ein Spezialfall der anforderungsbezogenen Verkehrslenkung (*constraint-based routing*) zu sehen, bei der jede Route bestimmten vorgegebenen Kriterien genügen muss.

Das auf absoluten Garantien basierende QoSR berücksichtigt im einfachsten Fall nur Bandbreitanforderungen. Weiter gehende Lösungen sind in der Lage, auch Maximalwerte der Verzögerung einzuhalten. Dabei muss entsprechend den Ausführungen in Abschnitt 3.1.6 zusätzlich unterschieden werden, ob die Anforderungen nur auf einen Abschnitt bezogen (*link constraint*) oder als Ende-zu-Ende-Bedingungen (*path constraint*) interpretiert werden. Die Erfüllung dieser Anforderungen, verbunden mit dem Ziel einer effizienten Nutzung der zur Verfügung stehenden Ressourcen, führt auf sehr komplexe Optimierungsprobleme. Zusätzlich gilt es, einen Kompromiss zwischen Effizienz und Realisierungsaufwand zu finden. Daneben spielen bei der Entwicklung von QoSR-Verfahren protokolltechnische Fragen, z. B. bzgl. der Zusammenarbeit mit Reservierungsprotokollen, sowie Fragen im Zusammenhang mit der Zustandshaltung eine Rolle.

Vorschläge für QoSR existieren sowohl im Interdomain- als auch im Intradomainbereich, wobei in letzterem wiederum OSPF als Ausgangspunkt für Erweiterungen dienen kann [9]. Eine Übersicht und Klassifizierung von QoSR-Verfahren ist in [67] zu finden.

### 3.2.5.3 QoSR ohne lokale Dienstgütemechanismen in Netzknoten

QoSR ist allgemein als Ergänzung zur Verwendung entsprechender Bedien- und Pufferverwaltungsstrategien in Routern zu sehen. Wenn Dienstgüteparameter auf aggregierte Verkehrsströme bezogen werden und vor allem wenn nur eine relative Differenzierung mit wenigen

Klassen angestrebt wird, ist prinzipiell aber auch ein alternativer Einsatz unter Verzicht auf spezielle Mechanismen in den Netzknoten denkbar. So könnte man sich z. B. im Fall zweier Prioritätsklassen vorstellen, dass Pakete mit höherer Priorität ggf. über einen Alternativweg mit geringerer Belastung der einzelnen Abschnitte geführt werden, während für Pakete mit geringerer Priorität immer der Standardweg gewählt wird.

Ein Vorteil einer solchen Lösung auf der Basis von QoSR ist, dass für eine Einführung lediglich eine Anpassung der Router-Software notwendig ist, während die Mechanismen in den Knoten oft einen Eingriff in die Routerarchitektur erfordern. Allerdings gibt es neben der bereits erwähnten Komplexität der Verkehrslenkungsalgorithmen auch eine Reihe von Nachteilen, die gegen eine ausschließliche Abstützung von Dienstgütearchitekturen auf QoSR sprechen:

- Um eine dynamische Anpassung an die Lastsituation zu ermöglichen, sind häufige Routenänderungen erforderlich, was möglicherweise Oszillationseffekte nach sich zieht. Außerdem wächst die Gefahr, dass sich Pakete gegenseitig überholen, was insbesondere im Fall von TCP-Verkehr zu erheblichen Leistungseinbußen führt [28].
- QoSR erfordert ein Zusammenwirken aller Knoten im Netz oder zumindest innerhalb eines bestimmten Netzbereichs (z. B. innerhalb eines AS). Das bedeutet zum einen, dass alle Knoten QoSR inklusive der erforderlichen Protokollerweiterungen unterstützen müssen, was die Migration zu einer QoSR-basierten Netzarchitektur erschwert. Zum anderen erhöht sich das Verkehrsaufkommen, da im Vergleich zur einfachen Verkehrslenkung zusätzliche Informationen ausgetauscht werden müssen.
- Die Einrichtung unterschiedlicher Wege für Verkehr aus unterschiedlichen Klassen bedeutet zwangsläufig eine Abkehr von der Verkehrslenkung entlang des kürzesten Weges. Untersuchungen von alternativen Verkehrslenkungsverfahren in durchschaltvermittelnden Netzen haben gezeigt, dass Umwege im Hochlastfall zu einer Verschlechterung des Gesamtverhaltens führen [257]. Ähnliche Effekte sind auch beim Einsatz von QoSR – zumindest in der o. g. Verwendungsform – zu erwarten.

### **3.2.6 Entgelterhebung**

Eine charakteristische Eigenschaft des Internets war lange Zeit der für die Nutzer kostenlose Zugang an Universitäten und Forschungseinrichtungen. Mit der zunehmenden Kommerzialisierung des Internets sowie der Ausdehnung hin zu Privathaushalten einerseits und Firmen andererseits konnte dieses Prinzip allerdings nicht mehr aufrechterhalten werden. Um die Kosten für Ausbau und Betrieb des Netzes zu tragen, wurde die Erhebung von Entgelten für die Nutzung erforderlich.

Die Entgelterhebung ist seit jeher eine wichtige Komponente beim Betrieb von Kommunikationsnetzen, die allerdings in klassischen Netzen weitgehend unabhängig von anderen Verkehrs-

management-Funktionen betrachtet werden konnte. Wenn es jedoch um die Entwicklung von Dienstgüte unterstützenden Architekturen geht, muss die Entgelterhebung als integraler Bestandteil des Verkehrsmanagements gesehen werden, denn sie sorgt dafür, dass nicht jeder Nutzer maximale Dienstgüteanforderungen stellt bzw. immer die Klasse mit der höchsten Priorität wählt, was Mechanismen zur Dienstgütedifferenzierung wirkungslos machen würde. Abgesehen von ihrer Aufgabe als Verkehrsmanagement-Funktion gehören zur Entgelterhebung auch rein technische ebenso wie rein ökonomische Aspekte.

### 3.2.6.1 Begriffe

Eng verbunden mit dem Begriff der Entgelterhebung sind eine Reihe von Begriffen<sup>9</sup>, die entweder mehr die technischen oder die ökonomischen Aspekte hervorheben. Obwohl eine klare Trennung in der Literatur kaum erkennbar ist, soll im Folgenden – unter teilweiser Bezugnahme auf Definitionen in verschiedenen Quellen [57, 95, 261] – für die wichtigsten dieser Begriffe der Schwerpunkt ihrer Bedeutung angegeben werden:

- Das *Accounting* umfasst die Sammlung von Daten (z. B. Dauer einer Verbindung, übertragenes Datenvolumen), die Aufschluss über die Nutzung eines Dienstes geben und als Basis für eine Entgelterhebung wichtig sind, in nutzerbezogenen Datensätzen (*accounting records*).
- Unter *Charging* kann im engeren Sinn die Zuordnung eines Entgelts zu den beim *Accounting* erfassten Daten gesehen werden. Daneben wird *Charging* auch häufig als Überbegriff für Entgelterhebung verwendet.
- Unter *Billing* wird die Erstellung einer Rechnung aus den erfassten Entgelten verstanden („Rechnungsstellung“).
- Der Begriff des *Pricing* betont mehr die ökonomischen Gesichtspunkte und beinhaltet – insbesondere in einem Netz, das Verkehrsdifferenzierung unterstützt – auch die Aufgabe, Preise so festzulegen, dass eine effiziente Ausnutzung von Ressourcen gefördert wird. Wie *Charging* wird auch *Pricing* außerdem als Überbegriff für das Themengebiet verwendet.
- Der Begriff der Tarifierung (*Tariffing*) hat grundsätzlich eine ähnliche Bedeutung wie *Pricing*, wobei der Schwerpunkt auf einer regulatorischen Preisgestaltung liegt. Daher ist dieser Begriff auch häufig im Zusammenhang mit klassischen Telefonnetzen und monopolistischer Betreiberstruktur zu finden [261].

Wenn im Rahmen dieser Arbeit von Entgelterhebung gesprochen wird, sind damit meist die Aufgaben des *Charging* und *Pricing* zu verstehen, während Fragen des *Accounting* und *Billing* weitgehend außer Acht gelassen werden.

---

<sup>9</sup>Für die meisten dieser englischen Begriffe gibt es im Deutschen keine eindeutige und die Bedeutung klar wiedergebende Entsprechung.

### 3.2.6.2 Anforderungen

Bei der Entwicklung von Konzepten für die Entgelterhebung in Kommunikationsnetzen, insbesondere in IP-Netzen mit Dienstgüteunterstützung, sind eine Reihe von Anforderungen zu beachten, die zu drei Hauptpunkten zusammengefasst werden können:

- **Anreizkompatibilität** (*incentive compatibility*): Die Preise müssen so gestaltet sein, dass ein Nutzer einen Anreiz für ein bzgl. des Ressourcenbedarfs günstiges Verhalten hat. Das Verhalten kann das übertragene Datenvolumen, die Charakteristik des Verkehrs oder die gewählte Dienstgütekategorie umfassen. Außerdem sollte ein Anreiz bestehen, sich fair zu verhalten. Dies heißt z. B. im Fall einer Architektur, die Ressourcenreservierung mit absoluten Garantien unterstützt, korrekte und möglichst genaue Angaben über das eigene Verkehrsprofil zu machen. Dazu gehört auch, dass sich Nutzer im Fall von Datenverkehr einer Selbstbeschränkung durch Mechanismen der Überlastvermeidung, wie sie z. B. in TCP implementiert sind, unterwerfen.
- **Benutzertransparenz**: Ein Verfahren zur Entgelterhebung muss klar und leicht verständlich sein. Für den Nutzer muss nachvollziehbar sein, inwiefern sein Verhalten Auswirkungen auf das anfallende Entgelt hat bzw. welche Auswirkung die Wahl einer Preiskategorie auf die Dienstgüte hat. Zur Benutzerfreundlichkeit gehört auch eine Flexibilität bzgl. Änderungswünschen des Nutzers (z. B. andere Werte für Verkehrs- oder Dienstgüteparameter bzw. Wechsel der Dienstgüte-/Preiskategorie).
- **Realisierbarkeit**: Ein Konzept zur Entgelterhebung sollte mit geringem technischen Aufwand umsetzbar sein. Dazu gehört die Abstimmung mit den vorhandenen Architekturen und Protokollen ebenso wie die Berücksichtigung von Sicherheitsanforderungen. Darüber hinaus gilt es auch, unternehmenspolitische Randbedingungen zu berücksichtigen, z. B. wenn eine Zusammenarbeit verschiedener Netzbetreiber erforderlich ist.

Daneben existieren in IP-Netzen spezielle Probleme, wie etwa die Fragestellung, welcher Partner innerhalb einer Kommunikationsbeziehung bezahlen soll. Das aus dem Telefonnetz bekannte Prinzip, dass – abgesehen von speziellen Rufnummern – immer der Anrufer bezahlt, ist nicht ohne weiteres auf die in IP-Netzen vorhandenen Dienste übertragbar. Dies liegt in erster Linie daran, dass innerhalb des Netzes aufgrund des verbindungslosen Charakters von IP die Kommunikationsbeziehungen überhaupt nicht bekannt sind, es sei denn, es werden zusätzliche Signalisierungsprotokolle, z. B. zur Reservierung von Ressourcen, eingesetzt. Geht man zu dem Prinzip über, dass jeder Partner für die von ihm gesendeten Daten bezahlt, widerspricht dies andererseits der Philosophie vieler Internet-Dienste, insbesondere des WWW-Dienstes. Hier können Dienstanutzer Informationen abrufen, die in der Regel kostenlos auf Servern angeboten werden, und es wäre wünschenswert, dass Kosten für die Übertragung der Daten komplett von den Nutzern übernommen werden. Die Realisierung eines solchen *Receiver Pays*-

Schemas bringt jedoch manche Schwierigkeiten (z. B. zusätzliche Signalisierung, erforderliche Sicherheitsmaßnahmen) mit sich, die sich weiter verschärfen, wenn verschiedene Dienstgüteklassen (mit unterschiedlichen Entgelten) angeboten werden, aus denen sowohl Client als auch Server wählen können.

### 3.2.6.3 Pauschales Nutzungsgentgelt

Die einfachste Variante der Entgelterhebung in Kommunikationsnetzen ist die Berechnung eines pauschalen Nutzungsgentgelts (*flat rate*), d. h. Nutzer bezahlen einen monatlichen oder jährlichen Betrag, um Zugang zum Netz zu erhalten. Diese Lösung hat den Vorteil, dass auf ein *Accounting* völlig verzichtet werden kann und das *Billing* sehr einfach wird. Allerdings existieren auch eine Reihe von Nachteilen, die teilweise in der Vergangenheit dazu geführt haben, dass Anbieter von Pauschaltarifen diese wieder zurücknehmen mussten:

- Es gilt als ökonomischer Grundsatz, dass Preise stets Kosten widerspiegeln sollten. Wie MacKie-Mason und Varian in [182] zeigen, sind das neben Kosten für den Aufbau der Netzinfrastruktur sowie die Bereitstellung von Nutzerzugängen, die beide durch ein Pauschalentgelt adäquat wiedergegeben werden, vor allem Kosten für Übertragungs- und Vermittlungskapazität sowie „soziale Kosten“ im Fall einer Überlast. Diese Komponenten sind nutzungsabhängig und legen daher auch eine nutzungsabhängige Entgelterhebung nahe.
- Durch pauschale Entgelte werden grundsätzlich Nutzer benachteiligt, die im gleichen Zeitraum weniger Ressourcen beanspruchen, da sie dafür den gleichen Betrag zahlen wie Vielnutzer.
- Eine Differenzierung von Verkehr kann nur unzureichend unterstützt werden. Prinzipiell ist es zwar möglich, unterschiedliche Dienstgüteklassen mit unterschiedlichen Preisen für den monatlichen Zugang anzubieten. Jedoch bedeutet dies eine deutliche Einschränkung in der Flexibilität, da nicht dynamisch zwischen Klassen gewechselt werden kann.
- Es besteht die Gefahr, dass Nutzer zu einem verschwenderischen Umgang mit den Ressourcen veranlasst werden, indem sie z. B. permanent am Internet angeschlossen bleiben (*always on*) und damit Zugangsnetzressourcen belegen, obwohl sie den Zugang nur ab und zu nutzen, oder Informationen abrufen, die nach der Übertragung sofort wieder verworfen werden.

Letztlich lassen sich alle diese Punkte als mangelnde Anreizkompatibilität der pauschalen Entgelterhebung zusammenfassen.

### 3.2.6.4 Nutzungsabhängige Entgelterhebung

Die nutzungsabhängige Entgelterhebung (*usage-based charging*<sup>10</sup>) orientiert sich an dem durch die Nutzung tatsächlich entstehenden Ressourcenverbrauch. Dabei kann der Grad der Kopplung zwischen Entgelt und Ressourcenverbrauch in einem weiten Bereich variieren.

Aus klassischen Kommunikationsnetzen, vor allem aus Fernsprechnetzen, ist diese Form der Entgelterhebung bereits bekannt. Dort findet in der Regel zusätzlich zur Berechnung einer pauschalen Gebühr eine Entgelterhebung u. a. auf Basis der Nutzungsdauer statt. Da im Fall von (konstantratigem) Sprachverkehr die Nutzdatenmenge proportional zur Nutzungsdauer ist, werden mit diesem Ansatz sowohl der direkt von der Verbindungsdauer abhängige Teil des tatsächlichen Ressourcenverbrauchs als auch der Kostenanteil, der von der Menge der übertragenen Daten abhängt, adäquat wiedergegeben.

IP-Verkehr ist allerdings im Unterschied zu Sprachverkehr durch starke Schwankungen in der Übertragungsrate gekennzeichnet, einerseits aufgrund der Verschiedenheit der Anwendungen, andererseits begründet durch die variablen Raten, die von den Anwendungen selbst produziert werden. Damit einher geht auch eine Schwankung im Ressourcenbedarf. Eine einfache Möglichkeit, dem Rechnung zu tragen, besteht in einer Entgelterhebung mittels eines Volumenpreises auf der Basis einzelner Pakete oder Bytes. Nutzer bezahlen also ein bestimmtes Entgelt für jede übertragene Informationseinheit. Alternativ dazu existieren Konzepte, die eine Entgelterhebung auf der Grundlage der effektiven Bandbreite von Verkehrsströmen vorsehen (siehe Abschnitt 3.2.6.5).

Im Fall eines Dienstgüte unterstützenden IP-Netzes müssen außerdem die in Form von absoluten oder relativen Garantien formulierten Dienstgüteanforderungen bei der nutzungsabhängigen Entgelterhebung mit einbezogen werden. Höhere Anforderungen bedeuten grundsätzlich höheren Ressourcenbedarf und damit einen höheren Preis.

Sowohl in klassischen Telefonnetzen wie auch in IP-Netzen hängt der Wert der in Anspruch genommenen Kapazitätsressourcen, ausgedrückt durch die Gefahr der Entstehung von Überlast, stark vom Zeitpunkt der Inanspruchnahme innerhalb des Tages- und Wochenablaufs ab. Daher bietet es sich prinzipiell an, den Preis pro Zeit- oder Volumeneinheit der Nutzung entsprechend der Tageszeit oder des Wochentages zu variieren. Es ist jedoch zu beobachten, dass eine Differenzierung der Preise nach diesen Kriterien in vielen Netzen zunehmend zugunsten der Einfachheit fallen gelassen oder auf wenige Zeitzonen reduziert wird.

Eine andere Komponente, die aus klassischen Netzen bekannt ist, deren Bedeutung aber selbst dort immer weiter abnimmt, ist die Abhängigkeit des Entgelts von der Entfernung der Kommunikationspartner. Diese Komponente ist in IP-Netzen nicht vorzufinden, obwohl auch hier eine solche Abhängigkeit des Ressourcenbedarfs von der Entfernung besteht. Allerdings ist bei den

---

<sup>10</sup> Gleichbedeutende englische Begriffe sind *Usage-based Pricing* sowie *Congestion Pricing* [181].

meisten auf IP basierenden Diensten, insbesondere im Fall von WWW, dem Nutzer oft nicht bewusst, wo sich sein Kommunikationspartner befindet, da im Gegensatz zur klassischen Festnetztelefonie dessen Adresse (Domain-Name, IP-Adresse) keine eindeutigen Rückschlüsse zulässt. Es ist daher kaum möglich, mit Hilfe einer entfernungsabhängigen Komponente auf das Nutzerverhalten so Einfluss zu nehmen, dass Nutzer bevorzugt mit Partnern in geringerer Entfernung kommunizieren. Außerdem wäre eine entfernungsabhängige Entgelterhebung nur mit sehr großem Aufwand zu realisieren.

### **3.2.6.5 Lösungsansätze für nutzungsabhängige Entgelterhebung**

In diesem Abschnitt sollen einige bekannte Vorschläge zur nutzungsabhängigen Entgelterhebung in IP-Netzen vorgestellt werden. Die meisten davon berücksichtigen die Möglichkeit der Bereitstellung von differenzierter Dienstgüte bzw. können leicht dahingehend erweitert werden. Für einen tieferen Einblick in die einzelnen Verfahren sei auf entsprechende Übersichtsliteratur verwiesen [29, 82, 96, 151, 193].

#### **Paris Metro Pricing**

Das von Odlyzko vorgeschlagene *Paris Metro Pricing* (PMP) [207] orientiert sich an einem früheren Verfahren zur Entgelterhebung in der Pariser Metro. Dort wurden Sitzplätze in der ersten und zweiten Klasse angeboten, die qualitativ identisch waren und sich nur im Preis für die jeweiligen Fahrscheine unterschieden. Dadurch, dass Fahrscheine der ersten Klasse teurer waren, wurden sie seltener gekauft, wodurch die Wahrscheinlichkeit, einen Sitzplatz zu bekommen, dort größer war als in der zweiten Klasse. Das System wurde in einer Art dynamischem Gleichgewicht betrieben, das nur durch die Fahrscheinpreise sowie das in den beiden Klassen verfügbare Platzangebot gesteuert wurde.

Im übertragenen Sinn bedeutet dies, dass verschiedene Dienstgüteklassen mit unterschiedlichen Preisen für die Nutzung angeboten werden. Für jede Dienstgütekategorie existiert ein (logisch) separates Netz, z. B. mit fest zugewiesenen Bandbreiten auf jedem Übertragungsabschnitt. Zunächst werden keinerlei Garantien über die Dienstgüte in den einzelnen Klassen gegeben. Man könnte allenfalls von relativen Garantien sprechen, da zu erwarten ist, dass in der teureren Klasse eine bessere Dienstgüte erzielt wird. Ein Gleichgewicht wird sich einstellen, weil Nutzer nur solange die teurere Klasse wählen werden, wie der Qualitätsunterschied im Vergleich zur billigeren Klasse signifikant ist und die Preisdifferenz rechtfertigt.

Aufgrund seiner Einfachheit stellt dieser Vorschlag einen attraktiven Ansatz zur Unterstützung von Dienstgütedifferenzierung dar. Allerdings werden Garantien auch nur in einer sehr schwachen Form gegeben. Insbesondere ist zu beachten, dass der Gleichgewichtszustand des Systems dynamischer Natur ist, sodass transiente Vorgänge vorübergehend sogar zu einer schlechteren Qualität in der teureren Klasse führen können. Darüber hinaus ist der Zusammen-

hang zwischen der Preisdifferenz sowie der Kapazitätsaufteilung einerseits und der in den einzelnen Klassen im Mittel erzielten Dienstgüte andererseits unklar. Es bleibt daher fraglich, ob ein Verfahren mit solch geringer Vorhersehbarkeit von Nutzern akzeptiert wird.

### **Priority Pricing**

In einem Netz, das Dienstgüte in Form von relativen Garantien bereitstellt, liegt die direkte Zuordnung eines Preises zu einer Prioritätsklasse sehr nahe [39, 74, 120]. Dies ist vor allem dann sinnvoll, wenn der Prioritätswert direkt vom Nutzer eingestellt wird oder durch Nutzung einer bestimmten Applikation festliegt. Das *Charging* kann entweder über ein Entgelt pro Paket [74] oder mittels eines bereits im Voraus erworbenen Kontingents an Datenvolumen, das pro Prioritätsklasse gesendet oder empfangen werden darf [39], erfolgen.

### **Expected Capacity Pricing**

Das von Clark vorgeschlagene *Expected Capacity Pricing* [70] ist eng verbunden mit der Diff-Serv-Dienstgütearchitektur, die in Abschnitt 3.3.3 vorgestellt wird. Der Nutzer gibt dabei in einem Verkehrsvertrag eine erwartete Bandbreite an, nach der sich die Höhe des zu bezahlenden Entgelts richtet. Im Netz wird dafür gesorgt, dass Pakete dieses Nutzers bevorzugt behandelt werden, solange seine Senderate die erwartete Bandbreite nicht übersteigt, wobei allerdings keine absolute Garantie gegeben wird, dass in diesem Fall keine Paketverluste auftreten. Um festzustellen, ob Pakete bevorzugt zu behandeln sind, werden Mechanismen zur Quellflussskontrolle am Netzrand eingesetzt.

In [70] wird außerdem beschrieben, wie das Schema erweitert werden kann, um eine Entgelterhebung auf der Basis der empfangenen Pakete zu realisieren. Diese Lösung, auf die hier nicht tiefer eingegangen werden soll, eignet sich allerdings nur für TCP-Verkehr unter Verwendung von expliziter Überlastanzeige (*explicit congestion notification*, ECN).

### **Smart Market Pricing**

Eine interessante und zumindest unter ökonomischen Gesichtspunkten viel versprechende Variante der Entgelterhebung stellen Auktionsverfahren dar. Der Grundgedanke ist dabei, dass Konflikte um Ressourcen im Netz mit Hilfe einer Auktion aufgelöst werden. Der bekannteste Ansatz aus dieser Gruppe ist das von MacKie-Mason und Varian vorgeschlagene *Smart Market Pricing*, bei dem jedes Paket mit einem bestimmten Gebot an einer Versteigerung der zur Verfügung stehenden Ressourcen (Speicherplatz, Bandbreite) teilnimmt [182]. Die Ressourcen werden an die Pakete mit dem höchsten Gebot vergeben. Der Preis für ein Paket richtet sich allerdings nicht nach dem jeweiligen Gebot, sondern wird – für alle Pakete gleich – auf einen Wert gesetzt, der dem höchsten Gebot eines Paketes entspricht, das nicht mehr zum Zug kam.

Neben dem *Smart Market Pricing* gibt es noch andere auf dem Konzept der Auktion basierende Verfahren zur Entgelterhebung, z. B. das von Semret vorgeschlagene PSP-Auktionsverfahren (*progressive second price*) [249]. Diese Ansätze geben allesamt Anlass für Untersuchungen mit Hilfe der Optimierungs- und Spieltheorie. Fragen im Zusammenhang mit der technischen Umsetzung von Auktionsverfahren in einem realen Netz sind jedoch weithin ungeklärt.

### **Effective Bandwidth Pricing**

Im Kontext von ATM-Netzen wurden einige Verfahren vorgeschlagen, die auf der Verkehrsbeschreibung mit Hilfe von effektiven Bandbreiten beruhen [43, 78, 154, 177]. Prinzipiell lassen sich diese Ansätze aber auch auf IP-Netze mit entsprechenden Reservierungsmöglichkeiten (siehe Abschnitt 3.3.2) übertragen.

Am meisten Beachtung unter diesen Lösungsvorschlägen haben die im Rahmen des Projektes CA\$hMAN entwickelten Ansätze gefunden [78, 154]. Dabei geht es nicht nur darum, den Einfluss der Verkehrs-Charakteristik (insbesondere der Spitzigkeit des Verkehrs) auf den Ressourcenbedarf zu berücksichtigen, sondern auch einen Kompromiss zwischen einer Ableitung des Preises aus den deklarierten Verkehrsparametern und einer Entgelterhebung auf der Basis des tatsächlich erzeugten Verkehrsvolumens zu schaffen. Hintergrund ist das Problem, dass die im Verkehrsvertrag angegebenen Parameter (z. B. Spitzenrate, mittlere Rate) oft sehr konservativ gewählt sind, weil sie durch die Quellflusskontrolle überwacht werden. Die tatsächlich auftretenden Werte, z. B. für die mittlere Rate, sind meist deutlich niedriger. Eine Entgelterhebung, die sich allein an gemessenen Werten orientiert, würde jedoch keinen Anreiz schaffen, die Verkehrsparameter, die von der Verbindungsannahmesteuerung benutzt werden, möglichst genau anzugeben.

Die Entgelterhebung auf Basis effektiver Bandbreiten bringt einige Probleme mit sich. An erster Stelle ist hier die mathematische Komplexität zu nennen, die sich hinter dem Begriff der effektiven Bandbreite verbirgt und die einem durchschnittlichen Nutzer kaum zu vermitteln ist. Darüber hinaus ist auch die Realisierung aufwändig, insbesondere wenn dem Nutzer die aktuellen Preise vom Netz mitgeteilt werden müssen, wie es einige der Vorschläge vorsehen.

## 3.3 Ansätze und Architekturen zur Dienstgüteunterstützung in IP-Netzen

### 3.3.1 Dienstgüteunterstützung unterhalb der IP-Schicht

Eine nahe liegende Möglichkeit der Dienstgüteunterstützung in IP-Netzen ist die Nutzung von Verkehrsmanagement-Funktionalität einer darunter liegenden Protokollschicht. Dabei bietet sich besonders das als Übermittlungstechnik des Breitband-ISDN (*integrated services digital network*) konzipierte ATM an. Obwohl die Ursprünge in der synchronen Zeitmultiplextechnik liegen, stellt ATM im Kern ein Paketvermittlungsverfahren dar, bei dem sich Pakete durch eine konstante Länge auszeichnen und als Zellen bezeichnet werden. Eine weitere charakteristische Eigenschaft ist die verbindungsorientierte Kommunikation über virtuelle Kanäle (*virtual channels*, VCs), die zu virtuellen Pfaden (*virtual paths*, VPs) zusammengefasst werden können.

Neben dem Vermittlungsverfahren selbst umfasst ATM die Definition zahlreicher Protokolle zur Nutzdatenübertragung ebenso wie zur Signalisierung. Außerdem wurde ein Rahmen für das Verkehrsmanagement spezifiziert, der eine Dienstgüteunterstützung ermöglicht [16, 138]. Teilweise wurden sogar die Verkehrsmanagementfunktionen selbst standardisiert, was den Einsatz in einem heterogenen Umfeld prinzipiell attraktiv macht.

#### 3.3.1.1 ATM-Dienstklassen

In ATM-Netzen erfolgt die Bereitstellung von Dienstgüte zumeist in Form von absoluten Garantien. Entsprechende Dienstgüteparameter werden zusammen mit Verkehrsparametern in einem Verkehrsvertrag beim Verbindungsaufbau festgelegt. Um den Anforderungen unterschiedlicher Anwendungen gerecht zu werden, wurden eine Reihe von Dienstklassen definiert. In Tabelle 3.1 sind die vom ATM Forum in [16] spezifizierten Dienstklassen (*service categories*) aufgelistet; eine ähnliche Unterteilung in so genannte Transferklassen (*transfer capabilities*) wurde auch von der ITU vorgenommen [138].

Als Dienstklassen für Echtzeitverkehr sind CBR (*constant bit rate*) und rt-VBR (*real-time variable bit rate*) vorgesehen. Der Verkehr wird dabei durch eine Spitzenzellrate (*peak cell rate*, PCR) beschrieben, zu der im Falle von rt-VBR außerdem eine mittlere durchsetzbare Rate (*sustainable cell rate*, SCR) sowie eine maximale Bündelgröße (*maximum burst size*, MBS) hinzukommt. Werden die Verkehrsparameter eingehalten, so wird garantiert, dass die im Verkehrsvertrag festgelegten Werte für die maximale Zellverzögerung (*maximum cell transfer delay*, MaxCTD), die Verzögerungsschwankung (*cell delay variation*, CDV) sowie die relative Häufigkeit von Zellverlusten (*cell loss ratio*, CLR) nicht überschritten werden. Im Unterschied zu rt-VBR werden bei der ursprünglich für Datenverkehr vorgesehenen Dienstklasse nrt-VBR (*non-real-time variable bit rate*) keine Garantien bezüglich CTD und CDV gegeben.

**Tabelle 3.1:** ATM-Dienstklassen nach der Definition des ATM Forum

Dienstklasse	typische Verkehrseigenschaften	Garantien
CBR	Echtzeitverkehr mit konstanter Rate	CLR, MaxCTD, CDV
rt-VBR	Echtzeitverkehr mit variabler Rate	CLR, MaxCTD, CDV
nrt-VBR	Nichtechtzeitverkehr mit variabler Rate	CLR
ABR	adaptiver Datenverkehr	MCR, ACR
GFR	Datenverkehr (v. a. TCP/IP)	MCR
UBR	Datenverkehr ( <i>Best Effort</i> )	keine

Für Datenverkehr stehen neben nrt-VBR hauptsächlich die Dienstklassen ABR (*available bit rate*), GFR (*guaranteed frame rate*) und UBR (*unspecified bit rate*) zur Verfügung. Während bei UBR keinerlei Garantien gegeben werden, sorgt bei ABR eine in den Endpunkten eingerichtete und von den Netzknoten unterstützte Flusssteuerung dafür, dass neben einer garantierten minimalen Zellrate (*minimum cell rate*, MCR) die von den Klassen CBR und VBR nicht genutzte Bandbreite dynamisch in Form einer verfügbaren Zellrate (*available cell rate*, ACR) bereitgestellt wird. Die Flusssteuerung basiert auf einem Rückkopplungsmechanismus ähnlich dem von TCP realisierten. Genau dies führt aber zu Problemen beim Transport von TCP/IP-Verkehr über ATM-ABR, da die Mechanismen von ABR und TCP interferieren, was die Leistungsfähigkeit stark beeinträchtigen kann. Bei GFR wird speziell der Problematik Rechnung getragen, dass Bandbreitanforderungen nicht unbedingt auf der Zellebene vorliegen, sondern auf einer übergeordneten Rahmenebene. Genauer ausgedrückt bedeutet dies, dass GFR versucht, Rahmen der ATM-Adaptionsschicht (*ATM adaptation layer*, AAL) vom Typ 5 als Ganzes zu behandeln. Außerdem verzichtet GFR im Vergleich zu ABR auf eine mit hoher Komplexität in den Endgeräten verbundene Flusssteuerung und geht davon aus, dass eine solche auf einer höheren Schicht implementiert ist. Damit zielt GFR insbesondere auf den effizienten Transport von TCP/IP-Verkehr über ein ATM-Netz ab.

### 3.3.1.2 Verkehrsmanagement-Funktionen in ATM

Das Verkehrsmanagement in ATM-Netzen umfasst sämtliche der in Abschnitt 3.2 genannten Funktionen. Da diese bereits in anderen Literaturquellen [53, 160] ausführlich beschrieben wurden, sollen hier nur einige besonders für ATM charakteristische Mechanismen herausgegriffen werden.

Die in Abhängigkeit von der Dienstklasse im Verkehrsvertrag anzugebenden Verkehrsparameter werden mit Hilfe des in [16] spezifizierten generischen Zellratenalgorithmus (*generic cell rate algorithm*, GCRA) überwacht, der in dem bereits in Abschnitt 3.2.2 erwähnten LB-Mechanismus eine spezielle Ausprägung findet. Bei mangelnder Konformität des Verkehrs-

stroms werden die Zellen mit Hilfe eines Prioritätsbits (*cell loss priority*, CLP) im Zellkopf markiert, um im Überlastfall verworfen werden zu können.

Innerhalb eines Netzknotens erfolgt häufig eine separate Pufferung der Zellen einzelner Verbindungen (*per VC queuing*), denen z. B. mit Hilfe von WFQ-Scheduling Anteile an der Linkbandbreite zugewiesen werden. Daneben muss bei Pufferung und Scheduling auch eine Unterscheidung nach Dienstklassen vorgenommen werden. Bei der Pufferung von Zellen, die zu einer Datenverbindung gehören (insbesondere im Zusammenhang mit den Dienstklassen UBR und GFR), hat es sich als effizient erwiesen, beim Verwerfen von Zellen auch die Zugehörigkeit zu einem AAL-Rahmen zu berücksichtigen, da immer der gesamte Rahmen (der z. B. einem TCP/IP-Paket entspricht) wiederholt werden muss, wenn mindestens eine Zelle daraus verloren gegangen ist. Dabei kommen die in Abschnitt 3.2.1.2 erwähnten selektiven Verwerfungsmechanismen wie EPD, PPD oder pPPD zum Einsatz.

Als Verkehrslenkungsprotokoll innerhalb des Netzes wurde vom ATM Forum das PNNI-Protokoll (*private network-network interface*<sup>11</sup>) definiert [15]. Es handelt sich dabei um ein *Link-State*-Protokoll mit stark hierarchischem Ansatz, d. h. es findet eine logische Zuordnung von Knoten zu Gruppen (*peer groups*) auf unterschiedlichen Hierarchie-Ebenen statt, was dem PNNI-Ansatz grundsätzlich eine große Flexibilität verleiht. Andererseits führt allerdings die Menge an Informationen, die zwischen Knoten ausgetauscht werden, zu Schwierigkeiten im Hinblick auf die Anwendbarkeit für große Netze. Im Übrigen definiert PNNI neben der Signalisierung zum Auf- und Abbau von Verbindungen nur den Austausch von Topologieinformationen zum Zwecke der Verkehrslenkung; ein standardisiertes Verkehrslenkungsschema für PNNI existiert nicht.

### 3.3.1.3 Ansätze und Probleme einer Nutzung für IP-Verkehr

Die Nutzung von ATM zum Transport von IP-Verkehr wirft eine Reihe von Fragen und Problemen auf, die zunächst protokolltechnischer Natur sind. Zunächst ist festzustellen, dass die Verwendung von Zellen mit fünf Byte Steuerinformation und 48 Nutzbytes einen nicht vernachlässigbaren Protokoll-Overhead mit sich bringt.<sup>12</sup> Hinzu kommt die Notwendigkeit der Aufteilung von IP-Paketen, deren Größe sich häufig an der Rahmengröße von LAN-Protokollen wie Ethernet orientiert, in Zellen und die Zusammensetzung des Paketes auf der Empfängerseite (*segmentation and reassembly*, SAR). Schwierigkeiten verursacht jedoch hauptsächlich die Diskrepanz zwischen dem verbindungslosen Charakter von IP und dem verbindungsorientierten Paradigma von ATM.

---

<sup>11</sup> PNNI steht gleichzeitig als Abkürzung für *Private Network Node Interface* [15].

<sup>12</sup> Der Protokoll-Overhead ist tatsächlich noch größer, da auch die Steuerinformation der ATM-Anpassungsschicht berücksichtigt werden muss.

Zur Lösung dieses Problems existieren eine Reihe von Ansätzen wie das von der IETF propagierte CLIP (*classical IP over ATM*) [170] oder die vom ATM Forum kommenden Vorschläge LANE (*LAN emulation*) [14] und MPOA (*multi-protocol over ATM*) [17]. Diese zielen mehr oder weniger alle darauf ab, durch Einführung von Protokoll-Zwischenschichten und zusätzlichen Netzkomponenten den Transport von IP-Verkehr über ATM ohne signifikante Änderungen an existierenden IP-Protokoll-Implementierungen zu realisieren. Erfolgt dabei, wie ursprünglich vorgesehen, der Zugang zum ATM-Netz bereits im Endgerät, erfordert dies allerdings entweder eine Verwaltung von sehr vielen gleichzeitig existierenden ATM-Verbindungen oder einen häufigen Auf- und Abbau von ATM-Verbindungen, was aufgrund der Signalisierung eine große Belastung der ATM-Komponenten bedeutet. Außerdem kann die Möglichkeit der Dienstgütedifferenzierung, die ATM prinzipiell bietet, auf diese Weise kaum genutzt werden. Eine vergleichende Darstellung der genannten Verfahren findet sich z. B. in [53, 68].

Einen völlig anders zu bewertendes Anwendungsgebiet stellt der Einsatz von ATM als reine Transportnetztechnologie im Kernnetz dar (vgl. auch Bild 2.1). Dabei wird vor allem die Effizienz von ATM bei der Vermittlung von Dateneinheiten ausgenutzt, auf eine hohe Dynamik bzgl. des Auf- und Abbaus von Verbindungen jedoch weitgehend verzichtet. Dies geht bis hin zur Verwendung permanenter VC-Verbindungen (*permanent virtual channel connections*, PVCs) anstelle geschalteter Verbindungen (*switched virtual channel connections*, SVCs). Die Zuordnung von IP-Paketen zu virtuellen ATM-Kanälen findet am Rand des Kernnetzes statt. Eine ATM-Verbindung dient dann zum Transport von aggregierten IP-Verkehrsströmen. Die Nutzung der ATM-Verkehrsmanagement-Funktionalität ist bei diesem Ansatz eher möglich, wobei sich Verkehrs- und Dienstgüteparameter immer auf ein Verkehrsaggregat und nicht auf einzelne IP-Verkehrsflüsse beziehen. Allerdings besteht die Gefahr, dass einzelne Verkehrsmanagement-Funktionen sowohl in der ATM-Schicht als auch in darüber liegenden Netzwerkschichten vorhanden sind und sich negativ beeinflussen. Beispiele hierfür sind die Verkehrslenkung sowie die Überlastkontrolle im Fall von TCP über ATM-ABR.

### 3.3.1.4 Multi-Protocol Label Switching

Eine Weiterentwicklung des im letzten Abschnitt genannten Konzeptes führte – über diverse Zwischenstufen – zur Entwicklung von MPLS (*multi-protocol label switching*) [241, 270, 272]. Diesem liegt ähnlich wie ATM der Gedanke zu Grunde, die Vermittlung von Paketen durch die Verwendung eines vordefinierten Kurzbezeichners (*label*), welcher im VCI (*virtual channel identifier*) und VPI (*virtual path identifier*) im ATM-Zellkopf seine Entsprechung hat, effizient zu gestalten. Im Gegensatz zu ATM handelt es sich jedoch bei MPLS im Kern um eine sehr schlanke und leicht integrierbare Zwischenschicht unterhalb der IP-Schicht. Die *Labels* werden lediglich dem IP-Paketkopf vorangestellt. Andererseits bietet MPLS mit der Option einer beliebig tiefen Schachtelung von *Labels* (*label stacking*) sogar weiter gehende Möglich-

keiten zur Hierarchiebildung, als dies ATM mit seinen virtuellen Kanälen und virtuellen Pfaden erlaubt.

Ähnlich wie bei ATM müssen auch im Falle von MPLS Pfade durch das Kernnetz festgelegt werden, die hier als *Label Switched Paths* (LSPs) bezeichnet werden. Dazu wird mit Hilfe eines geeigneten Signalisierprotokolls (*label distribution protocol*, LDP) jedem MPLS-Router (*label switch router*, LSR) entlang des Pfades mitgeteilt, welche *Labels* verwendet werden sollen und wie in den Randknoten die Abbildung zwischen IP-Verkehrsströmen und *Labels* zu erfolgen hat.

Die Festlegung der Pfade stellt – in Verbindung mit der Bildung so genannter *Forwarding Equivalency Classes* (FECs) – bei MPLS das Hauptinstrument des Verkehrsmanagements dar. Hierbei können auch Dienstgüteanforderungen berücksichtigt werden, was wie in den meisten Fällen bei QoSR letztlich auf ein komplexes Optimierungsproblem führt. Gerade im Zusammenhang mit MPLS hat sich dafür der Begriff des *Traffic Engineering* herausgebildet, hinter dem sich – abweichend von seiner ursprünglichen Bedeutung als Technik der Netzdimensionierung mit Mitteln der Verkehrstheorie – ein stark auf die Verkehrslenkung ausgerichtetes Verkehrsmanagement verbirgt.

### 3.3.2 Die Integrated-Services-Architektur

Parallel zur Definition von ATM und der Standardisierung seiner Verkehrsmanagement-Funktionen und Protokolle gab es seit Anfang der 1990er Jahre Bestrebungen in der IETF, eine Architektur zur Dienstgüteunterstützung in IP-Netzen zu entwickeln. Auslöser hierfür war die v. a. im Zusammenhang mit der Durchführung von Multimediakonferenzen über das Multicast-Testnetz *MBone* [164] gewonnene Erkenntnis, dass Echtzeitkommunikation (Audio, Video) in einem IP-Netz, das allein einen Best-Effort-Dienst anbietet, nur mit erheblichen Qualitätseinbußen möglich ist.

In der innerhalb der IETF gegründeten Arbeitsgruppe *Integrated Services* (IntServ) wurde in der Folgezeit ein Rahmen für eine gleichnamige Architektur geschaffen, die eine Bereitstellung von Dienstgüte insbesondere für die genannte Art von Anwendungen ermöglicht. Dieser Rahmen umfasst in erster Linie das in [47] definierte Dienstmodell (siehe Abschnitt 3.3.2.1), das mit geeigneten Verkehrsmanagementfunktionen umgesetzt werden kann (siehe Abschnitt 3.3.2.2). Der Kerngedanke hierbei ist – ähnlich wie in ATM – die Reservierung von Ressourcen für einzelne IP-Verkehrsflüsse (*per-flow reservation*) basierend auf einem Verkehrsvertrag. Dazu wurde das Reservierungsprotokoll RSVP (*resource reservation protocol*) entwickelt, ein Signalisierprotokoll, mit dessen Hilfe Ende-zu-Ende-Reservierungen durchgeführt werden können (siehe Abschnitt 3.3.2.3). Einen umfassenden Überblick über die IntServ-Architektur bieten neben anderen [132, 272, 273].

### 3.3.2.1 Dienstmodell

Die IntServ-Architektur sieht neben dem *Best Effort Service* (BE) einen Dienst mit absoluten Garantien (*Guaranteed Service*, GS) sowie einen Dienst mit kontrollierter Netzauslastung (*Controlled-Load Service*, CL) vor.

#### Guaranteed Service

Der *Guaranteed Service* wendet sich an Anwendungen mit harten Echtzeitbedingungen wie z. B. Videokonferenzapplikationen. Die Anwendungen müssen zunächst in der Lage sein, den von ihnen erzeugten Verkehr vorab zu charakterisieren. Die Verkehrsbeschreibung (*traffic specification*, TSpec) orientiert sich dabei an dem in Abschnitt 3.2.2 genannten LB-Modell und umfasst folgende Parameter:

- Spitzenrate (*peak rate*)  $R_{Peak}$
- Abflussrate (*bucket rate, token rate*)  $R_{LB}$
- Kapazität (*bucket depth*)  $S_{LB}$
- kleinste überwachbare Datagrammgröße (*minimum policed unit*)
- maximale IP-Datagrammgröße (*maximum datagram size*)

Die Parameter  $R_{Peak}$ ,  $R_{LB}$ , und  $S_{LB}$  entsprechen – abgesehen vom hier geltenden Bezug auf IP-Pakete anstelle von Zellen – den Parametern PCR, SCR und der aus der MBS ableitbaren *Burst Tolerance* (BT) bei ATM-VBR (siehe Abschnitt 3.3.1.1), was die große Ähnlichkeit von IntServ mit ATM unterstreicht.

Neben der Verkehrsbeschreibung muss beim garantierten Dienst eine Angabe über den Bedarf an Ressourcen (*reservation specification*, RSpec) erfolgen. Die RSpec enthält die gewünschte Bandbreite  $R_{Res}$  sowie einen Toleranzwert (*slack term*) für die Verzögerung von Paketen. Eine direkte Angabe über die maximale Paketverzögerung ist nicht vorgesehen. Jedoch kann für  $R_{Peak} \geq R_{Res} \geq R_{LB}$  ein Maximalwert für die Verzögerungskomponente, die am ehesten Schwankungen unterworfen ist, nämlich die Wartezeit  $T_Q$  in den durchlaufenen Puffern (*queueing delay*), mit Hilfe eines Flüssigkeitsflussansatzes aus den TSpec- bzw. RSpec-Parametern abgeschätzt werden:

$$T_Q \leq \frac{S_{LB}}{R_{Res}} \cdot \frac{R_{Peak} - R_{Res}}{R_{Peak} - R_{LB}} \quad (3.7)$$

Eine Erhöhung von  $T_Q$  ergibt sich, wenn wie in [250] angedeutet, außerdem eine zusätzliche Verzögerung in den Routern sowie der diskrete Charakter des Paketstroms berücksichtigt wird. Eine Aussage über die gesamte Verzögerung inklusive Ausbreitungsverzögerung sowie Verarbeitungszeit beim Netzzugang oder im Endgerät erlaubt das GS-Modell allerdings nicht.

Auf der Basis von TSpec und RSpec werden dann mit Hilfe von RSVP (siehe Abschnitt 3.3.2.3) auf den Routern entlang des Ende-zu-Ende-Pfades Ressourcenreservierungen vorgenommen, sofern die Router die Reservierungswünsche akzeptieren. Das heißt, es wird garantiert, dass die Pakete in jedem Knoten mit der in der RSpec angegebenen Rate bedient werden und dass außerdem keine Pakete verloren gehen.

### **Controlled-Load Service**

Die in [276] enthaltene Definition des *Controlled-Load Service* ist weitaus unschärfer als die des garantierten Dienstes. Grundsätzlich sollen die Nutzer stets den Eindruck eines schwach belasteten Netzes haben. Da jedoch im Gegensatz zum *Guaranteed Service* keinerlei Dienstgütegarantien gegeben werden, ist die eigentliche Bedeutung die eines Dienstes, der gegenüber BE eine „bessere“ Dienstgüte bietet (*better than best effort service*). Insofern folgt der *Controlled-Load Service* dem Konzept der relativen Dienstgüte (siehe Abschnitt 3.1.4.2). Eine etwas konkretere Forderung stellt die Aussage in [276] dar, dass beim *Controlled-Load Service* die durchschnittliche Verzögerung sowie die Verluste innerhalb eines Zeitraumes, der deutlich größer ist als die „Burstdauer“ (gegeben durch  $S_{LB}/R_{LB}$ ), sehr gering sein müssen.

Während eine RSpec entfällt, muss wie beim *Guaranteed Service* der erzeugte Verkehr mit Hilfe der oben erläuterten TSpec beschrieben werden. Damit wendet sich dieser Dienst an Applikationen, die in der Lage sind, eine solche Verkehrsbeschreibung zu liefern, jedoch ohne harte Garantien auskommen können, z. B. an „tolerante“ Multimediaanwendungen mit weichen Echtzeitbedingungen.

### **3.3.2.2 Verkehrsmanagementfunktionen**

Da die IntServ-Architektur auf einem Verkehrsvertrag für einzelne Verkehrsflüsse beruht, sind neben Pufferbedienstrategien zur Umsetzung der Dienstgütedifferenzierung in den Knoten Quellflusskontrolle und Verbindungsannahmesteuerung unverzichtbare Bestandteile des Verkehrsmanagements in IntServ-Netzen. Darüber hinaus sind noch Mechanismen zur Signalisierung von Reservierungswünschen (siehe Abschnitt 3.3.2.3) sowie zur Klassifizierung von Paketen notwendig, die in engem Zusammenhang mit den Verkehrsmanagement-Funktionen stehen, in [47] sogar als solche bezeichnet werden.

### **Scheduling und Puffermanagement**

Zur Erfüllung der Anforderungen von GS-Verkehrsflüssen sowie einer Besserbehandlung von CL-Verkehr gegenüber BE-Verkehr müssen in den IntServ-Knoten geeignete Scheduling-Mechanismen eingesetzt werden [72]. Insbesondere die strikten Anforderungen von GS-Verkehr erfordern dabei ein aufwändiges *Per-Flow Queueing*, d. h. für jeden Verkehrsfluss wird eine separate (logische) Warteschlange eingerichtet. Für die Bedienung der Warteschlangen wurden eine Reihe von Mechanismen vorgeschlagen, die alle mehr oder weniger in der Lage

sind, für ein vorgegebenes Verkehrsprofil auf Basis der TSpec-Parameter deterministische Leistungsgrößen zu liefern, insbesondere einen Maximalwert für die Verzögerung [279].

Ein großer Teil der Vorschläge basiert auf dem GPS-Prinzip. Der Gewichtungsfaktor  $\phi_i$  für eine Warteschlange  $i$  kann dabei im einfachsten Fall direkt aus der zur Verfügung stehenden Link-Bandbreite  $C$  und der mittels RSpec reservierten Bandbreite  $R_{Res,i}$  abgeleitet werden:

$$\phi_i = \frac{R_{Res,i}}{C}. \quad (3.8)$$

Solange gewährleistet ist, dass die Summe der Gewichte nicht größer als eins ist, erhält jeder Fluss mindestens die von ihm reservierte Rate. Die maximale Verzögerung folgt dann näherungsweise Gleichung (3.7). Allerdings fügen reale Scheduler-Implementierungen zusätzliche Verzögerung ein, sodass die maximale Paketverzögerung gegenüber (3.7) einen höheren Wert annimmt. In diesem Punkt unterscheiden sich auch die verschiedenen Scheduler-Implementierungen wie WFQ, WF<sup>2</sup>Q oder SCFQ (siehe Abschnitt 3.2.1.1). Eine Alternative zu den GPS-basierten Scheduling-Mechanismen stellen fristenbasierte Mechanismen nach dem Vorbild von EDD (z. B. *Delay-EDD* oder *Jitter-EDD*) dar, bei denen die Maximalwerte für Verzögerung oder Verzögerungsschwankung direkt als Parameter angegeben werden können [279]. Daneben spielen hier besonders auch ratengesteuerte Lösungen wie RCSP, das einen direkten Bezug zu den vereinbarten Verkehrsparametern herstellt, eine Rolle.

Für die Bedienung von CL-Paketen können ebenfalls separate Puffer für jeden Verkehrsfluss verwendet werden. Da jedoch hier keine konkreten Bandbreiteanforderungen formuliert werden, können die Scheduler-Parameter (z. B. die Gewichte bei FQ-Scheduling) lastabhängig gewählt werden, sodass die Anforderungen des GS-Verkehrs nicht verletzt werden und trotzdem eine Besserstellung im Vergleich zu BE-Verkehr gegeben ist. Alternativ besteht die Möglichkeit der Zusammenfassung des CL-Verkehrs zu einem Verkehrsstrom. Eine denkbare Umsetzung des zuletzt genannten Ansatzes ist die Verwendung eines FQ-Schedulers für alle drei Dienstklassen BE, GS und CL. Dabei muss dem aggregierten CL-Verkehr ein Gewichtungsfaktor

$$\phi_{CL} = 1 - \sum_{i \in GS} \phi_i \quad (3.9)$$

also das Restgewicht nach Abzug der summierten Gewichtungsfaktoren für GS-Verkehrsflüsse zugewiesen werden. Der BE-Verkehr erhält das Gewicht 0, wodurch eine Bedienung nur erfolgt, wenn keine GS- oder CL-Pakete vorhanden sind.

Eine andere Realisierungsmöglichkeit stellt ein hierarchischer Ansatz ähnlich dem in [1] vorgestellten dar, bei dem auf der obersten Ebene eine statische Priorisierung mit höchster Priorität für GS-Verkehr, mittlerer Priorität für CL-Verkehr und niedriger Priorität für BE-Verkehr

vorgenommen wird und auf einer zweiten Ebene eine Isolation einzelner GS-Verkehrsflüsse mit Hilfe eines ratenbasierten Schedulers stattfindet.

Zu der Reservierung von Bandbreite kommt als weitere Aufgabe die Bereitstellung von Speicherplatz in den Netzknoten hinzu. Insbesondere muss standardgemäß für GS-Verkehr garantiert werden, dass konforme Pakete nicht verloren gehen. Dies kann allerdings nur erreicht werden, wenn für jeden GS-Verkehrsfluss Pufferplatz entsprechend dem jeweiligen TSpec-Parameter  $b$  reserviert wird. Bei einer großen Zahl von GS-Flüssen würde das einen enorm hohen Speicherbedarf nach sich ziehen, sodass die Forderung absoluter Verlustfreiheit im Hinblick auf die praktische Umsetzung unrealistisch erscheint.

### **Quellflusskontrolle**

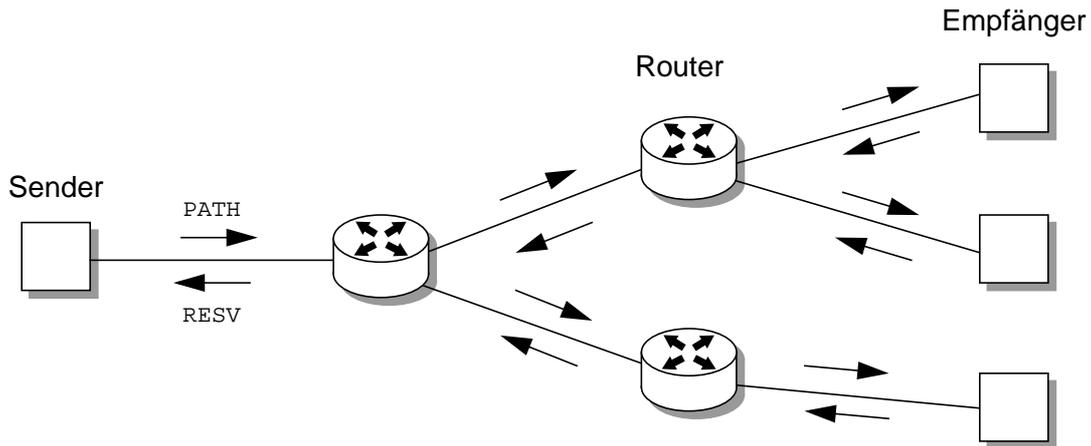
Für die Quellflusskontrolle muss am Netzzugang für jeden Verkehrsfluss ein LB implementiert sein, der mit den bei der Einrichtung einer Reservierung festgelegten TSpec-Parametern konfiguriert wird. Sowohl bei GS als auch bei CL werden Pakete, die nicht konform zur Verkehrsvereinbarung sind, markiert und anschließend weitgehend wie Pakete der BE-Klasse behandelt. Im Falle von nicht konformen CL-Paketen ist darüber hinaus zu beachten, dass laut [276] diese den meist TCP-basierten und damit adaptiven BE-Verkehr nicht unfair beeinflussen dürfen. Dies bedeutet, dass solche Pakete gegenüber BE-Paketen ggf. benachteiligt werden müssen.

### **Verbindungsannahmesteuerung**

Die Verbindungsannahmesteuerung, die bei einem Reservierungswunsch angestoßen wird, muss dafür sorgen, dass nur so viele Verkehrsflüsse akzeptiert werden, dass die Bandbreite- und Speicherplatzanforderungen von GS-Verkehrsflüssen, für die bereits Reservierungen durchgeführt wurden, nicht verletzt werden. Darüber hinaus kann es auch – in Abhängigkeit vom eingesetzten *Pricing*-Modell – sinnvoll sein, den Echtzeitverkehr zu begrenzen, damit noch Ressourcen für BE-Verkehr übrig bleiben [72]. Bei CL-Verkehrsflüssen muss entschieden werden, ob die für den *Controlled-Load Service* geltenden, in Abschnitt 3.3.2.1 genannten Kriterien bei einer Annahme noch erfüllt sind. Insbesondere in diesem Fall bietet sich die Einbeziehung von Messungen in die Annahmeentscheidung an [72, 105, 145].

#### **3.3.2.3 Reservierungsprotokoll**

Aufgrund des verbindungslosen Ansatzes von IP ist es nicht möglich, den Netzknoten direkt in der IP-Schicht Informationen und Anforderungen über einen beginnenden Verkehrsfluss mitzuteilen. Daher wurde von der IETF das entweder direkt auf IP oder auf UDP aufsetzende *Resource Reservation Protocol* (RSVP) definiert [48], das sowohl in den Kommunikationsendpunkten als auch in den dazwischen liegenden Netzknoten, in denen Ressourcen reserviert werden sollen, implementiert sein muss.



**Bild 3.7:** Ablauf einer Ressourcenreservierung mit RSVP [273]

RSVP ist im Wesentlichen durch folgende Eigenschaften charakterisiert:

- Reservierungen sind unidirektional, d. h. sie gelten nur für den Pfad vom Sender zum Empfänger. Ressourcen in der Rückrichtung müssen ggf. separat reserviert werden.
- RSVP arbeitet empfängerorientiert. Die Signalisierung wird zwar vom Sender initiiert; die Festlegung des Ressourcenbedarfs und die Durchführung der eigentlichen Reservierung erfolgen jedoch durch den Empfänger. Dieses Prinzip macht RSVP besonders geeignet für Multicast-Anwendungen.
- RSVP basiert auf dem Konzept „weicher“ Verbindungszustände (*soft states*), d. h. es wird nur der Aufbau einer Kommunikationsbeziehung/Reservierung durch entsprechende Signalameldungen angezeigt, während ein expliziter Verbindungsabbau entfällt. Reservierungen enden automatisch nach einer gewissen Zeit, wenn sie nicht aufgefrischt werden.

Die grundsätzliche Arbeitsweise von RSVP ist in Bild 3.7 dargestellt. Ein Reservierungsvorgang beginnt, indem der Sender eine PATH-Meldung an den Empfänger schickt. Die PATH-Meldung enthält unter anderem die Verkehrsbeschreibung (TSpec) sowie ein Feld, das Auskunft über den jeweiligen Vorgängerknoten (*previous hop*, PHop) gibt. Letzteres wird in den Netzknoten zwischengespeichert und in der weitergesendeten Meldung jeweils durch die eigene IP-Adresse ersetzt. Dies ist notwendig, um Reservierungsmeldungen entlang des gleichen Weges wieder zum Sender zurückzuschicken.

Die Reservierungsmeldungen (RESV) werden vom Empfänger nach Erhalt einer PATH-Meldung sowie später in regelmäßigen Abständen erzeugt. Sie enthalten im Wesentlichen eine Verkehrsflussbeschreibung (FlowSpec), bestehend aus einer TSpec und im Fall einer GS-Reservierung einer Beschreibung der gewünschten Ressourcen (RSpec). Die TSpec wird meist der vom Sender erhaltenen entsprechen, kann prinzipiell aber auch vom Empfänger modifiziert werden. Bei Ankunft einer RESV-Meldung in einem Knoten entlang des ursprünglich durch die PATH-Meldung festgelegten Weges wird die Flussbeschreibung an ein – prinzipiell von der

RSVP-Protokollsteuerung getrenntes – Verkehrssteuerungsmodul übergeben, wo über die Annahme der Reservierung entschieden wird. Fällt die Entscheidung positiv aus, wird eine entsprechende Ressourcenreservierung vorgenommen. Die RESV-Meldung wird an den Vorgängerknoten in Richtung des Senders weitergeschickt, der Empfänger erhält optional eine Bestätigungsmeldung. Im Fall einer Ablehnung wird eine RESV\_ERR-Meldung an den Empfänger gesendet.

Über den geschilderten Ablauf hinaus besteht die Möglichkeit, in der PATH-Meldung dem Empfänger zusätzliche Informationen (AdSpec) über die Eigenschaften des Kommunikationspfades (z. B. die in Abschnitt 3.3.2.1 erwähnten zusätzlichen Verzögerungen in einzelnen Routern) mitzuteilen. Außerdem kann im Fall mehrerer Sender und Empfänger mit Hilfe eines *Sender Template* in der PATH-Meldung sowie eines FilterSpec-Elementes in der RESV-Meldung über so genannte Filter eine Zusammenfassung von Reservierungen veranlasst werden. Schließlich existieren noch weitere RSVP-Meldungen, auf die jedoch hier nicht weiter eingegangen werden soll.

Aufgrund der angesprochenen Entkopplung von RSVP und eigentlicher Verkehrssteuerung in den Netzknoten ist der Einsatz von RSVP nicht nur auf die IntServ-Architektur beschränkt. Eine entsprechend angepasste Variante von RSVP namens RSVP-TE wurde z. B. als Verteilungsprotokoll für *Labels* bei MPLS (siehe Abschnitt 3.3.1) vorgeschlagen [19]. Andererseits gilt RSVP wegen seiner vielen Optionen als sehr komplex, was zur Definition alternativer Signalisierungsprotokolle für die Ressourcenreservierung in IntServ geführt hat [215].

#### 3.3.2.4 Bewertung

Der Ansatz, Dienstgüteunterstützung direkt in der IP-Schicht mit Hilfe eines auf IP aufsetzenden Signalisierungsprotokolls zu realisieren, hat gegenüber einer Lösung wie IP über ATM den Vorteil, dass eine Schicht entfällt und somit prinzipiell eine Vereinfachung stattfindet. Außerdem sind die Mechanismen für IP-basierte Anwendungen besser zugänglich.

Das ansonsten mit ATM eng verwandte IntServ-Prinzip einer Ressourcenreservierung für einzelne Verkehrsflüsse leidet jedoch unter dem Problem mangelnder Skalierbarkeit für große Netze. Im Hinblick auf die Skalierbarkeit sind die folgenden Punkte als kritisch zu bewerten:

- Signalisieraufwand (Verarbeitung von RSVP-Meldungen)
- Speicherung von Zustandsinformationen im Verkehrssteuerungsmodul
- Paketklassifizierung
- Scheduling

Beim letzten Punkt wiegen vor allem die Notwendigkeit der Einrichtung und Verwaltung separater logischer Warteschlangen für jede Reservierung sowie das im Zusammenhang mit der

Auswahl der jeweils als nächstes zu bedienenden Warteschlange auftretende Sortierproblem schwer. Zwar gibt es, wie in [167] aufgezeigt wird, hierzu wie auch zum Problem der Paketklassifizierung effiziente Algorithmen, mit denen auch noch Tausende gleichzeitig existierender Reservierungen beherrscht werden können. Doch letztendlich konnten dadurch die grundsätzlichen Bedenken im Hinblick auf die Skalierbarkeit nicht vollkommen ausgeräumt werden.

Daneben gibt es noch weitere Gründe dafür, dass sich IntServ bis heute nicht durchsetzen konnte. Dabei sind zunächst Defizite des in Abschnitt 3.3.2.1 vorgestellten Dienstmodells zu nennen. Dieses zielt in erster Linie auf Multimedia-Gruppenkommunikation ab, während WWW-Dienste, die zum Zeitpunkt der Festlegung der IntServ-Architektur erst in ihrer Anfangsphase waren, keine besondere Berücksichtigung finden. Der wesentliche Unterschied ist hierbei die deutlich kürzere Dauer von Ende-zu-Ende-Verbindungen im Fall von WWW, die dazu führt, dass sehr häufig Reservierungen vorgenommen werden müssen.

Aber auch bzgl. Echtzeitverkehr haben die in IntServ spezifizierten Dienstklassen Schwächen. Der *Guaranteed Service* konzentriert sich auf die Bereitstellung deterministischer Verzögerungsgarantien, was im Vergleich zu statistischen Garantien einen wenig effizienten Umgang mit Bandbreiteressourcen bedeutet, zumal die meisten der heutigen Echtzeitanwendungen mit geringer Wahrscheinlichkeit auftretende Überschreitungen der vorgegebenen maximalen Verzögerung tolerieren können. Die Definition des *Controlled-Load Service* andererseits ist sehr unscharf. Dies gilt sowohl aus Nutzersicht als auch aus dem Blickwinkel von Netzbetreibern, denen konkrete Anhaltspunkte für die Verbindungsannahme fehlen.

Hinzu kommen schließlich die grundsätzlichen Probleme einer Dienstgütearchitektur mit absoluten Garantien wie die Schwierigkeit, Verkehrsprofil und Ressourcenbedarf einer Anwendung anzugeben, oder die Notwendigkeit einer durchgängigen Unterstützung in allen (oder zumindest in den durch Überlast am meisten gefährdeten) Netzknoten.

### **3.3.3 Die Differentiated-Services-Architektur**

Nachdem die genannten Schwächen des IntServ-Ansatzes, insbesondere im Hinblick auf die Skalierbarkeit, innerhalb der IETF erkannt wurden, entstand immer mehr der Wunsch nach einer einfacheren Architektur für ein Internet, das wenigstens die Möglichkeit einer grundlegenden Dienstgütedifferenzierung und damit einer Verbesserung gegenüber Best-Effort-Verkehr gibt. Daraus entstand 1997 die Arbeitsgruppe *Differentiated Services* (DiffServ), in der eine gleichnamige Rahmenarchitektur definiert wurde [27, 126, 143, 204, 205].

#### **3.3.3.1 Kerngedanken**

Der wesentliche Unterschied zu IntServ ist, dass bei DiffServ im Kernnetz aggregierte Verkehrsströme behandelt werden, d. h. innerhalb einer DiffServ-Region finden keine Reservie-

rungen für einzelne Verkehrsflüsse mit den damit verbundenen Skalierungsproblemen statt. Es werden nur wenige Verkehrs- oder Dienstgüteklassen (*forwarding classes*) unterschieden. Die Zuordnung von Verkehrsflüssen zu Verkehrsklassen wird mit Hilfe des ToS-Feldes im IP-Paketkopf (siehe Abschnitt 2.3.1) festgehalten, sodass es im Kernnetz genügt, dieses Feld auszuwerten, um eine entsprechende Behandlung des Paketes zu veranlassen.

Auf diese Weise wird erreicht, dass die mit der Unterscheidung einzelner Verkehrsflüsse verbundene Komplexität (Klassifizierung, Überwachung von Verkehrsparametern) an den Netzrand verlagert wird, während das Kernnetz relativ einfach gehalten werden kann. Der Preis für die reduzierte Komplexität im Kernnetz ist der Verzicht auf garantierte Dienstgüte für einzelne Verkehrsflüsse, wie sie bei IntServ möglich ist. Dennoch bietet DiffServ ein breites Spektrum von Möglichkeiten zur Dienstgütendifferenzierung. Einerseits kann Dienstgüte in Form absoluter Garantien bereitgestellt werden, die sich allerdings auf einen aggregierten Verkehrsstrom beziehen. Andererseits sind auch relative Garantien in verschiedenen Ausprägungen möglich.

### **3.3.3.2 Elemente der DiffServ-Architektur**

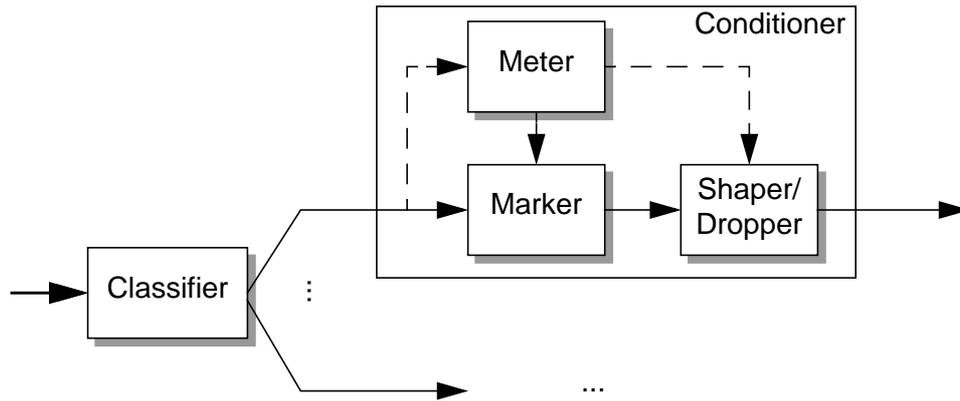
Im Rahmen zweier RFCs [27, 204] wurden die elementaren Bestandteile der DiffServ-Architektur festgelegt.

#### **Codierung der Dienstklasse im IP-Paketkopf**

Die Kennzeichnung einer Dienstklasse erfolgt bei DiffServ durch die ersten sechs Bits des ToS-Feldes im IPv4-Paketkopf bzw. des *Traffic-Class*-Feldes im IPv6-Paketkopf. Bei IPv4 war dieses Feld ursprünglich dafür vorgesehen, einerseits erhöhte Anforderungen an Verzögerung, Durchsatz und Zuverlässigkeit anzudeuten und andererseits eine Priorisierung bestimmter Anwendungstypen vorzunehmen (siehe Abschnitt 2.3.1). Es wurde aber in der Vergangenheit in realen Netzen kaum genutzt. Die genannten sechs Bits werden nun als *Differentiated Services Code Point* (DSCP) interpretiert. Von den 64 möglichen DSCP-Werten sind momentan 32 standardisiert (v. a. für die in Abschnitt 3.3.3.3 beschriebenen Weiterleitungsklassen) oder für eine zukünftige Verwendung reserviert. Die restlichen Werte sind für experimentelle Zwecke oder lokale Nutzung vorgesehen.

#### **Paketbehandlung im Kernnetz**

Durch den DSCP-Wert im Paketkopf wird die Behandlung des Pakets in jedem Knoten des DiffServ-Netzes festgelegt (*per-hop behavior*, PHB). Der aggregierte Verkehrsstrom, der sich aus der Zusammenfassung aller Pakete mit gleichem DSCP-Wert ergibt, wird als *Behavior Aggregate* (BA) bezeichnet. Die Definition eines PHBs umfasst in erster Linie Aspekte des Puffermanagements und des Scheduling. Der DSCP-Wert kann aber auch zur Richtungswahl herangezogen werden. Mit den PHBs wird zunächst nur eine lokale Differenzierung erreicht, was in einem gewissen Gegensatz zur Ende-zu-Ende-Philosophie von IntServ steht.



**Bild 3.8:** Komponenten zur Paketklassifizierung und Quellflusskontrolle in DiffServ [27]

Mehrere PHBs können logisch zu einer PHB-Gruppe (*PHB group*) zusammengefasst werden, um z. B. auszudrücken, dass diese PHBs zusammen zur Gestaltung eines bestimmten Dienstes verwendet werden.

### Dienstvereinbarungen

Die DiffServ-Architektur sieht wie andere Architekturen auch Dienstvereinbarungen (*service level agreements*, SLAs) vor. Diese stellen eine Verallgemeinerung des bekannten Verkehrsvertrags dar, indem sie auch Vereinbarungen bzgl. Sicherheit, Verfügbarkeit oder Entgelterhebung enthalten können. Als wichtigster und am häufigsten betrachteter Bestandteil eines SLAs gilt jedoch weiterhin der eigentliche Verkehrsvertrag, der bei DiffServ als *Traffic Conditioning Agreement* (TCA) bezeichnet wird und üblicherweise aus einer Verkehrsbeschreibung (z. B. auf der Basis von LB-Parametern) sowie Angaben zu Dienstgüteparametern besteht. Daneben kann noch festgelegt werden, wie nicht konforme Pakete behandelt werden, und es können Vereinbarungen zur Verkehrsformung getroffen werden.

Zu beachten ist in diesem Zusammenhang, dass sämtliche Komponenten eines SLAs optional und darüber hinaus auch nicht näher spezifiziert sind. Dies bedeutet insbesondere, dass auch DiffServ-Lösungen ohne SLAs und die nachfolgend beschriebenen Kontrollmechanismen möglich sind, was gerade im Kontext relativer Dienstgüte von Interesse ist. Ein Protokoll zur Einrichtung von SLAs (vergleichbar mit RSVP im Fall von IntServ) wurde ebenfalls nicht standardisiert.

### Paketklassifizierung und Quellflusskontrolle

Wurde ein Verkehrsvertrag definiert, muss dieser an der Grenze des DiffServ-Netzes mit Hilfe von Mechanismen zur Paketklassifizierung (*traffic classification*) und der Quellflusskontrolle (*traffic conditioning*, TC) überwacht bzw. umgesetzt werden. Dabei sind folgende logische Komponenten vorgesehen (siehe auch Bild 3.8):

- Der *Classifier* dient zur Klassifizierung des gesamten Verkehrsstromes in einzelne Ströme, für die getrennte TCAs definiert wurden. Dabei kann sich der *Classifier* am bisherigen DSCP-Wert (*behavior aggregate classifier*) oder an den Inhalten der Paketkopffelder für Quell- und Zieladresse, Quell- und Zielport sowie Transportprotokoll (*multi-field classifier*) orientieren. Zur weiteren Behandlung wird das Paket an die entsprechende Instanz des *Traffic Conditioners*, bestehend aus *Meter*, *Marker*, *Shaper* und *Dropper*, weitergereicht.
- Mit Hilfe des *Meter* wird die Charakteristik des Verkehrsstroms gemessen und mit den Parametern des zugehörigen Verkehrsvertrages verglichen. Dabei können wieder Mechanismen auf der Basis des LB (siehe Abschnitt 3.2.2.1) zum Einsatz kommen. Aber auch Verfahren, bei denen die mittlere Rate über einen EWMA abgeschätzt und mit der im SLA vereinbarten Rate verglichen wird, spielen hier eine große Rolle [71, 166]. Besonders erwähnenswert sind hier außerdem erweiterte Verfahren, die mit mehreren Konformitätsebenen arbeiten [128, 127, 166].
- Der *Marker* nimmt – unter Berücksichtigung der Konformitätsentscheidung des *Meter* – Eintragungen im DSCP-Feld des vom *Classifier* erhaltenen Pakets vor.
- Im Fall mangelnder Konformität kann der *Meter* einen *Shaper* oder einen *Dropper* anstoßen, um eine Verkehrsformung oder ein Verwerfen des Pakets zu veranlassen.

Entsprechend den Ausführungen zu SLAs gilt auch hier, dass jede der genannten Komponenten optional ist. Wenn kein Verkehrsvertrag festgelegt wurde und die DSCP-Markierung bereits vor dem Netzzugang (z. B. im Endgerät [100]) vorgenommen wurde, kann das *Traffic Conditioning* ganz entfallen. Allerdings ist es in diesem Fall sinnvoll, die klassifizierten Pakete einer geeigneten *Accounting*-Komponente zuzuführen.

### 3.3.3.3 Standardisierte PHBs

Neben einer Default-Klasse (DSCP-Wert 0) für herkömmlichen Verkehr (*best effort service*) wurden von der DiffServ-Arbeitsgruppe der IETF zwei Dienste mit den zugehörigen PHBs standardisiert: der *Assured Service*, der durch die AF-PHB-Gruppe (*assured forwarding*) realisiert wird [126], sowie der *Premium Service* [205], zu dessen Umsetzung das EF-PHB (*expedited forwarding*) definiert wurde [143].

#### Assured Forwarding

Der *Assured Service* geht auf einen Vorschlag von Clark und Fang zurück, bei dem der Nutzer im SLA eine erwartete ihm zur Verfügung stehende Bandbreite (*expected capacity*) mitteilen kann [71]. Solange die Senderate die „erwartete Kapazität“ nicht überschreitet – was mit Hilfe eines LB oder eines EWMA-basierten Mechanismus am Netzrand überprüft werden kann – werden von diesem Nutzer geschickte Pakete als *In* markiert. Wenn dauerhaft mehr gesendet wird, werden nichtkonforme Pakete als *Out* gekennzeichnet. Innerhalb des Netzes werden als

**Tabelle 3.2:** DSCP-Werte für AF

	AF-Klasse 1	AF-Klasse 2	AF-Klasse 3	AF-Klasse 4
niedere Verwerfungspriorität	001 010	010 010	011 010	100 010
mittlere Verwerfungspriorität	001 100	010 100	011 100	100 010
hohe Verwerfungspriorität	001 110	010 110	011 110	100 110

*In* markierte Pakete bevorzugt behandelt. Dabei kann zwar nicht garantiert werden, dass diese Pakete nicht verloren gehen und somit die vereinbarte Bandbreite wirklich zur Verfügung steht. Langfristig sollte jedoch – auch mit Hilfe geeigneter Dimensionierung – versucht werden, diese Bandbreite bereitzustellen (*provisioning*). Aus diesem Ansatz, bei dem im Kernnetz lediglich zwei Prioritäten unterschieden werden, hat sich schließlich AF entwickelt, das als PHB-Gruppe insgesamt 12 PHBs umfasst, die sich auf vier AF-Klassen mit jeweils drei Verwerfungsprioritäten (*drop preferences*) aufteilen. Die festgelegten DSCP-Werte für die AF PHBs sind Tabelle 3.2 zu entnehmen.

Genauere Aussagen zur Realisierung der AF PHBs werden im zugehörigen RFC [126] nicht gemacht; es werden lediglich einige Randbedingungen wie zu vermeidende Paketüberholungen innerhalb einer AF-Klasse genannt. Diese legen jedoch die Trennung der AF-Klassen mittels WFQ-ähnlicher Scheduling-Verfahren sowie die Verwendung von Paketverwerfungsmechanismen mit unterschiedlichen Schwellwerten zur Umsetzung der Verwerfungsprioritäten nahe. In [71] wird zur differenzierten Behandlung von *In*- und *Out*-Paketen das auf RED basierende RIO (*RED with In/Out bit*) vorgeschlagen.

### **Expedited Forwarding**

Der *Premium Service* soll vor allem im Hinblick auf Echtzeitanwendungen wie z. B. IP-Telefonie einen Kanal mit geringen Verlusten und minimaler Verzögerung bereitstellen, der wie eine virtuelle Mietleitung (*virtual leased line*, VLL) erscheint. Ähnlich wie beim *Guaranteed Service* im IntServ-Modell (siehe Abschnitt 3.3.2.1) muss für jeden Verkehrsfluss eine Spitzenrate angegeben werden, die nicht überschritten werden darf. Folglich muss am Netzrand mittels der im vorigen Abschnitt erwähnten Komponenten eine Quellflusskontrolle mit Verwerfung oder Degradierung nicht konformer Pakete oder einer entsprechenden Verkehrsformung (wie in [205] vorgeschlagen) stattfinden. Bei konformen Paketen wird im Paketkopf der DSCP-Wert 101 110 eingetragen, um in den Netzknoten eine Behandlung als EF-Verkehr zu erreichen.

Mit Hilfe des EF-PHBs wird dafür gesorgt, dass die für EF-Verkehr zur Verfügung stehende Bandbreite stets mindestens so groß ist wie die EF-Ankunftsrate im entsprechenden Knoten. Als einfache Lösungsmöglichkeit bietet sich die statische Priorisierung von EF-Verkehr gegenüber allen anderen Verkehrsaggregaten an [205], aber auch eine Verwendung eines FQ-Schedulers mit einer starken Gewichtung der EF-Warteschlange ist eine denkbare Alternative.

Um einerseits eine Einhaltung der Anforderungen bestehender EF-Verkehrsflüsse bzgl. Verlust und Verzögerung zu erreichen und andererseits zu vermeiden, dass andere Dienstgüteklassen überhaupt keine Ressourcen mehr erhalten, ist zur Realisierung des *Premium Service* eine Verbindungsannahmesteuerung erforderlich. In [205] wird eine Architektur vorgeschlagen, bei der diese Funktion in einer zentralen Instanz, dem so genannten *Bandwidth Broker* (BB), lokalisiert ist. Dieser wird von den Randknoten konsultiert, wenn dort neue Dienstanforderungen vorliegen, die z. B. mit Hilfe eines RSVP-ähnlichen Protokolls vom Endgerät zum Netzrand geschickt wurden. Der BB entscheidet dann – evtl. auf der Basis von Messungen, die in den Kernnetzknotten vorgenommen wurden –, ob genügend Ressourcen vorliegen, um zusätzliche *Premium-Service*-Verkehrsflüsse zulassen zu können.

Das BB-Konzept ist nicht auf den *Premium Service* beschränkt, sondern findet auch in weitergehenden Architekturvorschlägen Anwendung [231, 263]. Dabei kann die Aufgabe des BBs allgemein in der Verwaltung von Regeln für das Verkehrsmanagement (*policy control*) gesehen werden. Dazu gehören auch die Regeln, die die Arbeitsweise der PHBs in den Kernnetzknotten definieren (z. B. die Parameter von Scheduling-Algorithmen). Für die Kommunikation von Netzknoten mit dem BB bietet sich der Einsatz des in [91] definierten COPS-Protokolls (*common open policy service*) an. Schließlich spielt der BB auch eine wichtige Rolle, wenn Verkehrsflüsse über mehrere DiffServ-Netzbereiche gehen. Hierzu ist ein Zusammenwirken der BBs in den verschiedenen Bereichen erforderlich [205, 263].

#### **3.3.3.4 Relative Dienstgüte innerhalb von DiffServ**

Die Definition der DiffServ-Architektur auf Basis der in Abschnitt 3.3.3.1 formulierten Kerngedanken und der in Abschnitt 3.3.3.2 genannten Grundelemente bietet weitgehende Gestaltungsmöglichkeiten. Daher existiert eine Vielzahl von Vorschlägen, entweder als spezielle Ausprägungen der im vorigen Abschnitt beschriebenen vordefinierten Dienste und PHBs oder als Alternativlösungen ohne einheitlich festgelegte DSCPs, die sich in der IETF jedoch bislang nicht als Standard durchsetzen konnten. Grundsätzliche Unterscheidungskriterien stellen die Verwendung einer Annahmesteuerung sowie der Einsatz von Quellflusskontrolle oder Verkehrsformung am Netzrand dar. Das Vorhandensein oder Fehlen dieser Komponenten gibt Aufschluss über die der Architektur zugrunde liegende Dienstgütephilosophie.

Die extremste Form bildet dabei der mit Hilfe von EF realisierbare *Premium Service* nach der Definition in [205], bei dem mit Hilfe einer Annahmesteuerung und entsprechender TC-Komponenten am Netzrand absolute Garantien angeboten werden.

Beim *Assured Service* in seiner ursprünglichen Form [71] wird ebenfalls der DSCP-Wert am Netzrand eingestellt, um den im SLA vereinbarten absoluten Wert für die erwartete Kapazität zu erreichen. Ohne geeignete Mechanismen, um eine Annahme zu vieler SLAs mit entsprechend hohen Bandbreitanforderungen zu verhindern, erlangt dieser Wert jedoch automatisch

relativen Charakter. Das bedeutet, dass dann lediglich noch die Aussage gilt, dass ein Nutzer mit einer höheren erwarteten Kapazität mehr Bandbreiteressourcen erhalten wird als ein Nutzer mit einem geringen Wert. Wie sich diese Besserstellung quantitativ auswirkt, hängt sowohl vom Scheduling und Puffermanagement in den Kernnetzknotten als auch von der Reasilierung des TC-Mechanismus am Netzrand ab.

Große Ähnlichkeit zum *Assured Service* weist der USD-Ansatz (*user share differentiation*) auf, der sich selbst als Vertreter der Lösungen mit relativer Differenzierung sieht [271]. Wie beim *Assured Service* kann dabei vom Nutzer eine gewünschte Bandbreite angegeben werden, wobei unter einem „Nutzer“ auch ein Dienstanbieter verstanden werden kann, der die Bandbreite wiederum an Einzelnutzer aufteilt. Im Überlastfall ergibt sich der relative Anteil jedes Nutzers an der Gesamtbandbreite aus dem Verhältnis der von ihm angegebenen Bandbreite und der Summe der Bandbreiteanforderungen aller Nutzer. Zur Umsetzung dieses Konzepts wird allerdings nicht wie beim *Assured Service* ein *Traffic Conditioning* am Netzrand vorgenommen, sondern es werden in den Kernnetzknotten FQ-Scheduler eingesetzt, deren Gewichte entsprechend den Bandbreiteanteilen eingestellt werden. Dies bedeutet allerdings, dass in den Kernnetzknotten Verkehrsströme einzelner Nutzer identifiziert werden müssen, was sich nur im Falle einer Hierarchiebildung im oben angesprochenen Sinn vom IntServ-Modell unterscheidet. Außerdem ist eine Signalisierung erforderlich, die sich auf die Kernnetzknotten erstreckt.

Eine bzgl. des Steuerungsaufwands deutlich einfachere Lösung, um relative Differenzierung zu erreichen, ist der in [88] von Dovrolis vorgeschlagene, auf proportionaler Differenzierung basierende Ansatz, bei dem auf Annahmesteuerung, Quellflusskontrolle und Signalisierung im Kernnetz völlig verzichtet werden kann. Es werden lediglich eine Reihe von Prioritätsklassen angeboten, die durch Setzen des DSCP-Werts in der Regel im Endgerät ausgewählt werden. Dies kann entweder durch einen menschlichen Benutzer mit Hilfe einer geeigneten Schnittstelle oder automatisiert durch eine Anwendung mit Dienstgütereigenschaften erfolgen. Eine weitere Möglichkeit besteht in der Verwendung eines modifizierten Transportprotokolls, das die Priorität im Kopf der gesendeten Pakete in Abhängigkeit von den vom Netz erhaltenen Überlastsignalen (z. B. Paketverluste bei TCP) setzt und so für eine dynamische Anpassung an den Netzzustand sorgt [100]. Daneben ist weiterhin die Festlegung des DSCP-Werts am Netzrand in ähnlicher Weise wie bei AF möglich [87].

Die Komplexität ist bei diesem Ansatz hauptsächlich in den Scheduling- und Puffermanagement-Mechanismen der Kernnetzknotten enthalten, die für eine vorhersehbare Behandlung der unterschiedlichen Verkehrsströme sorgen. Eine mögliche Grundlage dafür bildet dabei das in Abschnitt 3.1.4.2 erwähnte Modell einer proportionalen Differenzierung. Die Untersuchung von geeigneten Lösungen für Scheduling und Puffermanagement auf Basis des Dovrolis-Vorschlags ist Hauptgegenstand von Kapitel 5.

### 3.3.4 Hybride Ansätze

Die vorigen Abschnitte haben gezeigt, dass einige der Ansätze zur Dienstgüteunterstützung teilweise Nachteile haben, die einen Einsatz in bestimmten Netzbereichen (z. B. dem Kernnetz) nicht empfehlenswert erscheinen lassen, in anderen Netzbereichen jedoch durchaus akzeptabel sind. Wenn es um eine Gesamtlösung der Dienstgüteproblematik geht, werden daher häufig Kombinationen verschiedener Ansätze vorgeschlagen:

- **IntServ/ATM:** Bei diesem Ansatz kommt im zugangsnahen Bereich die IntServ-Architektur mit RSVP als Signalisierprotokoll zum Tragen. Mit Hilfe eines Gateways wird eine Abbildung von RSVP-Reservierungen auf ATM-Verbindungen vorgenommen. Besonders interessant ist diese Option, wenn im Zuge dieser Abbildung gleichzeitig eine Aggregation von Verkehrsströmen stattfindet [1].
- **IntServ/MPLS:** Diese Lösung entspricht weitgehend der IntServ/ATM-Kombination mit einer Aggregation am Übergang, wobei an die Stelle von ATM-Verbindungen nun MPLS-Pfade treten. Aufgrund der in Abschnitt 3.3.1.4 genannten Vorteile von MPLS kann diese Lösung als die attraktivere angesehen werden.
- **IntServ/DiffServ:** Dieser Ansatz mit IntServ/RSVP im Zugangnetz und DiffServ im Kernnetz stellt die allgemein favorisierte Variante eines DiffServ-Einsatzes dar. Hierbei sollen die Möglichkeiten einer garantierten, auf einen Verkehrsfluss bezogenen Dienstgüte bei IntServ mit der besseren Skalierbarkeit der DiffServ-Architektur kombiniert werden. Auch hier sind wieder spezielle Router erforderlich, die den Übergang zwischen beiden Architekturen bewerkstelligen und dabei u. a. die Verbindungsannahme steuern [85].
- **DiffServ/MPLS:** Die Kombination dieser beiden Konzepte ist zunächst als eine Vereinigung innerhalb des Kernnetzes zu sehen. DiffServ definiert die Unterteilung des Verkehrs in Klassen und stellt Mechanismen zur Verkehrsdifferenzierung bereit. MPLS bietet die Möglichkeiten einer intelligenten Verkehrslenkung, um für Verkehrsaggregate einen Pfad durch das Kernnetz einzurichten. Darüber hinaus ist es aber auch möglich, DiffServ bis in den Zugangsbereich hinein zu betreiben, sodass von „DiffServ über MPLS“ im engeren Sinn gesprochen werden kann. Dieser Gedanke einer Überlagerung von DiffServ und MPLS findet seine Fortsetzung, wenn an die Stelle von MPLS das auf optische Pfade aufbauende GMPLS (*generalized multi-protocol label switching*) tritt [21]. In diesem Fall werden die DiffServ-Mechanismen vor allem benötigt, um beim Zugang zu den durchschaltevermittelten Kanälen der optischen Ebene eine entsprechende Differenzierung vorzunehmen.

Neben den genannten Hybridlösungen sind auch Kombinationen verschiedener Ansätze innerhalb einer Architektur denkbar. Beispielsweise kann im Kernnetz DiffServ mit absoluten Garantien verwendet werden (evtl. in Kombination mit MPLS oder GMPLS), während dem Nutzer relative Garantien angeboten werden.

# Kapitel 4

## Modellierung und Leistungsbewertung von IP-Netzen

Im Hinblick auf die Bewertung von Mechanismen zur Bereitstellung von Dienstgüte werden in diesem Kapitel Fragestellungen aufgegriffen, die mit der Modellierung und Leistungsuntersuchung im Kontext von IP-Netzen verbunden sind. Zunächst wird ein Überblick über Ziele und Methoden der Leistungsbewertung gegeben und dabei besonders die Anwendbarkeit verschiedener Methoden bei der Untersuchung von IP-Netzen berücksichtigt. Anschließend werden Verkehrsmodelle mit unterschiedlichem Abstraktionsgrad vorgestellt, wie sie teilweise auch im Rahmen dieser Arbeit eingesetzt wurden. Schließlich ist neben der Verkehrsmodellierung auch die Betrachtung geeigneter Leistungsmetriken notwendig, wie sie auf verschiedenen Ebenen der Kommunikation auftreten.

### 4.1 Ziele und Methoden der Leistungsbewertung

#### 4.1.1 Ziele

Die im Rahmen dieser Arbeit betrachtete Leistungsuntersuchung dient allgemein zur Unterstützung der Entwicklung von Systemen und Verfahren der Kommunikationstechnik. Im Speziellen geht es hier um die Architektur von Vermittlungsknoten in IP-Netzen sowie um Verfahren des Verkehrsmanagements, wie sie in Kapitel 3 beschrieben sind. Dabei soll in der Regel zunächst deren Leistungsfähigkeit ausgehend von einer gegebenen Konfiguration und festgelegten Randbedingungen nachgewiesen werden.

Darüber hinaus kann die Leistungsbewertung der genannten Kommunikationssysteme jedoch auch als Bestandteil eines umfassenden Netzplanungsprozesses gesehen werden, in dessen Rahmen einzelne Komponenten dimensioniert werden müssen. Typische Aufgaben sind z. B.

die Ermittlung von Puffergrößen in Netzknoten oder Bandbreite auf Verbindungsabschnitten oder auch die Festlegung sinnvoller Parameter für Verkehrsmanagement-Verfahren wie Scheduling- und Puffermanagement-Mechanismen.

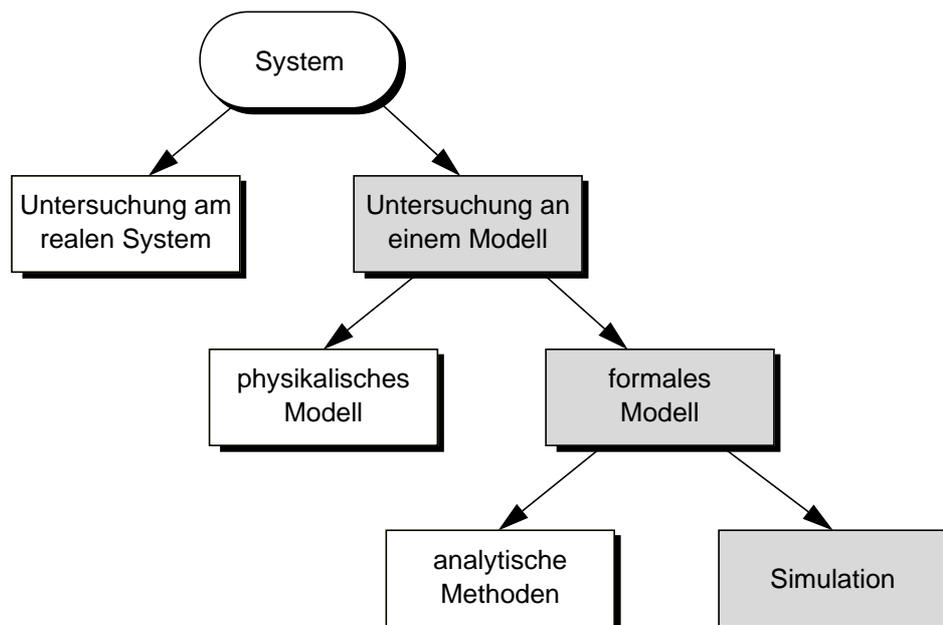
Neben der Beherrschung der eingesetzten Leistungsbewertungsmethode (siehe Abschnitt 4.1.2) bedarf es auch einer Festlegung der Anforderungen an das untersuchte System oder Verfahren und einer geeigneten Formulierung derselben, z. B. durch die Wahl entsprechender Leistungskenngrößen (siehe Abschnitt 4.3). Ebenso wichtig ist die richtige Interpretation der Ergebnisse, die im Falle der Integration in die Netzplanung entsprechende Modifikationen von Architektur oder Parametern nach sich zieht.

## 4.1.2 Methoden

Zur Bestimmung der Leistungsfähigkeit von Systemen stehen eine Reihe von Methoden zur Verfügung, die in sehr allgemeiner und vereinfachter Weise in Bild 4.1 dargestellt sind [171]. Methoden zur Leistungsuntersuchung lassen sich dabei in ein ähnliches Schema einordnen wie z. B. Methoden zur Untersuchung der Funktionsfähigkeit, sodass die Darstellung grundsätzlich zur Gliederung von Methoden zur Untersuchung eines System dienen kann.

### 4.1.2.1 Untersuchungen am realen System

Zunächst kann zwischen Untersuchungen an einem realen System und solchen, die auf der Grundlage eines mehr oder weniger abstrakten Modells durchgeführt werden, unterschieden werden. Eine Untersuchung an einem realen System kommt nur dann in Frage, wenn ein sol-



**Bild 4.1:** Methoden zur Untersuchung von Systemen nach [171]

ches System zum Zeitpunkt der Untersuchung überhaupt existiert, was bei Elementen von Kommunikationsnetzen eher die Ausnahme darstellt. Des Weiteren sind die beschränkte Zugänglichkeit fertiger Systeme für Untersuchungen sowie die mangelnde Reproduzierbarkeit von Ergebnissen als Probleme zu nennen. Dennoch findet diese Methode z. B. in Form von Verkehrs- und Leistungsmessungen in Kommunikationsnetzen, insbesondere auch in IP-Netzen Anwendung. Dabei handelt es sich zwar streng genommen meist nicht um Untersuchungen, die ein zu entwickelndes System direkt mit einbeziehen. Jedoch lassen solche Messungen häufig bereits Aussagen im Hinblick auf zukünftige Systeme zu und sind außerdem unverzichtbar zur Charakterisierung der Randbedingungen für spätere Untersuchungen an einem Modell.

Die Untersuchung an einem Modell andererseits erfordert ein gewisses Abstraktionsvermögen, d. h. die Konzentration auf die wesentlichen Eigenschaften des Systems und das Weglassen von Details. Hierbei stellt sich immer die Frage, ob ein gefundenes Modell gültig ist, was entweder eine genaue Kenntnis des Systems voraussetzt oder durch den Vergleich der Ergebnisse mit denen einer Untersuchung an einem realen System (z. B. von Messungen) – zumindest für Spezialfälle – nachgewiesen werden sollte (Validierung). Das Problem der Gültigkeit eines Modells verschärft sich mit zunehmendem Abstraktionsgrad, während die weitere Bearbeitbarkeit dadurch i. Allg. erleichtert wird. Somit stellt der Abstraktionsgrad grundsätzlich ein wichtiges Merkmal eines Modells dar.

#### **4.1.2.2 Untersuchungen mit Hilfe eines physikalischen Modells**

Ein relativ geringer Abstraktionsgrad liegt im Fall eines physikalischen Modells vor. Untersuchungen auf der Basis eines solchen Modells unterscheiden sich von Untersuchungen an einem realen System im Wesentlichen darin, dass Teile des Systems physikalisch nachgebildet werden, wobei irrelevante Eigenschaften vernachlässigt werden. Dabei ist jedoch der Übergang zwischen den beiden Untersuchungsmethoden fließend; eine klare Abgrenzung ist kaum möglich. Als Vertreter aus dem Bereich der Kommunikationsnetze für diese Kategorie sind Laboruntersuchungen mit Hilfe prototypischer Implementierungen – vor allem im Rahmen so genannter *Testbeds* – zu nennen. Gerade im Kontext von IP mit seinem verteilten Ansatz und der daraus erwachsenden einfachen Integrierbarkeit neuer Netzkomponenten hat diese Form der Untersuchung einen hohen Stellenwert. Häufig dienen solche Laboraufbauten zunächst dem Funktionsnachweis, können darüber hinaus aber auch für Leistungsuntersuchungen verwendet werden. Dies reicht bis hin zu Feldversuchen, bei denen menschliche Benutzer als Testpersonen mit einbezogen werden wie z. B. im Falle des INDEX-Feldversuchs zur Untersuchung des Nutzerverhaltens im Zusammenhang mit Dienstgütebereitstellung und Entgelterhebung [92].

Der große Vorteil dieser Methode liegt darin, dass auch Eigenschaften von Komponenten (insbesondere auf der Endgeräteseite) berücksichtigt werden, die für eine Modellierung nicht aus-

reichend bekannt sind, etwa weil sie im Bereich des Betriebssystems (z. B. TCP-Implementierungen) oder des Anwendungsprogramms (z. B. WWW-Browser, IP-Telefonie-Anwendung) liegen. Allerdings kommt es dadurch häufig zu einer Überlagerung von Effekten, die eine korrekte Interpretation der Ergebnisse meist deutlich erschwert. Darüber hinaus stellen – wie bei der Untersuchung am realen System – die Reproduzierbarkeit von Messungen sowie die Lastgenerierung Probleme dieser Methode dar.

#### **4.1.2.3 Untersuchungen auf der Basis eines formalen Modells**

Auf einem gegenüber dem physikalischen Modell hohen Abstraktionsgrad ist das formale Modell anzusiedeln. Im Zusammenhang mit der Leistungsuntersuchung handelt es sich hierbei in der Regel um ein verkehrstheoretisches Modell, bestehend aus Strukturelementen wie Warteschlangen und Bedieneinheiten sowie einer stochastischen Beschreibung von Ankunfts- und Bedienprozessen. Hinzu kommen meist noch Elemente zur funktionalen Beschreibung, z. B. im Zusammenhang mit der Bestimmung der Bedienreihenfolge.

Auf der Basis eines verkehrstheoretischen Modells kann nun entweder eine mathematische Analyse oder eine simulative Untersuchung des Systemverhaltens stattfinden. Mathematische Analyseverfahren sind häufig nur für relativ einfache Modelle mit klar definierten Randbedingungen (z. B. im Hinblick auf Ankunftsprozesse) verfügbar, während die Simulation auch die Behandlung komplizierterer Modelle mit geringerem Abstraktionsgrad erlaubt. Andererseits erfolgt dadurch bei der Analyse zwangsläufig eine Konzentration auf Kerneffekte, was zu einem besseren Systemverständnis beiträgt. Dies gilt hauptsächlich dann, wenn es gelingt, einfache Approximationen zu finden, während exakte analytische Lösungen oft sehr komplex sind, was unter anderem die numerische Auswertung erschwert. Bei der Simulation hingegen sind Probleme der statistischen Aussagesicherheit, lange Laufzeiten (besonders beim Fokus auf seltene Ereignisse) und ein erhöhter Implementierungsaufwand bei komplexeren Modellen – verbunden mit entsprechenden Anforderungen an den softwaretechnischen Entwurf – als Nachteile zu nennen.

Vergleicht man Analyse und Simulation bzgl. ihrer Anwendung in heutigen Untersuchungen, so kann man feststellen, dass mit der zunehmenden Konzentration auf IP-Netze im Bereich der Leistungsbewertung von Kommunikationsnetzen die Bedeutung der verkehrstheoretischen Analyse zugunsten der simulativen Untersuchung abgenommen hat. Dies ist v. a. auf folgende Ursachen zurückzuführen:

- IP-Verkehr hat eine andere Charakteristik als beispielsweise der aus Telefonnetzen bekannte Verkehr, der weitgehend mit Hilfe von Markoff-Prozessen modelliert werden kann. Bei einer Betrachtung auf der Paketebene sind es insbesondere die zu selbstständlichem Verkehr führenden Langzeitabhängigkeiten, die eine Analyse erschweren. Hinzu kommt der nicht zu vernachlässigende Einfluss höherer Protokollschichten, in erster Linie von TCP, das eine

Anpassung der Senderate an den aktuellen Netzzustand vornimmt und damit die Verkehrscharakteristik nachhaltig verändert.

- Das für die Leistungsbewertung relevante Systemverhalten ist als zunehmend komplex einzustufen, insbesondere wenn es um die Untersuchung von Verkehrsmanagement-Mechanismen geht. Dies gilt auch im Hinblick auf die in dieser Arbeit betrachteten Puffermanagement- und Scheduling-Mechanismen, die sich durch eine Vielzahl von Parametern und Optionen auszeichnen. Das Problem der Systemkomplexität verschärft sich weiter, wenn anstelle einzelner Netzknoten ein komplettes Netz betrachtet wird, um Ende-zu-Ende-Leistungsmaße zu bestimmen.

Die beiden genannten Punkte stellen allerdings auch in Bezug auf eine simulative Betrachtung eine nicht zu unterschätzende Herausforderung dar. Langzeitabhängigkeiten haben negative Auswirkungen in Bezug auf die statistische Aussagesicherheit. Zunächst gilt, dass in Abhängigkeit vom Hurst-Parameter  $H$  als Maß für den Grad der Selbstähnlichkeit sehr viel länger simuliert werden muss als in einem Szenario mit nicht selbstähnlichem Verkehr. Wird die gängige Teiltestmethode (*batch means*) [171] angewandt, die auch der in dieser Arbeit eingesetzten Simulationsumgebung [34] zugrunde liegt, und will man z. B. die Standardabweichung der Teiltestergebnisse um den Faktor 10 reduzieren, muss im Fall von Verkehr ohne Langzeitabhängigkeiten die Simulationsdauer um den Faktor 100 verlängert werden. Liegt hingegen selbstähnlicher Verkehr vor, ist eine Verlängerung der Simulationsdauer um den Faktor  $10^{1/(1-H)}$  (bei stark selbstähnlichem Verkehr mit  $H = 0.9$  also um den Faktor  $10^{10}$ ) erforderlich, um eine entsprechende Verringerung der Standardabweichung zu erreichen.

Erschwerend kommt hinzu, dass im Fall von selbstähnlichem Verkehr durch die verhältnismäßig starke Korrelation der Teiltests eine wesentliche Voraussetzung für den Einsatz des im Zusammenhang mit der Teiltestmethode üblichen statistischen Verfahrens zur Angabe von Vertrauensintervallen, welches auf der Anwendung der Student-t-Verteilung beruht, nicht erfüllt ist. Wendet man dieses Verfahren trotzdem an, erhält man zu kleine Vertrauensintervalle.

In vielen Forschungsbeiträgen, die sich mit Leistungsuntersuchungen in IP-Netzen beschäftigen, ist zu beobachten, dass die durch die Selbstähnlichkeit induzierten Probleme umgangen werden, indem auf stationäre Simulationen verzichtet wird und statt dessen die gemessenen Werte über der Simulationszeit aufgetragen werden. Während solche Traces durchaus ihre Berechtigung haben, wenn es um die Untersuchung von transientem oder Kurzzeitverhalten geht, wird dadurch die Ableitung von allgemeingültigen Aussagen über das langfristige Verhalten meist nur erschwert.

Auch die große Komplexität des untersuchten Systems, wie sie bei der Leistungsbewertung im Kontext IP-basierter Netze vorherrscht, wirkt sich auf die Methode der Simulation aus. So kommt man z. B. nicht umhin, gewisse Teile des TCP-Protokolls nachzuimplementieren, will

man die Auswirkungen von TCP untersuchen. Damit rückt die Simulation in Teilen in die Nähe der prototypischen Implementierung. Außerdem muss häufig aufgrund des ausufernden Parameterraums und der Schwierigkeit der Interpretation entsprechender Ergebnisse eine Konzentration auf eine eingeschränkte Auswahl von Szenarien und eine Fixierung vieler Parameter erfolgen. Eine solche Auswahl erfordert ein hohes Maß an Sorgfalt, um nicht wesentliche Effekte zu vernachlässigen oder andererseits Effekten eine zu große Bedeutung zuzumessen, obwohl sie vielleicht nur in Spezialfällen auftreten.

## 4.2 Verkehrsmodellierung

Die Verkehrsmodellierung nimmt als zweiter Bestandteil der Modellierung neben der Systemmodellierung eine wichtige Rolle bei der Leistungsuntersuchung von IP-Netzen ein. Sie umfasst i. d. R. eine stochastische Beschreibung des in einem System eintreffenden Verkehrs, die als Grundlage für eine Analyse ebenso herangezogen werden kann wie zur Nachbildung des Verkehrs in einem Verkehrsgenerator als Bestandteil einer Simulation.

Als Grundlage für ein Verkehrsmodell dienen häufig Verkehrsmessungen, die entweder in realen Netzen oder unter Laborbedingungen durchgeführt worden sind. Mit Hilfe einer statistischen Analyse kann aus den Messungen eine Verkehrscharakterisierung gewonnen werden. Gerade im Umfeld von IP-Netzen ist aber neben den Messungen auch ein Kenntnis der Protokolle und Anwendungen erforderlich, durch die der gemessene Verkehr erzeugt wird, um schließlich von der Charakterisierung zu einem Modell zu gelangen.

Bei der Modellierung von IP-Verkehr ist vor allem die Betrachtung verschiedener Zeit- und Protokollebenen notwendig, die einen Einfluss auf die Charakteristika des Verkehrs haben [66]. Eine solche hierarchische Verkehrsmodellierung ist bereits seit langem bekannt – z. B. aus Untersuchungen im Umfeld von ATM, wo eine Unterteilung in Verbindungs-, Bündel- und Zellebene mit der Bildung eventueller Unterebenen üblich ist [94, 160]. Im Vergleich zur Verkehrsmodellierung in ATM-Netzen ist im Kontext von IP-Netzen die Ausgangssituation wesentlich diffuser, da eine Vielzahl unterschiedlichster Anwendungen existiert, die als gemeinsame Basis nur die IP-Schicht haben. Eine Unterteilung in Ebenen muss daher stark von den jeweiligen Anwendungen und deren charakteristischen Eigenschaften abhängen. Hinzu kommt die u. a. durch das dominante TCP hervorgerufene stärkere Verwebung einzelner Ebenen, die eine dekompositorische Betrachtung erschwert. Dies drückt sich auch in dem bereits angesprochenen Phänomen der Selbstähnlichkeit aus.

Im Folgenden werden zunächst einige Modellierungstechniken vorgestellt, die im Wesentlichen die Paket- und Bündelebene berühren und die auch in dieser Arbeit zum Einsatz kommen. Darüber hinaus wird dann aber auch der Einfluss höherer Protokollschichten betrachtet, der insbesondere im Fall von TCP-basiertem Verkehr nicht vernachlässigt werden kann.

## 4.2.1 Modellierung auf Paket- und Büschelebene

Modelle, die sich auf eine Betrachtung der Paketebene beschränken, stellen häufig den Ausgangspunkt einer Verkehrsmodellierung dar. Der Vorteil dieser Modelle liegt zum einen in ihrer vergleichsweise einfachen Handhabbarkeit im Hinblick auf eine Verwendung in Analyse oder Simulation und zum anderen in der Nähe zu Messungen, bei denen Kenngrößen der Paketebene am leichtesten erfasst werden können.

Zusätzlich zur Paketebene ist meist die Betrachtung einer Büschelebene sinnvoll, um eine Autokorrelation im Paketankunftsprozess in das Modell einzubringen. Durch die Modellierung der Büschelebene in der nachfolgend dargestellten Form wird das durch höhere Protokollschichten beeinflusste Verhalten allerdings nur grob wiedergegeben. Insbesondere berücksichtigen die in diesem Abschnitt beschriebenen Modelle keine Rückkopplungen des aktuellen Netzzustands auf das Quellverhalten, womit eine wesentliche Eigenschaft von TCP vernachlässigt wird. Dennoch können unter bestimmten Voraussetzungen, z. B. wenn es um die Dimensionierung von Netzabschnitten geht, die nicht den Engpass auf einem Ende-zu-Endepfad darstellen [62], solche Modelle ohne Rückkopplung auch für die Modellierung von TCP-basiertem Verkehr eingesetzt werden.

### 4.2.1.1 Erneuerungsprozesse

Bei der Klassifizierung von Paketebenenmodellen muss zunächst zwischen einer deterministischen und einer stochastischen Modellierung unterschieden werden. Bei einem deterministischen Modell werden feststehende Muster für die Abstände aufeinander folgender Pakete angenommen. Dies kann z. B. zur Definition von Verkehrs nützlich sein, der unter gegebenen Bedingungen (z. B. Überwachung durch Quellflusskontrolle) zu Extremwerten der Leistungsparameter führt (*worst case traffic*).

Auf der Paketebene sollen im Rahmen dieser Arbeit nur stochastische Verkehrsmodelle betrachtet werden. Darunter ist als einfachster Fall die Modellierung eines Ankunftsprozesses als Erneuerungsprozess zu nennen. Dies bedeutet, dass Zwischenankunftszeiten aufeinander folgender Pakete durch statistisch unabhängige und der gleichen Verteilung gehorchende (*independent and identically distributed, iid*) Zufallsvariablen repräsentiert werden.

Eine besondere Bedeutung hat dabei der Poisson-Prozess, der mit einer negativ-exponentiellen Verteilungsfunktion der Zwischenankunftszeiten korrespondiert. Dazu tragen vor allem die folgenden beiden Eigenschaften bei:

- Die Überlagerung von Poisson-Prozessen resultiert wieder in einem Poisson-Prozess.
- Bei der Überlagerung von  $n$  Erneuerungsprozessen mit Intensität  $\lambda/n$  erhält man im Grenzfall  $n \rightarrow \infty$  einen Poisson-Prozess mit Rate  $\lambda$  (Palmsches Theorem).

Allerdings weisen die Verkehrseigenschaften von IP-Flüssen i. d. R. eine gewisse Büschelförmigkeit auf, weshalb eine Modellierung von Verkehrsflüssen als Erneuerungsprozess und damit auch eine Modellierung eines durch Überlagerung entstandenen Verkehrsstroms als Poisson-Prozess auf der Paketebene eine wesentliche Eigenschaft unberücksichtigt lässt.

#### **4.2.1.2 Modelle für büschelförmigen Verkehr mit Kurzzeitabhängigkeit**

Für die Modellierung der Büschelförmigkeit, bei der es sich in mathematischem Sinn um eine nichtverschwindende Autokorrelation der Zwischenankunftszeiten handelt, existieren eine Reihe von Ansätzen. Von diesen Verkehrsmodellen, die überwiegend im Zusammenhang mit der Untersuchung von ATM-Netzen entwickelt wurden [112, 162, 236, 258], sollen an dieser Stelle nur einige besonders relevante herausgegriffen werden.

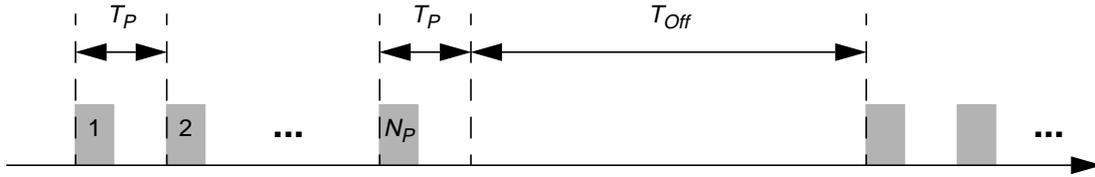
Zunächst werden einige Modelle aufgeführt, die eine einzelne Verkehrsquelle repräsentieren. Ein aggregierter Verkehrsstrom ergibt sich in diesem Fall aus einer Überlagerung solcher Einzelquellen. Daneben werden einige Modelle vorgestellt, die direkt für die Modellierung von aggregierten Verkehrsströmen herangezogen werden können.

#### **Modelle für Einzelquellen**

Ein großer Teil derartiger Verkehrsmodelle für büschelförmigen Verkehr kann mit Hilfe von Zustandsübergangsdiagrammen beschrieben werden. Charakteristisch sind hierbei einerseits die Verweildauern in den einzelnen Zuständen (modulierender Prozess), das Übergangsverhalten zwischen den Zuständen (Matrix der Übergangswahrscheinlichkeiten) sowie der jeweilige Ankunftsprozess in den einzelnen Zuständen (modulierter Prozess).

Die wichtigsten Vertreter für den modulierten Prozess sind der deterministische Prozess und der Poisson-Prozess, woraus die allgemeinen Modelle GMDP (*Generally Modulated Deterministic Process*) und GMPP (*Generally Modulated Poisson Process*) resultieren. Diese Modelle sind durch eine zunächst nicht näher definierte Anzahl von Zuständen und einen beliebigen modulierenden Prozess gekennzeichnet. In Bezug auf eine Konkretisierung des modulierenden Prozesses sind MMDP (*Markov Modulated Deterministic Process*) und MMPP (*Markov Modulated Poisson Process*) von besonderer Bedeutung, bei denen eine Markoff-Kette als modulierender Prozess wirkt. Ein MMPP ist dabei gleichzeitig eine spezielle Form eines MAP (*Markovian Arrival Process*). Zur Parametrisierung von MMDP und MMPP in Bezug auf den modulierenden Prozess können entweder die Mittelwerte der (negativ-exponentiell verteilten) Verweildauern in Kombination mit den Übergangswahrscheinlichkeiten oder die sich daraus ergebende Matrix der Übergangsraten herangezogen werden.

Orthogonal zur Konkretisierung im Hinblick auf den modulierenden Prozess stellen die jeweiligen Modellvarianten mit zwei Zuständen wichtige Spezialfälle dar, für die auch die Namen SDP (*Switched Deterministic Process*) bzw. SPP (*Switched Poisson Process*) verwendet wer-



**Bild 4.2:** Mögliche Realisierung einer On-Off-Quelle

den. Unter diesen sind wiederum die Sonderfälle IDP (*Interrupted Deterministic Process*) und IPP (*Interrupted Poisson Process*) von besonderem Interesse, bei denen einer der beiden Zustände ein Ruhezustand ohne Paketankünfte ist. Für den IDP hat sich als gebräuchlichere Bezeichnung der Begriff *On-Off-Quelle* eingebürgert. Weitere in der Literatur zu findende Bezeichnungen sind *sporadische Quelle* und – in Anlehnung an die Modellierung von Sprachverkehr – *Burst-Silence-* oder *Talkspurt-Silence-Quelle*.

Die genaue Beschreibung und Parametrisierung einer On-Off-Quelle kann auf unterschiedliche Weise erfolgen. Eine der Möglichkeiten ist in Bild 4.2 dargestellt. Dabei sind  $N_P$  und  $T_{Off}$  Zufallsvariablen für die Anzahl von Paketen in einem Burst bzw. die Dauer der Ruhephase, in der keine Pakete ankommen. Die On-Off-Quelle wird damit durch die Verteilungen dieser beiden Zufallsvariablen sowie durch den konstanten Paketabstand  $T_P$  innerhalb eines Bursts charakterisiert.

Im Gegensatz zu ATM-Netzen muss bei der Verwendung der On-Off-Quelle im Kontext von IP-Netzen prinzipiell berücksichtigt werden, dass die Paketlänge variabel ist. Dies kann dadurch geschehen, dass die Paketlänge als Zufallsvariable betrachtet wird, für die z. B. in Anlehnung an Messungen eine trimodale Verteilung zugrunde gelegt wird [191, 266]. Allerdings ist zu bezweifeln, ob damit die Realität angemessen wiedergegeben wird, da die gemessene Variabilität eher auf die Überlagerung von Quellen mit unterschiedlicher Paketlänge als auf Variationen der Paketlänge innerhalb eines Flusses zurückzuführen ist. Somit ist es sinnvoller, für eine einzelne On-Off-Quelle von einer konstanten Paketlänge  $L$  auszugehen. Daraus folgen dann weitere Kenngrößen der Quelle wie mittlere Rate  $m$ , Spitzenrate (*peak rate*)  $R_P$  und Büschelfaktor (*burstiness*)  $f_B$ :

$$m = \frac{E[N_P] \cdot L}{E[T_{Off}] + E[N_P] \cdot T_P} \quad (4.1)$$

$$R_P = \frac{L}{T_P} \quad (4.2)$$

$$f_B = \frac{R_P}{m} = 1 + \frac{E[T_{Off}]}{E[N_P] \cdot T_P} \quad (4.3)$$

Eine Markoff-modulierte On-Off-Quelle ergibt sich in diesem Fall, wenn für  $N_P$  eine (um eins verschobene) geometrische Verteilung und für  $T_{Off}$  eine negativ-exponentielle Verteilung gewählt werden.<sup>1</sup> Dabei gilt es allerdings zu beachten, dass eine solche Quelle keine Büschelförmigkeit mehr im Sinne nicht verschwindender Werte der Autokorrelationsfunktion besitzt, sondern einen Erneuerungsprozess darstellt. Wenn allerdings, wie unten dargestellt, eine Überlagerung von On-Off-Quellen betrachtet wird, geht die Erneuerungseigenschaft in Bezug auf den Überlagerungsprozess verloren.

Eine andere Modellvariante macht explizit von der Möglichkeit Gebrauch, dass die Paketlänge variabel sein kann. Anstatt eine diskrete Verteilung für die Anzahl von Paketen vorzugeben, wird dieses Modell mit einer Verteilung der Burstlänge  $B$  (im Sinne des im Burst enthaltenen Datenvolumens) parametrisiert. Dabei kann es sich entweder um eine kontinuierliche Verteilung oder aber wenigstens um eine diskrete Verteilung mit einer wesentlich feineren Granularität (Anzahl von Bytes statt Anzahl von Paketen) handeln. Alle Pakete im Burst bis auf das letzte haben eine vorgegebene Länge  $L_{max}$ . Die Gesamtzahl der Pakete im Burst ergibt sich zu

$$N_P = \left\lceil \frac{B}{L_{max}} \right\rceil \quad (4.4)$$

Das letzte Paket nimmt den Rest  $B - (N_P - 1) \cdot L_{max}$  des Burstvolumens auf und weist damit eine geringere Länge auf. Der Ausdruck

$$m = \frac{E[B]}{E[T_{Off}] + E\left[\left\lceil \frac{B}{L_{max}} \right\rceil\right] \cdot T_P} \quad (4.5)$$

liefert die mittlere Rate für diese Modellvariante.

### Modelle für aggregierte Verkehrsströme

Zur Modellierung aggregierter Verkehrsströme bietet sich zunächst sowohl in Analyse als auch Simulation eine Überlagerung einer endlichen Anzahl von Einzelquellen an. Dies hat jedoch v. a. den Nachteil, dass ein zusätzlicher Parameter (nämlich die Anzahl von Quellen) notwendig wird. Da in den meisten Fällen von einer großen Anzahl von Einzelquellen auszugehen ist, besteht eine Alternative in der Verwendung eines Modells, das die Überlagerung einer gegen unendlich strebenden Anzahl von Einzelquellen mit gleichbleibender mittlerer Summenrate beschreibt.

Die Überlagerung von  $n$  On-Off-Quellen führt im Grenzfall  $n \rightarrow \infty$  auf ein Modell, bei dem Burstankünfte als stochastischer Prozess, gekennzeichnet durch die Verteilung der Zwischenankunftszeit  $T_B$  von Bursts, beschrieben werden können. Voraussetzung dafür, dass die

---

<sup>1</sup>Die Verteilung der On-Dauer hat bei dieser Modellrealisierung allerdings nicht exakt einen negativ-exponentiellen Verlauf, sondern wird durch eine Treppenfunktion angenähert (siehe auch Abschnitt 6.1.1).

mittlere Rate des Summenprozesses endlich bleibt, ist allerdings, dass gleichzeitig  $T_{Off} \rightarrow \infty$  und damit die mittlere Rate jeder On-Off-Quelle gegen null geht. Da für die Einzelquelle der Ankunftsprozess der Bursts einen Erneuerungsprozess darstellt, dessen Zwischenankunftszeit sich aus  $T_{Off} + N_P \cdot T_P$  ergibt, ist aufgrund der o. g. Überlagerungseigenschaft von Erneuerungsprozessen der Burstankunftsprozess beim aggregierten Verkehrsstrom im Grenzfall ein Poisson-Prozess mit Ankunftsrate  $\lambda = 1/E[T_B]$ . Die Bursts selbst werden wie bei der On-Off-Quelle durch die Anzahl  $N_P$  von Paketen sowie den zeitlichen Abstand  $T_P$  von Paketankünften charakterisiert. Damit erhält man einen M/G/ $\infty$ -Burstprozess<sup>2</sup>, der auch als *Poisson Burst Process* (PBP) bezeichnet wird. Die mittlere Rate des PBP auf Basis der beschriebenen Realisierung ergibt sich zu:

$$m = \lambda \cdot E[N_P] \cdot L = \frac{E[N_P] \cdot L}{E[T_B]} \quad (4.6)$$

Eine andere Realisierungsmöglichkeit eines Burstprozesses leitet sich aus der zweiten der beiden genannten Varianten für die On-Off-Quelle ab, bei der das einer bestimmten Verteilung folgende Burstvolumen  $B$  auf  $N_P - 1$  Pakete maximaler Größe und ein kleineres Paket aufgeteilt wird. Auch in diesem Modell ergibt sich im Grenzfall ein Poisson-Burstankunftsprozess mit Rate  $\lambda$ , sodass man in diesem Fall als mittlere Rate erhält:

$$m = \lambda \cdot E[B] = \frac{E[B]}{E[T_B]} \quad (4.7)$$

Zu beachten ist, dass es sich bei den Ausdrücken nach Gleichung (4.6) und (4.7) jeweils um die mittlere Summenrate handelt, die – im Gegensatz zur mittleren Rate einer Einzelquelle – kleiner oder größer als die „Spitzenrate“  $R_P$  innerhalb eines Bursts sein kann. Das Verhältnis von mittlerer Rate und Spitzenrate stellt genauso wie beim Einzelprozess ein wichtiges Charakteristikum dieses Modells dar, kann nun aber Werte aus  $(0, \infty)$  annehmen.

### **Burstebenen-Modelle auf Basis des Flüssigkeitsflussansatzes**

Während die oben beschriebenen Modelle für Einzelquellen und aggregierte Verkehrsströme den Ankunftsprozess als Punktprozess beschreiben, existieren auch Modelle, bei denen nur noch die Burstebene betrachtet wird. Diese Modelle basieren auf dem Flüssigkeitsflussansatz (*fluid flow model*) und nehmen an, dass Daten nicht in Form diskreter Pakete, sondern – wie eine Flüssigkeit – in einem kontinuierlichen Strom eintreffen. Derartige Modelle werden häufig in analytischen Betrachtungen eingesetzt.

Sowohl die On-Off-Quelle als auch der PBP – jeweils in der Variante, bei der die Burstgröße durch eine kontinuierliche Zufallsvariable  $B$  beschrieben wird – können auch als Flüssigkeits-

---

<sup>2</sup> „G“ steht dabei für die beliebige Burstlängenverteilung, während „ $\infty$ “ die Überlagerung unendlich vieler Quellen symbolisiert.

flussmodell betrachtet werden, wenn an die Stelle von Zellankünften innerhalb eines Bursts ein kontinuierlicher Datenstrom mit Rate  $R_p$  tritt.

### 4.2.1.3 Modelle für selbstähnlichen Verkehr

Die im letzten Abschnitt vorgestellten Modelle erzeugen Verkehrsmuster, die durch Kurzzeitabhängigkeit (*short-range dependence*, SRD) gekennzeichnet sind. Dies äußert sich u. a. darin, dass die Werte der Autokorrelationsfunktion zwar von null verschieden sind, aber mit zunehmendem Zeitversatz (*lag*) rasch gegen null gehen. Bei zahlreichen Messungen von IP-Verkehr in LANs und in Weitverkehrsnetzen konnten jedoch auch Langzeitabhängigkeiten (*long-range dependence*, LRD) beobachtet werden, d. h. eine nicht verschwindende Autokorrelation über große Zeitbereiche hinweg [111, 172, 222]. Solche Langzeitabhängigkeiten führen zu selbstähnlichem Verkehr.

#### Eigenschaften selbstähnlicher Prozesse

Selbstähnlichkeit kann anschaulich als das Auftreten ähnlicher Verkehrsmuster auf allen Zeitebenen gedeutet werden. Aus der mathematischen Definition für Selbstähnlichkeit [219], die in verschiedenen Abstufungen auftreten kann (*distributional, second order, asymptotically second order*), können einige charakteristische Merkmale selbstähnlichen Verkehrs abgeleitet werden [94, 172]. Dazu gehört insbesondere, dass die Varianz des kumulierten Ankunftsprozesses  $A_t$ , also der in einem Zeitintervall der Länge  $t$  ankommenden Datenmenge, stärker mit  $t$  ansteigt als bei Erneuerungsprozessen oder Ankunftsprozessen mit Kurzzeitabhängigkeit. Quantitativ lässt sich dies mit Hilfe des Hurst-Parameters  $H$  folgendermaßen ausdrücken, wobei  $c$  einen nicht von  $t$  abhängigen Faktor bezeichnet:<sup>3</sup>

$$\text{VAR}[A_t] \rightarrow c \cdot t^{2H} \text{ für } t \rightarrow \infty \quad (4.8)$$

Die steigende Varianz von  $A_t$  korrespondiert mit einer sinkenden Varianz der in einem Intervall der Länge  $t$  zu beobachtenden mittleren Rate  $m_t$ :

$$\text{VAR}[m_t] \rightarrow c \cdot t^{2H-2} \text{ für } t \rightarrow \infty \quad (4.9)$$

Der Hurst-Parameter nimmt Werte zwischen 0.5 und 1 an, wobei  $H = 0.5$  für Verkehr gilt, der keine Selbstähnlichkeit aufweist, während  $H = 1$  einen Extremfall darstellt, bei dem überhaupt keine Varianzreduktion bzgl. der mittleren Rate auftritt.

---

<sup>3</sup> Gleichung (4.8) bezieht sich auf asymptotische Selbstähnlichkeit. Bei exakter Selbstähnlichkeit (zweiter Ordnung) gilt sogar Gleichheit.

### Büschelebenen-Modelle mit Langzeitabhängigkeiten

Es existiert eine Reihe von Modellen für selbstähnliche Verkehrsströme, von denen etliche als eine Erweiterung von Modellen für büschelartigen Verkehr, wie sie in Abschnitt 4.2.1.2 beschrieben wurden, gesehen werden können. So wird in [274] nachgewiesen, dass eine Überlagerung von On-Off-Quellen, bei denen die Burstgröße  $B$  einer *Heavy-Tail*-Verteilung genügt, selbstähnlichen Verkehr produziert. Heavy-Tail-Verteilungen sind dadurch gekennzeichnet, dass die komplementäre Verteilungsfunktion einem Potenzgesetz folgt und sich damit deutlich langsamer an null annähert als etwa die komplementäre Exponentialverteilung. Der bekannteste und für die Praxis mit Abstand relevanteste Vertreter dieser Klasse ist die Pareto-Verteilung mit der Verteilungsfunktion

$$P(X \leq x) = F(x) = \begin{cases} 0 & \text{für } x < k \\ 1 - \left(\frac{k}{x}\right)^\alpha & \text{für } x \geq k \end{cases} \quad (4.10)$$

und der Verteilungsdichtefunktion

$$f(x) = \begin{cases} 0 & \text{für } x < k \\ \alpha \cdot k \cdot \left(\frac{k}{x}\right)^{\alpha-1} & \text{für } x \geq k \end{cases} \quad (4.11)$$

Dabei repräsentieren  $k$  und  $\alpha$  den Minimalwert bzw. den Formparameter der Verteilung. Für  $\alpha \leq 2$  wird die Varianz unendlich; im Fall von  $\alpha \leq 1$  existiert zudem auch kein Erwartungswert. Der Hurst-Parameter des sich aus der Überlagerung ergebenden Verkehrsstroms hängt über die Beziehung

$$H = \frac{3 - \alpha}{2} \quad (4.12)$$

direkt mit dem Formparameter der Pareto-Verteilung zusammen.

Wie in Abschnitt 4.2.1.2 erläutert, resultiert eine Überlagerung von sehr vielen On-Off-Quellen im Grenzfall in einem M/G/ $\infty$ -Burstprozess. Wendet man dieses Konstruktionsprinzip auf On-Off-Quellen mit Pareto-verteilter Burstlänge an, ergibt sich folglich ein M/Pareto/ $\infty$ -Burstprozess, der ebenfalls selbstähnlichen Verkehr mit einem Hurst-Parameter entsprechend Gleichung (4.12) liefert. Dieses Modell, das auf Cox zurückgeht [79] und daher auch bisweilen als Cox-Modell, meist jedoch einfach als M/Pareto-Modell bezeichnet wird, ist vor allem im Zusammenhang mit Simulationen sehr praktikabel und wird z. B. in [31, 203] vorgeschlagen und für Leistungsuntersuchungen eingesetzt. Die Varianz des kumulierten Ankunftsprozesses, die aus

$$\text{VAR}[A_t] = 2 \cdot \frac{m}{b} \cdot R_P \cdot \int_0^t \int_0^u \int_{vh}^\infty \text{P}(B > s) ds \quad (4.13)$$

berechnet werden kann [206], nähert sich für  $t \rightarrow \infty$  folgendem Kurvenverlauf an [31]:

$$\text{VAR}[A_t] \rightarrow m \cdot \frac{R_P^{2H-1} \cdot \left(\frac{2-2H}{3-2H} \cdot b\right)^{2-2H}}{(3-2H) \cdot (2H-1) \cdot H} \cdot t^{2H} \quad (4.14)$$

wobei

$$b = k \cdot \frac{\alpha}{\alpha-1} \quad (4.15)$$

den Erwartungswert der Pareto-verteilten Burstgröße  $B$  bezeichnet und  $m$  und  $R_P$  die mittlere (Summen-)Rate bzw. die Spitzenrate des M/Pareto-Prozesses gemäß Abschnitt 4.2.1.2 darstellen. Im Vergleich dazu ergibt sich im Fall einer Burstlängenverteilung mit endlicher Varianz und somit einem nicht selbstähnlichen Verkehr

$$\text{VAR}[A_t] \rightarrow m \cdot \frac{\text{E}[B^2]}{\text{E}[B]} \cdot t \quad (4.16)$$

als Asymptote für die Varianz von  $A_t$ . Sowohl in Gleichung (4.14) als auch in (4.16) wird der mittlere Faktor des Terms auf der rechten Seite auch als Spitzigkeit (*peakedness*)<sup>4</sup> bezeichnet.

Die Modellierung von selbstähnlichem Verkehr als büschelförmiger Verkehr mit Heavy-Tail-Burstlängenverteilung liefert auch eine mögliche Erklärung für die Herkunft des Phänomens der Selbstähnlichkeit im Zusammenhang mit IP-Verkehr. So wurde in [81] durch Messungen von WWW-Verkehr und Web-Server-Inhalten herausgefunden, dass die Größe der auf einem WWW-Server verfügbaren Dateien und daher auch das Volumen und die Übertragungsdauer einzelner Transfers einer Heavy-Tail-Verteilung genügt.

### Fraktale Brownsche Bewegung

Ein sehr grundlegendes Modell, mit dem der Effekt der Selbstähnlichkeit wiedergegeben werden kann, ist die auf [185] zurückgehende fraktale Brownsche Bewegung (*fractional Brownian motion*, FBM). Der kumulative Ankunftsprozess wird dabei durch

$$A_t = m \cdot t + \sqrt{m \cdot a} \cdot Z_t \quad (4.17)$$

beschrieben, wobei  $m$  die mittlere Rate und  $a$  die Spitzigkeit des Verkehrs darstellen. Die Zufallsvariable  $Z_t$  repräsentiert eine normierte FBM mit Hurst-Parameter  $H$ . Neben verschie-

<sup>4</sup>Anstelle von Spitzigkeit wird bisweilen auch der Begriff Varianzkoeffizient (*variance coefficient*) verwendet [206], der jedoch leicht zu Verwechslungen mit dem Variationskoeffizienten führt.

denen weiteren Eigenschaften zeichnet sich  $Z_t$  vor allem durch Mittelwert null und Varianz  $t^{2H}$  aus, wodurch sich für  $A_t$  ergibt:

$$\text{VAR}[A_t] = m \cdot a \cdot t^{2H} \quad \text{für } t > 0 \quad (4.18)$$

FBM wird häufig in analytischen Untersuchungen verwendet, hat aber im Hinblick auf eine Verwendung in Simulationen gegenüber dem Burstebenen-Modell einige Nachteile. Zunächst besteht die Schwierigkeit, dass sich FBM aus (nicht-fraktalem) weißem Rauschen über ein nicht auflösbares Integral herleitet, so dass zur Erzeugung von Zufallsvariablen für den Inkrement-Prozess einer FBM, das fraktale weiße Rauschen (*fractional Gaussian noise*, FGN), auf Näherungsverfahren zurückgegriffen werden muss [94, 220]. Hinzu kommt dann aber noch das Problem, dass eine Beschreibung für die Ankunftszeitpunkte von Dateneinheiten, wie sie für eine ereignisgesteuerte Simulation benötigt werden, daraus nicht direkt abgeleitet werden kann. Für eine Erzeugung von Zufallszahlen für Zwischenankunftszeiten von Paketen sind daher weitere Näherungen erforderlich [220].

### Wavelet-basierte Modelle

Ein Hilfsmittel, das bei der Modellierung von selbstähnlichem Verkehr häufig zum Einsatz kommt, ist die Wavelet-Theorie. Dabei handelt es sich vereinfacht um eine integrierte Betrachtung von Zeit- und Frequenzbereich. Anwendung findet diese Theorie mit komplexem mathematischem Hintergrund zunächst bei der Auswertung von Verkehrsmessungen in Form einer Wavelet-Analyse. Die Wavelet-Analyse ermöglicht es, den Verkehrsstrom mittels einer Wavelet-Transformation durch eine Reihe von Koeffizienten zu charakterisieren und wichtige statistische Größen wie den Hurst-Parameter relativ genau zu bestimmen [268]. Die Koeffizienten können dann außerdem in umgekehrter Richtung wieder verwendet werden, um z. B. für eine Simulation einen Verkehrsstrom zu generieren, der die gleichen statistischen Eigenschaften hat wie der gemessene Verkehrsstrom, im Gegensatz zu diesem aber in seiner Länge nicht beschränkt ist [99]. Im Rahmen dieser Arbeit werden solche Ansätze allerdings nicht weiter verfolgt.

## 4.2.2 Modelle für elastischen IP-Verkehr

Die in Abschnitt 4.2.1 vorgestellten Modelle versuchen, die auf der Paketebene beobachtete Verkehrscharakteristik auf eher abstrakte Weise wiederzugeben. Demgegenüber stehen Modelle, bei denen Eigenschaften höherer Schichten explizit berücksichtigt werden. Dies ist vor allem im Fall von elastischem Verkehr, der auf der Anwendung von TCP beruht, wichtig, da TCP durch entsprechende Mechanismen in der Lage ist, die Senderate an den Netzzustand anzupassen, um Überlast zu vermeiden. Eine solche Rückkopplung ist in den Modellen aus Abschnitt 4.2.1 nicht berücksichtigt. In [12, 147] wird gezeigt, dass die TCP-Eigenschaften,



**Bild 4.3:** Unidirektionales TCP-Modell

insbesondere die Adaptivität, nicht vernachlässigt werden dürfen, weil sich andernfalls völlig unterschiedliche Resultate ergeben.

Nachfolgend werden zunächst aus den Eigenschaften von TCP, die ausführlich in [76, 259] beschrieben sind, diejenigen herausgegriffen, die für eine Modellierung relevant sind. Dabei wird die Vielfalt der Alternativen und Parameter aufgezeigt, die bei der Modellierung festgelegt werden müssen. Anschließend werden Möglichkeiten vorgestellt, die oberhalb von TCP ablaufenden Prozesse im Modell wiederzugeben.

#### 4.2.2.1 Modellierung von TCP-Eigenschaften

Die Modellierung von TCP in den Kommunikationsendpunkten stellt eine nicht triviale Aufgabe dar, die mit großer Sorgfalt angegangen werden muss [4]. Dies liegt nicht nur an der Komplexität der von TCP verwendeten Algorithmen, sondern vor allem auch an der Vielfalt an Versionen, Optionen und Parameter. Die Palette von Modellen reicht hier von der direkten Verwendung des Codes realer TCP-Implementierungen im Simulationsprogramm bis hin zu stark vereinfachten Modellen für analytische Untersuchungen.

Im Rahmen dieser Arbeit werden vorwiegend simulative Untersuchungen mit TCP-Verkehr durchgeführt, wobei das in [35, 180] beschriebene und auf [116] zurückgehende TCP-Simulationsmodell zum Einsatz kommt. Dieses stellt im Vergleich zu realen Implementierungen zunächst in folgenden Punkten eine Abstraktion dar:

- Es handelt sich um ein unidirektionales Modell. Dies bedeutet, dass ein Endgerät die Aufgabe des Senders (Versenden von Datenpaketen, Empfangen von Bestätigungen) übernimmt, während das andere Endgerät die Rolle des Empfängers (Entgegennahme der vom Sender kommenden Pakete, Versenden der Quittungen) ausfüllt. Dies stellt eine gängige Vereinfachung gegenüber dem Kommunikationsmodell der meisten TCP-basierten Dienste (insbesondere des WWW) dar, bei der beide Endstellen Datenpakete senden und empfangen können. Abgesehen von der Möglichkeit, in einer bidirektionalen Kommunikation Quittungen im Huckepack-Verfahren (*piggybacking*) mit Datenpaketen zu verschicken, können aber auch im bidirektionalen Fall Sende- und Empfangskomponenten weitgehend separiert werden, sodass die Verwendung eines unidirektionalen Modells keine wesentliche Einschränkung darstellt.

- Der Auf- und Abbau von TCP-Verbindungen mit Hilfe von Steuerpaketen sowie die TCP-Verbindungssteuerung, d. h. die Überwachung des Verbindungsauf- und abbaus durch Zustandsautomaten auf der Sender- und Empfängerseite, werden nicht modelliert. Dieser Verzicht ist gerechtfertigt, solange sowohl bei der Modellierung der Anwendungsebene (siehe Abschnitte 4.2.2.2 und 4.2.2.3) als auch in Bezug auf die Leistungsmetriken (siehe Abschnitt 4.3.2) die Betrachtung der Datenübertragungsphase im Vordergrund steht.
- Auf die Modellierung von Betriebssystemeigenschaften, wie z. B. eine Beschränkung des Sendepuffers oder nicht verschwindende Einschreib- und Auslesezeiten bei Sende- und Empfangspuffern, wird verzichtet, um eine Konzentration auf TCP-Eigenschaften zu erleichtern. Ein Modell, das solche Eigenschaften berücksichtigt, wurde in [116] entworfen.
- Mechanismen wie *Persistence Timer* oder *Keep-Alive Timer*, die nur in speziellen, hier nicht relevanten Situationen zum Tragen kommen [259], werden im Modell nicht berücksichtigt.
- Zahlreiche Optionen und Erweiterungen, die experimentelle TCP-Implementierungen teilweise bieten, z. B. die Verwendung von expliziter Überlastanzeige (*explicit congestion notification*, ECN) oder von Zeitstempeln zur Messung der Umlaufzeit (*round trip time*, RTT), werden im hier eingesetzten TCP-Modell nicht berücksichtigt.

Trotz dieser Einschränkungen stehen bei der Modellierung von TCP noch eine Reihe von Optionen zur Auswahl, die z. T. einen nicht unerheblichen Einfluss auf die Ergebnisse von Leistungsuntersuchungen haben.

### **Modellierung der Flusssteuerung**

In Simulationswerkzeugen ist häufig eine paketorientierte TCP-Flusssteuerung implementiert, d. h. es wird angenommen, dass Pakete eine feste Länge  $L_{MSS}$  entsprechend der maximalen Segmentgröße (*maximum segment size*, MSS) haben und jedes Paket durch eine Sequenznummer gekennzeichnet ist. Demgegenüber arbeitet die Flusssteuerung in realen Implementierungen byteorientiert, wodurch Paketlängen auch kleiner als  $L_{MSS}$  sein können und Fenstergrößen nicht unbedingt Vielfache von  $L_{MSS}$  sein müssen. Letzteres gilt auch für das hier verwendete Modell.

### **Überlaststeuerungs-Algorithmen**

Um Überlast zu vermeiden oder auf eine Überlastsituation zu reagieren, stehen auf der TCP-Senderseite Algorithmen bereit, die am Beginn der Übertragung, nach einem Timeout sowie nach Paketverlusten die Größe des Sendefensters begrenzen. Die Realisierung erfolgt mit Hilfe des Überlastfensters (*congestion window*, CWnd), dessen Größe mit  $W_C$  bezeichnet werden soll. Die Größe des Sendefensters  $W$  bestimmt sich aus dem Minimum von  $W_C$  und der Größe  $W_R$  des vom Empfänger mitgeteilten Fensters (*receiver advertised window*, RWnd).

Folgende Algorithmen beeinflussen das CWnd [5, 107, 142]:

- *Slow Start*: Am Beginn der Übertragung sowie nach dem Auftreten eines Timeouts hat  $W_C$  einen sehr niedrigen Wert ( $2 \cdot L_{MSS}$  bzw.  $L_{MSS}$ ). In der anschließenden Phase wird  $W_C$  beim Erhalt einer Bestätigung für bislang unquittierte Daten um die maximale Segmentgröße erhöht. Werden mit der Bestätigung z. B.  $L_{MSS}$  neue Daten quittiert, dürfen anschließend zwei neue Pakete maximaler Länge gesendet werden. Dieser Algorithmus führt zu einem exponentiellen Ansteigen von  $W_C$ .
- *Congestion Avoidance*: Wenn durch den *Slow-Start-Mechanismus* das CWnd eine bestimmte Größe erreicht hat, wird ein weiteres exponentielles Ansteigen von  $W_C$  verhindert, um die Entstehung von Überlast zu vermeiden. Ab diesem Schwellwert  $\theta_S$  von  $W_C$ , der als *Slow Start Threshold* (SSThresh) bezeichnet wird, bewirkt jede eintreffende Quittung nur noch eine Erhöhung von  $W_C$  um  $L_{MSS}^2/W_C$ , was einen näherungsweise linearen Anstieg des CWnd zur Folge hat.
- *Timeout Recovery*: Beim Auftreten eines Timeouts beim Sender wird das erste unbestätigte Paket wiederholt. Anschließend folgt eine *Slow-Start-Phase*, wobei  $W_C$  den Wert  $L_{MSS}$  erhält. SSThresh wird ebenfalls reduziert, wobei folgende Formel zur Anwendung kommt:

$$\theta_S = \max\left(2 \cdot L_{MSS}, \frac{W_F}{2}\right) \quad (4.19)$$

Dabei bezeichnet  $W_F$  das Volumen der zu diesem Zeitpunkt bereits gesendeten, aber noch unbestätigten Daten (*flight size*). In den meisten Fällen ist  $W_F$  identisch mit dem Wert des CWnd vor dessen Erniedrigung.

- *Fast Retransmit*: Wenn beim Sender nacheinander drei Bestätigungen eintreffen, mit denen keine neuen Daten quittiert werden (*duplicate acknowledgement*, DupACK), d. h. die nächste vom Empfänger erwartete Sequenznummer unverändert geblieben ist, geht der Sender von einem Paketverlust aus und wiederholt das letzte noch unbestätigte Segment. Anschließend folgt in älteren TCP-Implementierungen („TCP Tahoe“) ein Zurücksetzen von  $W_C$  auf den Wert  $L_{MSS}$  mit anschließendem *Slow Start*, während in neueren Implementierungen („TCP Reno“ und „TCP New Reno“) eine *Fast Recovery* durchgeführt wird.
- *Fast Recovery*: Beim Auftreten dreier DupACKs werden gleichzeitig auch  $W_C$  und  $\theta_S$  erniedrigt. Während die Reduktion von  $\theta_S$  Gleichung (4.19) folgt, wird  $W_C$  (nach der Verkleinerung von SSThresh) auf folgenden Wert gesetzt:

$$W_C = \theta_S + 3 \cdot L_{MSS} \quad (4.20)$$

Während der *Fast-Recovery-Phase* eintreffende DupACKs bewirken ein weiteres Ansteigen von  $W_C$  um jeweils  $L_{MSS}$ . Dieser Vorgang wird auch als *Conservation of Packets* bezeichnet.

net. Die „künstliche“ Vergrößerung des CWnd soll ein Versenden von neuen Daten auch in der *Fast-Recovery*-Phase ermöglichen, was allerdings nur unter bestimmten Voraussetzungen (große Anzahl von DupACKs im Vergleich zu der auf  $L_{MSS}$  normierten Größe des CWnd) tatsächlich geschieht. Das Ende der *Fast-Recovery*-Phase ist im Fall von „TCP Reno“ erreicht, wenn eine Bestätigung für bislang unquittierte Daten eintrifft. Nun wird  $W_C$  auf den Wert der SStresh gesetzt, was gleichzeitig den Übergang in den *Congestion-Avoidance*-Modus signalisiert.

- *Partial Acknowledgement Handling*: In der Version „TCP New Reno“ wurde die *Fast-Recovery*-Prozedur modifiziert, um auf gehäuft auftretende Paketverluste angemessen reagieren zu können. Wenn im Fall von „TCP Reno“ z. B. zwei aufeinander folgende Segmente verloren gegangen sind und das erste per *Fast Retransmit* wiederholt und bestätigt worden ist, wird der zweite Verlust erst durch einen Timeout mit anschließender *Timeout Recovery* erkannt und behoben, was sich meist leistungsmindernd auswirkt.

Bei „TCP New Reno“ hingegen führt eine teilweise Bestätigung (*partial acknowledgement*), d. h. eine Quittung, mit der zwar neue Daten bestätigt werden, jedoch nicht alle bisher gesendeten, zu einer sofortigen Wiederholung des nächsten unbestätigten Segments. Im genannten Beispiel würde der Sender eine solche teilweise Bestätigung als Antwort auf die Wiederholung des ersten verloren gegangenen Segments erhalten. Das CWnd wird dabei nicht weiter reduziert und der Sender bleibt im *Fast-Recovery*-Modus mit dem oben beschriebenen Effekt, dass weitere DupACKs zu einer Vergrößerung des CWnd führen.

Neben den erwähnten Varianten „TCP Tahoe“, „TCP Reno“ und „TCP New Reno“ gibt es noch weitere Formen der Überlaststeuerung (z. B. „TCP Vegas“), die aber eher experimentellen Charakter haben und deswegen hier nicht betrachtet werden.

Darüber hinaus ergeben sich an einigen Stellen Modifikationen der oben beschriebenen Algorithmen, wenn anstelle kumulativer Quittierung eine selektive Bestätigung (*Selective Acknowledgement*, SACK) empfangener Pakete eingesetzt werden [189]. Diese Option ist bereits in vielen TCP-Implementierungen enthalten, erfordert aber eine Unterstützung durch beide Verbindungsendpunkte.

## **Wiederholungsverhalten**

Eine Fragestellung, die in der Literatur kaum thematisiert wird, ist das Wiederholungsverhalten von TCP z. B. nach einem Timeout. In [76] wird vorgeschlagen, grundsätzlich nach einem Timeout nur das nächste unbestätigte Segment zu wiederholen. Sind allerdings mehrere aufeinander folgende Segmente verloren gegangen, kann das z. B. im Fall von „TCP Reno“ dazu führen, dass nur das erste Segment per *Fast Retransmit* wiederholt und für jedes weitere dieser Segmente ein Timeout ausgelöst wird. Die Abstände der Timeouts wachsen dabei aufgrund des Karn-Algorithmus (siehe unten) exponentiell an.

In realen Implementierungen findet sich hingegen häufig eine andere Variante, bei der nach einem Timeout eine Art *Go-Back-N* stattfindet. Dies bedeutet, dass im o. g. Szenario mit der Wiederholung aller bereits gesendeten Segmente ab dem ersten unbestätigten begonnen wird. Ein solches Verhalten führt zwar zu einer schnelleren Erholung im Fall gehäuft auftretender Verluste, hat aber andererseits möglicherweise unnötige Segmentwiederholungen (*false retransmits*) und damit die Erzeugung von Blindlast zur Folge. Die *Go-Back-N*-Strategie kommt vor allem bei „TCP Tahoe“ und „TCP Reno“ zum Tragen, während bei „TCP New Reno“ durch die spezielle Behandlung von teilweisen Bestätigungen in der *Fast-Recovery*-Phase ohnehin deutlich seltener Timeouts auftreten.

### **Rücksetzen nach Inaktivität**

In manchen TCP-Implementierungen wird das CWnd nach einer länger dauernden Inaktivitätsphase auf den Wert  $L_{MSS}$  zurückgesetzt, sodass ein *Slow Start* stattfindet, wenn wieder Daten zu senden sind [142]. Dieser Mechanismus wird auch als *Slow Start Restart* (SSR) bezeichnet und ist vor allem von Bedeutung, wenn innerhalb einer TCP-Verbindung einzelne Bursts mit dazwischen liegenden Pausen gesendet werden. Abgesehen davon, dass nicht alle TCP-Implementierungen dieses Verfahren unterstützen, gibt es auch unterschiedliche Varianten, um eine Inaktivitätsphase festzustellen.

### **Fensterbeschränkungen**

Da in der Realität der Empfangspuffer eine endliche Größe hat (die i. d. R. im Bereich von 8 KByte bis 64 KByte liegt), wird über das RWnd auch das Sendefenster begrenzt. Der gleiche Effekt wird durch eine Beschränkung des CWnd erzielt, wobei beachtet werden muss, dass während der *Fast Recovery* durch das künstliche Aufweiten des CWnd beim Eintreffen von DupACKs auch Werte möglich sind, die über dem Maximalwert liegen.

Die Fensterbeschränkung ist insofern von Bedeutung, als sie eine Begrenzung des Durchsatzes bewirkt, falls die maximale Fenstergröße kleiner ist als das vom Systemmodell abhängige Bandbreite-Verzögerungsprodukt. Umgekehrt bedeutet dies, dass das Systemmodell entsprechend angepasst werden muss, wenn dieser Fall nicht eintreten soll.

### **RTT-Messung**

Um den Wiederholungstimer beim Aussenden von Daten zu setzen, wird auf Messungen der RTT zurückgegriffen [142]. Dabei kommt ein von Karn vorgeschlagener und auch nach ihm benannter Algorithmus zum Einsatz. Dieser verhindert einerseits, dass der Schätzwert für die RTT von wiederholten Segmenten verfälscht wird. Andererseits gewährleistet er durch die Multiplikation des RTT-Schätzwerts mit einem bei aufeinander folgenden Timeouts exponentiell ansteigenden Faktor (*exponential backoff*) eine angemessene Reaktion auf einen sprunghaften Anstieg der RTT [150]. Da allerdings keine exakte Spezifikation des Karn-Algorithmus

existiert, ist nicht klar, welche Segmente genau in eine RTT-Messung mit einbezogen werden und wann ein Rücksetzen des Backoff-Wertes erfolgt. Somit ist die hier verwendete Implementierung nur eine von mehreren möglichen.

Hinzu kommen noch weitere Freiheitsgrade, die sich bei der Wahl einer Obergrenze für den Wiederholungs-Timer und den Backoff-Wert sowie in Bezug auf die Granularität des Timer-Wertes ergeben.

### **SWSA- und Nagle-Algorithmus**

Zur Vermeidung einer Vergrößerung des Sendefensters in unnötig kleinen Intervallen (*silly window syndrome avoidance*, SWSA) und somit unnötig kleiner Pakete werden sowohl auf Sender- als auch auf Empfängerseite spezielle Maßnahmen getroffen [45], die hier nicht näher beschrieben werden.

Ähnliche Bedeutung hat der mit der senderseitigen SWSA eng verknüpfte Nagle-Algorithmus [201]. Er bewirkt, dass auf der Senderseite kleine Dateneinheiten, die von der darüber liegenden Applikation kommen, nicht als getrennte Segmente gesendet werden, sondern nach Möglichkeit zu einem Segment mit Maximalgröße zusammengefasst werden. Nur wenn nach Ablauf eines Timers die Bedingungen gemäß SWSA- und Nagle-Algorithmus immer noch nicht erfüllt sind, wird ein kleineres Segment gesendet.

Während SWSA fester Bestandteil realer Implementierungen ist, kann es in manchen Fällen (z. B. bei Terminal-Anwendungen mit harten Echtzeitanforderungen) vorteilhaft sein, wenn der Nagle-Algorithmus ausgeschaltet ist. Deshalb wird auch in dem hier eingesetzten Simulationsmodell nur der SWSA-Algorithmus verwendet. Bei den Szenarien, die im Rahmen dieser Arbeit untersucht werden, hat allerdings insbesondere der Nagle-Algorithmus höchstens einen vernachlässigbaren Einfluss auf die Leistungsfähigkeit.

### **Verzögerte Quittierung**

Ein Mechanismus, der einen nicht unerheblichen Einfluss auf TCP-Leistungsparameter haben kann, ist das verzögerte Versenden und Zusammenfassen von Quittungen (*delayed acknowledgements*, *Delayed ACK*) auf der Empfängerseite [45]. Bestätigungen werden dabei – vereinfacht ausgedrückt – nur für jedes zweite Segment maximaler Größe oder nach Ablauf einer Zeitüberwachung beim Empfänger gesendet. Dadurch wird die Anzahl von Quittungspaketen verringert, was vor allem im Fall von Terminal-Anwendungen vorteilhaft ist. Andererseits hängt auch die Entwicklung des CWnd auf der Senderseite von der Anzahl der empfangenen Bestätigungen ab. Dies bedeutet, dass bei Verwendung von *Delayed ACK* das CWnd wesentlich langsamer ansteigt und somit eher eine geringere Leistung resultiert (siehe auch Anhang A).

## Initialwerte

Im Fall von kurzen TCP-Verbindungen haben die Initialwerte für die Größen von  $CW_{nd}$ ,  $SSThresh$  und Wiederholungstimer-Intervall nach dem Beginn der Verbindung einen nicht zu vernachlässigenden Einfluss auf die Leistung. Dies gilt vor allem für den Startwert  $w$  von  $W_C$ , der gemäß [5] nicht größer als  $2 \cdot L_{MSS}$  sein darf [5]. Untersuchungen in [213] zeigen, dass die allermeisten der dort betrachteten Server  $w = 2 \cdot L_{MSS}$  als Initialwert verwenden, es aber auch einige Server gibt, bei denen  $W_C$  am Verbindungsbeginn auf  $L_{MSS}$  gesetzt wird, und sogar größere Initialwerte als  $2 \cdot L_{MSS}$  in der Realität auftreten.

Im Falle von  $SSThresh$  spielt der Initialwert keine Rolle, wenn er über der Obergrenze des Sendefensters liegt, die sich wie oben beschrieben aus dem Minimum von Empfangspuffergröße und Maximalgröße des  $CW_{nd}$  ergibt. Wie in [5] angedeutet, ist in realen Implementierungen diese Bedingung meist erfüllt, da diese  $SSThresh$  auf das beim Verbindungsaufbau mitgeteilte Empfangsfenster, das zu diesem Zeitpunkt der Größe des Empfangspuffers entspricht, setzen.

Da zum Beginn der Übertragung noch keine RTT-Messwerte vorliegen, muss der Wiederholungstimer zunächst mit einem vorgegebenen Wert gesetzt werden. Drastische Auswirkungen hat es, wenn dieser Wert geringer ist als die tatsächlich auftretende RTT, da dann bereits für das erste Paket ein Timeout mit anschließender Paketwiederholung und Reduktion von  $CW_{nd}$  und  $SSThresh$  erfolgt. Andererseits führt ein großer Wert im Fall eines Verlustes des ersten Pakets dazu, dass eine Wiederholung erst sehr spät erfolgt. In der Realität auftretende Initialwerte sind implementierungsabhängig, liegen aber i. d. R. im Bereich weniger Sekunden.

## Paketgröße

In realen Netzen dominieren zwei Größen für die maximale Segmentgröße. Der am häufigsten vorzufindende Wert ist  $L_{MSS} = 1460$  Byte, was sich aus der maximalen Länge des Nutzdatenteils in einem Ethernet-Rahmen abzüglich 40 Byte für IP- und TCP-Steuerinformation im Paketkopf ergibt. Vielfach findet man außerdem – wie in einigen RFCs empfohlen –  $L_{MSS} = 536$  Byte, was zuzüglich TCP/IP-Paketkopf in einer IP-Paketlänge von 576 resultiert [191, 266].

Allerdings ist der Nutzen einer Übernahme einer Mischung von Implementierungen mit diesen Werten in ein Modell fraglich, da dadurch Konfiguration des Modells und Interpretation der Ergebnisse eher erschwert werden. Als Alternative bietet sich an, wie z. B. bei dem von vielen Forschern eingesetzten Simulator ns-2 [49] von einem abstrakten Wert von 1000 Byte für die maximale Segmentgröße auszugehen.

Ebenso kann auch der Verzicht auf eine explizite Berücksichtigung der Paketverlängerung durch den TCP/IP-Kopfteil eine sinnvolle Abstraktion darstellen.

**Tabelle 4.1:** Standardwerte für wichtige Optionen und Parameter von TCP in dem in dieser Arbeit verwendeten Simulationsmodell

Option/Parameter	Standardwert
Flusssteuerung	byteorientiert
SACK	nein
Überlaststeuerung	New Reno
SWSA-Algorithmus	ja
Nagle-Algorithmus	nein
<i>Delayed ACK</i>	nein
Karn-Algorithmus	ja
<i>Go-Back-N</i> nach Timeout	ja
SSR nach Inaktivitätsphase	ja (wie in [142])
$L_{MSS}$	1 000 Byte
Berücksichtigung des Paketkopfs	nein
Empfangspuffergröße	32 000 Byte
Timer-Granularität	200 ms
senderseitiger SWSA-Timer-Wert	200 ms
<i>Delayed ACK</i> -Timer-Wert	200 ms
Maximalwert Wiederholungstimer	10 000 ms
maximaler Backoff-Wert	64
Initialwert des CWnd	$2 \cdot L_{MSS}$
Initialwert SStresh	$32 \cdot L_{MSS}$
Initialwert RTT-Schätzung	1 000 ms

### Zusammenfassung

Die Vielzahl der Optionen wirft das Problem eines nahezu unüberschaubaren Parameterraums auf. Hinzu kommt eine gewisse Unsicherheit bzgl. der Implementierung bestimmter Algorithmen (z. B. der RTT-Messung), da diese in den zugehörigen RFCs nicht exakt spezifiziert sind. Daraus ergeben sich weitere Unterschiede zwischen TCP-Implementierungen verschiedener Hersteller, die zudem noch teilweise fehlerhaft sind [221]. Insgesamt folgt also eine überaus große Menge an TCP-Varianten, die in einem Modell unmöglich wiedergegeben werden können. Im Rahmen der vorliegenden Arbeit erfolgt daher eine Fixierung der meisten Parameter und eine Festlegung auf bestimmte Implementierungsvarianten. Eine Übersicht über die wichtigsten TCP-Optionen und -Parameter und deren Standardwerte gibt Tabelle 4.1.

#### 4.2.2.2 Ungesättigte TCP-Quellen

Die überwiegende Zahl von TCP-Untersuchungen, die in der Literatur zu finden sind, basieren auf einem einfachen Modell für die auf TCP aufsetzende Anwendungsebene, bei dem auf der Senderseite immer Daten bereitstehen. Es handelt sich somit um ungesättigte Quellen (*greedy sources*) auf der Grundlage permanenter TCP-Verbindungen.<sup>5</sup>

Mitunter wird ein solches Modell auch als FTP-Modell bezeichnet, was aber aus zwei Gründen irreführend ist. Einerseits werden dabei die charakteristischen Eigenschaften von FTP selbst, wie z. B. der Aufbau eines separaten Kontrollkanals, gar nicht berücksichtigt, andererseits sind auch TCP-Verbindungen in FTP genauso wie im Fall von WWW dynamisch, d. h. innerhalb einer TCP-Verbindung wird nur eine begrenzte Datenmenge übertragen. Allerdings ist aus Messungen bekannt, dass Dateien, die über FTP übertragen werden, tatsächlich tendenziell größer sind als Objekte, die im WWW abgerufen werden [266], sodass diese Bezeichnung in gewissem Sinn gerechtfertigt ist.

Während in analytischen Modellen häufig eine einzelne permanente TCP-Verbindung in Kombination mit einem stark vereinfachten Netzmodell betrachtet wird [212], basieren simulative Untersuchungen zumeist auf einer Überlagerung einer festen Anzahl von ungesättigten TCP-Quellen, unter denen die im Netz verfügbare Bandbreite aufgeteilt wird.

Ein Nachteil eines solchen Überlagerungsmodells ist dessen deterministischer Charakter. Dieser rührt daher, dass sämtliche TCP-Algorithmen deterministischer Natur sind und sich daher im Falle eines Netzmodells ohne stochastische Komponenten ein vorhersehbarer Ablauf von Ereignissen (z. B. Paketankünfte/-verluste) ergibt. Prinzipiell bedeutet dies, dass in Bezug auf den Gesamtsystemzustand auch Zyklen vorkommen können, was die Gültigkeit der aus einer Simulation abgeleiteten statistischen Größen (z. B. mittlerer Durchsatz) in Frage stellen würde. Andererseits scheint es gerechtfertigt, das System aufgrund des immens großen Zustandsraums als pseudozufällig zu betrachten, was wiederum die Anwendung statistischer Auswertung erlaubt.

Eine einfache Möglichkeit, um wenigstens die durch einen gemeinsamen Sendebeginn verursachte Gefahr einer Synchronität der überlagerten TCP-Quellen zu vermeiden, ist eine Verschiebung des Sendebeginns jeder Quelle um ein zufälliges Zeitintervall. Dabei bietet sich z. B. eine Gleichverteilung dieses Offsets an, deren Obergrenze so gewählt ist, dass am Ende der Warmlaufphase alle Quellen aktiv sind.

Darüber hinaus gibt es weitere Möglichkeiten, Zufall in das Modell zu bringen. So können in Anlehnung an das Phasensprungverfahren bei Modellen für periodische Quellen [61] Mechanismen eingesetzt werden, bei denen in großen Abständen kurze Aktivitätspausen eingestreut werden. Hierbei ist entweder der Abstand der Pausen oder die Pausendauer als Zufallsvariable

---

<sup>5</sup> Gelegentlich wird auch die Bezeichnung persistente Quellen (*persistent sources*) verwendet.

mit bestimmter Verteilung (z. B. Gleichverteilung) modelliert. Eine Anwendung auf ein TCP-basiertes Modell ist allerdings im Vergleich zu den angesprochenen periodischen Quellen etwas aufwändiger, da sich solche Pausen nur auswirken, wenn sie dazu führen, dass im Falle einer Sendeberechtigung (d. h. eines ausreichend großen Sendefensters) für kurze Zeit keine ungesendeten Daten im Sendepuffer vorliegen.

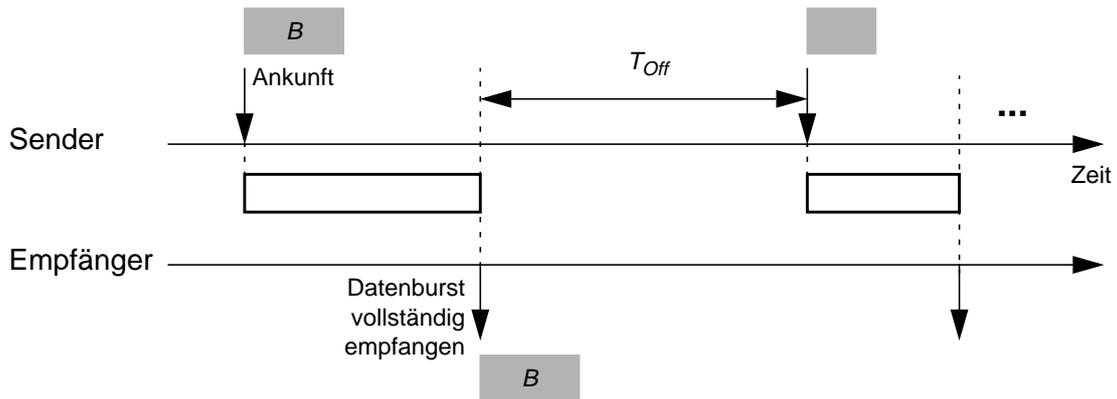
#### **4.2.2.3 Modelle für dynamischen TCP-Verkehr**

Während das im vorigen Abschnitt skizzierte Modell mit ungesättigten TCP-Quellen von sehr langen Verbindungen ausgeht, ist bekannt, dass ein Großteil des TCP-basierten Verkehrs auch einen wesentlichen Anteil von TCP-Verbindungen mit kurzer Dauer beinhaltet. Dies gilt insbesondere für WWW-Verkehr, der heute den größten Anteil am IP-Verkehrsaufkommen ausmacht [191, 266, 269]. Das vorgestellte Modell ist daher für die Modellierung von WWW-Verkehr eher ungeeignet. Statt dessen bieten sich zunächst zwei Modellvarianten an, in denen TCP-Transfers mit endlichem Datenvolumen, wie sie im Falle von WWW-Verkehr anzutreffen sind, berücksichtigt werden.

#### **Überlagerung von TCP-Quellen mit büschelartigem Verkehr**

In einer ersten Variante gibt es wie bei dem in Abschnitt 4.2.2.2 vorgestellten Modell eine statische Anzahl von TCP-Verbindungen. Innerhalb der TCP-Verbindungen werden allerdings einzelne Datenbursts übertragen, unterbrochen von längeren Ruhephasen. Ein Modell einer einzelnen Verbindung, das eine starke Ähnlichkeit mit der in Abschnitt 4.2.1.2 vorgestellten On-Off-Quelle aufweist, kann z. B. als Repräsentant für eine WWW-Sitzung gesehen werden. Ein Datenburst kann dabei eine einzelne HTML-Seite oder ein darin referenziertes Objekt (z. B. eine Bilddatei) repräsentieren. Allerdings wird bei dieser Interpretation davon ausgegangen, dass die durch einen Seitenabruf ausgelösten Transfers nicht über parallele TCP-Verbindungen abgewickelt werden. Ob es sich tatsächlich um eine durchgehende TCP-Verbindung oder um nacheinander auf- und wieder abgebaute Verbindungen handelt, spielt allerdings auch nur eine untergeordnete Rolle, da der Verbindungsauf und -abbau im Modell nicht berücksichtigt wird.

Ausgehend von dem dargestellten Anwendungsfall ist es sinnvoll, dass im Modell eine Ruhephase dann beginnt, wenn der vorhergehende Datentransfer abgeschlossen ist, die Daten also komplett beim Empfänger eingetroffen sind (Bild 4.4). Die Burstgröße  $B$  und die Dauer der Ruhephase  $T_{Off}$  sind im einfachsten Fall unabhängige Zufallsvariablen, durch deren Verteilungen das Modell parametrisiert wird. Ein derartiges Modell wird zum Beispiel in [218] eingesetzt, wo in Anlehnung an Messungen [81] eine Pareto-Verteilung für  $B$  angenommen wird, was zu selbstähnlichem Verkehr führt.



**Bild 4.4:** TCP-Quelle mit büschelartigem Verkehr

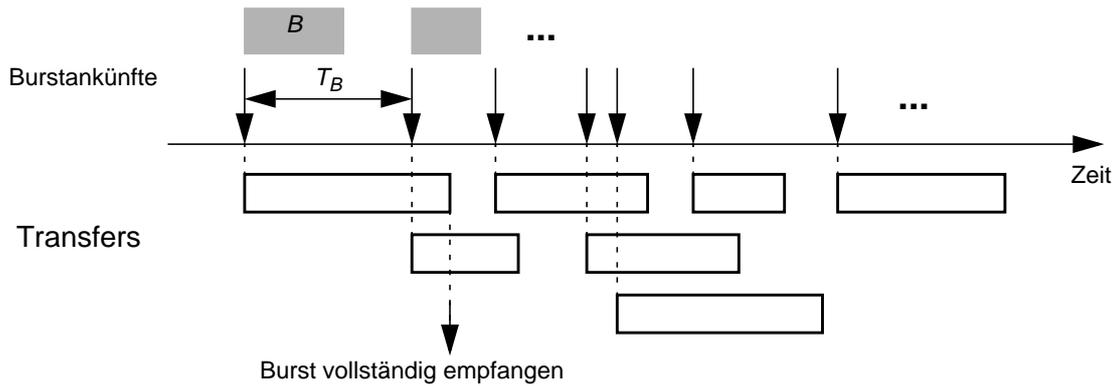
Problematisch ist bei diesem Modell, dass die mittlere Datenrate nicht wie etwa beim On-Off-Modell gemäß Gleichung (4.5) über die Parameter eingestellt werden kann. Somit ist es auch nicht möglich, ein Angebot für das darunter liegende System anzugeben. Der Grund liegt darin, dass die Dauer einer Burstübertragung zwar mit der Burstgröße zusammenhängt, gleichzeitig aber eine starke Abhängigkeit vom aktuellen Lastzustand im Netz besteht. Insbesondere hängt die Burstdauer und damit die mittlere Rate einer einzelnen Quelle von der Gesamtzahl an Quellen ab, die auf einem Verbindungsabschnitt um die verfügbare Bandbreite konkurrieren.

### M/G/∞-Modell für aggregierten TCP-Verkehr

In Abschnitt 4.2.1.2 wurde gezeigt, dass die Überlagerung sehr vieler On-Off-Quellen asymptotisch zu einem PBS führt. Entsprechend kann auch hier das Ankunftsverhalten von Datenbüscheln aus einer großen Zahl von TCP-Verbindungen als PBS modelliert werden, wobei im Unterschied zu dem Modell in Abschnitt 4.2.1.2 die „Aufteilung“ in einzelne Pakete durch TCP vorgenommen wird. Dieses TCP-Lastmodell (*workload model*), das erstmals in [222] vorgeschlagen wird, ist somit durch die Verteilungen der Burstgröße  $B$  sowie des Ankunftsabstandes  $T_B$  von Bursts charakterisiert (Bild 4.5).

Ein großer Vorteil dieses Modells gegenüber der oben beschriebenen Überlagerung einzelner TCP-Quellen mit büschelartiger Verkehrscharakteristik liegt neben der reduzierten Parameterzahl (an die Stelle der Verteilung der Ruhedauer und der Anzahl der Quellen tritt die Verteilung des Zwischenankunftsabstandes) in der bereits durch die Erwartungswerte von  $B$  und  $T_B$  gemäß Gleichung (4.7) festgelegten mittleren Datenrate. Daraus kann dann z. B. im Fall eines zentralen Links mit Bedienrate  $C$  als Systemmodell ein Verkehrsangebot<sup>6</sup>

<sup>6</sup>Das Verkehrsangebot ist ursprünglich auf Systeme ohne Rückkopplung bezogen. Eine Anwendung dieses Begriffes im Zusammenhang mit den hier beschriebenen TCP-Modellen ist jedoch üblich. Allerdings wird dabei nicht berücksichtigt, dass sich die Verkehrslast durch unnötige Wiederholungen erhöhen kann.



**Bild 4.5:** Modell für aggregierten dynamischen TCP-Verkehr

$$A = \frac{E[B]}{C \cdot E[T_B]} \quad (4.21)$$

abgeleitet werden. Dies bedeutet, dass sich Stationarität nur für  $A < 1$  ergibt, während ein höheres Angebot zu einem stetigen Anwachsen der Anzahl von Verbindungen und damit zu einem gegen null gehenden Durchsatz für jede Einzelverbindung führt.

Um das Modell trotzdem auch für stationäre Überlastuntersuchungen verwenden zu können, wird zusätzlich eine Begrenzung  $n_{max}$  der Gesamtzahl aktiver Verbindungen eingeführt, d. h. eine ankommende Verbindung wird nur akzeptiert, wenn zu diesem Zeitpunkt weniger als  $n_{max}$  Verbindungen bereits bestehen. Der Wert wird dabei so gewählt, dass im Fall  $A < 1$  die Zahl der Verbindungen diesen Wert nur mit vernachlässigbarer Wahrscheinlichkeit erreicht. Für  $A \geq 1$  repräsentiert  $A$  gemäß Gleichung (4.21) dann aber nicht mehr das Angebot, da die Ankunftsabstände tatsächlich akzeptierter Verbindungen sich dem Netzzustand anpassen und daher nicht mehr von einem Verkehrsangebot gesprochen werden kann.

Wie das Modell in Abschnitt 4.2.1.2 kann auch dieses Modell als Repräsentant für einen aggregierten Verkehrsstrom angesehen werden. Im Gegensatz zu dem oben erwähnten TCP-Quellmodell wird die Zuordnung zu einer Sitzung, Anwendung oder einem Nutzer hier nicht berücksichtigt, da angenommen wird, dass diese in einem aggregierten Verkehrsstrom nicht mehr sichtbar ist. Noch viel weniger als beim statischen Modell spielt es damit eine Rolle, ob nach Ablauf von  $T_B$  tatsächlich eine neue TCP-Verbindung ankommt oder ob nur über eine bereits existierende, aber momentan inaktive TCP-Verbindung ein neuer Datenblock übertragen wird.<sup>7</sup> Somit kann ein Burst sowohl eine per FTP übertragene Datei als auch eine HTML-Textseite oder ein eingebettetes Objekt (z. B. eine Graphik- oder Audiodatei) im Fall eines

<sup>7</sup> Diese und die nachfolgenden Aussagen gelten nur unter der Voraussetzung, dass ein SSR nach Inaktivität gemäß Abschnitt 4.2.2.1 durchgeführt wird, wovon im Rahmen dieser Arbeit ausgegangen wird. Hinzu kommt die Einschränkung, dass im Fall von SSR das CWnd stets auf  $L_{MSS}$  gesetzt wird, was nicht zwangsläufig dem Initialwert des CWnd beim Verbindungsstart entspricht.

WWW-Seitenabrufs repräsentieren. Charakteristisch ist, dass nach jeder Burst-Ankunft ein *Slow Start* durchgeführt wird. Insbesondere dann, wenn die Größe des zu übertragenden Bursts gering ist, kann beobachtet werden, dass Verbindungen die *Slow-Start*-Phase häufig gar nicht verlassen.

In Anlehnung an Messungen von FTP- und WWW-Verkehr [63, 81, 183, 209, 222, 223] kann für  $B$  eine Pareto-Verteilung angenommen werden, wodurch, wie in Abschnitt 4.2.1.3 diskutiert, der Verkehr die Eigenschaft der Selbstähnlichkeit erhält. Assoziiert man die Bursts mit einzelnen Objekten einer WWW-Seite, erhält man aus den Messungen, die sich auf WWW-Verkehr konzentrieren, Werte für die mittlere Burstgröße im Bereich von einigen KByte.

Weitaus weniger Übereinstimmung als bei der Burstgrößenverteilung herrscht allerdings im Hinblick auf die Annahme, dass die Ankunftszeitpunkte von TCP-Verbindungen durch einen Poisson-Prozess beschrieben werden können. Die Annahme der Poisson-Ankunft wird zwar in [222] anhand von Messungen gerechtfertigt, allerdings wird dort nur Telnet- und FTP-Verkehr zugrunde gelegt. Im Gegensatz zu FTP wird jedoch bei WWW-Verkehr die TCP-Verbindungs-Charakteristik nicht nur vom Nutzerverhalten, sondern viel mehr noch von der Struktur der abgerufenen Seite und vom Browser-Verhalten bestimmt. In [98] wird mit Hilfe von Messungen auf Weitverkehrslinks gezeigt, dass im Fall von aggregiertem WWW-Verkehr die Poisson-Annahme für TCP-Verbindungen nicht mehr gilt, sondern dass auch auf der Ebene der Verbindungsankünfte Selbstähnlichkeit vorliegt. Dabei muss allerdings betont werden, dass die statistische Aussagekraft von Ergebnissen bzgl. der Verbindungsebene gegenüber denen auf der Paketebene deutlich reduziert ist, d. h. entweder ist die Zahl der Stichproben nicht allzu hoch oder das Messintervall ist so groß, dass Instationarität nicht mehr ausgeschlossen werden kann. Die Selbstähnlichkeit wird darauf zurückgeführt, dass die Anzahl von Objekten, die im Rahmen einer WWW-Sitzung übertragen werden, einer Heavy-Tail-Verteilung gehorcht, während die Anfangszeitpunkte von Sitzungen weiterhin durch einen Poisson-Prozess beschrieben werden können. Die Messungen und Untersuchungen in [73, 209] bestätigen diese Erkenntnisse insoweit, als dass auch dort festgestellt wird, dass die empirische Verteilung der Zwischenankunftszeiten von TCP-Verbindungen am besten durch eine Weibull-Verteilung und nicht durch eine negativ-exponentielle Verteilung approximiert werden kann. Allerdings wird in [209] nur eine geringe Autokorrelation der Zwischenankunftszeiten beobachtet, was nicht auf selbstähnlichen Verkehr schließen lässt. Das Modell in [73] liefert für höhere Ankunftsraten von TCP-Verbindungen einen Form-Parameter der Weibull-Verteilung nahe eins, sodass auch hier der Unterschied zur Poisson-Ankunft nicht sehr groß ist.

Insgesamt ergeben die genannten Untersuchungen damit ein eher uneinheitliches Bild, das Zweifel an der Widerlegung der Poisson-Annahme für den Verbindungsankunftsprozess hinterlässt. Abgesehen davon weisen Bonald et al. in [40] darauf hin, dass die Leistungsparameter in einem idealisierten, aber durchaus realitätsnahen System, bei dem die Linkbandbreite unter

den existierenden TCP-Verbindungen fair aufgeteilt wird, insensitiv gegenüber der genauen Charakteristik des TCP-Verbindungs-Ankunftsprozesses ist, solange der Ankunftsprozess auf einer übergeordneten Sitzungsebene ein Poisson-Prozess ist. Das  $M/G/\infty$ -Modell zur Beschreibung des TCP-Verbindungsprozesses kann somit als repräsentativ angesehen werden und wird daher auch im Rahmen dieser Arbeit zur Modellierung von dynamischem TCP-Verkehr verwendet.

#### 4.2.2.4 Anwendungs- und nutzerbezogene WWW-Verkehrsmodelle

Während die im letzten Abschnitt genannten Modelle die Verkehrscharakteristik oberhalb von TCP in eher allgemeiner Weise modellieren, sind insbesondere für WWW-Verkehr aus der Literatur zahlreiche Modelle bekannt, die versuchen, die Vorgänge innerhalb einer Nutzersitzung mehr oder weniger exakt im Modell wiederzugeben [23, 66, 69, 98, 183]. Dabei gilt es – häufig gestützt durch Messungen – sowohl das Verhalten von Anwendungen (z. B. WWW-Server und -Browser) als auch das der Nutzer (z. B. deren Ungeduld [41]) nachzubilden. Berücksichtigt werden in solchen WWW-Anwendungs- und Nutzermodellen neben dem eigentlichen Transfer von Objekten meist auch weitere Komponenten wie DNS-Abfrage, TCP-Verbindungsaufbau, Bearbeitung der Anfrage im WWW-Server und Darstellung der Ergebnisse durch den Browser.

Solche anwendungsbezogenen Modelle weisen jedoch gegenüber eher universellen Modellen wie den in Abschnitt 4.2.2.3 vorgestellten einige Nachteile auf:

- Eines großes Problem ist die wachsende Diversifizierung der Modelllandschaft. Allein im Beispiel WWW existiert eine Vielzahl unterschiedlicher Anwendungen und Anwendungsversionen auf Client- und Server-Seite mit teilweise signifikant unterschiedlichen Eigenschaften. Darüber hinaus ist festzuhalten, dass neben WWW noch andere Anwendungen und Dienste, z. B. E-Mail, Dateitransfer, Rechnernetzung und in weiten Teilen auch *Streaming Media*, elastischen Verkehr produzieren.
- Zu den bereits zahlreichen Optionen, die auf der Ebene des Transportprotokolls vorzufinden sind (TCP-Parameter), kommen zahlreiche Modellparameter der Nutzer- und Anwendungsebene. Dies liegt nicht nur an der angesprochenen Vielfalt von Anwendungen, sondern häufig bereits an der Komplexität der Anwendermodelle selbst. Insgesamt wird dadurch die Problematik eines im Rahmen von Untersuchungen kaum mehr zu überschauenden Parameterraums weiter vergrößert.
- Hinzu kommt die Schwierigkeit, aus Messungen repräsentative Werte für die verschiedenen Parameter zu gewinnen, was vor allem bei Parametern problematisch ist, die das Nutzerverhalten wiedergeben (z. B. Betrachtungszeit für eine WWW-Seite). Die in der Literatur zu findenden Modelle basieren i. d. R. auf unterschiedlichen Heuristiken, mit deren Hilfe von gemessenen Größen auf Werte von Modellparametern geschlossen wird.

- Trotz der Komplexität mancher Modelle werden dort Korrelationen einzelner Parameter (z. B. Umfang einer HTML-Seite und anschließende Betrachtungszeit) selten berücksichtigt. Dies liegt zum einen daran, dass diese Korrelationen grundsätzlich sehr schwierig auf ein Modell abzubilden sind. Zum anderen sind bislang hierzu kaum Ergebnisse von Messungen bekannt.
- Selbst wenn es gelingt, einen WWW-Nutzer und die Client-Anwendung detailgetreu zu modellieren, bleibt die Frage, welcher Teil des von diesem Nutzer erzeugten Verkehrs an einer bestimmten Stelle im Netz überhaupt sichtbar ist. Dies liegt zum einen daran, dass an einer Nutzersitzung (insbesondere im Fall von WWW) häufig mehr als ein Server beteiligt ist. Zum anderen bewirken Caching-Mechanismen, dass nicht jede Datei bei einer Anfrage über das Netz geladen werden muss. Die Frage muss vor allem vor dem Hintergrund eines – berechtigterweise – eher einfachen Netzmodells gesehen werden, bei dem z. B. nur ein Engpasslink modelliert wird.

Insgesamt bleibt festzuhalten, dass WWW-Anwendungs- und Nutzermodelle häufig sehr komplex und wenig robust gegenüber bereits kleinen Änderungen von Randbedingungen wie neuen Programm- oder Protokollversionen sind. Hinzu kommt, dass der eigentliche Vorteil solcher Modelle, nämlich die Nähe zur Wirklichkeit, erst bei der Definition von darauf angepassten Leistungsmetriken (siehe Abschnitt 4.3) zum Tragen kommt.

### **4.2.3 Modelle für echtzeitkritischen IP-Verkehr**

Neben elastischem Verkehr wird in zukünftigen IP-Netzen auch verstärkt Echtzeitverkehr eine Rolle spielen, der somit auch bei der Modellierung mitberücksichtigt werden muss. Wie in Kapitel 2 angesprochen, gibt es bei Echtzeitanwendungen große Unterschiede bzgl. der Härte von Echtzeitanforderungen. Insbesondere bei *Streaming-Media*-Applikationen sind auch relativ große Verzögerungen noch tolerierbar, was dazu führt, dass diese Anwendungen teilweise als TCP-basierte Lösungen realisiert werden können. In diesem Fall ist eine Abgrenzung zu Datenanwendungen, die elastischen Verkehr produzieren, kaum möglich. Das Problem wird dadurch erschwert, dass zu der Adaptivität auf der Transportebene oft noch Anpassungsmechanismen auf der Anwendungsebene kommen, die zudem für unterschiedliche Applikationen und Implementierungen sehr weit auseinander liegen.

Aber auch in Bezug auf interaktive Anwendungen wie Sprach- oder Bildtelefonie, die meist auf der Verwendung von UDP als Transportprotokoll und RTP zur Synchronisation beruhen, ist es nicht möglich, ein allgemeingültiges Modell anzugeben, das alle Eigenschaften wiedergibt. Dies gilt insbesondere wieder im Hinblick auf adaptive Anwendungen, bei denen z. B. je nach Netzzustand unterschiedliche Audio- oder Videokompressionsalgorithmen eingesetzt werden [246]. Da solche adaptiven Multimedia-Anwendungen zum gegenwärtigen Zeitpunkt

jedoch noch kaum Verbreitung gefunden haben, ist es gerechtfertigt, diese im Rahmen der vorliegenden Arbeit unberücksichtigt zu lassen.

Allerdings ist selbst bei der Beschränkung auf nicht-adaptive Anwendungen die Palette immer noch groß. Zunächst muss zwischen Audio- und Videoverkehr unterschieden werden, deren Charakteristik wiederum stark vom jeweils verwendeten Codec geprägt ist. Die Nachbildung der Codec-Eigenschaften bei Videoanwendungen hat in der Vergangenheit zur Entwicklung zahlreicher Modelle mit oft mehreren Hierarchieebenen geführt [94, 236, 240, 258], wobei meist MPEG-codierter Videoverkehr im Mittelpunkt des Interesses stand. Allerdings ist fraglich, inwieweit eine solch detaillierte Modellierung wirklich Auswirkungen auf die relevanten Leistungsmaße hat. Bei Audioanwendungen stellt sich vor allem die Frage, ob der Codec mit Sprachpausenunterdrückung arbeitet. In diesem Fall weist der Verkehr eine On-Off-Charakteristik auf, während sich beim Verzicht auf Sprachpausenunterdrückung ein konstantratiger Verkehrsstrom ergibt.

In dieser Arbeit sollen zur Modellierung von Echtzeitanwendungen weitgehend nur einfache Burstebenen-Modelle herangezogen werden, insbesondere das in Abschnitt 4.2.1.2 vorgestellte  $M/G/\infty$ -Modell. Dieser pragmatische Ansatz wird z. B. auch in [203] angewandt. Es zeigt sich, dass die Wahl der Parameter dieses einfachen Modells bereits einen bedeutenden Einfluss auf die Leistungsgrößen hat. Zudem repräsentiert das Modell einen aggregierten Verkehrsstrom, während bei den detaillierteren Modellen für Videoanwendungen die Einzelquelle im Vordergrund steht.

Ein wichtiger Parameter beim  $M/G/\infty$ -Modell ist die Verteilung der Burstgröße. Während eine Interpretation von Bursts als Sprachblöcke eine negativ-exponentielle Verteilung nahelegt [244], wird in [203] mit Bezug auf Messungen eine Pareto-Verteilung gewählt. Neben diesen Verteilungen kommt außerdem im Rahmen der Arbeit eine empirische Verteilung zum Einsatz, die aus Labormessungen mit Videokonferenzanwendungen auf Basis des Codecs H.261 gewonnen wurde [86].

### 4.3 Leistungskenngrößen

Im vorhergehenden Abschnitt wurde auf die Bedeutung der Verkehrsmodellierung bei der Leistungsuntersuchung von IP-Netzen hingewiesen. Ebenso wichtig ist die Verwendung aussagekräftiger Leistungsmetriken zur Beurteilung der Leistungsfähigkeit. Im Vergleich zur Verkehrsmodellierung wird der Definition von Metriken in der Literatur allerdings weit weniger Aufmerksamkeit gewidmet.

Wie bei der Verkehrsmodellierung ist auch bei der Wahl von Leistungs- und Dienstgütemetriken für IP-Verkehr eine Betrachtung auf verschiedenen Ebenen der Kommunikation erforderlich. Das Ziel ist dabei, Maße zu finden, welche die vom Anwender wahrgenommene Dienst-

güte möglichst gut widerspiegeln. Gleichzeitig sollen die Maße aber auch als Optimierungskriterien (z. B. aus der Sicht eines Netzbetreibers) anwendbar sein. Somit spielen auch die Messbarkeit in einem realen Netz sowie die Möglichkeit, die jeweiligen Effekte und Zusammenhänge auf den verschiedenen Ebenen der Kommunikation zu erklären, eine Rolle.

In den folgenden Abschnitten sollen in Anlehnung an [32] Metriken auf unterschiedlichen Ebenen vorgestellt werden, angefangen von der Paketebene bis hinauf zur Anwendungs- und Nutzerebene. Insbesondere in letzterem Fall ist festzustellen, dass die Metriken eng verknüpft sind mit bestimmten Modellen. Dies heißt einerseits, dass solche Metriken nur dann betrachtet werden können, wenn ein entsprechendes Verkehrsmodell einer höheren Ebene bei der Leistungsuntersuchung zum Einsatz kommt, was entsprechend den Ausführungen in Abschnitt 4.2 auch mit Nachteilen verbunden sein kann. Andererseits spricht es auch dafür, Verkehrsmodelle zu verwenden, die über eine Betrachtung von Paket- und Büschelebene hinausgehen, wenn dadurch eine Beurteilung der Qualität aus Nutzersicht erst möglich wird.

Ähnlich wie bei der Verkehrsmodellierung kann oberhalb der Paketebene (Abschnitt 4.3.1) wieder grundsätzlich zwischen elastischem Verkehr (Abschnitt 4.3.2) und echtzeitkritischem Verkehr (Abschnitt 4.3.3) unterschieden werden, wobei diese Trennung wieder aufgehoben wird, wenn es um Möglichkeiten zur Bestimmung einer subjektiven Dienstgüte (Abschnitt 4.3.4) geht. Darüber hinaus wird in Abschnitt 4.3.5 verdeutlicht, dass in vielen Fällen neben den üblichen stationären Maßen auch eine Betrachtung unter Berücksichtigung transienter Vorgänge von Nutzen sein kann.

### 4.3.1 Grundlegende Leistungsmaße

Als grundlegende Leistungsmaße können im Zusammenhang mit paketvermittelnden Netzen solche bezeichnet werden, die sich auf die Paketebene beziehen. Dazu gehören in erster Linie folgende Metriken:

- **Paketverlustwahrscheinlichkeit**<sup>8</sup>: Dabei kann unterschieden werden zwischen Verlusten, die in einem Netzknoten (z. B. einem Router) stattfinden, und solchen, die im Endgerät auftreten (z. B. im Fall von Echtzeitanwendungen in dem Fall, dass ein Paket verspätet ankommt). Darüber hinaus ist eine Unterteilung möglich in Verluste, die aktiv herbeigeführt wurden (z. B. durch spezielle Verwerfungsmechanismen), und Verluste, die zwangsläufig aufgrund von Ressourcenmangel (z. B. Pufferüberlauf) aufgetreten sind.
- **Paketverzögerung**: Statistisch relevante Größen sind Mittelwert, Verteilung und Quantile der Paketverzögerung, die im Wesentlichen aufgeteilt werden kann in die konstante Aus-

---

<sup>8</sup> Streng genommen muss zwischen Verlustwahrscheinlichkeit und Verlusthäufigkeit unterschieden werden, je nach dem, ob es sich um eine a priori bestimmbare Größe oder um einen aus Messungen (a posteriori) gewonnenen Wert handelt. Im Rahmen dieser Arbeit wird aber entsprechend dem in der Literatur üblichen Sprachgebrauch stets von Verlustwahrscheinlichkeit gesprochen.

breitungsverzögerung (*propagation delay*) sowie die variable Verzögerung durch Pufferung in den Netzknoten (*queueing delay*). Hinzu kommen noch eventuelle Verzögerungen in den Endgeräten.

- **Verzögerungsschwankung** (*jitter*): Dieses vor allem im ATM-Umfeld häufig verwendete und dort auch als *Cell Delay Variation* (CDV) bezeichnete Maß kann am einfachsten aus der Verteilung der Paketverzögerung abgeleitet werden. Daneben gibt es auch andere Definitionen der Verzögerungsschwankung, denen bestimmte Messverfahren zugrunde gelegt sind [16].

Außer diesen Metriken sind auf der Paketebene auch systembezogene Maße wie die Auslastung von Links oder Warteschlangenlängen von Bedeutung. Diese sind, obgleich die oben genannten Maße eng mit ihnen zusammenhängen, für den Nutzer nicht direkt sichtbar. Sie spielen aber für den Netzbetreiber eine große Rolle, weil sie Aussagen über Effizienz und Leistungsfähigkeit des Netzes zulassen. In diesem Zusammenhang ist auch die effektive Bandbreite zu nennen. Mit Hilfe dieser auch als äquivalente Bandbreite oder effektive Bedienrate bezeichneten Größe kann angegeben werden, welche Kapazität bereitgestellt werden muss, um einen bestimmten Wert eines primären Leistungsmaßes, z. B. eine vorgegebene Verlustwahrscheinlichkeit, zu erreichen. Damit handelt es sich bei der effektiven Bandbreite um ein Sekundärmaß, das vor allem in Bezug auf Netzdimensionierung, Verbindungsannahmesteuerung und Entgelterhebung von großer Bedeutung ist. Im Gegensatz zu den Primärmaßen kann die effektive Bandbreite in einem System oder einem Simulationsmodell nicht direkt gemessen werden, sondern muss aus dem analytisch oder simulativ bestimmten Zusammenhang zwischen Kapazität und Verlustwahrscheinlichkeit oder Durchlaufzeitverteilung gewonnen werden [31].

Die klassischen Maße, die sich auf die Paketebene beziehen, haben weiterhin ihre Bedeutung. Dies gilt nicht zuletzt deshalb, weil im Netz nur die IP-Schicht sichtbar ist und somit dort nur diese Maße bestimmt und z. B. für adaptive Verkehrsmanagement-Mechanismen genutzt werden können. Außerdem sind diese Maße von einer konkreten Anwendung und einem bestimmten Modell unabhängig und damit universell einsetzbar. Für eine Beschreibung der Leistungsfähigkeit eines Systems reichen sie aber nicht aus, weil sie nur ungeeignet die Dienstgüte widerspiegeln, wie sie vom Nutzer schließlich wahrgenommen wird.

### 4.3.2 Kenngrößen für elastischen Verkehr

Im Fall von elastischem Verkehr, der auf der Verwendung von TCP basiert, ist es einerseits notwendig, die Eigenschaften von TCP wie z. B. seine Adaptivität an den Netzzustand zu modellieren (siehe Abschnitt 4.2.2), andererseits aber auch die Leistung aus der Sicht des TCP-Nutzers anzugeben. Die sich daraus ergebenden Metriken hängen allerdings stark von dem für TCP-Verkehr verwendeten Modell ab. Daher werden im Folgenden die relevanten

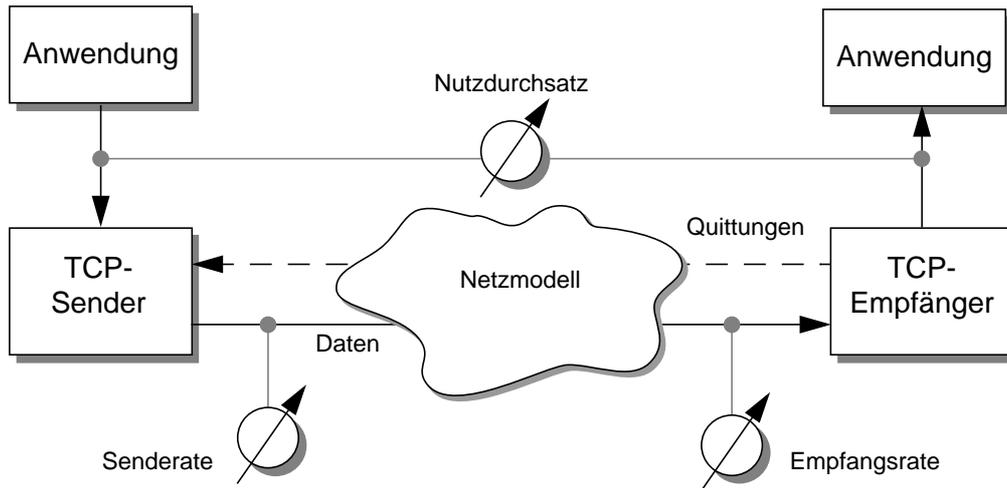
Metriken für die in Abschnitt 4.2.2 beschriebenen statischen und dynamischen TCP-Modelle diskutiert.

#### 4.3.2.1 Ungesättigte Quellen

Das aussagekräftigste Dienstgütemaß auf der Basis eines Modells mit ungesättigten Quellen (siehe Abschnitt 4.2.2.2) ist der mittlere Durchsatz. Zwar kann unter vereinfachenden Annahmen im Hinblick auf das Netzmodell (unkorrelierte Verluste) der Zusammenhang zwischen mittlerem Durchsatz und Metriken der Paketebene (Verlusthäufigkeit, mittlere Verzögerung) angegeben werden [212]. Jedoch sind im Allgemeinen Verlusthäufigkeit und Verzögerung für sich allein wenig aussagekräftig und dürfen vor allem im Hinblick auf eine Systemoptimierung nicht fehlinterpretiert werden. So müssen z. B. Versuche scheitern, durch eine Puffervergrößerung bei unveränderter Bandbreite eines Engpass-Links eine Reduktion der Verluste und damit eine Leistungssteigerung herbeizuführen. Zwar wird die Verlusthäufigkeit tatsächlich zurückgehen, allerdings zum einen nicht so drastisch wie im Fall von Systemen ohne Rückkopplung und zum anderen auf Kosten einer erhöhten Verzögerung, sodass – zumindest über einen weiten Bereich der Puffergröße – der Durchsatz und damit die für den Nutzer wahrnehmbare Leistung völlig unbeeinflusst bleiben (siehe auch Anhang B).

Bei einer detaillierteren Betrachtung des Durchsatzes kommt man zu einer weiteren Unterteilung in unterschiedliche Maße. Für die Dienstgüte von oberstem Interesse ist der Nutzdurchsatz, d. h. die Datenmenge, die im Mittel pro Zeiteinheit über TCP übertragen werden kann. Der Nutzdurchsatz kann oberhalb von TCP sowohl auf Sender- als auch auf Empfängerseite gemessen werden (Bild 4.6). Abgesehen vom Protokolloverhead ist der Nutzdurchsatz weitgehend identisch mit der mittleren Rate, mit der Daten beim Empfänger eintreffen. Geringe Unterschiede ergeben sich jedoch durch unnötige Wiederholungen (*false retransmits*). Eine weitere Größe, die in diesem Zusammenhang häufig verwendet wird, ist die Senderate. Diese kann sich, bedingt durch Verluste im Netz, von der Rate beim Empfänger erheblich unterscheiden. Schließlich geben auch Verteilung und Mittelwert der CWnd-Größe Aufschluss über die Leistungsfähigkeit des Systems. Der Quotient aus mittlerer Größe des CWnd und mittlerer Umlaufzeit stellt dabei eine gute Näherung für die mittlere Senderate dar.

Neben dem Durchsatz spielt in einem Modell mit Überlagerungsverkehr aus ungesättigten TCP-Quellen außerdem die Fairness eine Rolle. Als Maß für die Unterschiede in dem von einzelnen Verbindungen erfahrenen Durchsatz kann der in [144] definierte Fairness-Index oder auch der damit zusammenhängende Variationskoeffizient der Einzeldurchsätze herangezogen werden. In einem Modell mit persistenten Verbindungen wird man allerdings feststellen, dass die Fairness – im Fall von Quellen mit gleichen Parametern – mit zunehmender Dauer der Untersuchung zunimmt. Daher empfiehlt es sich, den Fairness-Index auf ein bestimmtes Zeitintervall zu beziehen, was gleichzeitig die Aufdeckung von kurzzeitiger Unfairness ermöglicht



**Bild 4.6:** Messgrößen im statischen TCP-Modell mit ungesättigten Quellen

(siehe Abschnitt 4.3.5). Ein auf den Durchsatz sowie eine Intervallbreite  $\Delta t$  bezogener mittlerer Fairness-Index kann dann wie folgt formuliert werden kann:

$$f_F(\Delta t) = E \left[ \frac{\left( \sum_{i=1}^n G_i(\Delta t) \right)^2}{n \cdot \sum_{i=1}^n G_i^2(\Delta t)} \right] \quad (4.22)$$

Darin bezeichnet  $G_i(\Delta t)$  eine Zufallsvariable für den in einem beliebigen Zeitintervall der Länge  $\Delta t$  von Quelle  $i$  erfahrenen Nutzdurchsatz und  $n$  die Gesamtzahl der Quellen.<sup>9</sup>

#### 4.3.2.2 Dynamischer TCP-Verkehr

Wie im Fall der persistenten Quellen stellt auch in Bezug auf die in Abschnitt 4.2.2.3 beschriebenen Modelle für dynamischen TCP-Verkehr der Durchsatz eine wichtige Größe dar. Jedoch macht hier die Betrachtung des Summendurchsatzes keinen Sinn, da sich dieser im stationären Fall nach Gleichung (4.7) direkt aus den Modellparametern ergeben wird. Stattdessen bietet sich der innerhalb einer Verbindung beobachtete Durchsatz  $G$  als Zufallsvariable an. Für eine statistische Auswertung liegt zunächst der Mittelwert von  $G$  nahe, der jedoch den Nachteil hat, dass die für Bursts unterschiedlicher Größe erhaltenen Werte gleich gewichtet in die Mittelwertbildung eingehen. Mehr Aufschluss gibt daher meist der bedingte Mittelwert

$$g(s) = E[G|B = s] \quad (4.23)$$

unter der Voraussetzung, dass die Burstgröße den Wert  $s$  hat. Dieser hat allerdings den Nachteil, dass er sich in einer Simulation im Fall einer kontinuierlichen oder feingranularen diskre-

<sup>9</sup> In dem hier betrachteten stationären Fall spielt die genaue Lage des Zeitintervalls keine Rolle.

ten Verteilung von  $B$  nicht bzw. nur schwer exakt bestimmen lässt. Als Näherung wird – ähnlich wie beim Histogramm-Verfahren zur Ermittlung von Verteilungen – der Wertebereich von  $B$  in  $n_B$  diskrete Bereiche unterteilt, deren Ränder durch  $\hat{B}_0, \hat{B}_1, \dots, \hat{B}_{n_B}$  gegeben sind. Simulativ können dann die bedingten Mittelwerte unter der Voraussetzung  $\hat{B}_{i-1} < B \leq \hat{B}_i$  für  $i \in \{1, \dots, n_B\}$  bestimmt werden. Dabei hat eine grobe Rasterung den Vorteil, dass in jeden Bereich ausreichend Messwerte fallen, sodass die statistische Aussagefähigkeit erhöht wird, während eine feinere Rasterung einen genaueren Einblick liefert und grundsätzlich die Darstellung in einem Diagramm vereinfacht.

Für die Dienstgüte aus Sicht des Anwenders ist mehr noch als der Durchsatz die Transferzeit  $T$  für ein Objekt relevant, da sie eine wichtige Komponente seiner Wartezeit darstellt. Die Transferzeit hängt mit dem Durchsatz über die Beziehung

$$T = \frac{B}{G} \quad (4.24)$$

zusammen, wobei  $B$  die Zufallsvariable für die Burstgröße darstellt. Auch hier gibt neben Mittelwert und Verteilung der bedingte Mittelwert

$$t(s) = E[T|B = s] \quad (4.25)$$

mit Bezug auf die Burstgröße einen guten Einblick in das Systemverhalten.

Während es sich im Fall der Transferzeit und des Durchsatzes um absolute Metriken handelt, wird in [62] mit dem so genannten *Fun Factor* ein relatives Gütemaß eingeführt. Als Bezugsgröße wird dabei allgemein der Wert einer absoluten Größe herangezogen, der sich im Idealfall, i. d. R. also unter der Voraussetzung eines unbelasteten Systems, ergeben würde. Der *Fun Factor* nimmt somit stets Werte im Bereich zwischen 0 und 1 an, wobei der Wert 1 „maximalen Fun“ anzeigt.<sup>10</sup>

Bei der Anwendung auf das betrachtete Modell für dynamischen TCP-Verkehr ergeben sich eine Reihe möglicher Definitionen [33]. Neben der Zufallsvariablen für den *Fun Factor*

$$\Phi = \frac{G}{G_{max}} = \frac{T_{min}}{T} \quad (4.26)$$

wird auch die reziproke Zufallsvariable

$$\Delta = \frac{1}{\Phi} = \frac{T}{T_{min}}, \quad (4.27)$$

---

<sup>10</sup> Im ursprünglichen Verwendungszusammenhang bezieht sich „Fun“ auf den Spaß beim Surfen im WWW. Allgemein kann dahinter jedoch die Zufriedenheit mit einer beliebigen Anwendung gesehen werden.

betrachtet, die in Anlehnung an [178] als *Delay Factor* bezeichnet wird. Dabei bezeichnen  $G_{max}$  und  $T_{min}$  die Referenzwerte für Durchsatz bzw. Transferzeit, die sich im Idealfall (d. h. in einem unbelasteten System) ergeben würden.

Insbesondere in Bezug auf die Bestimmung des mittleren *Fun Factor* macht es einen Unterschied, ob die Mittelwertbildung über  $\Phi$  oder über  $\Delta$  erfolgt. Bei Bestimmung des Mittelwerts von  $\Delta$  und anschließende Kehrwertbildung erhält man den mittleren *Fun Factor*

$$\varphi_{\Delta} = \frac{1}{\mathbb{E}[\Delta]} = \frac{1}{\mathbb{E}[T/T_{min}]} \quad (4.28)$$

Wird andererseits über  $\Phi$  gemittelt, ergibt sich als mittlerer *Fun Factor*

$$\varphi_{\Phi} = \mathbb{E}[\Phi] = \mathbb{E}\left[\frac{T_{min}}{T}\right] \quad (4.29)$$

Wie bei den absoluten Metriken hat auch hier der bedingte Mittelwert

$$\varphi_{\Delta}(s) = \frac{1}{\mathbb{E}[\Delta|B=s]} = \frac{T_{min}(s)}{t(s)} \quad (4.30)$$

bzw. bei Mittelwertbildung über  $\Phi$

$$\varphi_{\Phi}(s) = \mathbb{E}[\Phi|B=s] = T_{min}(s) \cdot \mathbb{E}\left[\frac{1}{T} \middle| B=s\right] \quad (4.31)$$

unter der Voraussetzung einer gegebenen Burstgröße  $s$  seine Bedeutung, wobei  $T_{min}(s)$  für die minimale Transferzeit für einen Burst gegebener Größe steht (siehe Anhang A).

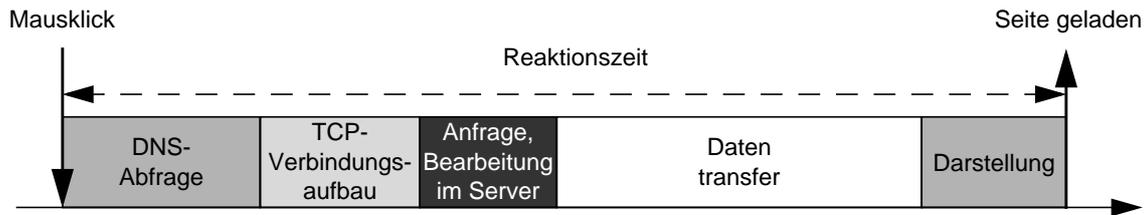
Neben Mittelwert und bedingtem Mittelwert lässt außerdem die *Fun-Factor*-Verteilungsfunktion Aussagen über die Leistungsfähigkeit zu. Dabei gilt der Zusammenhang

$$F_{\Phi}(x) = \mathbb{P}(\Phi \leq x) = \mathbb{P}\left(\Delta \geq \frac{1}{x}\right) = 1 - F_{\Delta}\left(\frac{1}{x}\right) \quad (4.32)$$

zwischen den Verteilungsfunktionen von  $\Phi$  und  $\Delta$ . Weitere mögliche Definitionen für relative Leistungsmetriken auf der Basis des dynamischen TCP-Verkehrsmodells finden sich in [33].

Ein Nachteil der genannten Metriken stellt die Überbewertung kurzer Verbindungen dar. Diesem Problem kann auf zweierlei Weise begegnet werden:

- durch eine Gewichtung mit der Burstgröße bei der Betrachtung des Mittelwerts oder der Verteilung [33]



**Bild 4.7:** Reaktionszeit beim WWW-Zugriff

- durch einen modifizierten *Fun Factor* bzw. *Delay Factor*, bei dem nur der über einen bestimmten Schwellwert  $T_0$  hinausragende Transferzeiten zu einer Reduktion bzw. Erhöhung führen:

$$\Phi' = \frac{\max(T_{min}, T_0)}{\max(T, T_0)} \quad (4.33)$$

$$\Delta' = \frac{\max(T, T_0)}{\max(T_{min}, T_0)} = \frac{1}{\Phi'} \quad (4.34)$$

Die modifizierten Faktoren gemäß Gleichung (4.33) haben allerdings den Nachteil, dass eine Abhängigkeit von einem weiteren Parameter ( $T_0$ ) besteht. Um ein Ausufern der Menge an Metriken zu verhindern, sollte also nur ein zuvor festgelegter Wert von  $T_0$  mit anschaulicher Bedeutung (z. B.  $T_0 = 500$  ms als Mindestgröße für eine vom Benutzer wahrgenommene Verzögerung) betrachtet werden.

### 4.3.2.3 Anwendungs- und Nutzermodelle

Die im letzten Abschnitt definierten Dienstgütemaße stellen, ausgehend von den Metriken auf der Paketebene, einen ersten Schritt dar in Richtung einer Dienstgüte, wie sie vom Nutzer wahrgenommen wird. Ein weiterer möglicher Schritt ist die Einbeziehung der Anwendungs- und Nutzerebene in die Betrachtung. Dazu gehören neben speziellen Anwendungsmodellen (siehe Abschnitt 4.2.2.4) auch auf diese Modelle abgestimmte Metriken.

Auf dieser Ebene spielen vor allem Reaktionszeiten auf vom Anwender initiierte Ereignisse eine Rolle. Im Beispiel WWW-Zugriff kann das die Zeit vom Mausclick auf einen Link im WWW-Browser bis zur vollständigen Darstellung der neuen Seite sein (Bild 4.7). Die Reaktionszeit enthält zwar die Transferzeit als wichtige Komponente. Hinzu kommen aber weitere Bestandteile wie Zeiten für DNS-Abfrage, den Aufbau einer TCP-Verbindung, Senden einer HTTP-GET-Anfrage und Bearbeitung im WWW-Server sowie Darstellung der Ergebnisse durch den Browser.

Häufig lassen sich die einzelnen Zeiten nicht wie in Bild 4.7 angedeutet einfach addieren. Dies gilt vor allem, wenn eine Seite noch weitere Objekte enthält. Dann werden je nach Browser

und HTTP-Version evtl. mehrere parallele TCP-Verbindungen aufgebaut. Außerdem können die eingebetteten Objekte erst angefordert werden, nachdem die Grundseite zumindest teilweise geladen worden ist. Schließlich sind die heutigen Browser in der Lage, bereits Teile einer Seite oder auch eines darin eingebetteten Bildes darzustellen, bevor der Ladevorgang abgeschlossen ist, sodass sich die Zeiten für Datentransfer und Darstellung überlappen. Dies wirft auch Schwierigkeiten bei der genauen Definition der Reaktionszeit auf. Enthält die Seite z. B. viele Bilder, ist auch die Zeit bis zur Darstellung des Textes ohne Bilder bereits eine wichtige Größe.

Wie bei den einfachen Modellen für dynamischen TCP-Verkehr können auch hier neben absoluten Metriken wieder von der Erwartungshaltung des Nutzers abhängige relative Maße zum Einsatz kommen. In [64] werden verschiedene Erweiterungen für den in Abschnitt 4.3.2.2 präsentierten *Fun Factor* vorgeschlagen. Die Definitionen enthalten als Bezugsgrößen neben der minimalen Transferzeit auch vom Nutzer erwartete Werte für die Verzögerung beim Verbindungsaufbau.

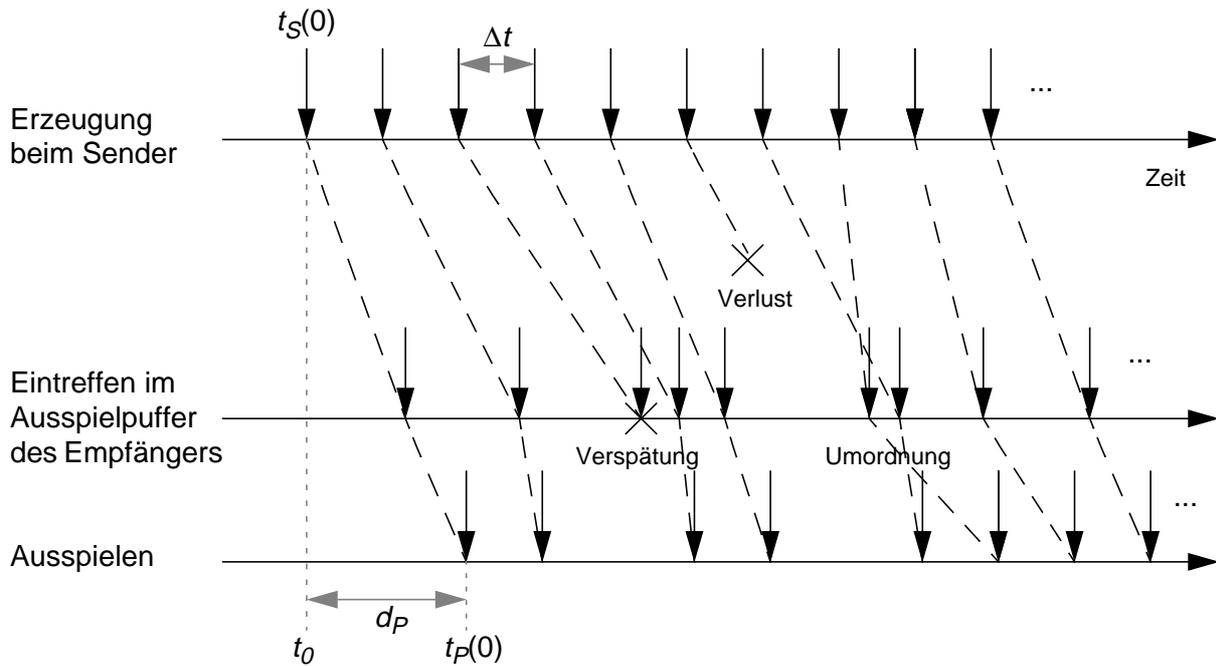
### 4.3.3 Kenngrößen für Echtzeitverkehr

Ähnlich wie bei elastischem Verkehr geben auch bei Echtzeitanwendungen die grundlegenden Maße der Paketebene noch kein klares Bild über die letztendlich vom Nutzer wahrgenommene Dienstgüte. Allerdings ist deren Aussagekraft allgemein größer als bei Anwendungen, die elastischen Verkehr produzieren. Dies gilt zumindest, solange man auf die Betrachtung adaptiver Anwendungen, wie sie z. B. in [246] beschrieben werden und bei denen eine Anpassung der Bandbreite an den Netzzustand erfolgt, verzichtet.

Multimedia-Applikationen erzeugen auf der Senderseite einen kontinuierlichen Datenstrom, der im einfachsten Fall aus Paketen konstanter Länge besteht, die zu äquidistanten Zeitpunkten  $t_0, t_0 + \Delta t, t_0 + 2\Delta t, \dots$  gesendet werden (Bild 4.8). Beim Empfänger wird zur Synchronisation der Daten ein Ausspielpuffer (*playout buffer*) verwendet, um Verzögerungsschwankungen im Netz auszugleichen und ein Ausspielen zu äquidistanten Zeitpunkten zu ermöglichen.<sup>11</sup> Dies bedeutet, dass das Paket mit der Sequenznummer  $i$ , das zum Zeitpunkt  $t_S(i) = t_0 + i \cdot \Delta t$  beim Sender erzeugt wurde, zu einem um einen konstanten Wert verschobenen Zeitpunkt  $t_P(i) = t_S(i) + d_P = t_0 + i \cdot \Delta t + d_P$  ausgespielt, d. h. in ein Audio- oder Videosignal umgewandelt wird. Die zeitliche Verschiebung  $d_P$  wird dabei als Auspielverzögerung (*playout delay*) bezeichnet. Ist eine Auspielverzögerung festgelegt, kann der Empfänger den Auspielzeitpunkt mit Hilfe der über RTP mitgelieferten Zeitstempel und einer über NTP (*network time protocol*) synchronisierten Uhr bestimmen. Ein Ausspielen kann jedoch nur erfolgen, wenn das Paket spätestens zum Zeitpunkt  $t_P(i)$  beim Empfänger eingetroffen ist

---

<sup>11</sup> Dies gilt zumindest für Audioanwendungen. Bei Videoapplikationen ist teilweise die Reihenfolge innerhalb eines Rahmens nicht maßgeblich. Doch auf Rahmenebene erfolgt auch hier i. d. R. eine Synchronisation.



**Bild 4.8:** Prinzipielle Abläufe bei der Multimedia-Kommunikation

und die Decodierstufe durchlaufen hat. Dies ist dann der Fall, wenn die Verzögerung im Netz kleiner war als  $d_p$ . Kommt das Paket zu spät an, muss es beim Empfänger verworfen werden.

Für den Nutzer werden Qualitätseinbußen auf zweierlei Weise wahrnehmbar. Zum einen wird insbesondere bei Konversationsanwendungen wie Sprach- oder Bildtelefonie eine große Verzögerung des Audio- oder Videosignals als störend empfunden. Laut [135] kann bei Sprachverkehr dieser Einfluss nahezu vernachlässigt werden, wenn die Verzögerung unter 150 ms liegt, während eine Verzögerung von mehr als 400 ms als nicht mehr akzeptabel gilt. Neben Verarbeitungszeiten in den Endgeräten (z. B. für Analog-Digitalwandlung) macht dabei die Ausspielverzögerung  $d_p$  den größten Anteil aus. Zum anderen beeinträchtigt eine schlechte Qualität des Sprach- oder Videosignals selbst (Rauschen, Knacken, fehlende Bildpunkte) die Kommunikation. Ursache sind Lücken in dem an den Decoder übergebenen Datenstrom. Dies wiederum lässt sich auf Pakete zurückführen, die entweder im Netz verloren gegangen sind oder aufgrund ihrer Verzögerung im Netz beim Empfänger verworfen werden mussten. Die Wahrscheinlichkeit, dass ein Paket beim Empfänger nicht ausgespielt werden kann, ergibt sich also bei einer stationären Betrachtung zu

$$p_g = p_N + P(T_N > d_p - d_C) \quad (4.35)$$

Hierin bezeichnen  $p_N$  die Verlustwahrscheinlichkeit im Netz,  $T_N$  die Zufallsvariable für die Paketverzögerung im Netz und  $d_C$  die als konstant angenommene Zeit für Codierung und Decodierung in den Endgeräten. Wie stark sich eine bestimmte Gesamtverlustwahrscheinlichkeit auf die Signalqualität auswirkt, hängt u. a. davon ab, ob Verfahren zur Fehlerkorrektur

(*forward error correction*, FEC) eingesetzt werden [158]. In diesem Fall kann es sein, dass bis zu einem gewissen Wert von  $p_g$  überhaupt keine Beeinträchtigung zu beobachten ist. Andererseits gibt es Anwendungen, die bei zu hoher Verlustrate gar nicht mehr genutzt werden können. In beiden Szenarien spielt neben dem stationären Wert der Verlustwahrscheinlichkeit auch das Kurzzeitverhalten in Bezug auf eine Häufung der Verluste eine Rolle (siehe auch Abschnitt 4.3.5).

Aus Gleichung (4.35) wird deutlich, dass die beiden Einflusskomponenten auf die vom Anwender wahrgenommene Qualität nicht unabhängig voneinander sind, da sowohl Signalverzögerung als auch Gesamtverlustwahrscheinlichkeit entscheidend von der Ausspielverzögerung  $d_p$  abhängen. Dabei handelt es sich um gegenläufige Effekte.

Die Ausspielverzögerung kann entweder fest eingestellt werden, sodass zumindest während einer Verbindung keine Änderung mehr möglich ist, oder dynamisch an den Netzzustand angepasst werden. Im Falle einer festen Ausspielverzögerung sollte diese so gewählt werden, dass keiner der beiden Einflüsse die Dienstgüte dominant beeinträchtigt. Adaptive Playout-Algorithmen versuchen, die Wahl der Ausspielverzögerung entsprechend der Verkehrscharakteristik des beim Empfänger ankommenden Paketstroms ständig zu optimieren [197, 232]. Eine Anpassung der Playout-Verzögerung findet entweder kontinuierlich oder stufenweise zu bestimmten Zeitpunkten (z. B. nach einer Sprachpause) statt. Dies kann aber unter Umständen wiederum vom Anwender wahrgenommen werden und stellt damit selbst wieder eine Komponente einer Dienstgüteeinträchtigung dar. Außerdem zeigen Untersuchungen, dass die Verbesserung gegenüber einer festen Playout-Zeit nicht immer gegeben ist [196].

#### 4.3.4 Subjektive Qualität

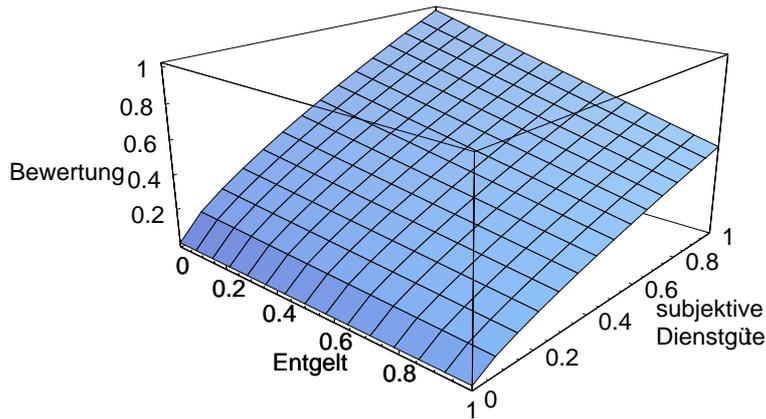
Bei den in den vorhergehenden Abschnitten vorgestellten Maßen, die häufig an mehr oder weniger abstrakten Modellen festgemacht sind, handelt es sich um objektive Metriken, die in einem System messbar sind. Was für den Anwender letztlich zählt, ist jedoch die von ihm wahrgenommene Dienstgüte. Ein solches subjektives Maß kann entweder durch eine geeignete Abbildung der objektiven Maße oder durch Tests mit Nutzern gewonnen werden.

Ansätze zur Abbildung objektiver Maße auf ein subjektives Maß finden sich z. B. in [50]. Dort wird versucht, eine Beziehung zwischen dem Durchsatz und der Nützlichkeit (*utility*) einer Anwendung herzustellen. Allerdings wird es kaum möglich sein, für diese Abbildung eine exakte Funktion anzugeben, zumal der Zusammenhang von Zufriedenheit und objektiven Maßen auch stark vom persönlichen Empfinden der einzelnen Nutzer abhängt. Allerdings kann oft für eine bestimmte Anwendung der grobe Verlauf einer Nützlichkeitsfunktion (*utility function*) abgeschätzt werden. So gibt es einerseits Anwendungen, bei denen die subjektive Dienstgüte rapide abfällt, sobald der Wert eines bestimmten objektiven Maßes einen bestimmten Wert über- oder unterschreitet. Dazu gehören z. B. Anwendungen mit harten Echtzeitbedin-

gungen, die nicht mehr nutzbar sind, sobald Verzögerungen einen Maximalwert überschreiten. Andererseits ist der Großteil der in IP-Netzen vorzufindenden Dienste dadurch gekennzeichnet, dass die subjektive Qualität eher allmählich abnimmt, wenn sich die Werte für objektive Maße verschlechtern. Das bedeutet, dass ein Dienst wie z. B. WWW-Zugriff selbst dann noch von Nutzen sein kann, wenn die Reaktionszeit auf vom Anwender ausgelöste Ereignisse groß ist. Ein noch extremeres Beispiel, bei dem auch noch Verzögerungen im Stundenbereich bedingt akzeptabel sind, ist die Übertragung von E-Mails.

Wird die subjektive Qualität nicht aus objektiven Maßen, sondern durch eine Befragung von Nutzern bestimmt, bedeutet dies i. d. R., dass ein groß angelegter Feldversuch durchgeführt wird, bei dem verschiedene Nutzer unter identischen und festgelegten Bedingungen die Qualität eines Dienstes beurteilen. Als Grundlage für die Beurteilung kann dann eine Skala mit diskreten Werten für verschiedene Qualitätsstufen dienen. Bei dem ursprünglich für Sprachdienste konzipierten *Mean Opinion Score* (MOS) umfasst die Skala z. B. fünf Stufen [139]. Bei anderen Untersuchungen müssen die Testpersonen nach der Dienstnutzung verschiedene Fragen beantworten, wobei ihnen eine Auswahl an möglichen Antworten zur Verfügung gestellt wird [44]. Eine weitere Möglichkeit zeigt der in [92] beschriebene INDEX-Feldversuch auf, bei dem die Testnutzer aus verschiedenen Konfigurationen mit unterschiedlichen Leistungsparametern wählen können, die zu unterschiedlichen Preisen angeboten werden. Eine Auswertung der Benutzung der einzelnen Konfigurationen gibt dann einen Eindruck bzgl. der Wertschätzung durch die Nutzer und der Bereitschaft, für ein Mehr an Dienstgüte zu bezahlen (*willingness to pay*).

Tests wie der INDEX-Feldversuch zeigen, dass es nicht genügt, nur die subjektive Dienstgüte zu bestimmen. Ob ein Dienst sich der Beliebtheit der Nutzer erfreuen wird, hängt auch von dem Entgelt ab, das für die Nutzung erbracht werden muss. Diese beiden Aspekte getrennt zu betrachten, z. B. indem einerseits von einer minimalen Anforderung an die (subjektive) Dienstgüte und andererseits von einem festen Budget eines Nutzers ausgegangen wird, stellt eine starke Vereinfachung dar. In der Realität wird ein Nutzer die Bewertung eines Dienstes sowohl von der wahrgenommenen Dienstgüte als auch dem dafür zu bezahlenden Preis abhängig machen. Eine solche integrierte Bewertung könnte dann so aussehen, wie in Bild 4.9 für normierte Größen dargestellt. Wenn der Nutzer mehrere Angebote, d. h. Paare von Werten für Entgelt und QoS, zur Auswahl hat, wird er sich für dasjenige entscheiden, bei dem die Bewertungsfunktion den höchsten Wert liefert. Allerdings sind in der Wirklichkeit derartige Profile stark vom Nutzertyp abhängig, sodass Abbildungsfunktionen wie in Bild 4.9 auf weitgehenden Annahmen basieren.



**Bild 4.9:** Integrierte Bewertung von Dienstgüte und Entgelt

### 4.3.5 Leistungsmaße zur Beschreibung von transientem Verhalten

Bei der Leistungsuntersuchung wird meist die Annahme der Stationarität zugrunde gelegt. Dies bedeutet einerseits, dass von einer stationären Verkehrsbeschreibung ausgegangen wird, und andererseits, dass stationäre Maße betrachtet werden. Daraus ergeben sich zwei Probleme: Zum einen ist der Verkehr in Kommunikationsnetzen allgemein tageszeitlichen und saisonalen Schwankungen unterworfen. Daher wird bei der klassischen verkehrstheoretischen Dimensionierung nur ein zeitlicher Ausschnitt, die so genannte Hauptverkehrsstunde (*busy hour*), betrachtet und angenommen, dass der Verkehr innerhalb dieses Zeitraums stationär ist. Zum anderen wurde vor allem in IP-Netzen das Phänomen der Selbstähnlichkeit beobachtet. Dadurch können Hochlastphasen sehr lange andauern, sodass Instationarität und Selbstähnlichkeit als Ursachen nur schwer zu unterscheiden sind. Dies berührt nicht nur die Verkehrsmodellierung, sondern auch die Aussagekraft von Leistungsmetriken. Denn ebenso wie es angesichts der tageszeitlichen Schwankungen sinnvoll ist, die Dimensionierung und die dafür verwendeten Metriken auf den Abschnitt des Tages zu beziehen, in dem im Mittel die stärkste Belastung auftritt, so ist es auch angebracht, die Leistung innerhalb der durch Selbstähnlichkeit verursachten Hochlastphasen zu bestimmen. In Bezug auf eine Simulation heißt dies, dass zusätzlich zu Langzeitmittelwerten und -häufigkeiten auch die auf verschiedene Intervallgrößen bezogenen Mittelwerte und Häufigkeiten gemessen werden.

Einen Eindruck davon, ob das Langzeitverhalten auch auf kurze Zeitabschnitte übertragbar ist, geben dabei zunächst Traces. Wird beispielsweise die Verlustwahrscheinlichkeit betrachtet, so kann der Trace aus aufeinander folgenden ( $i = 1, 2, \dots$ ) Wertepaaren  $(i \cdot \Delta t, p((i-1) \cdot \Delta t, i \cdot \Delta t))$  bestehen, wobei  $p(t_1, t_2)$  die im Intervall  $[t_1, t_2]$  gemessene Verlusthäufigkeit bezeichnet. Die gleichzeitige Darstellung von Traces mit unterschiedlichen Intervallbreiten  $\Delta t$  ermöglicht den Vergleich verschiedener Zeitebenen.

Eine Alternative zu Traces ist die Bestimmung der Varianz der zeitlichen Mittelwerte. Einen guten Einblick gibt dabei die Darstellung der Varianz über der Intervallgröße. Eine solche Varianz-Zeit-Darstellung (*variance time plot*) wird häufig bei der statistischen Analyse von Verkehrsmessungen eingesetzt. Wie in Abschnitt 4.2.1.3 angedeutet wurde, kann man mit Hilfe dieser Darstellung selbstähnliches Verhalten feststellen. Darüber hinaus ist aber auch interessant, inwieweit der Verlauf der Varianz von Leistungskenngrößen wie Verlusthäufigkeiten ein vergleichbares Verhalten zeigt. Aus der Sicht des Anwenders ist es in diesem Zusammenhang wünschenswert, dass die Varianz mit steigender Intervallgröße rasch abnimmt.

Anstatt zeitlich begrenzte Leistungsgrößen auf in ihrer Lage und Dauer vordefinierte Zeitintervalle zu beziehen, können auch aus dem Verkehrsmodell abgeleitete Intervalle verwendet werden. Lassen sich z. B. in einem Modell einzelne Verbindungen wiederfinden, so bietet es sich an, Leistungsmetriken auf die Verbindungsdauer zu beziehen [200] und z. B. Verteilung und Varianz zu betrachten. Ein Beispiel hierfür ist der in Abschnitt 4.3.2.2 als Metrik für dynamischen TCP-Verkehr vorgeschlagene verbindungsbezogene Durchsatz. Solche Metriken stehen allerdings nur zur Verfügung, wenn das Verkehrsmodell eine Verbindungsebene oder Vergleichbares kennt. Umgekehrt heißt dies, dass es auch durchaus sinnvoll sein kann, eine Verbindungsebene im Modell zu berücksichtigen, um entsprechende Dienstgütemetriken anwenden zu können, auch wenn es für die Charakterisierung des Verkehrs nicht erforderlich wäre.

# Kapitel 5

## Ein neues Verfahren zur relativen Dienstgütedifferenzierung in IP-Netzknoten

In diesem Kapitel wird ein Verfahren vorgestellt, das mit Hilfe von Scheduling und Puffermanagement eine relative Differenzierung von Verkehrsströmen innerhalb eines IP-Netzknotens ermöglicht. Das Verfahren wird qualitativ mit anderen Mechanismen verglichen, die aus der Literatur bekannt sind und für vergleichbare Anforderungen entworfen wurden. Schließlich wird aufgezeigt, wie das Verfahren in einem Ende-zu-Ende-Szenario eingesetzt werden kann.

### 5.1 Randbedingungen und Anforderungen

Das dem hier vorgestellten Verfahren zugrunde liegende Konzept der Dienstgüteunterstützung folgt weitgehend der DiffServ-Rahmenarchitektur (siehe Abschnitt 3.3.3). Dies bedeutet insbesondere, dass im betrachteten Netzbereich keine Informationen zu einzelnen Verkehrsflüssen gehalten werden. Statt dessen erfolgt jeweils eine Aggregation von Verkehrsflüssen, die der gleichen Dienstgütekategorie angehören. Eine differenzierte Behandlung im Netz muss sich also nach dieser Dienstgütekategorie richten, die im ToS-Feld im Kopf der IP-Pakete als DSCP-Wert eingetragen ist. Da das ToS-Feld eine Breite von acht Bit hat, ist damit die Anzahl der Klassen auf 256 begrenzt; werden nur die in [204] für den DSCP vorgesehenen sechs Bit genutzt, liegt eine Beschränkung auf 64 Klassen vor. Darüber hinaus besteht noch die unten angesprochene Möglichkeit, die Behandlung von Verkehrsströmen zusätzlich vom Transportprotokoll-Feld abhängig zu machen.

DiffServ wird hier allerdings nur als Rahmen betrachtet, der prinzipiell noch Platz für Erweiterungen lässt. So finden die sowohl im Hinblick auf ihre inhaltliche Bedeutung als auch auf ihre

DSCP-Werte festgelegten PHBs oder PHB-Gruppen AF und EF (siehe Abschnitt 3.3.3.3) keine explizite Berücksichtigung. Das bedeutet, dass die hier definierten Regeln zur differenzierten Behandlung im DiffServ-Sinn als neue PHB-Gruppe anzusehen sind. Grundsätzlich besteht jedoch die Möglichkeit, eine Abbildung auf entsprechende AF-/EF-PHBs herzustellen.

Eine weitere Eingrenzung stellt die Voraussetzung dar, dass keinerlei absolute Garantien gegeben werden sollen. Dies erlaubt den Verzicht auf einige Verkehrsmanagement-Funktionen. So wird angenommen, dass weder eine Quellflusskontrolle existiert, die z. B. nach dem Prinzip des *Leaky Bucket* Pakete abhängig von der Verkehrscharakteristik bestimmten Klassen zuordnet (siehe Abschnitt 3.2.2), noch eine Annahmesteuerung, durch die das Verkehrsaufkommen insgesamt begrenzt wird. Das hier betrachtete Schema sieht lediglich vor, die in den Netzknoten vorhandenen Ressourcen (gemäß Abschnitt 3.2.1) per Scheduling und Puffermanagement an aggregierte Verkehrsströme zu vergeben, sodass eine relative Differenzierung ermöglicht wird.

Das Schema ist dabei nicht zwangsläufig auf das Kernnetz beschränkt, sondern kann bis hin zum Nutzer sichtbar sein, der aus einer Reihe von Dienstgüteklassen wählen kann. In diesem Fall muss durch ein *Priority Pricing* (siehe Abschnitt 3.2.6.5) sichergestellt werden, dass durch höhere Entgelte für Klassen mit besserer Behandlung ein Anreiz zur Benutzung der „schlechteren“ Klassen geschaffen wird. Dabei ist auch vorstellbar, dass Nutzer (selbst oder mittels Anwendungen mit entsprechenden Adaptionsmechanismen) während einer Verbindung die Dienstgüteklasse wechseln, um schließlich die aus ihrer Sicht bestmögliche Kombination aus QoS und Preis zu finden (vgl. Bild 4.9 in Abschnitt 4.3.4). Der Fall, dass wie bei der Anwendung des *Leaky Bucket* aufeinander folgende Pakete des gleichen Verkehrsflusses mit ständig wechselnden DSCP-Werten markiert werden, soll jedoch hier nicht betrachtet werden.

Eine weitere Voraussetzung, von der hier ausgegangen werden soll, ist die häufig vorgeschlagene Trennung von elastischem und Echtzeitverkehr [86, 235, 242]. Der Grund für eine solche Trennung liegt in den unterschiedlichen Anforderungen dieser beiden Verkehrsarten. Wie oben angedeutet, kann die Erkennung der Zugehörigkeit zu einer der beiden Verkehrsarten über die Kennzeichnung des Transportprotokolls im Kopf von IP-Paketen erfolgen. Dabei kann vereinfachend angenommen werden, dass elastischer Verkehr als Transportprotokoll TCP verwendet, während der in Echtzeitanwendungen erzeugte Verkehr UDP-basiert ist. Alternativ dazu können für die beiden Verkehrsarten unterschiedliche Bereiche für DSCP-Werte festgelegt werden, sodass bei der Klassifizierung im Kern nur noch der DSCP berücksichtigt werden muss.

Zusätzlich zu der Möglichkeit, elastischen und echtzeitkritischen Verkehr zu trennen, sollen innerhalb jeder der beiden Verkehrsarten mehrere Dienstgüteklassen unterschieden werden. Damit ist es möglich, den gleichen Dienst (z. B. WWW oder VoIP) in unterschiedlicher Qualität anzubieten oder auf die Anforderungen verschiedener Dienste innerhalb der gleichen Gruppe (z. B. Echtzeitanwendungen mit unterschiedlich starken Verzögerungsanforderungen)

einzugehen. Insbesondere im ersten Anwendungsfall ist es wünschenswert, wenn nicht nur allgemeine relative Garantien („Klasse  $i$  ist besser als Klasse  $j$ “) gegeben werden, sondern wenn proportionale Differenzierung („Dienstgüte in Klasse  $i$  ist um den Faktor  $q_{ij}$  höher als in Klasse  $j$ “) angeboten wird. Damit kommt das bereits in Abschnitt 3.1.4.2 vorgestellte Modell von Dovrolis [88] zur Anwendung, dessen Umsetzung allerdings nicht unproblematisch ist. Einerseits wird proportionale Differenzierung im o. g. Sinn für den Nutzer erst dann wirklich attraktiv, wenn er weiß, dass die von ihm empfundene Dienstgüte beim Klassenwechsel sich um einen bestimmten „Faktor“ ändert. Andererseits können die in den Netzknoten implementierten Mechanismen niemals direkt auf diese nutzerbezogene Dienstgüte Einfluss nehmen, da nicht einmal die Zugehörigkeit einzelner Pakete zu einem bestimmten Verkehrsfluss, geschweige denn die mit diesem Verkehrsfluss assoziierten Randbedingungen (z. B. Anwendung, Parameter wie absolute Verzögerungen oder Zugangsbandbreite) bekannt sind.

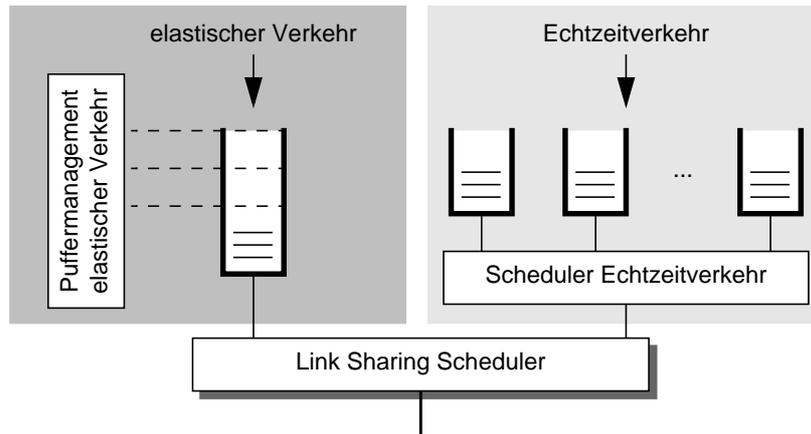
Eine proportionale Differenzierung ist prinzipiell möglich beim Bezug auf grundlegende Leistungsmaße wie Verlustwahrscheinlichkeit oder Verzögerung. Während sich damit für Echtzeitverkehr bereits eine ausreichende Nähe zur empfundenen Qualität erzielen lässt (siehe Abschnitt 4.3.3), ist der Einfluss auf die Dienstgüte bei elastischem Verkehr weitaus schwerer abzuschätzen. Im Fall von persistenten, ungesättigten Quellen, wie sie z. B. häufig als Modell für FTP-Verkehr verwendet werden, ist eine Differenzierung hinsichtlich des Nutzdurchsatzes relevant. Für dynamischen TCP-Verkehr mit kurzen Verbindungen (WWW-Verkehr) hingegen sollte eine unterschiedliche Behandlung zum Ziel haben, dass sich Dienstgüteklassen in Bezug auf Kenngrößen wie den in Abschnitt 4.3.2.2 beschriebenen *Fun Factor* unterscheiden.

Für den Fall, dass eine angemessene proportionale Differenzierung unter den gegebenen Randbedingungen nicht zu realisieren ist, sollten in Anlehnung an [38] zumindest die beiden folgenden Kriterien für eine relative Differenzierung erfüllt sein:

- Die Differenzierung sollte signifikant sein, sodass ein Wechsel in eine höhere Dienstgütekategorie wahrgenommen wird. Insbesondere sollte der Nutzer vorhersehen können, welche Klasse ihm eine bessere Dienstgüte beschert.
- Die Differenzierung sollte nur so stark sein, dass kein „Aushungern“ (*starvation*) in niederpriorisierten Klassen stattfindet.

Beide Punkte sind nur im Fall eines stark belasteten Netzes relevant, während sich bei schwacher Belastung ohnehin in allen Klassen eine hohe Dienstgüte ergeben wird und somit keine Mechanismen zur Differenzierung erforderlich sind.

Als weitere Randbedingungen für ein Verfahren zur relativen Dienstgütedifferenzierung gilt die Implementierbarkeit. Angesichts der begrenzten Anzahl von Dienstgüteklassen sind hier allerdings die Voraussetzungen günstiger als etwa im Fall von IntServ, wo Skalierbarkeitsprobleme ein Hindernis für die praktische Umsetzung von Modellen und Mechanismen darstellen.



**Bild 5.1:** Trennung von elastischem und Echtzeitverkehr durch hierarchisches Scheduling

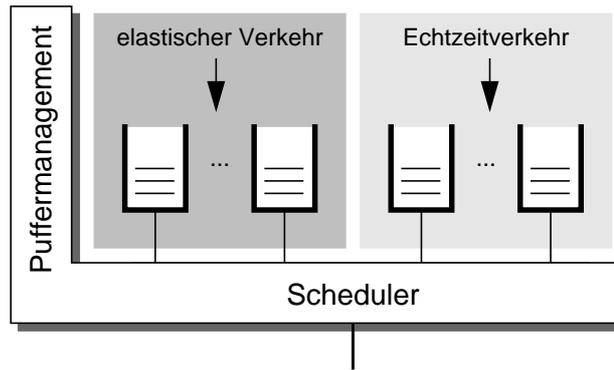
Schließlich ist unter den Anforderungen die Möglichkeit zur Integration in ein Schema zu nennen, das nicht nur eine Differenzierung innerhalb eines einzelnen Knotens bietet, sondern Dienstgüte über ein ganzes Netz hinweg bereitstellt. Diesbezügliche Aspekte werden in Abschnitt 5.5 näher betrachtet.

## 5.2 Prinzipielle Design-Alternativen

Zur Trennung von elastischem und echtzeitkritischem Verkehr bietet sich zunächst ein hierarchischer Ansatz wie der in Bild 5.1 dargestellte an. Dabei wird mit Hilfe eines *Link Sharing Scheduler*, der z. B. nach dem FQ-Prinzip arbeitet (siehe Abschnitt 3.2.1.1), die Linkbandbreite auf die beiden Gruppen von Klassen aufgeteilt. Statt einer Aufteilung nach bestimmten Gewichten kann auch eine statische Priorisierung einer der beiden Verkehrsarten (z. B. von Echtzeitverkehr) vorgenommen werden, was dann dem in [143] vorgeschlagenen Schema zur bevorzugten Behandlung von EF-Verkehr nahe kommt (vgl. Abschnitt 3.3.3.3).

Für die weitere Differenzierung innerhalb der beiden Gruppen werden untergeordnete Scheduling- und Puffermanagement-Mechanismen eingesetzt. Im einfachsten Fall wird die Differenzierung von elastischem Verkehr – in Anlehnung an das für AF-Klassen spezifizierte Verhalten – mit Hilfe eines Verwerfungsmechanismus (z. B. WRED) realisiert. Sollen allerdings mehrere Echtzeitklassen mit unterschiedlichen Verzögerungsanforderungen unterschieden werden, genügt es nicht, wie bei elastischem Verkehr nur entsprechende Puffermanagement-Mechanismen einzusetzen, sondern es ist ein geeigneter Scheduling-Mechanismus erforderlich. Zusätzlich können auch beim Echtzeitverkehr unterschiedliche Verwerfungsprioritäten eingeführt werden (in Bild 5.1 nicht dargestellt).

Die Verwendung eines FQ-Schedulers zur Trennung von elastischem und unelastischem Verkehr hat den Vorteil, dass damit implizit das Auftreten des bekannten Verdrängungseffekts, bei



**Bild 5.2:** Lösung mit integriertem Scheduling und Puffermanagement

dem TCP-Quellen in Anwesenheit von nicht adaptiven UDP-Strömen ihre Senderate reduzieren, verhindert werden könnte. Allerdings bedarf es dann zur Festlegung der Aufteilung der Linkbandbreite (d. h. der Gewichte des FQ-Schedulers) eines Adaptionmechanismus. Wird statt dessen Echtzeitverkehr statisch priorisiert, kann das zum völligen „Aushungern“ der elastischen Verkehrsströme führen. Darüber hinaus hat ein hierarchischer Ansatz grundsätzlich den Nachteil höherer Komplexität in Bezug auf die Realisierung. Aus diesen Gründen ist es vorteilhaft, wenn eine Architektur eingesetzt wird, wie sie in Bild 5.2 skizziert ist. Dabei gibt es nur einen Scheduler und ein gemeinsames Puffermanagement für alle Klassen. Die Klassen für elastischen Verkehr unterscheiden sich dann von denen für Echtzeitverkehr nur durch andere Werte für die Parameter des gleichen Verfahrens. Allerdings ist es erforderlich, dass – wie im Bild angedeutet – die beiden Teile, Scheduler und Puffermanagement, eng miteinander kooperieren.

### 5.3 Weighted Earliest Due Date

Zur Lösung des in den vorigen Abschnitten geschilderten Problems der relativen Differenzierung wird ein Verfahren vorgeschlagen, das als *Weighted Earliest Due Date* (WEDD) bezeichnet wird. Die Grundzüge von WEDD wurden vom Autor erstmals in [30] im Zusammenhang mit der Differenzierung von Echtzeitverkehr präsentiert. Dieser Anwendungsfall entspricht dem Szenario nach Bild 5.1, wobei durch WEDD der rechte Teil der dort skizzierten Architektur abgedeckt wird. Im Rahmen dieser Arbeit kommt WEDD als integriertes Scheduling- und Puffermanagement-Verfahren entsprechend Bild 5.2 zur Anwendung, d. h. es wird auch zur Differenzierung von elastischem Verkehr eingesetzt.

#### 5.3.1 Grundsätzliches Verhalten

Wie der Name andeutet, kann WEDD als eine Erweiterung der in Abschnitt 3.2.1.1 beschriebenen EDD-Strategie angesehen werden. Genau wie bei EDD kann jeder Dienstgütekategorie eine bestimmte maximale Verzögerung  $\delta_i$  zugeordnet werden. Während allerdings im Fall von

EDD unter den gegebenen Randbedingungen (keine Begrenzung des Angebots durch eine Annahmesteuerung) die Wahrscheinlichkeit  $p_i$ , dass die Frist und damit die maximale Verzögerung für ein Paket überschritten wird, in allen Klassen gleich ist, können bei WEDD unterschiedliche Überschreitungswahrscheinlichkeiten erzwungen werden. Die Formulierung dieser Regel verwendet dabei das Modell zur proportionalen Differenzierung von Dovrolis [88]:

$$\frac{p_i}{p_j} = \frac{w_i}{w_j} \quad (5.1)$$

Das Verhältnis der Überschreitungswahrscheinlichkeiten folgt also dem Verhältnis vorgegebener Gewichtungparameter  $w_i$ , wobei ein geringerer Wert von  $w_i$  eine geringere Überschreitungswahrscheinlichkeit und damit – bei gleicher Maximalverzögerung – eine höhere Dienstgüte anzeigt.

In Analogie zu der in Abschnitt 3.2.1.1 genannten EDD-Erweiterung STE [216] besteht außerdem auch bei WEDD die Möglichkeit, Pakete zu verwerfen, die ihre Frist überschritten haben. Diese Option wird im Folgenden als LPD (*late packet discard*) bezeichnet. Sie ist insbesondere dann Gewinn bringend, wenn – wie im Fall von Echtzeitanwendungen mit Ausspielpuffer – verspätete Pakete ohnehin im Endgerät verworfen werden würden. Damit ist WEDD in der Lage, sowohl eine Verzögerungs- als auch eine Verlustdifferenzierung zu realisieren und stellt daher eine Lösung für ein integriertes Scheduling und Puffermanagement dar. Als Vorteil z. B. gegenüber Lösungen mit statischer Priorisierung ist dabei zu werten, dass der Grad der Differenzierung über entsprechende Parameter eingestellt werden kann, und zwar getrennt für Verzögerungs- und Verlustdifferenzierung. Dies ist besonders auch im Hinblick auf Echtzeitverkehr interessant. Denn dort sind, wie in Abschnitt 4.3.3 verdeutlicht wurde, Verluste und Verzögerungen zwei Formen von Qualitätsbeeinträchtigungen, die vom Nutzer auf unterschiedliche Weise wahrgenommen werden.

Die proportionale Verlustdifferenzierung kann – mit gewissen Einschränkungen – auch auf TCP-Verkehr angewandt werden. Im Fall ungesättigter Quellen gilt unter der Bedingung unabhängiger Verluste näherungsweise, dass die mittlere Senderate proportional zum Kehrwert der Wurzel aus der Verlustwahrscheinlichkeit ist [190, 212]. Dieses Gesetz kann auch als Approximation für den Fall korrelierter Verluste, wie sie z. B. bei WEDD auftreten, angewandt werden (vgl. Anhang B.1.2). Wird außerdem angenommen, dass sich mittlere Senderate und mittlerer Nutzdurchsatz nur geringfügig unterscheiden, ist es damit möglich, das Verhältnis der Mittelwerte  $g_i$  und  $g_j$  für den Nutzdurchsatz pro Verbindung in zwei Klassen  $i$  bzw.  $j$  direkt über die Gewichtungparameter einzustellen, wenn in beiden Klassen gleiche Werte für die maximale Verzögerung gewählt werden (d. h.  $\delta_i = \delta_j$ ):

$$\frac{g_i}{g_j} = \sqrt{\frac{w_j}{w_i}} \quad (5.2)$$

Zusätzlich kann auch noch eine Differenzierung hinsichtlich der maximalen Verzögerung innerhalb der Klassen für elastischen Verkehr nützlich sein, um z. B. den besonderen Gegebenheiten im Falle von dynamischem TCP-Verkehr Rechnung zu tragen (siehe Kapitel 6).

### 5.3.2 Realisierung

Die Differenzierung der Fristüberschreitungswahrscheinlichkeiten wird dadurch erreicht, dass WEDD zwei Betriebsmodi kennt. Im „Normalbetrieb“ arbeitet WEDD exakt gleich wie EDD: Ein Paket aus einer beliebigen Klasse  $i$  erhält zum Ankunftszeitpunkt  $t_A$  einen Zeitstempel (*tag*), der seiner Bearbeitungsfrist  $t_A + \delta_i$  entspricht; Pakete werden in der Reihenfolge steigender Fristen bearbeitet. Wenn jedoch in mindestens einer Klasse Überlast auftritt, geht der Scheduler in den „Überlastbetrieb“. Überlast in einer Klasse  $i$  bedeutet konkret, dass zum aktuellen Zeitpunkt  $t$  dem ersten Paket in der zugehörigen Warteschlange (d. h., dem nächsten Klasse- $i$ -Paket, das gemäß EDD-Disziplin zur Bearbeitung ansteht) eine Frist zugeordnet ist, die in naher Zukunft abläuft:

$$t_A + \delta_i - t < \varepsilon_i \quad (5.3)$$

Dabei bezeichnet  $\varepsilon_i$  eine Art zeitlichen „Sicherheitsabstand“ (*congestion margin*), dessen Wert beispielsweise relativ zur maximalen Verzögerung angegeben werden kann (z. B.  $\varepsilon_i = \delta_i/10$ ).

Im Überlastbetrieb werden bei der Auswahl des nächsten zu bedienenden Pakets nur noch diejenigen Klassen berücksichtigt, die als überlastet im o. g. Sinn gelten. Anschaulich bedeutet dies, dass Pakete, die Gefahr laufen, ihre Frist zu überschreiten, bevorzugt behandelt werden. Falls es mehr als eine überlastete Klasse gibt, muss nun entschieden werden, welche Klasse davon bedient wird. Dazu wird für jede dieser Klassen ein so genannter Überlaststempel (*congestion tag*)  $c_i$  berechnet:

$$c_i = \frac{w_i}{E_i(t)} \quad (5.4)$$

Darin repräsentiert  $E_i(t)$  den Wert einer Funktion zum aktuellen Zeitpunkt  $t$ , die eine Abschätzung der Häufigkeit von Fristüberschreitungen in Klasse  $i$  liefert. Betrachtet man die Überlaststempel zweier Klassen  $i$  und  $j$ , bedeutet  $c_i < c_j$ , dass das Verhältnis der Häufigkeiten von Verlusten in Klasse  $i$  und Klasse  $j$  größer ist als das vorgegebene Verhältnis der Gewichtungparameter  $w_i$  bzw.  $w_j$ :

$$c_i < c_j \Leftrightarrow \frac{E_i(t)}{E_j(t)} > \frac{w_i}{w_j} \quad (5.5)$$

Das Verfahren wählt schließlich das erste Paket in der Klasse mit dem kleinsten Überlaststempel zur Bedienung aus. Auf diese Weise wird ein Regelmechanismus geschaffen, der am Ende dafür sorgt, dass das Verhältnis der Überschreitungswahrscheinlichkeiten dem Verhältnis der Gewichtungparameter folgt.

Wird WEDD mit der Option LPD eingesetzt, so muss für jedes Paket, das bei der Auswahl zur Bedienung berücksichtigt wird, zunächst geprüft werden, ob die Frist nicht bereits überschritten ist. Ist dies der Fall, so wird das Paket verworfen und das nächste Paket in der gleichen Warteschlange geprüft usw. Insgesamt ergibt sich damit der in Bild 5.3 mittels Pseudocode in C-ähnlicher Syntax dargestellte Ablauf für die Auswahl des nächsten zu bedienenden Pakets. Die genannte Realisierung von LPD bedeutet allerdings, dass ein Verwerfen erst erfolgen kann, wenn das Paket als nächstes in dieser Klasse zur Bedienung ansteht. Damit nehmen solche Pakete während ihrer Wartezeit unnötig Speicherressourcen in Anspruch, was einen gewissen Nachteil darstellt gegenüber Puffermanagementverfahren, bei denen die Entscheidung über ein Verwerfen gleich bei der Ankunft des Pakets getroffen wird. Allerdings muss betont werden, dass die Überlast in erster Linie durch einen Bandbreite-Engpass verursacht wird und die Hauptaufgabe bei der Zuteilung der Speicherressourcen darin besteht, eine angemessene Differenzierung hinsichtlich der Verluste zu erzielen. Darüber hinaus würde ein Ausnutzen des gesamten verfügbaren Puffers in vielen Fällen zu lange Wartezeiten verursachen, weshalb Verfahren wie RED schon vor Erreichen einer vollständigen Pufferauslastung Pakete verwerfen.

### 5.3.3 Abschätzung der Überschreitungshäufigkeit

Wie im letzten Abschnitt erläutert wurde, basiert der Mechanismus für die Differenzierung der Verwerfungswahrscheinlichkeiten auf einer Abschätzung der Häufigkeit bisheriger Fristüberschreitungen in jeder Klasse. Hierfür existieren verschiedene Realisierungsmöglichkeiten.

Der einfachste Ansatz zur Bestimmung von  $E_i(t)$  verwendet zwei Zähler für jede Klasse. Mit  $N_i$  werden alle ankommenden Pakete gezählt, die zur Klasse  $i$  gehören. Die Variable  $M_i$  erfasst nur diejenigen Pakete, die ihre Frist überschreiten. Zu jedem Zeitpunkt gibt dann das aktuelle Verhältnis

$$E_i(t) = \frac{M_i}{N_i} \quad (5.6)$$

die Häufigkeit von Fristüberschreitungen in Klasse  $i$  an.

Allerdings muss in einer realen Implementierung darauf geachtet werden, dass der Wertebereich von  $N_i$  und  $M_i$  beschränkt bleibt. Eine Möglichkeit, dieses zu erreichen, besteht darin, bei jeder Aktualisierung von  $N_i$  die alten Werte beider Zähler (zur Verdeutlichung mit  $N_i'$  bzw.  $M_i'$  bezeichnet) mit einem Faktor  $1 - \beta$  zu multiplizieren, der im Bereich  $[0, 1]$  liegt:

```

min_tag = infinity;
min_congestion_tag = infinity;
next_class = undefined;
congestion = false;
for all class identifiers i {
  if (LPD is enabled) {
    while ((queue i is not empty) &&
           (tag of first packet in class i < current time) {
      discard first packet in class i queue;
      update estimator for class i;
    }
  }
  if (queue i is not empty) {
    tag = tag of first packet in class i queue;
    if (tag - congestion_margin[i] < current time) {
      congestion = true;
      congestion_tag = w[i]/current_estimator_value[i];
      if (congestion_tag < min_congestion_tag) {
        min_congestion_tag = congestion_tag;
        next_class = i;
      }
    }
    if (!congestion) {
      if (tag < min_tag) {
        min_tag = tag;
        next_class = i;
      }
    }
  }
}
if (next_class != undefined) {
  serve first packet in next_class queue;
  update estimator for next_class;
}

```

**Bild 5.3:** Ablauf der Auswahl des nächsten zu bedienenden Pakets durch WEDD in Pseudocode-Darstellung

$$N_i = (1 - \beta) \cdot N_i' + 1$$

$$M_i = \begin{cases} (1 - \beta) \cdot M_i' & \text{falls Paket fristgerecht abgearbeitet} \\ (1 - \beta) \cdot M_i' + 1 & \text{falls Frist überschritten} \end{cases} \quad (5.7)$$

Für  $\beta > 0$  ergibt sich damit eine Obergrenze

$$M_i \leq N_i < \frac{1}{\beta} \quad (5.8)$$

die im Fall von  $N_i$  gleichzeitig den Grenzwert für  $t \rightarrow \infty$  darstellt. Damit ergibt sich im Grenzfalle für Gleichung (5.6):

$$E_i(t) \rightarrow M_i \cdot \beta \text{ für } t \rightarrow \infty \quad (5.9)$$

Dies bedeutet, dass es bei diesem Verfahren nicht notwendig ist, für  $N_i$  eine eigene Zählvariable zu führen. Um außerdem eine doppelte Multiplikation zu vermeiden, wie sie bei Anwendung der Gleichungen (5.7) und (5.9) auftreten würden, bietet es sich an, direkt  $P_i = M_i \cdot \beta$  als Variable anstelle von  $M_i$  einzuführen und regelmäßig zu aktualisieren:

$$P_i = \begin{cases} (1 - \beta) \cdot P_i' & \text{falls Paket fristgerecht abgearbeitet} \\ (1 - \beta) \cdot P_i' + \beta & \text{falls Frist überschritten} \end{cases} \quad (5.10)$$

Der Schätzwert  $E_i(t)$  ist dann identisch mit dem aktuellen Wert von  $P_i$ . Aus Gleichung (5.10) wird deutlich, dass es sich bei diesem Verfahren um eine Anwendung des bekannten Autoregressionsprinzips handelt und somit eine exponentiell abfallende Gewichtung von Überschreitungsereignissen in der Vergangenheit stattfindet.

Alternativ dazu kann ein schwellenbasierter Mechanismus eingesetzt werden, bei dem eine Reduktion von  $M_i$  und  $N_i$  in größeren Abständen durchgeführt wird, z. B. wenn  $N_i$  einen bestimmten Wert  $\hat{N}$  erreicht hat. Folglich wird die Reduktion durch einen größeren Wert  $\hat{\beta}$  bestimmt (z. B.  $\hat{\beta} = 0.5$ ). Da für WEDD nicht die Absolutwerte von  $E_i(t)$  ausschlaggebend sind, besteht außerdem eine Verbesserungsmöglichkeit darin, bei Erreichen des Schwellenwerts  $\hat{N}$  nicht beide Zähler der jeweiligen Klasse zu reduzieren, sondern nur den jeweiligen Zähler ( $M_i$  oder  $N_i$ ), der die Schwelle erreicht hat. Um aber das Verhältnis der Schätzwerte beizubehalten, muss diese Reduktion dann auch für den entsprechenden Zähler in allen anderen Klassen durchgeführt werden.

Darüber hinaus gibt es noch weitere Optionen bei der Realisierung der Häufigkeitsabschätzung:

- Anstatt die Pakete zu zählen, besteht die Möglichkeit, aggregierte Paketlängen zu summieren. Allerdings ist nur in extremen Szenarien (z. B. bei stark erhöhtem Anteil von kurzen Paketen in temporären Überlastsituationen) ein wahrnehmbarer Einfluss auf die resultierenden Verwerfungswahrscheinlichkeiten zu erwarten.
- Mit  $M_i$  könnten nicht nur aufgrund einer Fristüberschreitung verworfene Pakete, sondern gleichzeitig auch Pakete gezählt werden, die aufgrund eines Pufferüberlaufs verloren gehen. Damit kann in Fällen geringer Pufferkapazität mit einer resultierenden Verlustwahrscheinlichkeit am Puffereingang, die in der Größenordnung der Fristüberschreitungen liegt, verhindert werden, dass die Unterschiede der Gesamtverluste in den einzelnen Klassen zu gering sind.

### 5.3.4 Verhinderung des Aushungerns von Klassen

Ein Problem von WEDD stellt die Realisierung der nach Gleichung (5.1) geforderten Differenzierung bei starker Überlast dar. Angenommen werde z. B. ein System mit zwei Klassen ( $i \in \{0, 1\}$ ) und einem Verhältnis der Gewichtungparameter  $w_0/w_1 = 10$ . Wenn das Angebot in Klasse 1 einen gewissen Wert überschreitet, wird es nicht mehr möglich sein, das gewünschte Verhältnis der Verwerfungswahrscheinlichkeiten zu erreichen, selbst wenn alle Pakete in Klasse 0 verworfen werden. Konkret wird das spätestens der Fall sein, wenn  $p_1 > w_1/w_0$ . Der Scheduler wird ab diesem Lastpunkt eine statische Priorisierung von Verkehr der Klasse 1 durchführen, was ein komplettes Aushungern von Klasse 0 bedeutet. Ein solches Verhalten ist insbesondere bei der Anwendung auf TCP-Verkehr unerwünscht.

Aus diesem Grund wird eine als CSA (*class starvation avoidance*) bezeichnete Verbesserung berücksichtigt, bei der anstelle einer proportionalen Differenzierung wie im Fall von Gleichung (5.1) ein Verhältnis

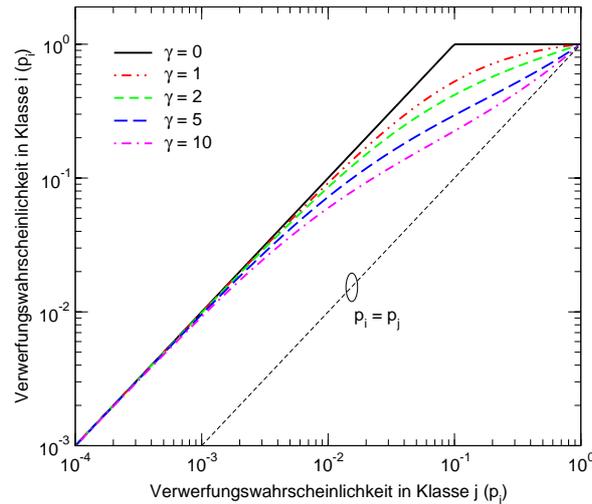
$$\frac{p_i \cdot (1 - p_i)^{-\gamma}}{p_j \cdot (1 - p_j)^{-\gamma}} = \frac{w_i}{w_j} \quad (5.11)$$

angestrebt wird, das von einem Parameter  $\gamma \geq 0$  abhängt. Gleichung (5.1) ergibt sich dabei als Spezialfall für  $\gamma = 0$ . Wird ein Wert  $\gamma > 0$  gewählt, heißt dies, dass die Differenzierung allmählich verschwindet, wenn die Gesamtverlustwahrscheinlichkeit gegen eins geht.

Wie die Werte in Tabelle 5.1 zeigen, hat der Faktor  $(1 - p_i)^{-\gamma}$  nur einen nennenswerten Einfluss, wenn  $\gamma$  größere Werte annimmt oder  $p_i$  in die Größenordnung von 0.1 kommt, sodass über einen weiten Bereich die proportionale Differenzierung nahezu unberührt bleibt. Dies zeigt sich auch in Bild 5.4, das den Zusammenhang zwischen  $p_i$  und  $p_j$  nach Gleichung (5.11) für unterschiedliche Werte von  $\gamma$  veranschaulicht. Aus dieser Darstellung wird außerdem die unterschiedlich starke Annäherung an die Kurve  $p_i = p_j$  für  $p_i \rightarrow 1$  deutlich. Damit kann dieses Vorgehen als guter Kompromiss zwischen einer signifikanten, einstellbaren Differenzierung und der Verhinderung des Aushungerns niedrig priorisierter Klassen betrachtet werden.

**Tabelle 5.1:** Werte von  $(1 - p_i)^{-\gamma}$

$\gamma$	$p_i = 0.0001$	$p_i = 0.001$	$p_i = 0.01$	$p_i = 0.1$	$p_i = 0.5$
1	1.0001	1.0010	1.0101	1.1111	2
2	1.0002	1.0020	1.0203	1.2346	4
5	1.0005	1.0050	1.0515	1.6935	32
10	1.0010	1.0101	1.1057	2.8680	1024



**Bild 5.4:** Zusammenhang zwischen  $p_i$  und  $p_j$  bei CSA nach Gleichung (5.11) in Abhängigkeit von  $\gamma$  für  $w_i/w_j = 10$

Eine Realisierung von CSA kann z. B. dadurch erfolgen, dass zur Berechnung des Überlaststempels

$$c_i = \frac{w_i}{E_i(t) \cdot (1 - E_i(t))^{-\gamma}} \quad (5.12)$$

an die Stelle von Gleichung (5.4) tritt. Die Auswahl des Pakets mit dem kleinsten Überlaststempel bewirkt dann wie im Fall ohne CSA eine Minimierung der Unterschiede zwischen den Überlaststempeln und damit eine Regelung der Verwerfungswahrscheinlichkeiten, sodass Gleichung (5.11) erfüllt wird.

### 5.3.5 Implementierungsaspekte

Nachdem die charakteristischen Eigenschaften von WEDD vorgestellt worden sind, sollen abschließend die im Hinblick auf eine Implementierung des Schedulers relevanten Aspekte beleuchtet werden.

Wie bereits in Abschnitt 3.2.1.1 erwähnt wurde, ist es bei allen fristenbasierten Verfahren erforderlich, zu jedem Paket einen individuellen Zeitstempel zu generieren und zu speichern, der die Bearbeitungsfrist für dieses Paket enthält. Damit ist prinzipiell ein erhöhter Speicherbedarf verbunden, der allerdings angesichts einer Paketlänge, die üblicherweise im Durchschnitt mehrere 100 Bytes beträgt [191], relativ gesehen als vernachlässigbar einzustufen ist. Das Problem liegt eher darin, dass die Routerarchitektur grundsätzlich die Möglichkeit vorsehen muss, jedem Paket ein *Tag* zuzuordnen.

Ein weiteres Problem, das im Zusammenhang mit Scheduling-Mechanismen auftritt und auch im Fall von EDD prinzipiell eine Rolle spielt, ist der Sortieraufwand, der notwendig ist, um das Paket mit der kleinsten Frist zu bestimmen. Da sich die vorgeschlagene Lösung allerdings im DiffServ-Rahmen bewegt, ist die Anzahl der Klassen und damit der zu sortierenden Warteschlangen eher gering, sodass die mit dem Sortieraufwand verbundene Skalierbarkeitsproblematik im Gegensatz zu einer IntServ-basierten Architektur hier nicht zum Tragen kommt.

Eine größere Schwierigkeit bei der Implementierung von WEDD stellen die arithmetischen Operationen z. B. zur Berechnung des Überlaststempels oder bei der Abschätzung der Fristüberschreitungshäufigkeit dar. Solange eine Realisierung in Software in Betracht kommt, was angesichts der größeren Länge von Paketen im Vergleich zu ATM-Zellen auch bei höheren Datenraten noch eher möglich ist als im Fall von ATM, sollte die Komplexität der Algorithmen beherrschbar sein. Wenn aber, wie im Fall von Hochgeschwindigkeits-Routern, eine Hardware-Lösung unumgänglich wird, sind Operationen wie Multiplikation, Division und Potenzierung nur mit großem Aufwand zu realisieren. Vor allem Divisionen kommen bei der Auswahl des nächsten Pakets häufig vor (vgl. Bild 5.3), nämlich jeweils eine für die Abschätzung der Verwerfungshäufigkeit (bei Verwendung eines Verfahrens mit zwei Zählern) sowie für die Berechnung des Überlaststempels in jeder überlasteten Klasse. Daher muss bei einer Realisierung angestrebt werden, Divisionen zu vermeiden und durch Multiplikationen zu ersetzen und anstelle von Multiplikationen nach Möglichkeit Schiebeoperationen zu verwenden. Als mögliche Ansatzpunkte für eine Optimierung sind in diesem Zusammenhang zu nennen:

- Anstatt die Überlaststempel  $c_i$  und  $c_j$  zweier Klassen  $i$  und  $j$  zu vergleichen, zu deren Berechnung nach Gleichungen (5.4) und (5.6) jeweils eine Division erforderlich ist, wird geprüft, ob die folgende Relation erfüllt ist, die nur Produktausdrücke enthält.

$$w_i \cdot N_i \cdot M_j > w_j \cdot N_j \cdot M_i \quad (5.13)$$

- Sofern für alle Größen in Gleichung (5.13) eine Ganzzahldarstellung gewählt wird und bei der Wahl der Gewichtungsfaktoren eine Beschränkung auf 2er-Potenzen erfolgt, kann die Multiplikation mit  $w_i$  durch ein einfaches Linksschieben realisiert werden.
- Bei der Aktualisierung der Zähler  $M_i$  und  $N_i$  auf die in Gleichung (5.7) beschriebene Weise wird man in der Praxis bestrebt sein, eine Multiplikation mit einem Fließkommawert  $(1 - \beta)$  bei jeder Aktualisierung zu vermeiden. In Abschnitt 5.3.3 wurde bereits die Alternative angedeutet,  $M_i$  und  $N_i$  bei Überschreiten einer vorgegebenen Grenze  $\hat{N}$  durch Multiplikation mit einem Faktor  $\hat{\beta}$  zu reduzieren. Wird z. B. für beide Zähler eine binäre Ganzzahldarstellung mit  $\hat{z}$  bit (z. B.  $\hat{z} = 16$ ) eingesetzt, bietet sich an, als Grenzwert  $\hat{N} = 2^{\hat{z}} - 1$  und als Reduktionsfaktor  $1 - \hat{\beta} = 2^{-\hat{k}}$  (z. B.  $\hat{\beta} = 0,5$ ) zu wählen. Eine Reduktion könnte in diesem Fall bei Erreichen des Wertes  $2^{\hat{z}} - 1$  (alle Bits von  $N_i$  gesetzt) durch einfaches Rechtsschieben der Zähler um  $\hat{k}$  bit (z. B.  $\hat{k} = 1$ ) erfolgen.

Inwiefern derartige Maßnahmen einerseits notwendig und andererseits ausreichend sind, hängt sehr stark von den gegebenen Randbedingungen wie beispielsweise der Datenrate auf dem Link oder zur Verfügung stehenden Hardware-Technologien und -Ressourcen ab.

## **5.4 Qualitativer Vergleich mit anderen Verfahren**

In Kapitel 3 wurden im Rahmen der Klassifizierung von Scheduling- und Puffermanagementverfahren bereits einige aus der Literatur bekannte Mechanismen erwähnt, die auch für eine relative Differenzierung unter den gegebenen Randbedingungen nach Abschnitt 5.1 in Frage kommen. In diesem Abschnitt soll nun eine ausführlichere Vorstellung sowie eine qualitative Bewertung im Vergleich zu WEDD erfolgen.

### **5.4.1 Scheduling**

Zunächst werden Verfahren für das zur Differenzierung hinsichtlich der Verzögerung notwendige Scheduling betrachtet. Neben der statischen Verzögerungspriorisierung und den FQ-Verfahren, die bereits in Abschnitt 3.2.1.1 vorgestellt wurden, spielen unter den gegebenen Randbedingungen vor allem Mechanismen eine Rolle, die speziell zur Unterstützung proportionaler Differenzierung entwickelt wurden.

#### **5.4.1.1 Statische Prioritäten**

Als einfaches Verfahren, um eine relative Differenzierung v. a. im Hinblick auf Verzögerungen zu realisieren, bietet sich eine statische Priorisierung bei der Abarbeitung von Paketen aus unterschiedlichen Klassen an. Ein solches Scheduling wird z. B. im Rahmen von DiffServ (siehe Abschnitt 3.3.3) vorgeschlagen, um Verkehr der EF-Klasse, die hauptsächlich für echtzeitkritische Anwendungen wie VoIP gedacht ist, von AF- und BE-Verkehr zu trennen [143]. Solange der Anteil des priorisierten Verkehrs gering ist, sind statische Prioritäten eine attraktive Lösung, da sie einfach zu implementieren sind – viele Router bieten eine priorisierte Behandlung bereits als Option an – und keine weiteren Parameter benötigen. Wenn allerdings dieser Anteil signifikant ist, kann die statische Priorisierung dazu führen, dass Klassen mit niedrigerer Priorität stärker benachteiligt werden als gewünscht – bis hin zu einem völligen Aushungern dieser Klassen. Dies kann nur dann ausgeschlossen werden, wenn der Zugang zu den Klassen mit höherer Priorität mit Hilfe einer Annahmesteuerung beschränkt wird, was aber den hier zugrunde liegenden Voraussetzungen gemäß Abschnitt 5.1 widersprechen würde.

Darüber hinaus kann das Fehlen von Parametern auch als Nachteil ausgelegt werden, da dem Netzbetreiber keinerlei Anpassungsmöglichkeiten zur Verfügung stehen. Negative Auswirkungen sind z. B. in Fällen denkbar, in denen ein Benutzer aus einer ganzen Reihe von Klassen wählen kann, die im Netz durch statische Prioritäten differenziert werden. Hier ist allgemein

zu beobachten, dass sich die Leistungsparameter innerhalb der niederpriorien ebenso wie innerhalb der hochpriorien Klassen jeweils kaum unterscheiden, während beim Übergang zur nächsthöheren Klasse ab einer bestimmten Klasse eine schlagartige Verbesserung auftritt. Ein solches Verhalten ist insbesondere dann wenig wünschenswert, wenn dem Nutzer oder einer adaptiven Anwendung die Möglichkeit gegeben werden soll, die Wahl der Prioritätsklasse dynamisch vorzunehmen und ggf. anzupassen.

Schließlich hängt die Auswirkung einer statischen Priorisierung sehr stark vom Puffermanagement ab. Während z. B. bei getrennten Puffern oder einer verdrängenden Pufferstrategie die Gefahr des Aushungerns niederpriorer Klassen besteht, kann es insbesondere im Fall von verlustsensitiven Anwendungen vorkommen, dass die Differenzierung durch Verzögerungsprioritäten allein kaum greift (siehe Kapitel 6).

#### 5.4.1.2 FQ-Verfahren

Eine weitere Alternative, die wie die statische Priorisierung bereits heute in vielen Routern implementiert ist, stellt die Verwendung von FQ-Mechanismen dar, wie sie in Abschnitt 3.2.1.1 vorgestellt wurden. Eine derartige Lösung ist insbesondere im Hinblick auf eine Architektur wie die in Bild 5.1 dargestellte interessant, da sich dort für den *Link Sharing Scheduler* ohnehin ein FQ-Mechanismus anbietet. Somit könnte man bei einer Verwendung von FQ zur Differenzierung innerhalb einer Gruppe (Echtzeit- bzw. elastischer Verkehr) prinzipiell leicht zu einer integrierten Lösung (vgl. Bild 5.2) gelangen.

Während allerdings FQ-Verfahren bei der Anwendung auf einzelne Verkehrsflüsse den Vorteil haben, dass für jeden Verkehrsfluss eine Mindestrate garantiert werden kann, wird unter den gegebenen Randbedingungen nur eine Zuteilung von Anteilen an der Linkbandbreite an aggregierte Verkehrsströme vorgenommen. Das bedeutet zunächst, dass die Fähigkeiten von FQ im Hinblick auf eine Isolation einzelner Verkehrsströme nicht zum Tragen kommen und somit z. B. kein Schutz von TCP-Verbindungen vor nicht-adaptiven Verkehrsflüssen in der gleichen Klasse besteht. Da eine Identifikation einzelner Verkehrsflüsse im Aggregat gemäß den in Abschnitt 5.1 genannten Voraussetzungen im Netzknoten nicht möglich und somit dort auch die Anzahl von Verkehrsflüssen innerhalb des Aggregats unbekannt ist, kann zudem die Auswirkung einer bestimmten Einstellung der Scheduler-Gewichte nicht vorhergesehen werden. Insbesondere kann eine statische Zuweisung der Gewichte dazu führen, dass in bestimmten Lastsituationen, z. B. bei einer vergleichsweise großen Anzahl von Verkehrsflüssen in den Klassen, die eigentlich eine bevorzugte Behandlung erfahren sollten, die einzelnen Flüsse in einer dieser Klassen trotz eines höheren Scheduler-Gewichts einen geringeren Durchsatz erfahren als die in einer niederpriorien Klassen.

Aufgrund dieser Problematik ist es erforderlich, die Gewichte  $\Phi_j$  der einzelnen Klassen dynamisch an die Lastsituation anzupassen. Im einfachsten Fall könnten als Basis dafür mit einem

EWMA-Mechanismus gemessene Ankunftsrate dienen. Dies führt jedoch im Fall von TCP-Verkehr nicht zum erwünschten Erfolg, da hier die Quellen ihre Senderate entsprechend reduzieren, sodass nicht von der Summenankunftsrate im aggregierten Verkehrsstrom auf die Anzahl der Verkehrsflüsse darin sowie deren Durchsatz geschlossen werden kann.

In [90] wird ein Mechanismus (*backlog-proportional rate scheduler*, BPR-Scheduler) vorgeschlagen, bei dem die momentane Bedienrate  $R_i(t)$  in Klasse  $i$  proportional zur aktuellen Warteschlangenlänge  $Q_i(t)$  in dieser Klasse und zu einem Differenzierungsparameter  $s_i$  eingestellt wird. Damit wird näherungsweise erreicht, dass das Verhältnis der mittleren Wartezeiten in zwei beliebigen Klassen dem Kehrwert des Verhältnisses der Differenzierungsparameter entspricht.<sup>1</sup> Allerdings zeigen die Untersuchungen in [90], dass diese Aussage nicht gilt, wenn die mittleren Verzögerungen innerhalb kürzerer Intervalle betrachtet werden.

Der in [198] beschriebene *Proportional Queue Control Mechanism* (PQCM) weist große Ähnlichkeit zu dem genannten BPR-Scheduler auf, konzentriert sich allerdings auf die Trennung der DiffServ-Klassen von EF- und AF-Verkehr, d. h. er unterstützt nur zwei Prioritäten.

#### 5.4.1.3 Scheduler mit proportionaler Differenzierung der mittleren Verzögerung

Ausgehend von der grundlegenden Arbeit von Dovrolis et al. [88, 90] wurden eine Reihe von Schemen vorgeschlagen, die eine proportionale Differenzierung bzgl. der mittleren Verzögerung von Paketen ermöglichen (*proportional delay differentiation*, PDD). Neben dem im letzten Abschnitt genannten BPR-Scheduler, der als eine Variante eines FQ-Schedulers angesehen werden kann, gibt es noch weitere Scheduling-Mechanismen mit speziellem Fokus auf diese Art der Differenzierung.

Als besonders geeignet hat sich in diesem Fall der ursprünglich von Kleinrock [155] vorgestellte und in [87, 88, 90] im Zusammenhang mit relativer Dienstgüte in IP-Netzen vorgeschlagene und untersuchte WTP-Scheduler erwiesen. Bei diesem Mechanismus wird diejenige Klasse  $i$  bedient, für die  $T_{Q,i}(t) \cdot s_i$ , also das Produkt aus der momentanen Wartezeit des nächsten Paketes in dieser Klasse und dem zugehörigen Differenzierungsparameter, am größten ist. Wie beim BPR-Scheduler ergibt sich ein Verhältnis der mittleren Wartezeiten, das dem umgekehrten Verhältnis der Differenzierungsparameter folgt<sup>1</sup>. Allerdings trifft dies weit genauer zu als beim BPR-Scheduler und gilt vor allem auch in Bezug auf das Kurzzeitverhalten. Ein Nachteil des Verfahrens ist jedoch, dass unter gewissen Bedingungen (büschelförmige Ankünfte in den höher priorisierten Klassen mit einer Ankunftsrate, die größer ist als die Bedienrate) ein temporäres Aushungern der nieder priorisierten Klassen zu beobachten ist [90]. Außerdem ist – wie bei WEDD – für jedes Paket ein Zeitstempel erforderlich.

---

<sup>1</sup> Ein höherer Wert von  $s_i$  bedeutet hier also – im Gegensatz zu den Gewichtungsparemtern bei WEDD – eine Besserstellung von Klasse  $i$ .

Der in [202] beschriebene MDP-Scheduler (*mean delay proportional*) arbeitet ähnlich wie WTP, benutzt jedoch anstelle der Wartezeit  $T_{Q,i}(t)$  des nächsten Pakets in Klasse  $i$  eine Abschätzung für die mittlere Verzögerung von Paketen, die auf der Verwendung mehrerer Zählvariablen in jeder Klasse basiert. Dafür kann auf Zeitstempel für jedes Paket verzichtet werden. Das Verfahren soll eine noch bessere Annäherung an das PDD-Modell in Situationen mit mäßiger Last und bei Betrachtung kurzer Zeitintervalle liefern [202].

Eine weitere Alternative stellt der in [174] präsentierte PLQ-Scheduler (*probabilistic longest queue first*) dar. Hierbei wird mit einer von der aktuellen Warteschlangenlänge  $Q_i(t)$  in Klasse  $i$  und dem zugehörigen Differenzierungsparameter  $s_i$  abhängigen Wahrscheinlichkeit

$$p_i = \frac{s_i \cdot Q_i(t)}{\sum_{k=0}^{K-1} s_k \cdot Q_k(t)} \quad (5.14)$$

ein Paket der Klasse  $i$  ( $i \in \{0, \dots, K-1\}$ ) zur Bedienung ausgewählt. Wie beim WTP- und beim MDP-Scheduler ergibt sich näherungsweise ein Verhältnis der mittleren Verzögerungen, das dem umgekehrten Verhältnis der Differenzierungsparameter entspricht. Die Abweichungen von diesem Idealverhältnis sind allerdings größer als bei den oben genannten Bedienstrategien. Jedoch weist PLQ den Vorteil einer einfacheren Implementierung auf. Mit dem PLQ-Scheduler ist außerdem ein Verwerfungsmechanismus verbunden, der ebenfalls auf der Basis der Warteschlangenlängen arbeitet (siehe Abschnitt 5.4.2.3).

Die beschriebenen Scheduling-Verfahren eignen sich prinzipiell zur proportionalen Differenzierung von TCP-Verkehr, da dort der Durchsatz umgekehrt proportional zur mittleren RTT ist [212]. Allerdings gilt dies nur im Fall von langen TCP-Verbindungen, die als ungesättigte Quellen modelliert werden können, und auch dann nur, wenn die RTT maßgeblich von der Wartezeit in den Netzknoten bestimmt wird. Im Allgemeinen setzt sich die mittlere RTT aus einer konstanten Verzögerung  $\tau$  und der von der Klassenzugehörigkeit abhängigen mittleren Wartezeit  $d_i$  zusammen. Damit erhält man, wie in [87] gezeigt, als Verhältnis der Werte für den mittleren Nutzdurchsatz  $g_i$  bzw.  $g_j$  in zwei Klassen  $i$  und  $j$ :

$$\frac{g_i}{g_j} = \frac{\tau + d_j}{\tau + d_i} = \frac{\frac{\tau}{d_i} + s_{ij}}{\frac{\tau}{d_i} + 1} \rightarrow s_{ij} \quad \text{für } \frac{\tau}{d_i} \rightarrow 0 \quad (5.15)$$

wobei  $s_{ij} = s_i/s_j$  das Verhältnis der eingestellten Differenzierungsparameter angibt und angenommen wird, dass die absolute Verzögerung  $\tau$  ebenso wie die Verlustwahrscheinlichkeit für alle Verkehrsflüsse gleich ist. Das bedeutet, dass eine proportionale Durchsatzdifferenzierung ohne Kenntnis von  $\tau$  und  $d_i$ , von der i. d. R. nicht ausgegangen werden kann, nicht exakt erreicht werden kann.

Neben der Anwendung auf elastischen Verkehr ist auch die Nutzung zur Differenzierung von Echtzeitverkehr denkbar. Dabei ist jedoch aus Anwendungssicht nicht der Mittelwert der Verzögerung entscheidend, sondern die Wahrscheinlichkeit, dass die Verzögerung einen bestimmten Wert nicht überschreitet. Aus dieser Sicht erscheint für Echtzeitverkehr der fristbasierte Ansatz von WEDD besser geeignet.

## **5.4.2 Puffermanagement**

In Bezug auf das Puffermanagement müssen für einen Vergleich zunächst die in Abschnitt 3.2.1.2 als „klassische“ Verfahren bezeichneten Mechanismen herangezogen werden. Darüber hinaus spielen im Zusammenhang mit relativer Differenzierung RED-basierte Verwerfungsmechanismen sowie die speziell unter dem Blickwinkel der proportionalen Differenzierung entwickelten Verfahren eine große Rolle.

### **5.4.2.1 Klassische Verfahren**

Die drei klassischen Verfahren (Puffertrennung, Verdrängung und Schwellwertverfahren) wurden ursprünglich zur differenzierten Behandlung von ATM-Zellen mit unterschiedlichem CLP-Bit vorgeschlagen und in diesem Kontext untersucht [160, 161]. Eine Anwendung auf das gegebene Szenario mit potenziell mehr als zwei Verlustprioritäten ist aber bei allen drei Verfahren ohne weiteres möglich.

#### **Puffertrennung**

In Analogie zu den FQ-Strategien kann durch eine Aufteilung des zur Verfügung stehenden Speicherplatzes in unterschiedlich große Anteile, die exklusiv von den jeweiligen Klassen genutzt werden können, Einfluss auf die jeweiligen Verlustwahrscheinlichkeiten genommen werden. Allerdings tritt wie bei FQ das Problem auf, dass die Auswirkung einer bestimmten Aufteilung von der Lastverteilung abhängt und damit schwer vorhersehbar ist. Wie im Fall der Gewichte bei FQ (siehe Abschnitt 5.4.1.2) garantiert auch hier ein größerer Pufferanteil für eine bestimmte Klasse nicht, dass die Verluste in dieser Klasse geringer sind. Puffertrennung mit jeweils gleich großen Anteilen spielt jedoch eine Rolle, wenn sie im Zusammenhang mit einem differenzierenden Scheduler (z. B. einem FQ-Scheduler) verwendet wird, da in einem solchen Fall eine vollständig gemeinsame Nutzung des Puffers annähernd gleiche Verlustwahrscheinlichkeiten für alle Klassen ergeben würde und somit die Ziele des Scheduling (z. B. eine Durchsatzdifferenzierung im Fall von FQ) zunichte machen würde.

#### **Verdrängung**

Der Verdrängungsmechanismus ist von seiner prinzipiellen Arbeitsweise her vergleichbar mit der statischen Priorisierung beim Scheduling und weist daher auch ähnliche Vor- und Nach-

teile auf. Dies bedeutet insbesondere, dass Verdrängung im Fall eines hohen Angebots an priorisiertem Verkehr ein vollständiges Aushungern in den Klassen mit geringer Priorität bewirken kann. Wie in [160] gezeigt wird, ist außerdem die Implementierung des Verdrängungsmechanismus im Vergleich zu den beiden anderen klassischen Puffermanagementverfahren deutlich komplexer. Allerdings wiegt dieser Nachteil angesichts der heute gegebenen Möglichkeiten, auch komplexere Algorithmen in Hardware zu realisieren, nicht mehr so schwer [60].

### Schwellwertverfahren

Das Schwellwertverfahren stellt in doppelter Hinsicht einen Kompromiss zwischen Puffertrennung und Verdrängung dar. Zum einen ist es in Bezug auf Komplexität zwischen den beiden anderen Verfahren einzustufen. Zum anderen garantiert es im Gegensatz zur Puffertrennung zumindest im statistischen Sinn eine Besserstellung der Klasse mit höherem Schwellwert, ohne dass Pakete mit niedriger Priorität bei Ressourcenknappheit völlig aus dem Puffer verdrängt werden. Allerdings hängt das konkrete Verhalten sehr stark von den Schwellwerten in den einzelnen Klassen ab, sodass auch hier bei unbekannter Lastverteilung und Verkehrscharakteristik sowohl absolute Verluste als auch das Verhältnis der Verlustwahrscheinlichkeiten kaum vorhersehbar sind. Die Zusammenhänge sind noch schwieriger zu beherrschen, wenn eine teilweise gemeinsame Nutzung des Puffers mit unterschiedlichen Schwellwerten zusammen mit einem differenzierenden Scheduling eingesetzt wird [118].

#### 5.4.2.2 Verfahren auf der Basis von RED

Im Kontext von IP-Netzen haben RED-basierte Puffermanagementverfahren große Beachtung gefunden. Geht es dabei um eine differenzierte Behandlung, ist in erster Linie das in Abschnitt 3.2.1.2 vorgestellte WRED zu nennen, bei dem die Parameter  $min_{th,i}$ ,  $max_{th,i}$  und  $max_{p,i}$  in den einzelnen Klassen unterschiedliche Werte annehmen können. Eine proportionale Verlustdifferenzierung kann mit WRED näherungsweise erreicht werden, indem für  $min_{th,i}$  und  $max_{th,i}$  in allen Klassen identische Werte gewählt und nur für  $max_{p,i}$  unterschiedliche Werte gesetzt werden [38]. Für das Verhältnis der Verwerfungswahrscheinlichkeiten erhält man dann:

$$\frac{p_i}{p_j} = \frac{max_{p,i}}{max_{p,j}} \quad (5.16)$$

Voraussetzung dafür ist allerdings, dass für die Schwellenparameter ebenso wie für weitere RED-Parameter (z. B. die Gewichtung bei der Mittelwertbildung in Bezug auf die Warteschlangenlänge) angemessene Werte eingesetzt werden. Die idealen Werte hierfür hängen u. a. von der Puffergröße sowie der Charakteristik des Verkehrs ab.

Ein großer Vorteil von WRED ist seine weite Verbreitung und Bekanntheit, die dazu geführt haben, dass es bereits heute in vielen Routern implementiert ist. Allerdings zielen RED-basierte Verfahren in erster Linie auf TCP-Verkehr ab, wobei Untersuchungen in der Literatur

fast ausschließlich auf der Annahme ungesättigter TCP-Quellen (siehe Abschnitt 4.2.2.2) beruhen. Unter dieser Annahme gilt – wie bereits in Abschnitt 5.3.1 erwähnt – näherungsweise, dass der Durchsatz proportional zum Kehrwert der Wurzel aus der Verlustwahrscheinlichkeit ist, sodass sich wie bei WEDD eine proportionale Durchsatzdifferenzierung ergibt:

$$\frac{g_i}{g_j} = \sqrt{\frac{\max_{p,j}}{\max_{p,i}}} \quad (5.17)$$

Eine zusätzliche Unterscheidung hinsichtlich der Verzögerung erlaubt WRED im Gegensatz zu WEDD allerdings nicht. Damit eignet sich WRED ohne zusätzlichen Scheduler praktisch ausschließlich zur Differenzierung von elastischem Verkehr.

### 5.4.2.3 Verfahren zur proportionalen Verlustdifferenzierung

Neben WRED sind aus der Literatur noch weitere Verfahren bekannt, die eine proportionale Verlustdifferenzierung ermöglichen und die teilweise – im Unterschied zu WRED – speziell vor diesem Hintergrund entwickelt wurden. Die meisten dieser Verfahren sind entweder parallel zu WEDD entstanden [37, 89, 174] oder wurden in einem anderen Kontext entwickelt [277]. Diese Verfahren weisen z. T. große Ähnlichkeit mit einem WEDD-Scheduler auf, bei dem für die maximale Verzögerung in allen Klassen gleiche Werte gewählt werden.

Eines der bekanntesten Verfahren in diesem Zusammenhang ist der von Dovrolis in [89] vorgestellte PLR-Verwerfungsmechanismus (*proportional loss rate*). Hier wird die Entscheidung, wann ein Paket verworfen wird, von einem separaten Modul getroffen, das die Gesamtwarteschlangenlänge beobachtet. Eine einfache Realisierung dieses Moduls stößt eine Verwerfung an, wenn der Puffer voll ist. Die Auswahl der Klasse, aus der ein Paket verworfen werden soll, erfolgt wie bei WEDD durch Vergleich von Schätzwerten der bisherigen Verluste in jeder Klasse, die jeweils durch einen Differenzierungsparameter dividiert werden. Die Abschätzung kann analog zu den in Abschnitt 5.3.3 beschriebenen Lösungen erfolgen. Alternativ dazu wird in [89] ein Algorithmus vorgeschlagen, bei dem für die  $M$  zuletzt angekommenen Pakete in einer Tabelle gespeichert wird, zu welcher Klasse jedes Paket gehörte und ob es verworfen wurde. Auf Basis der Informationen in dieser Tabelle werden Schätzwerte für die Verlusthäufigkeiten in den verschiedenen Klassen ermittelt. Ein weiteres Verfahren, das auf einer EWMA-basierten Abschätzung der mittleren Abstände zwischen Verlustereignissen in verschiedenen Klassen beruht, wird in [37] präsentiert.

Das bereits in Abschnitt 5.4.1.3 genannte PLQ-Verfahren [174] umfasst neben einem differenzierenden Scheduling auch einen Verwerfungsmechanismus, der die aktuellen Warteschlangenlängen  $Q_i(t)$  in den einzelnen Klassen berücksichtigt. Die Entscheidung, aus welcher Klasse das zu verwerfende Paket gewählt wird, erfolgt zufällig mit unterschiedlichen Wahrscheinlichkeiten für jede Klasse, die proportional zu  $\sqrt{Q_i(t)/s_i}$  sind. Näherungsweise erhält

man dadurch ein Verhältnis der Verwerfungswahrscheinlichkeiten, das umgekehrt proportional zum Verhältnis der Differenzierungsparameter  $s_i$  ist. Allerdings ist diese Näherung, wie in [174] gezeigt, weitaus weniger exakt als z. B. bei Anwendung des PLR-Verfahrens.

### 5.4.3 Integrierte Verfahren

In jüngster Zeit wurden zwei Verfahren vorgeschlagen [124, 175], die wie WEDD Lösungen darstellen, bei denen Scheduling und Puffermanagement integriert werden. Beide Verfahren ermöglichen eine proportionale Differenzierung und weisen daher eine enge Verwandtschaft zu WEDD auf.

Im Fall des von Liebeherr und Christin vorgestellten JoBS (*joint buffer management and scheduling*) [175] werden in Bezug auf die Verzögerung sowohl relative als auch absolute Garantien gegeben. Relative Garantien bedeuten hier – wie bei den in Abschnitt 5.4.1.3 beschriebenen Scheduling-Algorithmen – eine proportionale Differenzierung im Hinblick auf die mittlere Verzögerung von Paketen. Unter absoluten Garantien sind zusätzlich definierbare Werte für die Maximalverzögerung (wie im Fall von WEDD) zu verstehen. Darüber hinaus wird eine proportionale Verlustdifferenzierung ermöglicht, die ebenfalls durch Angabe eines Maximalwerts der Verlustwahrscheinlichkeit in jeder Klasse ergänzt werden kann. Die gegebenen Randbedingungen werden auf ein Optimierungsproblem abgebildet, dessen Lösung die Bedienreihenfolge der im System befindlichen Pakete und eine Entscheidung über zu löschende Pakete liefert. Da die Implementierung einer solchen Optimierung in einem Router kaum realisierbar ist, wird außerdem eine heuristische Approximation vorgeschlagen. Neben der Art der Realisierung ist ein Unterschied zu WEDD, dass bei JoBS stets davon ausgegangen wird, dass eine gültige Lösung existiert, d. h. dass die Maximalwerte von Verzögerungen und Verlustwahrscheinlichkeiten eingehalten werden.

Ebenso wie WEDD stellt auch das von Hei und Tsang in [124] vorgeschlagene und als E-LOSS bezeichnete Verfahren eine Erweiterung von EDD dar, bei der eine proportionale Differenzierung der Fristüberschreitungswahrscheinlichkeiten ermöglicht wird. In Analogie zur LPD-Option bei WEDD ist auch hier vorgesehen, Pakete zu verwerfen, die bereits ihre Frist überschritten haben. Auch bzgl. der Realisierung sind die Verfahren sehr ähnlich: Wie bei WEDD kann zwischen einem „Normalbetrieb“, in dem weitgehend die EDD-Strategie angewandt wird, und einem „Überlastmodus“ unterschieden werden. Während allerdings bei WEDD ein Übergang in den Überlastmodus stattfindet, wenn der zeitliche Abstand zur Frist einen bestimmten Wert unterschritten hat, findet im Fall von E-LOSS der Übergang bei Überschreitung eines klassenspezifischen Schwellwertes der jeweiligen Wartschlängellänge statt. Darüber hinaus ist E-LOSS nur für die Anwendung auf echtzeitkritischen Verkehr vorgesehen.

## 5.5 Ende-zu-Ende-Betrachtung

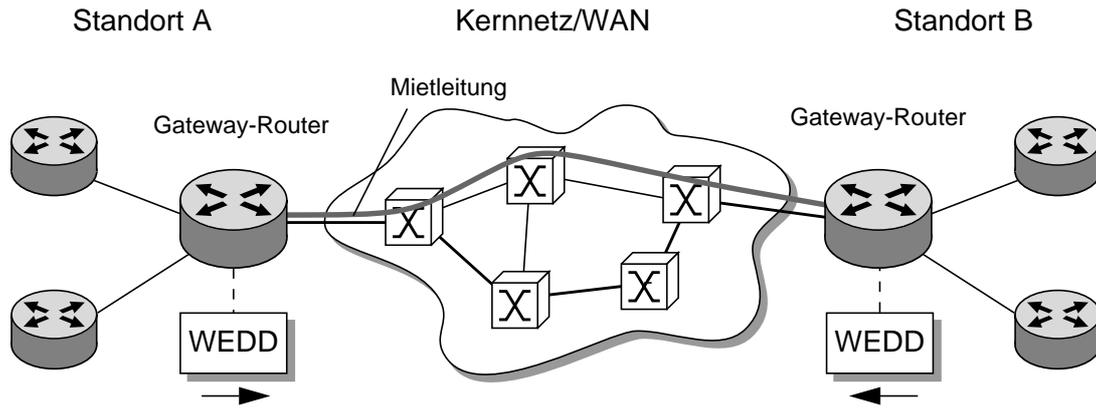
Das in diesem Kapitel vorgestellte WEDD-Verfahren bietet – ebenso wie die in Abschnitt 5.4 diskutierten Alternativen – zunächst nur eine differenzierte Behandlung von Verkehrsströmen innerhalb eines Knotens. Aus Nutzersicht hingegen liegen Dienstgütereigenschaften auf einer Ende-zu-Ende-Basis vor, was die Forderung nach einer durchgängigen Dienstgüteunterstützung nach sich zieht. Allerdings ist angesichts der Erfahrungen der letzten Jahre die Annahme, dass sich in allen IP-Netzen eine einheitliche Dienstgütearchitektur durchsetzen wird, eher unrealistisch. Viel wahrscheinlicher ist eine Kombination verschiedener Ansätze; diverse Möglichkeiten wurden bereits in Abschnitt 3.3.4 aufgezeigt. Für die hier im Fokus stehende relative Differenzierung sind vor allem zwei Varianten interessant:

- DiffServ wird – entsprechend seiner ursprünglichen Intention – als Kernnetztechnologie eingesetzt, wobei relative Dienstgütegarantien gegeben werden. Nutzer können dann mit Hilfe eines Reservierungsprotokolls wie RSVP Ressourcen reservieren. Die Reservierungen werden im Zugangsbereich durch Knoten realisiert, die einzelne Verkehrsflüsse unterscheiden können. Beim Übergang in das DiffServ-Netz erfolgt die Abbildung auf geeignete DiffServ-Klassen, die entweder statisch oder – wie in [87, 202] vorgeschlagen – dynamisch erfolgt mit dem Ziel, Verzögerungs- und Verlustanforderungen befriedigen zu können.
- Relative Differenzierung wird nur im Zugangsbereich angewandt, während im Kernnetz für die Verkehrsströme der unterschiedlichen Klassen – z. B. über ATM oder MPLS – Bandbreite mit absoluten Garantien reserviert wird oder insgesamt so viel Kapazität bereitgestellt wird, dass keine Engpässe zu erwarten sind (*overprovisioning*). Ein solches Szenario ist beispielsweise im Fall eines Firmennetzes (*virtual private network*, VPN) realistisch, bei dem nur der Zugangsbereich unter der direkten Kontrolle der Firma ist und in dem Mietleitungen (*leased lines*)<sup>2</sup> zur Verbindung der Firmenstandorte eingesetzt werden (Bild 5.5). Relative Differenzierung ist dabei vor allem in den Knoten wichtig, die den Zugang zu diesen Mietleitungen ermöglichen.

Allerdings bleibt selbst bei einer Beschränkung der Betrachtung auf einen Teilbereich des Netzes, indem eine einheitliche Dienstgütearchitektur (mit relativer Differenzierung) angewandt wird, das Problem, wie mit Hilfe der beschriebenen Scheduling- und Puffermanagementverfahren relative Garantien über mehrere Netzknoten hinweg gegeben werden können. In Bezug auf WEDD stellt sich konkret die Frage nach der kumulierten Verzögerung und der Gesamtverwerfungswahrscheinlichkeit  $p_{i,g}$ . Letztere ergibt sich aus den Verwerfungswahrscheinlich-

---

<sup>2</sup>Hierbei kann es sich auch um die Emulation einer Mietleitung (vergleichbare Eigenschaften, begrenzte Bandbreite, aber kein fester Kanal), eine so genannte virtuelle Mietleitung (*virtual leased line*, VLL, vgl. Abschnitt 3.3.3.3), handeln [103].



**Bild 5.5:** Einsatz von WEDD in einem VPN-Szenario

keiten  $p_i^{(v)}$  für Verkehr der Klasse  $i$  in den einzelnen Vermittlungsknoten zwischen Quelle und Ziel, deren Anzahl mit  $V$  bezeichnet werde:

$$p_{i,g} = 1 - \prod_{v=1}^V (1 - p_i^{(v)}) \approx \sum_{v=1}^V p_i^{(v)} \quad (5.18)$$

Die Näherung ist um so genauer, je kleiner die Einzelverwerfungswahrscheinlichkeiten sind. Unter dieser Voraussetzung folgt dann, dass sich auch die Gesamtwahrscheinlichkeiten proportional zu den Gewichtsparametern verhalten, falls in allen Knoten die gleichen Gewichte gewählt werden. In diesem Zusammenhang muss allerdings erwähnt werden, dass ein Verwerfen von TCP-Paketen (z. B. anstelle einer weiteren Verzögerung derselben) in einem Netzszenario nicht ganz unproblematisch ist. Dadurch dass ein Verwerfen im Knoten  $v$  eine Paketwiederholung zur Folge hat, erhöht sich prinzipiell die Last in allen Vorgängerknoten von  $v$ . Das bedeutet, dass eine Erhöhung von  $p_i^{(v)}$  in Knoten  $v$  auch ein Ansteigen der Verwerfungswahrscheinlichkeiten in den Vorgängerknoten zur Folge haben kann. In ungünstigen Szenarien kann das zu einer Ausbreitung von Überlast im Netz beitragen.

Ein anderes Problem stellt der Übergang von einer maximalen Verzögerung in den einzelnen Knoten zu einer maximalen Gesamtverzögerung dar. Aus der Sicht des Netzbetreibers bedeutet dies: Welche Verzögerungsklassen – mit jeweiliger maximaler Gesamtverzögerung  $\delta_{i,g}$  – können dem Dienstinutzer angeboten werden und wie müssen dafür die maximalen Verzögerungen  $\delta_i^{(v)}$  in den einzelnen Knoten eingestellt werden?

Am einfachsten lässt sich dieses Problem für den Fall lösen, dass kein Engpass gleichzeitig in mehr als einem Knoten im Netz auftritt. Dies kann z. B. im Fall des oben beschriebenen VPN-Szenarios mit einem stark belasteten Gateway-Knoten zutreffend sein. In diesem Fall wird man für die Verzögerungsparameter von WEDD jeweils einen Wert wählen, welcher der gewünschten maximalen Gesamtverzögerung in der jeweiligen Klasse entspricht, reduziert um

einen konstanten Wert, mit dem die Verzögerung in den anderen Teilen des Netzes berücksichtigt wird.

Ein anderer Ansatz besteht darin, die WEDD-Parameter so zu wählen, dass auf allen Wegen durch das Netz die Summe der Einzelverzögerungen in den Knoten entlang des Weges die gewünschte Gesamtverzögerung nicht überschreitet. Dieser Ansatz ist als sehr konservativ zu bewerten, was insbesondere bei Verwendung der LPD-Option dazu führen kann, dass häufig im Einzelknoten verspätete Pakete verworfen werden, obwohl die maximale Gesamtverzögerung nicht überschritten ist. Außerdem besteht bei dieser Lösung auch eine große Abhängigkeit von den gewählten Routen durch das Netz und damit vom Verkehrslenkungsverfahren.

Schließlich besteht noch die Möglichkeit, einen Ansatz vergleichbar zu dem in [173] vorgeschlagenen zu wählen, bei dem in jedem Knoten entlang des Weges zum Ziel die bereits erfahrene Verzögerung eines Paketes berücksichtigt wird. Angewandt auf den gegebenen Fall eines DiffServ-Netzes mit Verwendung von WEDD bedeutet dies, dass ein Paket, das im Knoten  $v$  der Klasse  $i$  mit maximaler Verzögerung  $\delta_i^{(v)}$  angehört und eine tatsächliche Verzögerung  $T_D^{(v)}$  erfahren hat, nach dem Durchlaufen von Knoten  $v$  einer anderen Dienstgütekategorie  $j$  zugeordnet wird. Dabei wird  $j$  so gewählt, dass für die maximale Verzögerung  $\delta_j^{(v+1)}$  im nächsten Knoten in dieser Klasse gilt:

$$\delta_j^{(v+1)} \leq \delta_i^{(v)} - T_D^{(v)} \quad (5.19)$$

Dazu müssen allerdings die im Nachfolgeknoten eingestellten WEDD-Parameter bekannt sein, was sich leicht realisieren lässt, wenn grundsätzlich in allen Knoten gleiche Werte für die Parameter gewählt werden. Außerdem ist diese Lösung nur dann effizient, wenn eine ausreichend große Anzahl an Klassen mit unterschiedlichen Werten für die maximale Verzögerung existiert. Als Beispiel sei ein System mit 32 Klassen genannt, die durch Kombination von 16 Werten für die maximale Verzögerung und zwei unterschiedlichen Werten für die Gewichtungsparemeter gebildet werden.

Insgesamt muss die Entwicklung einer Lösung, die konkrete Ende-zu-Ende-Anforderungen an die maximale Verzögerung erfüllt und gleichzeitig effizient arbeitet, als sehr schwierig angesehen werden. Die Probleme werden durch die Heterogenität einer realen Netz- und Dienstgütearchitektur und durch die in Abschnitt 3.1.6 angesprochenen Verzögerungen innerhalb der Endgeräte, die im Netz weder beeinflussbar noch bekannt sind, weiter verschärft. Allerdings muss betont werden, dass Verfahren wie WEDD nicht nur interessant sind, wenn es um die Einhaltung einer fest vorgegebenen Ende-zu-Ende-Verzögerung geht. Sie sind zunächst v. a. dazu geeignet, die Verzögerung in einzelnen Knoten zu begrenzen und damit die grundsätzlichen Voraussetzungen für die Nutzung des Netzes für echtzeitkritische Dienste durch Bereitstellung von Ressourcen im Sinne eines *Provisioning* zu schaffen.

## Kapitel 6

# Leistungsuntersuchung eines IP-Netzknotens mit Dienstgütedifferenzierung

Dieses Kapitel präsentiert die Ergebnisse einer Leistungsbewertung des in Kapitel 5 vorgestellten WEDD-Verfahrens zur relativen Dienstgütedifferenzierung in IP-Netzen. Die Betrachtung beschränkt sich dabei auf einen Ausgang eines einzelnen Netzknotens (vgl. Bild 3.1), der für den dort ankommenden Verkehr aus unterschiedlichen Dienstgüteklassen einen Engpass in Bezug auf die Bandbreite darstellt.

Zunächst werden Studien durchgeführt, in denen nur die Differenzierung von Echtzeitverkehr untersucht wird, basierend auf einem Verkehrsmodell, das keine Adaptivität vorsieht. Anschließend erfolgt eine Bewertung für TCP-basierten, also elastischen Verkehr. Dabei werden sowohl ungesättigte TCP-Quellen als auch dynamischer Verkehr mit kurzen TCP-Verbindungen berücksichtigt. Abschließend findet eine Leistungsuntersuchung für ein integriertes Szenario statt, das sowohl Echtzeit- als auch TCP-Verkehr enthält.

Die nachfolgenden Untersuchungen erfolgen zumeist simulativ, wobei die Ergebnisse teilweise in Relation zu analytisch bestimmten Bezugsgrößen betrachtet werden (siehe Anhang A). Neben der Variation des Verkehrsmodells wird der Einfluss von Verkehrsangebot und -zusammensetzung, von Systemparametern wie Linkrate und Puffergröße sowie von diversen Parametern und Realisierungsoptionen von WEDD gemäß Abschnitt 5.3 untersucht. An einigen Stellen werden Vergleiche mit anderen Verfahren zur relativen Differenzierung vorgenommen. Außerdem wird im Fall von elastischem Verkehr auf Ergebnisse zurückgegriffen, die sich im Fall eines IP-Netzknotens ohne Dienstgütedifferenzierung ergeben (siehe Anhang B).

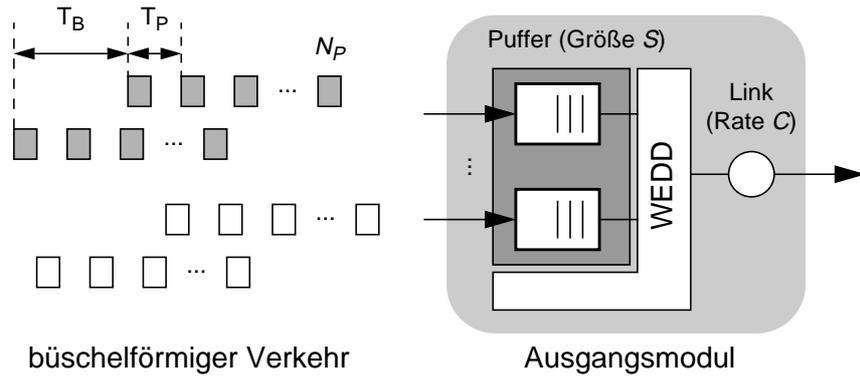
## 6.1 Differenzierung von Echtzeitverkehr

Eine der Aufgaben von WEDD besteht darin, verschiedene Klassen von Echtzeitverkehr mit unterschiedlichen Anforderungen an maximale Verzögerung und relative Verwerfungswahrscheinlichkeit zu unterstützen. Dazu wird zunächst nur dieser Teil des Verkehrs betrachtet, der als unelastischer Verkehr modelliert werden kann, und gleichzeitig angenommen, dass hierfür die gesamte Linkbandbreite (im integrierten Modell gemäß Bild 5.2) oder ein fest vorgegebener Anteil davon (z. B. realisiert durch einen *Link Sharing Scheduler* zur Trennung von TCP- und Echtzeitverkehr wie in Bild 5.1) zur Verfügung steht. Nach der Vorstellung des verwendeten System- und Verkehrsmodells (Abschnitt 6.1.1) wird zunächst die Fähigkeit von WEDD unter Beweis gestellt, den grundsätzlichen Anforderungen einer proportionalen Differenzierung gerecht zu werden (Abschnitt 6.1.2). Anschließend erfolgt in Abschnitt 6.1.3 eine Untersuchung des Einflusses der diversen in Abschnitt 5.3 vorgestellten Realisierungsoptionen und Parameter.

### 6.1.1 Modell

Für die Untersuchungen mit unelastischem Verkehr wird das in Bild 6.1 dargestellte Modell zugrunde gelegt. Dabei wird nur das Ausgangsmodul betrachtet, das den Zugang zu einem Ausgangslink mit Rate  $C$ . Der zugehörige Ausgangspuffer mit Gesamtgröße  $S$  ist logisch unterteilt in  $K$  Warteschlangen, die mit Hilfe von WEDD abgearbeitet werden. In allen Untersuchungen mit unelastischem Verkehr wird von einer Linkrate von  $C = 10$  Mbit/s ausgegangen. Die Puffergröße sei – mit Ausnahme der Untersuchungen in Abschnitt 6.1.3.5 – so gewählt, dass kein Pufferüberlauf und somit keine Verluste am Puffereingang auftreten. Als Standardkonfiguration für WEDD wird in allen Studien, in denen keine gegenteiligen Angaben gemacht werden, angenommen, dass die LPD-Option genutzt wird, während das CSA-Verfahren sowie eine Begrenzung der Zählvariablen bei der Schätzung der Überschreitungshäufigkeit nicht angewandt werden (siehe Abschnitt 5.3).

Für die Beschreibung des Echtzeitverkehrs sollen, beziehend auf Abschnitt 4.2.3, nur einfache Modelle für büschelförmigen Verkehr verwendet werden. Soweit in den nachfolgenden Studien keine anderen Aussagen getroffen werden, wird die Ankunft der Pakete mit Hilfe des in Abschnitt 4.2.1.2 beschriebenen  $M/G/\infty$ -Burstmodells charakterisiert, das auch als Überlagerung einer großen Zahl von On-Off-Quellen interpretiert werden kann. Dabei kommen Bursts mit zufälligem Abstand  $T_B$  im System an, wobei  $T_B$  einer negativ-exponentiellen Verteilung folgt. Innerhalb eines Bursts treffen Pakete mit konstantem Abstand  $T_P$  ein. Für die Burstgröße  $B$  bzw. die Anzahl  $N_P$  der Pakete im Burst werden unterschiedliche Verteilungen untersucht.



**Bild 6.1:** Modell für die Untersuchungen mit unelastischem Verkehr

Das  $M/G/\infty$ -Modell wird jeweils für den Ankunftsprozess der Pakete einer bestimmten Klasse verwendet. Das bedeutet insbesondere, dass alle Pakete innerhalb eines Bursts zur selben Klasse gehören. Die mittlere Rate  $m_i$  in Klasse  $i$  ( $i = 0, 1, \dots, K - 1$ ) ergibt sich je nach Realisierungsform des  $M/G/\infty$ -Modells mit Hilfe von Gleichung (4.6) bzw. Gleichung (4.7) aus den Erwartungswerten von  $T_B$  und  $N_P$  sowie der Paketlänge  $L$  bzw. dem Erwartungswert von  $B$ . Eine Variation von  $m_i$  wird dabei durch veränderte Werte für  $E[T_B]$  erreicht, während  $E[N_P]$  sowie  $L$  bzw.  $E[B]$  festgehalten werden. Das Angebot an Verkehr der Klasse  $i$  wird durch

$$A_i = \frac{m_i}{C} \quad (6.1)$$

bestimmt. Die mittlere Gesamtrate  $m$  sowie das Gesamtangebot  $A$  ergeben sich aus der Summation der Werte für die mittlere Raten  $m_i$  bzw. der Angebote  $A_i$  der einzelnen Klassen.

Die genaue Charakteristik des Verkehrs wird von der Paketlänge, der Paketrate im Burst sowie von der Verteilung der Burstgröße bestimmt. In Tabelle 6.1 sind die Parameter von fünf Szenarien genannt, die in den nachfolgenden Untersuchungen verwendet werden. Dabei liegen die beiden unterschiedlichen Realisierungen des  $M/G/\infty$ -Modells gemäß Abschnitt 4.2.1.2 zugrunde. In den Szenarien  $V_A$ ,  $V_C$  und  $V_D$  ist die Paketlänge  $L$  konstant; die Burstgröße  $B$  ergibt sich als Produkt von  $L$  und der Anzahl  $N_P$  an Paketen im Burst, die der angegebenen Verteilung folgt. Die hier verwendete um eins verschobene geometrische Verteilung für  $N_P$  erhält man anschaulich aus einer fiktiven Burstgröße  $\tilde{B}$  mit negativ-exponentieller Verteilung durch

$$N_P = \lceil \tilde{B}/L \rceil \quad (6.2)$$

Dabei ist zu beachten, dass der Mittelwert von  $B = N_P \cdot L$  größer ist als der von  $\tilde{B}$ :

**Tabelle 6.1:** Verkehrsszenarien

Bezeichnung	$L$	$T_P$	$R_P$	Burstgröße	
				Verteilung <sup>a</sup>	Mittelwert
$V_A$	200 Byte	8 ms	200 kbit/s	geometrisch	$40 \cdot 200$ Byte
$V_B$	200 Byte	8 ms	200 kbit/s	Pareto, $\alpha = 1.6$	8000 Byte
$V_C$	200 Byte	20 ms	80 kbit/s	geometrisch	$4000 \cdot 200$ Byte
$V_D$	1000 Byte	40 ms	200 kbit/s	geometrisch	$8 \cdot 1000$ Byte
$V_E$	$\leq 1000$ Byte	10.42 ms	768 kbit/s	empirisch	4706 Byte
$V_F$	$\leq 1000$ Byte	20.83 ms	384 kbit/s	empirisch	2232 Byte

<sup>a</sup> Im Fall der Verkehrsszenarien  $V_A$ ,  $V_C$  und  $V_D$  ist die Verteilung der Anzahl  $N_P$  von Paketen im Burst gemeint.

$$E[B] = \frac{1}{1 - \exp\left(-\frac{L}{E[\tilde{B}]}\right)} \rightarrow E[\tilde{B}] + \frac{L}{2} \quad \text{für } E[\tilde{B}] \rightarrow \infty \quad (6.3)$$

Demgegenüber wird bei  $V_B$ ,  $V_E$  und  $V_F$  direkt die Verteilung für die Burstgröße  $B$  angegeben. Ein Burst besteht in diesem Fall aus  $N_P - 1$  Paketen mit maximaler Größe  $L_{max}$  (hier 1000 Byte) und einem kleineren Paket. Im Fall von  $V_E$  und  $V_F$  werden empirische Verteilungen der Burstgröße  $B$  herangezogen, die aus den in [86] beschriebenen Labormessungen von Video-konferenzverkehr gewonnen wurden. Die Rate  $R_P$  innerhalb eines Bursts („Spitzenrate“) ergibt sich in allen Modellvarianten gemäß Gleichung (4.2) direkt aus  $L$  bzw.  $L_{max}$  und  $T_P$ .

Die meisten der nachfolgenden Untersuchungen basieren auf einem System mit zwei Klassen und den WEDD-Parametern gemäß Tabelle 6.2. Aus den Werten der Parameter folgt, dass Pakete der Klasse 1 sowohl eine zeitliche Priorisierung als auch eine Besserstellung hinsichtlich der Verwerfungswahrscheinlichkeiten erfahren. Für den Sicherheitsabstand in den Klassen 0 und 1 gelte  $\varepsilon_i = \delta_i/10$ , soweit anders lautende Aussagen fehlen. Standardmäßig soll das Verkehrsszenario  $V_A$  für beide Klassen Anwendung finden.

Die Leistungsuntersuchung erfolgt – abgesehen von einigen analytischen Randbetrachtungen – simulativ unter Verwendung der in [34] beschriebenen Simulationsumgebung. Der Verzicht auf eine analytische Untersuchung hat mehrere Gründe:

- Es existiert zwar eine Vielzahl von analytischen Untersuchungen, die ein FIFO-System mit büschelförmigem Verkehr betrachten [160, 224, 236]. Die Ergebnisse können aber nicht ohne weiteres auf Systeme übertragen werden, in denen andere Scheduling-Mechanismen eingesetzt werden. Außerdem können damit nicht die Auswirkungen von speziellen Scheduling-Parametereinstellungen und Realisierungsoptionen untersucht werden.

**Tabelle 6.2:** Parameter des Zwei-Klassen-Systems für Untersuchung mit unelastischem Verkehr

Parameter	Klasse 0	Klasse 1
$\delta_i$	40 ms	20 ms
$w_i$	10	1

- Viele der analytischen Ansätze sind entweder schwer handhabbar (z. B. im Hinblick auf numerische Stabilität) oder stellen lediglich Approximationen dar. So wird z. B. bei dem auf [130] zurückgehenden REM-Ansatz (*rate envelope multiplexing*) der Effekt der Pufferung vernachlässigt, während bei der Anwendung des Flüssigkeitsflussansatzes [8] häufig anstelle der Verlustwahrscheinlichkeit die Überschreitungswahrscheinlichkeit der Warteschlangenlänge berechnet wird, die jedoch deutlich größer sein kann [31].
- Die analytischen Lösungen für die Verlust- oder Überschreitungswahrscheinlichkeit unter büschelförmigem Verkehr gehen vielfach von einer negativ-exponentiell verteilten Burstdauer aus. Wenn die Leistungsfähigkeit unter der Voraussetzung von selbstähnlichem Verkehr bestimmt werden soll, wie er beispielsweise bei Verwendung einer Pareto-Verteilung für die Burstgröße entsteht, sind meist nur Näherungen auf der Basis des FBM-Modells (siehe Abschnitt 4.2.1.3) oder Lösungen mit stark eingeschränktem Wertebereich der Parameter verfügbar [3, 31, 52, 146, 206, 267].

Ein Nachteil einer simulativen Bewertung ist allerdings die Abhängigkeit der statistischen Aussagesicherheit von der Simulationsdauer. Die statistische Aussagesicherheit wird nachfolgend bei den Untersuchungen, die auf einer stationären Betrachtung des Systems basieren, durch die Angabe von Vertrauensintervallen untermauert, die sich auf ein Vertrauensniveau von 95% beziehen. Ausnahmen bilden Untersuchungen mit selbstähnlichem Verkehr, wo die übliche Methode zur Berechnung von Vertrauensintervallen nicht angewandt werden kann (siehe hierzu die entsprechenden Bemerkungen in Abschnitt 4.1.2), sowie einige Diagramme, in denen aus Gründen der Übersichtlichkeit auf die Darstellung der Vertrauensintervalle verzichtet wird.

### 6.1.2 Untersuchung der Grundeigenschaften von WEDD

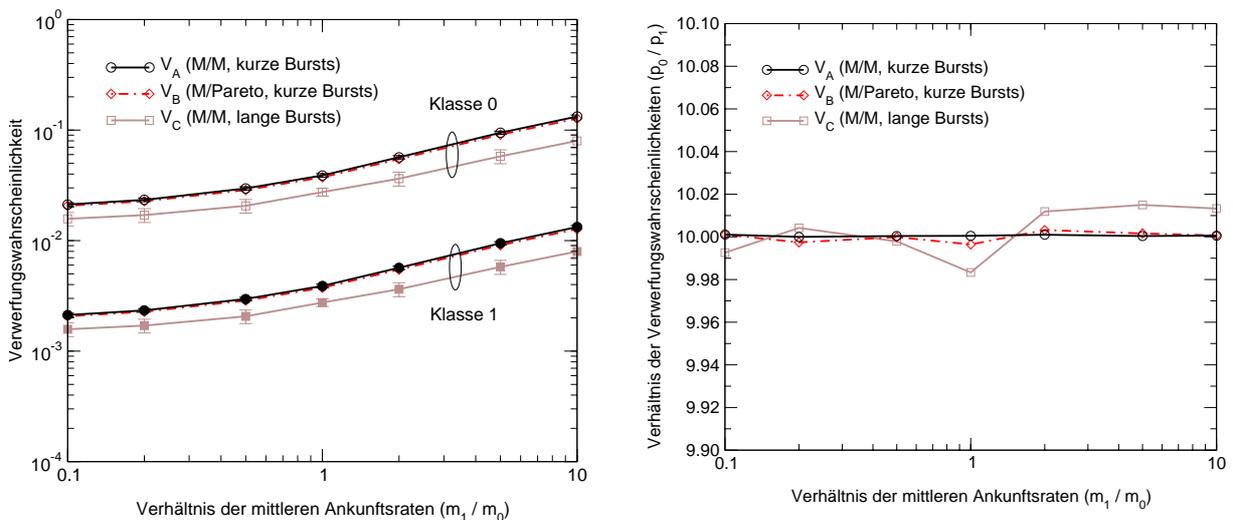
In diesem Abschnitt wird in erster Linie die Funktionstauglichkeit von WEDD unter Beweis gestellt. Dabei wird zunächst das Langzeitverhalten unter variierenden Verkehrsrandbedingungen untersucht. Anschließend erfolgt eine Betrachtung des Verhaltens in kurzen Zeitintervallen.

### 6.1.2.1 WEDD-Differenzierung bei unterschiedlichen Verkehrsrandbedingungen

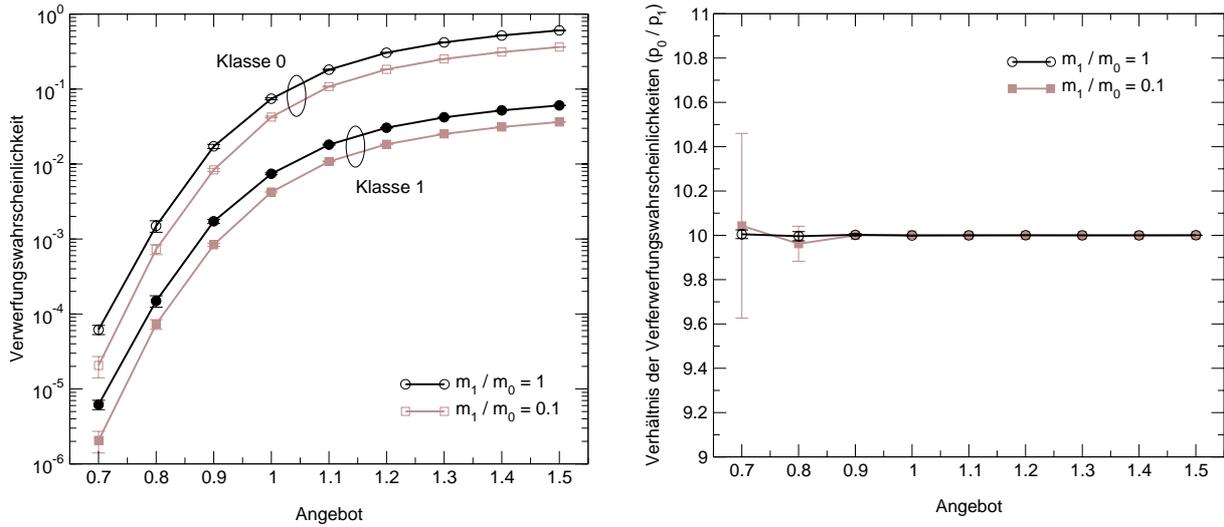
Die Untersuchungen basieren zunächst auf dem in Abschnitt 6.1.1 beschriebenen Zwei-Klassen-System mit den WEDD-Parametern gemäß Tabelle 6.2. In einer ersten Studie wird bei einem gleich bleibenden Angebot von  $A = 0.95$  das Verhältnis der Ankunftsraten  $m_1/m_0$  variiert. Dabei kommen unterschiedliche Verkehrsszenarien ( $V_A$ ,  $V_B$  und  $V_C$  entsprechend Tabelle 6.1) zur Anwendung, wobei innerhalb einer Simulation für beide Klassen die gleichen Verkehrsparameter eingesetzt werden und somit homogener Verkehr vorliegt.

In Bild 6.2 sind die sich ergebenden Verwerfungswahrscheinlichkeiten  $p_i$  in den beiden Klassen sowie deren Verhältnis ( $p_0/p_1$ ) für die unterschiedlichen Verkehrsrandbedingungen aufgetragen. Es ist deutlich zu erkennen, dass  $p_0/p_1$  mit sehr guter Genauigkeit dem erwarteten Wert  $w_0/w_1 = 10$  entspricht. Dabei ist zu beachten, dass zwar das Verhältnis der Verwerfungswahrscheinlichkeiten konstant bleibt, sich die absoluten Werte aber erwartungsgemäß mit zunehmendem Anteil der besser behandelten Klasse 1 am Gesamtangebot erhöhen. Die Parameter des Verkehrsmodells haben keinen Einfluss auf das prinzipielle Verhalten, lediglich die Absolutwerte von  $p_i$  weisen leichte Änderungen auf. Im Einzelnen kann festgestellt werden, dass die geringere Spitzenrate bei  $V_C$  gegenüber  $V_A$  trotz der größeren Burstlänge zu reduzierten Verwerfungswahrscheinlichkeiten führt. Hinsichtlich der Fähigkeit, das Verhältnis der Gewichtungparameter möglichst exakt wiederzugeben, weist das Modell mit kurzen Bursts und geometrisch verteilter Anzahl von Paketen im Burst die größte Stabilität auf.

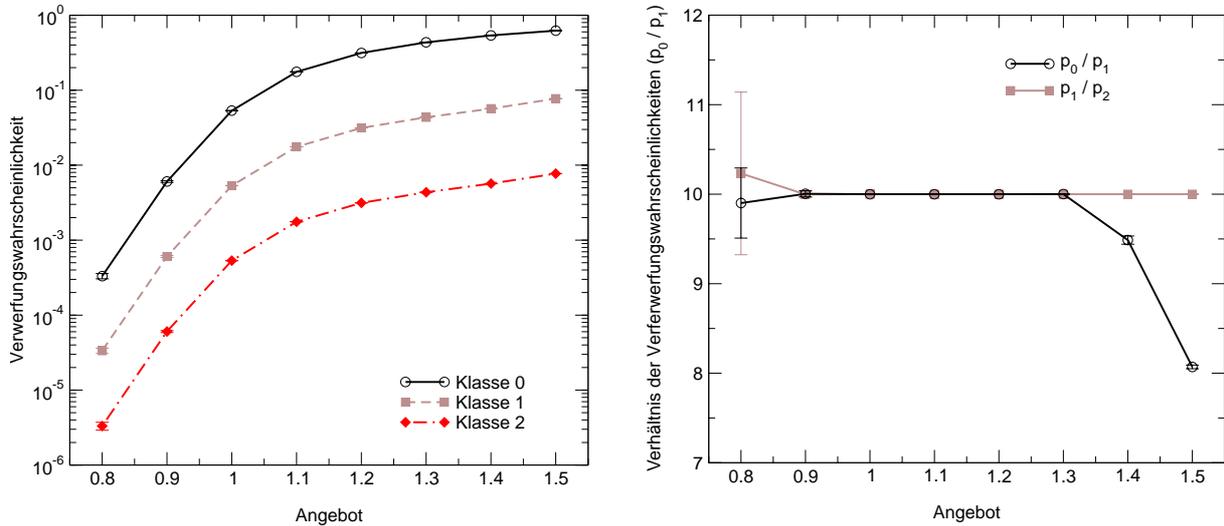
Ein ähnlich robustes Verhalten im Hinblick auf die Proportionalität der Verwerfungswahrscheinlichkeiten zeigt sich, wenn anstatt des Verhältnisses der Ankunftsraten in den einzelnen Klassen das Gesamtangebot bei festgehaltenem  $m_1/m_0$  variiert wird (Bild 6.3). Selbst bei deutlicher Überlast entspricht  $p_1/p_0$  noch mit sehr guter Genauigkeit dem über die Gewichtungparameter voreingestellten Wert 10. Den Abweichungen im Fall eines geringen Angebots



**Bild 6.2:** Verwerfungswahrscheinlichkeiten in absoluter (links) und relativer Darstellung (rechts) über der Lastverteilung mit unterschiedlichen Verkehrsszenarien ( $A = 0.95$ )



**Bild 6.3:** Verwerfungswahrscheinlichkeiten in absoluter (links) und relativer Darstellung (rechts) über dem Angebot im Zwei-Klassen-System mit Szenario  $V_A$



**Bild 6.4:** Verwerfungswahrscheinlichkeiten in absoluter (links) und relativer Darstellung (rechts) über dem Angebot im Drei-Klassen-System mit Szenarien  $V_E$  und  $V_F$

kann angesichts der großen Vertrauensintervalle wenig Bedeutung zugemessen werden. Abgesehen davon sind im Fall  $A = 0.7$  die Verwerfungswahrscheinlichkeiten in beiden Klassen so klein, dass eine Differenzierung ohnehin kaum notwendig erscheint.

Anstelle des Zwei-Klassen-Systems soll nun ein Knotenmodell mit drei Klassen und den WEDD-Parametern  $\delta_0 = \delta_1 = 40$  ms und  $\delta_2 = 20$  ms für die Maximalverzögerungen sowie  $w_0 = 100$ ,  $w_1 = 10$  und  $w_2 = 1$  für die Gewichtungparameter verwendet werden. Als Verkehrsmodelle werden an dieser Stelle  $V_E$  (entspricht Videoverkehr mittlerer Qualität) in den Klassen 0 und 1 sowie  $V_F$  (entspricht Videoverkehr hoher Qualität) in Klasse 2 eingesetzt, wobei die Ratenverhältnisse konstant gehalten werden:  $m_1 = m_2 = m_0/2$ . Die obigen Aussagen werden durch die Ergebnisse in Bild 6.4 weitgehend bestätigt. Allerdings ist festzustel-

len, dass bei einem Angebot von mehr als 1.3 deutlichere Abweichungen vom gewünschten Verhältnis der Verwerfungswahrscheinlichkeiten zu Tage treten, da offensichtlich in der Klasse mit der geringsten Priorität zu wenig Pakete verworfen werden. Dies liegt einerseits daran, dass in diesem Bereich  $p_0$  schon sehr nahe bei 1 liegt, sodass eine weitere Benachteiligung dieser Klasse kaum mehr möglich ist. Zum anderen hängt der Effekt mit dem eingestellten Sicherheitsabstand  $\varepsilon_i$  zusammen, der in Abschnitt 6.1.3.2 untersucht wird.

Insgesamt kann jedoch festgehalten werden, dass WEDD unabhängig von den zugrunde liegenden Verkehrsrandbedingungen in der Lage ist, eine proportionale Differenzierung durchzuführen. Aufgrund dieser Erkenntnis wird in den nachfolgenden Studien innerhalb von Abschnitt 6.1 in fast allen Untersuchungen das gleiche Verkehrsszenario ( $V_A$ ) verwendet.

### 6.1.2.2 Auswirkungen der klassenspezifischen WEDD-Parameter

In diesem Abschnitt wird verdeutlicht, wie sich die klassenspezifischen Parameter  $\delta_i$  und  $w_i$  des WEDD-Schedulers auf die elementaren Leistungskenngrößen auswirken. Zunächst wird der Einfluss einer Änderung der Gewichtungsfaktoren im Zwei-Klassen-System untersucht, wobei die Werte für die Maximalverzögerung (gemäß Tabelle 6.2) konstant bleiben.

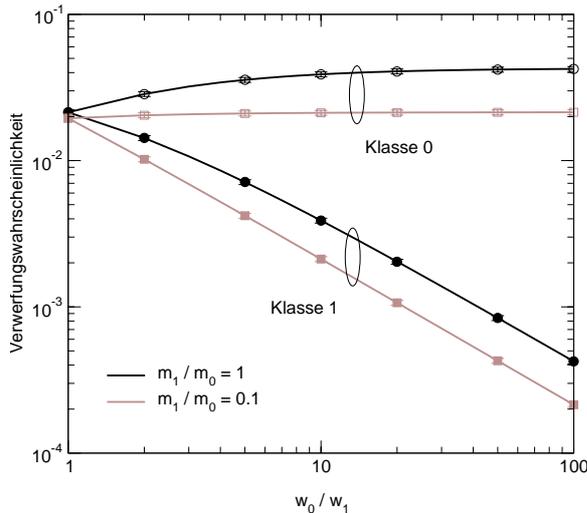
Dazu sind in Bild 6.5 für ein Gesamtangebot von 0.95 die Verwerfungswahrscheinlichkeiten über dem Verhältnis  $w_0/w_1$  der Gewichtungsparameter, deren absolute Werte ohne Bedeutung sind, aufgetragen. Mit zunehmendem Gewichtungsverhältnis ergibt sich eine starke Reduktion von  $p_1$  zu Lasten von Klasse 0. Allerdings ist die Erhöhung von  $p_0$  nur signifikant, wenn ein wesentlicher Anteil des Gesamtverkehrs der Klasse 1 zuzuordnen ist. Schon bei einem Verhältnis  $m_1/m_0 = 0.1$  zeigen sich kaum noch negative Auswirkungen für die Leistung in der niederpriorien Klasse.

Darüber hinaus kann festgestellt werden, dass die Gesamtverwerfungswahrscheinlichkeit  $p$  für unterschiedliche Werte von  $m_1/m_0$  bei einer Erhöhung von  $w_0/w_1$  nahezu konstant bleibt (in Bild 6.5 nicht dargestellt). Unter dieser Voraussetzung gilt allgemein für ein System mit  $K$  Klassen der Zusammenhang (*conservation law*):

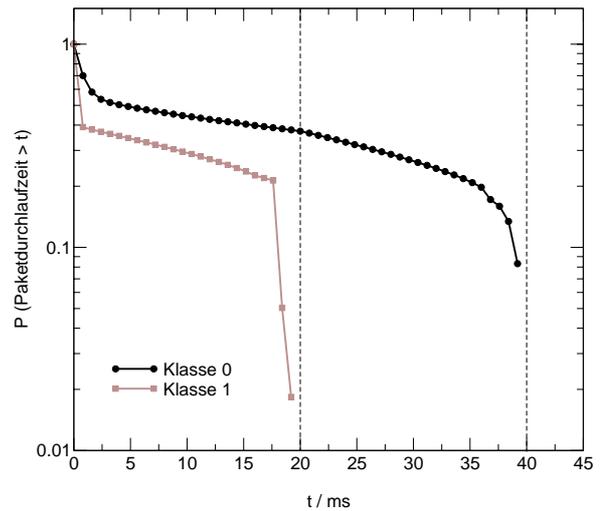
$$\sum_{i=0}^{K-1} \left( p_i \cdot \frac{m_i}{m} \right) = p = \text{const.} \quad (6.4)$$

Damit können die Verwerfungswahrscheinlichkeiten in den einzelnen Klassen aus der Gesamtverwerfungswahrscheinlichkeit berechnet werden:

$$p_i = p \cdot \frac{m}{\sum_{j=0}^{K-1} \left( m_j \cdot \frac{w_j}{w_i} \right)} \quad (6.5)$$



**Bild 6.5:** Auswirkung der Gewichte  
(Szenario  $V_A$ ,  $A = 0.95$ )



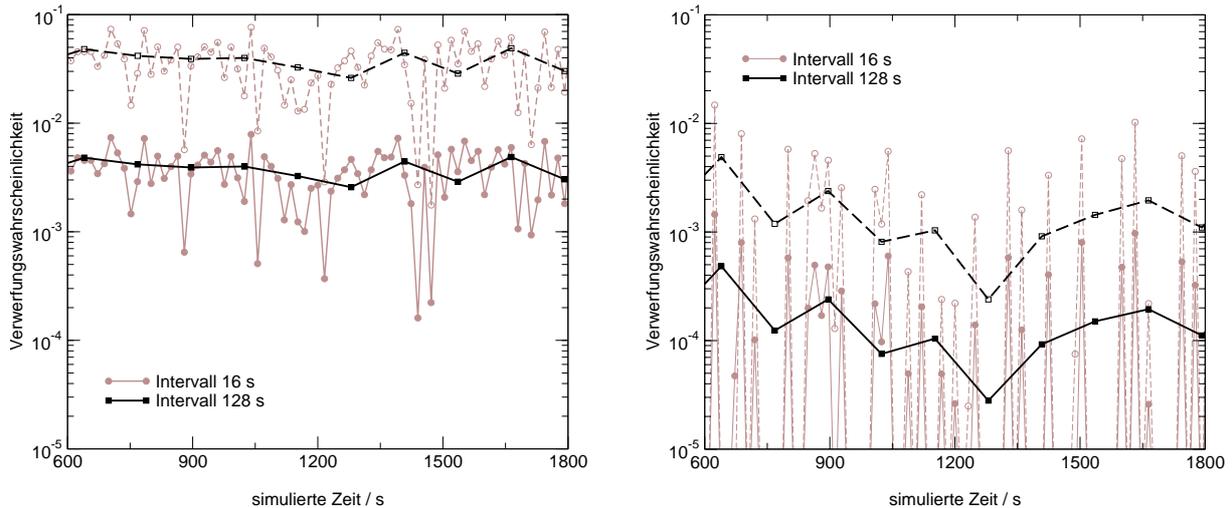
**Bild 6.6:** Durchlaufzeit-CCDF  
(Szenario  $V_A$ ,  $A = 0.95$ ,  $m_1 = m_0$ )

Dies kann auch anhand von Bild 6.5 nachgewiesen werden. Dort stammen nur die Einzelpunkte direkt aus der Simulation, während die Kurven selbst mit Hilfe von Gleichung (6.5) berechnet wurden – ausgehend von einer Gesamtverwerfungswahrscheinlichkeit, die aus der Simulation für  $w_0/w_1 = 1$  gewonnen wurde. Ein Unterschied ist optisch nicht festzustellen.

Die Auswirkung der eingestellten Maximalverzögerungen  $\delta_i$  zeigt sich am besten bei Betrachtung der komplementären Verteilungsfunktion (*complementary cumulative distribution function*, CCDF) der Paketdurchlaufzeit  $T_D$ . In Bild 6.6 ist diese für das bekannte Zwei-Klassen-System mit Parametern entsprechend Tabelle 6.2 für ein Angebot von  $A = 0.95$  bei  $m_1 = m_0$  aufgetragen. Deutlich zu erkennen ist zunächst der steile Abfall beider Kurven bei Annäherung an den jeweiligen Wert von  $\delta_i$  (senkrechte Linie). Pakete mit einer größeren Durchlaufzeit als  $\delta_i$  tragen hierzu nicht bei, da sie aufgrund der eingeschalteten LPD-Option verworfen werden. In Klasse 1 beginnt dieser Abfall abrupt an der Stelle  $t = \delta_1 - \varepsilon_1$ .

Darüber hinaus ist festzustellen, dass im Bereich  $t < \delta_1 - \varepsilon_1$  die Kurve für Klasse 1 zwar unterhalb der von Klasse 0 liegt, der Unterschied aber nur einen Faktor von ca. 2 ausmacht. Dass in diesem Bereich Klasse 1 besser gestellt ist als Klasse 0, liegt vorwiegend an der fristbasierten Abarbeitung. Diese ist auch für den geringeren Abfall der CCDF für Klasse 0 im Bereich  $t < \delta_1$  verantwortlich. Interessanterweise knickt die Kurve für Klasse 0 an der Stelle  $t = \delta_1$  leicht ab, um im Bereich  $\delta_1 \leq t \leq \delta_0 - \varepsilon_0$  die gleiche Steigung aufzuweisen wie die Kurve für Klasse 1 im Bereich  $t < \delta_1 - \varepsilon_1$ .

Auffallend ist außerdem der insgesamt eher flache Verlauf beider Kurven vor Annäherung an die Maximalverzögerung. Daraus kann auf eine geringe Wahrscheinlichkeitsdichte der Durchlaufzeit in diesem Bereich geschlossen werden. Umgekehrt bedeutet dies, dass sowohl sehr geringe Durchlaufzeiten als auch solche im Bereich  $\delta_i - \varepsilon_i \leq T_{D,i} \leq \delta_i$  sehr wahrscheinlich



**Bild 6.7:** Verwerfungswahrscheinlichkeiten in Intervallen unterschiedlicher Länge für  $m_1 = m_0$  und  $A = 0.95$  (links) bzw.  $A = 0.8$  (rechts)

sind. Letzteres wiederum zeigt, dass bei WEDD Pakete häufig so lange verzögert werden, bis eine Fristüberschreitung droht und folglich der Übergang in den Überlastmodus greift.

### 6.1.2.3 Kurzzeitverhalten

In den vorangegangenen Abschnitten wurden Untersuchungen auf der Basis einer stationären Systembetrachtung durchgeführt, die in der Lage ist, das Langzeitverhalten wiederzugeben. Darüber hinaus ist aber auch das Verhalten innerhalb kurzer Zeitabschnitte von Interesse. Vor allem stellt sich die Frage, ob auch in begrenzten Intervallen das angestrebte Verlustverhältnis erreicht wird. Um diese Frage zu beantworten, bietet sich eine Betrachtung von Traces an.

In Bild 6.7 sind für das Zwei-Klassen-System nach Abschnitt 6.1.1 mit  $m_1 = m_0$  und unterschiedlichen Angeboten die beobachteten Verwerfungshäufigkeiten innerhalb aufeinander folgender Intervalle der Länge  $\Delta t$  über einen simulierten Zeitraum von 20 min aufgezeichnet. Als Intervallbreite  $\Delta t$  wurden die Werte 16 s und 128 s gewählt. Der größere der beiden Werte liegt dabei etwa in der Größenordnung der mittleren Dauer eines Telefongesprächs, sodass Maße, die sich auf ein Intervall dieser Breite beziehen, z. B. als die von einem VoIP-Nutzer während einer Verbindung wahrgenommene Leistung interpretiert werden können. Andererseits macht die Betrachtung von Intervallen mit einer Länge von weniger als 16 s kaum Sinn, da schon bei  $\Delta t = 16$  s maximal 125 Pakete einer Verbindung pro Intervall ankommen und bei einer langfristigen Verwerfungswahrscheinlichkeit von  $10^{-2}$  im Mittel nur etwa ein Paket einer Verbindung pro Intervall verloren geht.

Bei einem Angebot von 0.95 ergeben sich signifikante Schwankungen für die kleinere Intervallbreite, während für  $\Delta t = 128$  s bereits ein deutlicher Glättungseffekt eintritt. Entscheidend ist jedoch, dass für beide Werte von  $\Delta t$  die Kurven für Klasse 0 bzw. 1 annähernd parallel ver-

laufen, d. h. das Verhältnis bleibt selbst im Fall von  $\Delta t = 16$  s weitgehend konstant entsprechend dem voreingestellten Wert  $w_0/w_1 = 10$ .

Wird das System bei einem Angebot von 0.8 betrachtet, stellt man fest, dass die relativen Schwankungen für beide Intervallbreiten deutlich zunehmen. Allerdings muss beachtet werden, dass die Verwerfungswahrscheinlichkeiten hier insgesamt um eine Größenordnung niedriger ausfallen. Dies ist auch der Grund dafür, dass innerhalb von Intervallen der Länge  $\Delta t = 16$  s kein konstantes Verhältnis der Verwerfungshäufigkeiten mehr erzielt werden kann. Dabei handelt es sich jedoch nicht um eine Unzulänglichkeit von WEDD, sondern angesichts einer langfristigen Verwerfungswahrscheinlichkeit im Bereich von  $10^{-4}$  für Klasse 1 einerseits (vgl. Bild 6.3) und im Mittel  $4 \cdot 10^4$  ankommenden Paketen pro Klasse innerhalb von 16 s andererseits ist es aufgrund der Verkehrsschwankungen nicht verwunderlich, wenn es eine beträchtliche Anzahl von Intervallen gibt, in denen überhaupt kein Paket aus Klasse 1 verworfen wird. Bei solch geringen Verwerfungswahrscheinlichkeiten ist es also prinzipiell durch die geringe Granularität der Häufigkeiten nahezu unmöglich, in kleinen Intervallen ein bestimmtes Verlustverhältnis zu gewährleisten. Wird die Intervallbreite hingegen größer gewählt (z. B. 128 s in Bild 6.7), so kann auch wieder bei  $A = 0.8$  ein paralleler Verlauf der Kurven für die Verlusthäufigkeiten beobachtet werden.

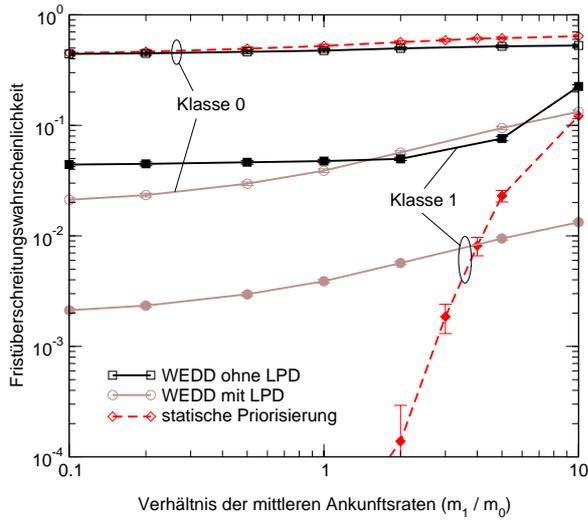
### **6.1.3 Einfluss von WEDD-Optionen und Systemparametern**

Die Umsetzung von WEDD bietet eine Reihe von Freiheitsgraden, z. B. im Hinblick auf Optionen wie das Verwerfen von verspäteten Paketen (LPD) oder den Einsatz des CSA-Mechanismus oder bzgl. der Wahl von Parametern wie Sicherheitsabstand, Puffergröße oder Abschätzungsverfahren der Überschreitungshäufigkeiten (siehe Abschnitt 5.3). Der Einfluss dieser Optionen und Parameter wird in diesem Abschnitt untersucht.

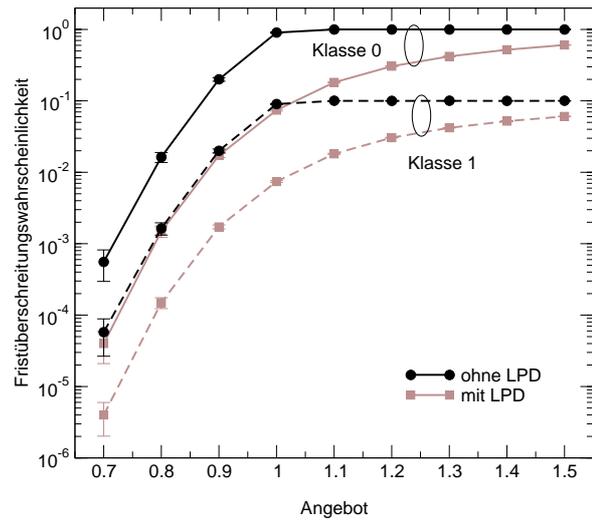
#### **6.1.3.1 Verwerfen von verspäteten Paketen**

Die bisherigen Untersuchungen verwendeten das WEDD-Verfahren in der Version, bei der verspätete Pakete, die zur Abarbeitung anstehen, verworfen werden (LPD, siehe Abschnitt 5.3.1). Hier soll nun ein Vergleich mit einem WEDD-System mit zwei Klassen stattfinden, das auf diese Option verzichtet.

Zunächst werden die Fristüberschreitungswahrscheinlichkeiten, die im Fall von WEDD mit LPD gleichbedeutend mit den Verwerfungswahrscheinlichkeiten sind, wieder für ein konstantes Gesamtangebot  $A = 0.95$  über dem Verhältnis der mittleren Datenraten  $m_1/m_0$  aufgetragen (Bild 6.8). Dabei zeigt sich zum einen, dass die Fristüberschreitungswahrscheinlichkeiten deutlich höher sind als im Fall mit LPD. Dies liegt daran, dass bei WEDD ohne LPD verspätete Pakete Bandbreiteressourcen belegen, sodass andere Pakete warten müssen und dadurch selbst ihre Frist überschreiten. In dem Fall, dass Pakete bei Verspätung im Endgerät ohnehin



**Bild 6.8:** Einfluss von LPD bei variabler Lastverteilung für  $A = 0.95$

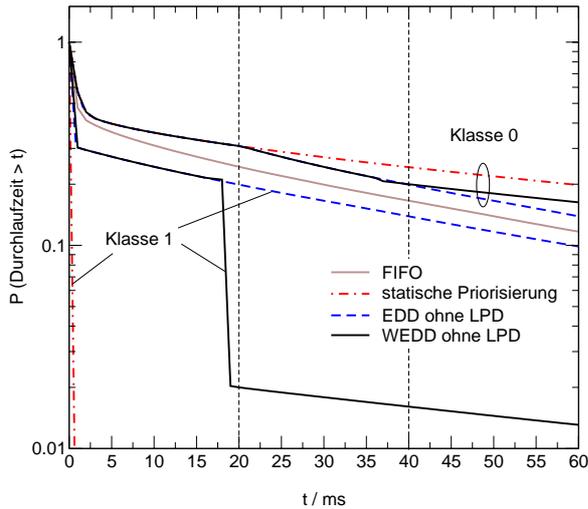


**Bild 6.9:** Einfluss von LPD in Abhängigkeit vom Angebot für  $m_1 = m_0$

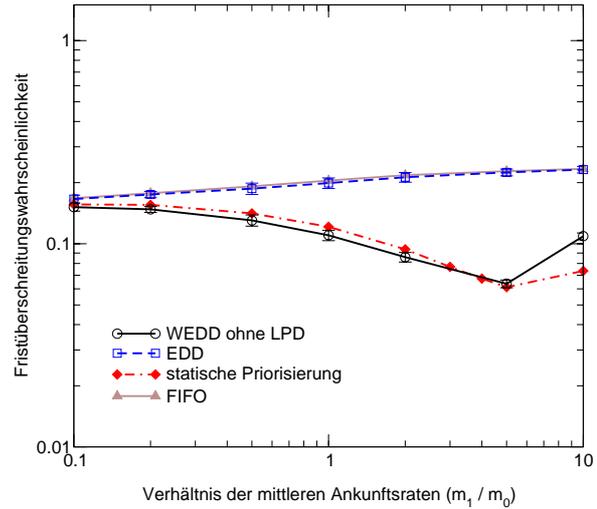
verworfen werden, bringt es also bereits bei Betrachtung eines einzelnen Knotens deutliche Vorteile, wenn solche Pakete frühzeitig verworfen werden. In einem Netzscenario mit mehreren Knoten ist sogar noch eine Verstärkung dieses Effekts zu erwarten.

Zum anderen fällt auf, dass im System ohne Verwerfung mit steigendem Anteil von Verkehr der Klasse 1 das Verhältnis der Überschreitungswahrscheinlichkeiten abnimmt. Der Grund ist darin zu sehen, dass bei einer insgesamt hohen Überschreitungswahrscheinlichkeit selbst bei der extremsten Form der Besserstellung von Klasse 1, nämlich der statischen Priorisierung, ab einem gewissen Lastpunkt das Verhältnis  $p_0/p_1$  den Wert 10 überschreitet. Somit kann auch WEDD in diesem Bereich keine stärkere Differenzierung erreichen. Allerdings wird deutlich, dass bei WEDD die Unterschreitung des gewünschten Abstands bereits früher einsetzt als bei der statischen Priorisierung. Dies hängt wesentlich mit dem Sicherheitsabstand zusammen, der in Abschnitt 6.1.3.2 untersucht wird.

Die drastische Erhöhung der Fristüberschreitungswahrscheinlichkeiten beider Klassen um etwa den Faktor 10 im Fall von WEDD ohne LPD zeigt sich auch bei der Variation des Gesamtangebots (Bild 6.9). Sie tritt insbesondere auch schon bei kleineren Werten von  $A$  auf. Da ohne LPD der WEDD-Mechanismus keine aktive Verwerfung vornimmt, kommt es für  $A \geq 1$  – unabhängig von der Puffergröße – zu Verlusten am Puffereingang (in Bild 6.9 nicht dargestellt), die ohne zusätzliche Maßnahmen beide Klassen gleichermaßen treffen (siehe Abschnitt 6.1.3.5). Von den Paketen, die im Puffer Platz finden, überschreiten in diesem Angebotsbereich alle Pakete der Klasse 0 ihre Frist, während diejenigen der Klasse 1 eine Überschreitungswahrscheinlichkeit von 10% erfahren. Somit gilt – zumindest im betrachteten Fall mit  $m_1 = m_0$  – die Aussage, dass WEDD auch ohne LPD-Option über den gesamten Angebotsbereich seiner Grundanforderung der proportionalen Differenzierung von Überschreitungswahrscheinlichkeiten gerecht wird.



**Bild 6.10:** Vergleich hinsichtlich der Durchlaufzeit-CCDF für  $A = 0.9$ ,  $m_1 = m_0$



**Bild 6.11:** Vergleich der Gesamtüberschreitungswahrscheinlichkeiten für  $A = 0.9$

Wenn auch der Verzicht auf das Verwerfen von Paketen die Leistungsfähigkeit von WEDD reduziert, so ermöglicht er dennoch einen guten Einblick in die Arbeitsweise des Schedulers und einen Vergleich mit anderen Mechanismen, die sich ebenfalls nur mit der Festlegung der Abarbeitungsreihenfolge ohne Berücksichtigung des Puffermanagements beschäftigen. In Bild 6.10 ist daher die CCDF der Paketdurchlaufzeit für unterschiedliche Verfahren bei einem Angebot von  $A = 0.9$  dargestellt. Bei WEDD fällt erneut (vgl. Bild 6.6) der steile Abfall bei  $\delta_i - \varepsilon_i$  auf, der in diesem Fall jedoch nur für Klasse 1 – also diejenige mit der geringeren Überschreitungswahrscheinlichkeit – zu beobachten ist. Deutlich zu sehen ist der Unterschied zu EDD, das gleiche Wahrscheinlichkeiten für die Überschreitung der maximalen Verzögerung in beiden Klassen liefert. Im Bereich  $t < \delta_0$  sind die Kurven für die Durchlaufzeit-CCDF in Klasse 0 für EDD bzw. WEDD identisch – inklusive des schon für WEDD mit LPD in Bild 6.6 beobachteten Knicks bei  $t = \delta_1$ . Allerdings ergibt sich nun für WEDD auch noch an der Stelle  $t = \delta_0 - \varepsilon_0$  ein leichtes Abknicken, sodass die Wahrscheinlichkeit für große Durchlaufzeiten in Klasse 0 bei WEDD ohne LPD größer ist als bei EDD. Im Vergleich zur statischen Priorisierung ergeben sich leichte Vorteile bzgl. der Durchlaufzeit in Klasse 0. Andererseits liefert die statische Priorisierung für Klasse 1 sehr geringe Durchlaufzeiten.

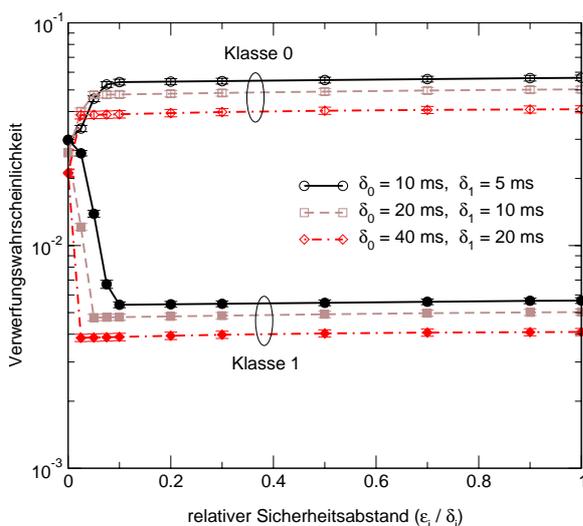
Interessant ist in diesem Zusammenhang auch die Betrachtung der Gesamtüberschreitungswahrscheinlichkeit für Pakete aus beiden Klassen (Bild 6.11). Diese ist offensichtlich für EDD und FIFO deutlich höher als für WEDD ohne LPD und statische Priorisierung. Im Vergleich zu EDD bewirkt WEDD also nicht nur eine Differenzierung hinsichtlich der klassenspezifischen Überschreitungswahrscheinlichkeiten, sondern auch eine Verbesserung im Hinblick auf das Gesamtverhalten.

### 6.1.3.2 Sicherheitsabstand

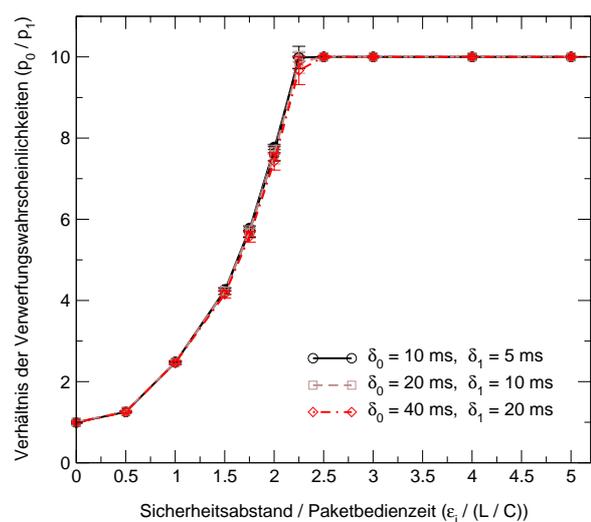
Ein Parameter, der in den bisherigen Untersuchungen unverändert geblieben ist, aber in einigen Szenarien bereits als wichtige Einflussgröße identifiziert werden konnte, ist der Sicherheitsabstand  $\epsilon_i$ , bei dessen Unterschreiten ein Eintritt in den Überlastmodus stattfindet (siehe Abschnitt 5.3.2). Für die nachfolgenden Studien wird wieder ein WEDD-Scheduler zugrunde gelegt, der die LPD-Option verwendet und dabei zwei Klassen mit  $w_1/w_0 = 10$  differenziert.

Zunächst wird der Sicherheitsabstand so variiert, dass das Verhältnis  $\epsilon_i/\delta_i$  in beiden Klassen gleich ist. Für ein Angebot von 0.95, gleiche Lastanteile in beiden Klassen und unterschiedliche Werte für  $\delta_i$  ergeben sich die in Bild 6.12 dargestellten Verwerfungswahrscheinlichkeiten. Diese sind über einen weiten Bereich nahezu unabhängig von  $\epsilon_i/\delta_i$ , wobei ein ganz leichter Anstieg in beiden Klassen zwischen  $\epsilon_i/\delta_i = 0.1$  und  $\epsilon_i/\delta_i = 1$  zu erkennen ist. Für sehr kleine Werte des Sicherheitsabstandes wird jedoch der Unterschied zwischen den Klassen immer geringer. Offensichtlich hängt dabei der Knickpunkt, ab dem das gewünschte Verhältnis von  $p_0/p_1 = 10$  näherungsweise erreicht wird, nicht von dem Verhältnis zur Maximalverzögerung ab.

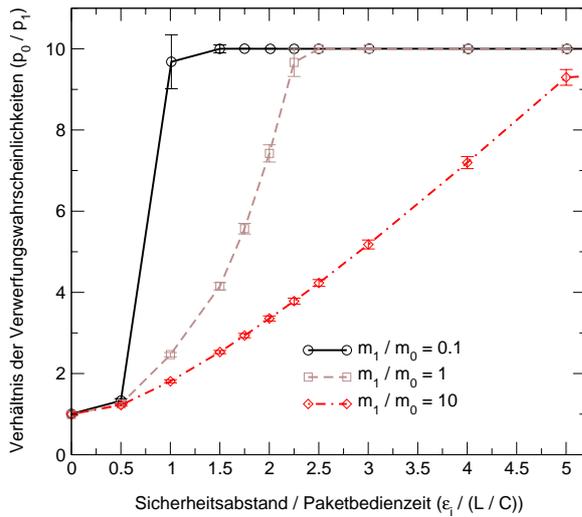
Bild 6.13 zeigt daher die Ergebnisse einer Untersuchung mit gleichem, von  $\delta_i$  unabhängigen Sicherheitsabstand in beiden Klassen. Hier ergibt sich ein fast identischer Verlauf von  $p_0/p_1$  für unterschiedliche Werte der Maximalverzögerungen. Die Untersuchung wurde dabei nicht nur für das Verkehrsszenario  $V_A$ , sondern auch für Szenario  $V_D$  durchgeführt, bei dem Pakete mit einer Länge von 1000 Byte (anstelle von  $L = 200$  Byte bei  $V_A$ ) bei gleicher Burstgröße und Spitzenrate erzeugt werden. Die Auftragung der relativen Verwerfungswahrscheinlichkeiten in Bild 6.13 erfolgt daher über einem Sicherheitsabstand, dessen Wert auf die Paketbe-



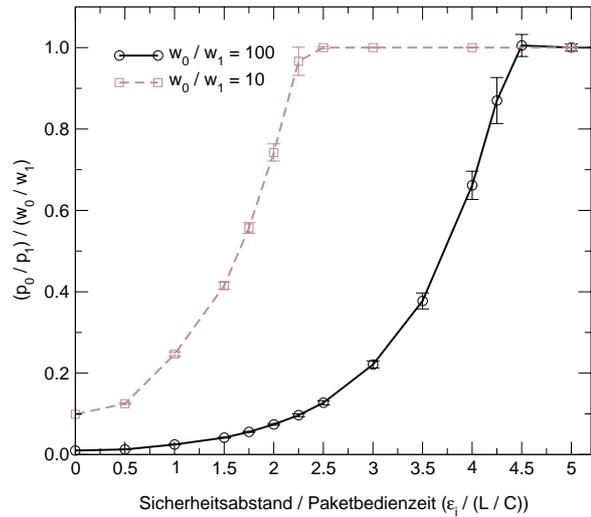
**Bild 6.12:** Absolute Verwerfungswahrscheinlichkeiten über  $\epsilon_i/\delta_i$  ( $V_A$ ,  $A = 0.95$ ,  $m_1 = m_0$ )



**Bild 6.13:** Relative Verwerfungswahrscheinlichkeiten über  $\epsilon_i/(L/C)$  (gefüllt:  $V_A$ , offen:  $V_D$ ,  $A = 0.95$ ,  $m_1 = m_0$ )



**Bild 6.14:** Einfluss der Lastverteilung ( $V_A$ ,  $A = 0.95$ ,  $\delta_0 = 40$  ms,  $\delta_1 = 20$  ms)



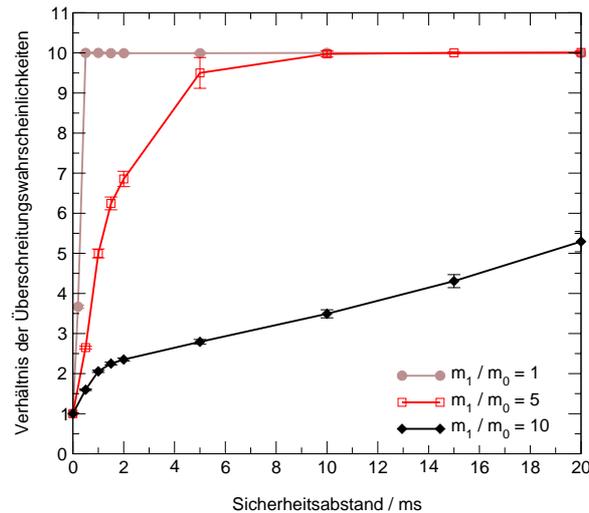
**Bild 6.15:** Einfluss der Gewichtungsfaktoren ( $V_A$ ,  $A = 0.95$ ,  $\delta_0 = 40$  ms,  $\delta_1 = 20$  ms)

diendauer  $L/C$  normiert wurde. Diese hat offensichtlich entscheidenden Einfluss auf den o. g. Knickpunkt, der in beiden Szenarien bei  $\epsilon_i \approx 2.5 \cdot L/C$  auftritt.

Darüber hinaus sind Abhängigkeiten des Verlaufs von der Lastverteilung (Bild 6.14) und dem Verhältnis der Gewichtungsfaktoren (Bild 6.15) festzustellen. Sowohl ein zunehmender Lastanteil in Klasse 1 als auch eine Erhöhung von  $w_0/w_1$  haben eine Vergrößerung des Bereiches zur Folge, innerhalb dessen das gewünschte Verhältnis der Verwerfungswahrscheinlichkeiten nicht erreicht wird. Absolut gesehen entspricht der Wert  $5 \cdot L/C$ , bis zu dem sich der Bereich in den beiden Bildern ausdehnt bei dem hier verwendeten Verkehrsszenario  $V_A$  lediglich einem Wert von 0.8 ms für  $\epsilon_i$ . Allerdings legen die genannten Abhängigkeiten und die daraus erwachsende Gefahr, dass sich diese Grenze weiter verschiebt, den Gedanken nahe, einen Sicherheitsabstand  $\epsilon_i = \delta_i$  zu wählen und dafür die leichte Erhöhung der Verwerfungswahrscheinlichkeiten (vgl. Bild 6.12) in Kauf zu nehmen. Das würde bedeuten, dass der Scheduler immer im Überlastmodus arbeiten würde (vgl. Abschnitt 5.3.2). Diese Option ist besonders interessant, wenn WEDD ohne LPD betrieben wird. In diesem Fall kann das in Abschnitt 6.1.3.1 beobachtete Problem, dass bei hohem Lastanteil der Grad der Differenzierung abnimmt (vgl. Bild 6.8), zumindest teilweise behoben werden, wenn für den Sicherheitsabstand ein Wert im Bereich der maximalen Verzögerungen gewählt wird (Bild 6.16).

### 6.1.3.3 Realisierungsformen für die Schätzung von Überschreitungshäufigkeiten

In Abschnitt 5.3.3 wurden verschiedene Möglichkeiten vorgestellt, um die von WEDD benötigte Abschätzung der Überschreitungshäufigkeiten in den einzelnen Klassen zu realisieren. Bisher wurde nur die einfachste Lösung berücksichtigt, bei der zwei als unbegrenzt angenommene Zähler  $N_i$  und  $M_i$  für die insgesamt angekommenen und die verworfenen Pakete verwendet werden. Nun sollen die Auswirkungen der Nutzung des autoregressiven Verfahrens

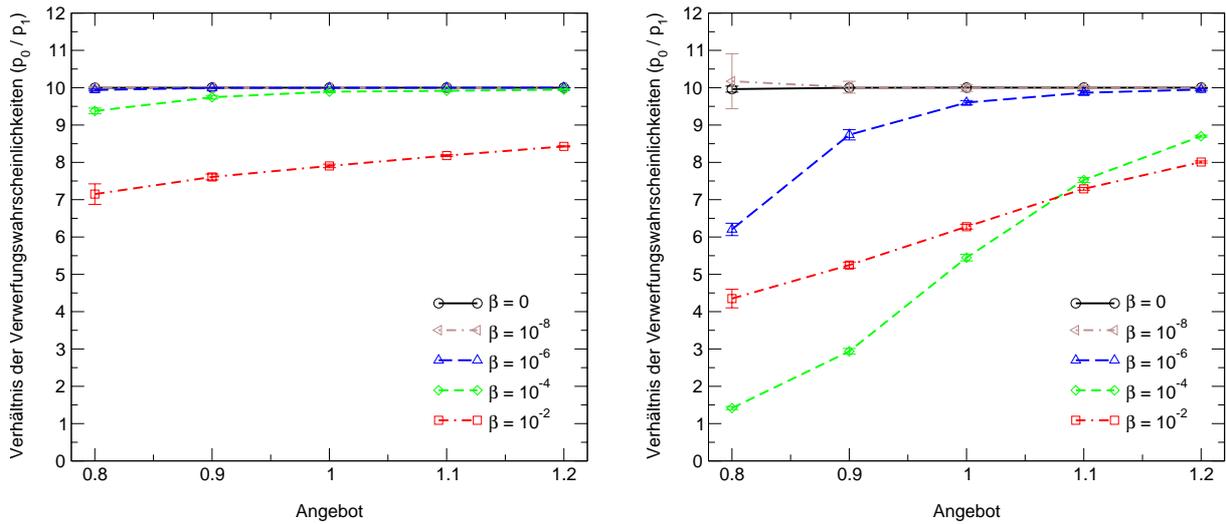


**Bild 6.16:** Auswirkung des Sicherheitsabstands bei WEDD ohne LPD  
 ( $V_A, A = 0.95, \delta_0 = 40 \text{ ms}, \delta_1 = 20 \text{ ms}$ )

gemäß Gleichung (5.10) zur Häufigkeitsbestimmung sowie des besonders unter Implementierungsaspekten (siehe Abschnitt 5.3.5) attraktiven schwellenbasierten Verfahrens zur Zählerbegrenzung untersucht werden.

Wird das autoregressive Verfahren eingesetzt, gilt es vor allem, den Einfluss des Parameters  $\beta$  zu bestimmen, wobei  $\beta = 0$  dem bisher angenommenen Schema entspricht und gleichzeitig den Idealfall repräsentiert. Dazu ist in Bild 6.17 das Verhältnis  $p_0/p_1$  der Verwerfungswahrscheinlichkeiten für das bekannte Zwei-Klassen-System mit Parameterwerten gemäß Tabelle 6.2 und  $V_A$  als Verkehrsszenario dargestellt. Im Fall gleicher Lastanteile ( $m_1 = m_0$ ) zeigt sich, dass bereits ein geringer Wert von  $\beta = 10^{-6}$  zu Abweichungen gegenüber dem Idealfall  $p_0/p_1 = 10$  führt. Bis zu einem Wert von  $\beta = 10^{-4}$  können diese Abweichungen, die für ein geringes Angebot am stärksten sind, noch als tolerierbar angesehen werden, während sie bei  $\beta = 10^{-2}$  nicht mehr akzeptabel sind. Die Effekte verstärken sich, wenn eine ungleiche Lastverteilung ( $m_1/m_0 = 0.1$ ) betrachtet wird. Hier treten bereits für  $\beta = 10^{-6}$  signifikante Abweichungen vom gewünschten Verhältnis der Verwerfungswahrscheinlichkeiten auf. Für  $\beta = 10^{-2}$  geht in diesem Szenario bei geringem Angebot die Differenzierung sogar komplett verloren. Insgesamt lässt sich folgern, dass die autoregressive Häufigkeitsabschätzung gemäß Gleichung (5.10) nur für sehr kleine Parameterwerte (z. B.  $\beta = 10^{-8}$ ) annehmbar ist.

Als Alternative zu dem auch aus Implementierungssicht problematischen autoregressiven Schätzungsmechanismus wurde in Abschnitt 5.3.3 ein Verfahren präsentiert, das eine Reduktion der Zähler  $N_i$  und  $M_i$  vornimmt, wenn  $N_i$  einen Schwellwert  $\hat{N}$  erreicht. Gemäß der Überlegungen in Abschnitt 5.3.5 wird in den nachfolgenden Untersuchungen von einer ganzzahligen Darstellung von  $N_i$  und  $M_i$  mit  $\hat{z}$  bit ausgegangen, wobei dann  $\hat{N}$  dem größten mit  $\hat{z}$  darstellbaren Wert, also  $2^{\hat{z}} - 1$ , entspricht. Beim Erreichen des Schwellwerts erfolgt hier ein Rechtsschieben beider Zählerinhalte um ein bit, was einer Division durch zwei entspricht.



**Bild 6.17:** Relative Verwerfungswahrscheinlichkeit bei Autoregressionsverfahren für  $m_1/m_0 = 1$  (links) bzw.  $m_1/m_0 = 0.1$  (rechts)

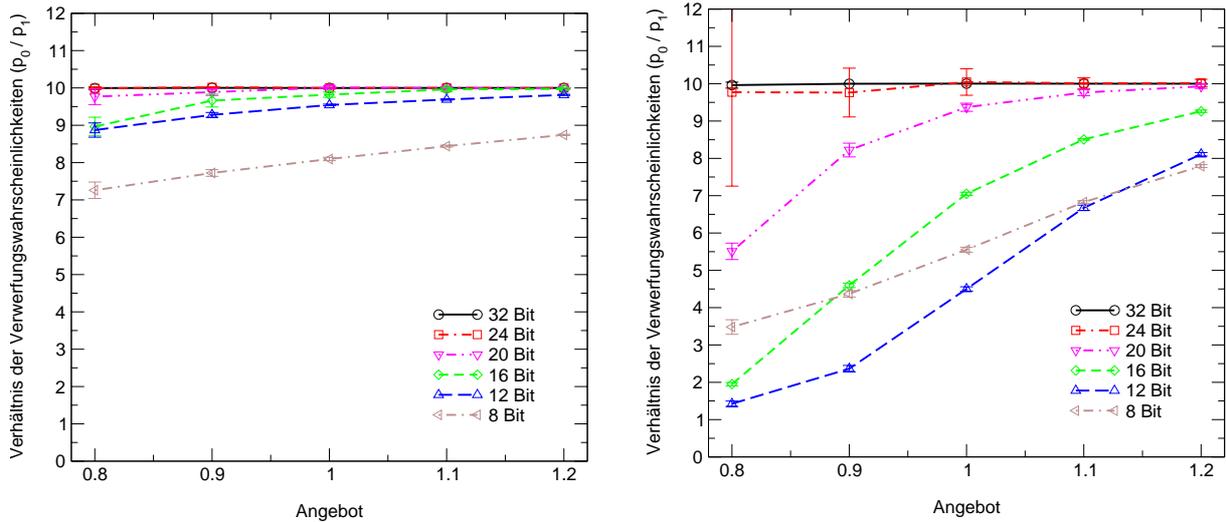
Bevor die Ergebnisse von Studien präsentiert werden, müssen zunächst einige hier auftretende Schwierigkeiten im Zusammenhang mit der simulativen Untersuchungsmethodik genannt werden. Eine Zählerreduktion in Klasse  $i$  findet bei dem schwellwertbasierten Verfahren im Mittel nach

$$\Delta N_i = \hat{N} \cdot \frac{m}{m_i} = (2^{\hat{z}} - 1) \cdot \frac{m}{m_i} \quad (6.6)$$

im System angekommenen Paketen der Klasse  $i$  statt. Dies bedeutet, dass im Fall einer Zählerbreite von 32 bit selbst bei einer Gesamtzahl von  $10^9$  simulierten Paketen<sup>1</sup> während eines Simulationslaufs keine einzige Aktualisierung stattfindet. Auch bei  $\hat{z} = 24$  nimmt  $\Delta N_1$  im Fall von  $m_1/m_0 = 0.1$  noch einen Wert von mehr als  $10^8$  an, sodass bei  $10^9$  simulierten Paketen, die sich auf zehn Teiltests aufteilen, im Mittel weniger als eine Aktualisierung innerhalb eines Teiltests auftritt. Für größere Werte von  $\hat{z}$ , die aber durchaus als realistisch anzusehen sind, ist also unter den gegebenen Randbedingungen keine stationäre Simulation möglich. Hinzu kommen die bereits in den bisherigen Untersuchungen teilweise beobachteten Probleme mit der statistischen Aussagesicherheit bei geringem Angebot.

In Bild 6.18 ist der Einfluss der Zählerbreite auf das Verhältnis der Verwerfungswahrscheinlichkeiten für unterschiedliche Lastanteile beider Klassen zu sehen, wobei die Ergebnisse für  $\hat{z} = 32$  und  $\hat{z} = 24$  im Licht der obigen Aussagen zur Stationarität betrachtet werden müssen. Für den Fall  $m_1 = m_0$  machen sich nennenswerte Abweichungen vom Idealverhalten nur bemerkbar, wenn 16 bit oder weniger für die Zähler zur Verfügung stehen. Ein geringerer

<sup>1</sup> Dies kann unter den gegebenen Randbedingungen (Leistungsfähigkeit des Simulationsrechners) als gerade noch realisierbarer Wert angesehen werden. In den meisten Studien in diesem Abschnitts werden  $10^8$  Pakete simuliert.



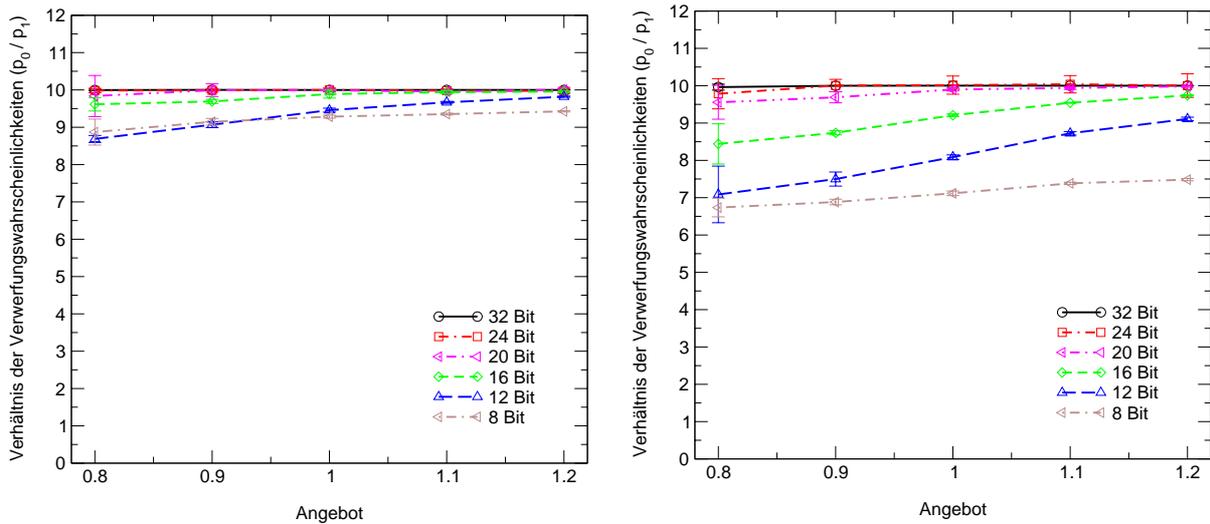
**Bild 6.18:** Relative Verwerfungswahrscheinlichkeit bei schwellwertbasierter Zählerreduktion für  $m_1/m_0 = 1$  (links) bzw.  $m_1/m_0 = 0.1$  (rechts)

Anteil von Verkehr der Klasse 1 bedeutet wie beim autoregressiven Verfahren eine Verschlechterung der Ergebnisse. Bemerkenswert ist dabei, dass für sehr kleine Werte von  $\hat{z}$  die Ergebnisse kaum noch vorhersehbar sind, sodass z. B.  $\hat{z} = 12$  über den gesamten Angebotsbereich hinweg eine größere Abweichung vom Idealverhalten liefert als  $\hat{z} = 8$ . Für eine Zählerbreite von  $\hat{z} = 20$  bit sind bei einem Angebot unterhalb von 1 die Abweichungen immer noch signifikant. Erst bei  $\hat{z} = 24$  ist die Verringerung des Verlustverhältnisses vernachlässigbar, wobei in diesem Fall die geringe statistische Aussagesicherheit zu beachten ist.

Im Zusammenhang mit der schwellenbasierten Zählerreduktion wurde in Abschnitt 5.3.3 die Möglichkeit angesprochen, bei Erreichen des Schwellwerts in einer Klasse zwar alle entsprechenden Zähler in anderen Klassen zu reduzieren, dafür aber nur den Zähler modifizieren, für den die Schwelle erreicht wurde (also entweder alle  $N_i$  oder alle  $M_i$ ). Dies erfordert, dass die Einheiten, die für die Abschätzung in den einzelnen Klassen zuständig sind, miteinander kooperieren, hat aber den Vorteil, dass auch im Fall  $M_i \ll N_i$  für  $M_i$  die volle Zählerbreite genutzt werden kann und somit die Auflösung der Häufigkeitsabschätzung steigt.

Bei der Betrachtung der Ergebnisse in Bild 6.19 muss zunächst wieder die Problematik der Stationarität der Untersuchung angesprochen werden, die sich gegenüber dem obigen Fall mit Gleichung (6.6) insoweit verschärft, als dass jetzt die Reduktion der  $M_i$ -Zähler erst stattfindet, wenn in einer Klasse  $M_i$  den Schwellwert  $\hat{N}$  erreicht hat. Damit ist der mittlere Abstand zwischen zwei Reduktionsvorgängen dieser Zähler durch

$$\Delta M \approx \hat{N} \cdot \frac{m}{\max_i(p_i \cdot m_i)} = (2^{\hat{z}} - 1) \cdot \frac{m}{\max_i(p_i \cdot m_i)} \quad (6.7)$$



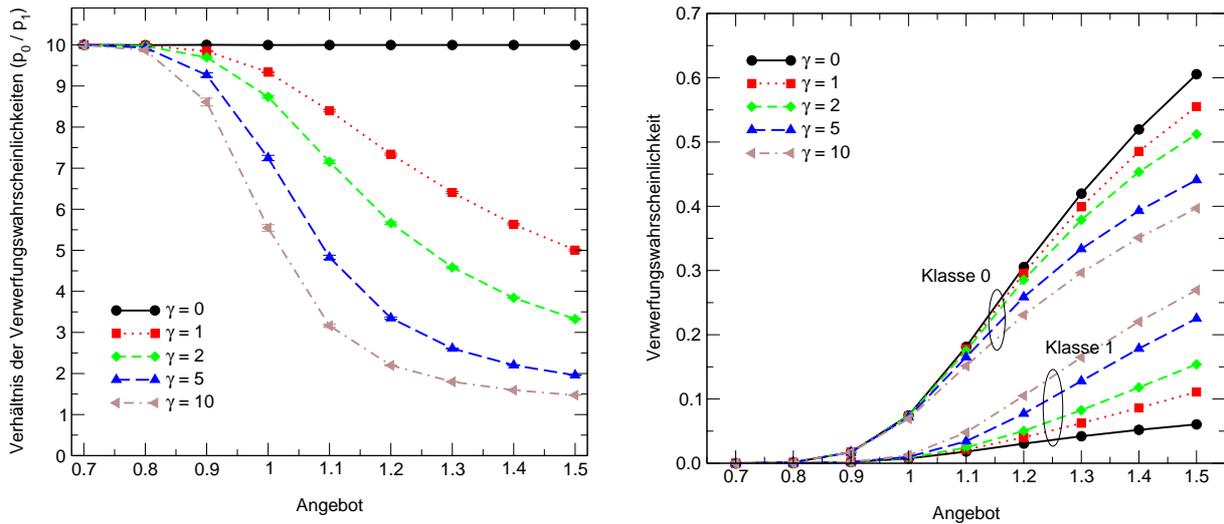
**Bild 6.19:** Relative Verwerfungswahrscheinlichkeit bei kooperativem Schwellwertverfahren für  $m_1/m_0 = 1$  (links) bzw.  $m_1/m_0 = 0.1$  (rechts)

gegeben und damit bei geringen Verwerfungswahrscheinlichkeiten  $p_i$  noch größer als der Wert, der sich aus Gleichung (6.6) ergibt. Dennoch wird aus Bild 6.19 auch bei einer entsprechend vorsichtigen Beurteilung der Resultate für geringes Angebot klar ersichtlich, dass dieses Verfahren eine deutliche Verbesserung bringt. Vor allem im Fall  $m_1/m_0 = 0.1$  werden die großen Abweichungen vom Idealwert  $p_0/p_1 = 10$  bei geringem Angebot und kleinen Werten von  $\hat{z}$  vermieden. Für  $\hat{z} = 24$  und  $\hat{z} = 20$  ergeben sich nahezu keine Abweichungen mehr; selbst eine Zählerbreite von 16 bit kann noch mit gewissen Einschränkungen akzeptiert werden.

#### 6.1.3.4 Auswirkungen der CSA-Erweiterung

Wenn der Verkehrsanteil von Klassen mit geringerem Verwerfungsgewicht zunimmt, kommt es im Standard-WEDD-Verfahren zu einem „Aushungern“ der Klassen mit geringerer Priorität. Um dies zu vermeiden, kann die in Abschnitt 5.3.4 vorgestellte CSA-Erweiterung von WEDD genutzt werden. Über den Parameter  $\gamma$  kann dabei Einfluss darauf genommen werden, wie sehr in den genannten Überlastsituationen von dem über  $w_i$  eingestellten Verhältnis der Verwerfungswahrscheinlichkeiten abgewichen wird.

Nachfolgend wird wieder von dem Zwei-Klassen-System nach Abschnitt 6.1.1 und Verkehr mit Parameterkombination  $V_A$  ausgegangen. Für ein variables Gesamtangebot bei gleich bleibendem Ratenverhältnis  $m_1/m_0 = 1$  sind in Bild 6.20 die Verwerfungswahrscheinlichkeiten sowie deren Verhältnis für unterschiedliche Werte von  $\gamma$  aufgetragen. Erwartungsgemäß nimmt der Grad der Differenzierung zwischen den beiden Klassen, also  $p_0/p_1$ , mit zunehmendem Angebot ab. Die Abnahme ist bei größeren Werten von  $\gamma$  stärker, setzt aber bei allen Werten erst im Bereich von  $A = 0.9$  ein. Der Blick auf die absoluten Verwerfungswahrscheinlichkeiten, die hier ausnahmsweise in linearem Maßstab dargestellt sind, zeigt, dass sich  $p_0$



**Bild 6.20:** Auswirkungen von  $\gamma$  auf die relative (links) bzw. absolute (rechts) Verwerfungswahrscheinlichkeit für  $m_1 = m_0$

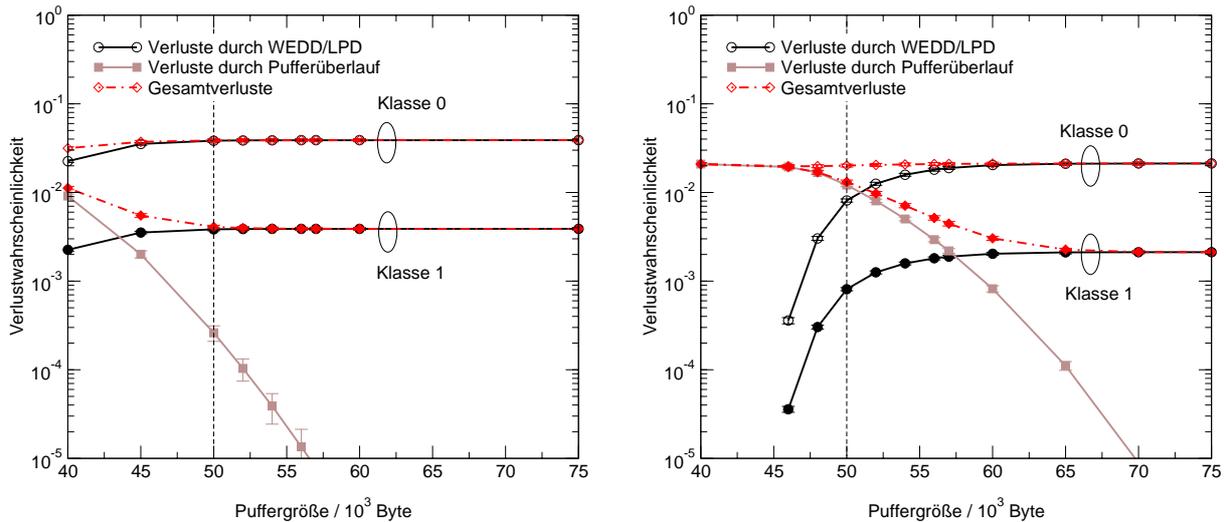
etwa um den gleichen Absolutwert verringert, um den sich andererseits  $p_1$  erhöht, d. h. der flachere Anstieg von  $p_0$  geht stark zu Lasten von Klasse 1.

Der Nutzen der CSA-Erweiterung aus Anwendungssicht wird aus dieser Untersuchung nicht unmittelbar ersichtlich und hängt insbesondere von der Auswirkung der Verlustwahrscheinlichkeit auf Dienstgütemaße der höheren Modellierungsebenen ab. Damit sei an dieser Stelle auf die Untersuchung von CSA in Abschnitt 6.2.2.2 im Zusammenhang mit TCP-Verkehr verwiesen.

### 6.1.3.5 Einfluss der Puffergröße

Eine charakteristische Eigenschaft von WEDD bei Verwendung der LPD-Option besteht darin, dass ein Verwerfen von Paketen immer am Pufferausgang erfolgt. Angesichts einer endlichen Puffergröße sind zusätzlich aber auch stets Paketverluste am Eingang möglich, die durch einen Pufferüberlauf verursacht werden. In den bisherigen Untersuchungen wurde die Puffergröße so groß gewählt, dass es im Rahmen der Simulationen nie zu einem Pufferüberlauf kam. Nun soll der Einfluss der Puffergröße betrachtet werden, der sich dann bemerkbar macht, wenn die Überlaufwahrscheinlichkeit mindestens in der gleichen Größenordnung wie die Verwerfungswahrscheinlichkeit am Pufferausgang liegt. Der Nachteil ist hierbei, dass die Überlaufwahrscheinlichkeit bei homogenem Verkehr in allen Klassen nahezu identisch ist, sodass im geschilderten Fall das Verhältnis der Gesamtverlustwahrscheinlichkeiten in einzelnen Klassen nicht mehr dem Verhältnis folgt, das durch die WEDD-Gewichtungparameter vorgegeben ist.

Einen Anhaltspunkt bei der Pufferdimensionierung gibt der Wert der Puffergröße, bei dem die maximale Verzögerung im Fall einer FIFO-Abarbeitung der größten Maximalverzögerung in irgendeiner Klasse bei WEDD-Scheduling entspricht:



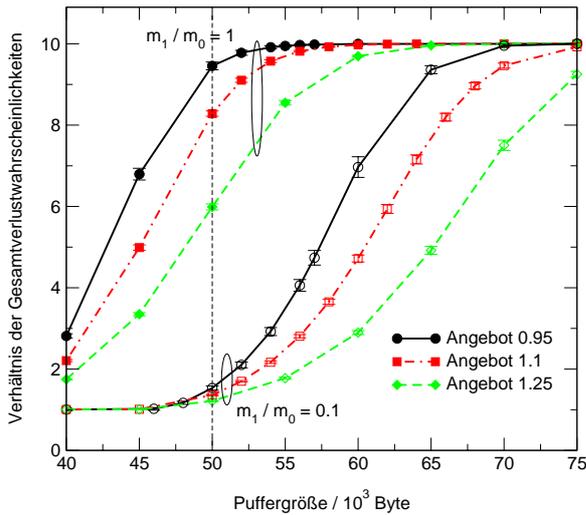
**Bild 6.21:** Wahrscheinlichkeiten für Verlust durch Verwerfung oder Pufferüberlauf bei  $A = 0.95$  und  $m_1/m_0 = 1$  (links) bzw.  $m_1/m_0 = 0.1$  (rechts)

$$S_{min} = \max_i(\delta_i) \cdot C \tag{6.8}$$

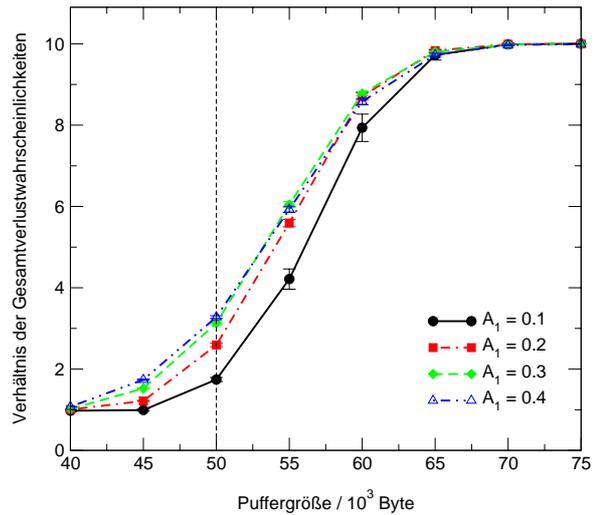
Dieser Wert kann als Mindestpuffergröße für ein funktionierendes WEDD-Scheduling angesehen werden. Im Fall einer Linkrate von  $C = 10$  Mbit/s und eines WEDD-Schedulers mit zwei Klassen und den Parametern gemäß Tabelle 6.2 ergibt sich  $S_{min}$  aus dem Produkt von  $\delta_0$  und  $C$  zu  $50 \cdot 10^3$  Byte.

Die Ergebnisse der Untersuchung mit variierender Puffergröße bei einem Angebot von  $A = 0.95$  zeigen, dass im Fall gleicher Lastanteile in beiden Klassen die am Eingang auftretenden Verluste bei einer Puffergröße von  $S = S_{min}$  bereits um eine Größenordnung unter der Verwerfungswahrscheinlichkeit der von WEDD höher priorisierten Klasse 1 liegen (Bild 6.21) und damit für  $S \geq S_{min}$  praktisch ohne Auswirkung bleiben. Anders sieht es dagegen aus, wenn der Anteil an Verkehr der Klasse 0, d. h. der Klasse mit höherer Maximalverzögerung, überwiegt. In diesem Fall liegt die Wahrscheinlichkeit eines Paketverlusts am Puffereingang bei  $S = S_{min}$  im Bereich der WEDD/LPD-Verwerfungswahrscheinlichkeit von Klasse 0 und damit deutlich über der von Klasse 1. Dies bedeutet, dass bei  $S = S_{min}$  die Gesamtverlustwahrscheinlichkeit noch von der für beide Klassen gleichen Verlustwahrscheinlichkeit durch Pufferüberlauf dominiert wird, sodass das gewünschte Verhältnis von 10 nicht erreicht wird. Erst bei einer Puffergröße von  $65 \cdot 10^3$  Byte ist die Verlustwahrscheinlichkeit am Eingang so klein, dass sie deutlich unter der WEDD-Verwerfungswahrscheinlichkeit für Klasse 1 liegt und damit keinen Einfluss mehr auf die Gesamtverlustwahrscheinlichkeiten und deren Verhältnis hat.

Der Einfluss der Puffergröße hängt außer von der Lastverteilung auch vom Gesamtangebot ab, wie in Bild 6.22 zu erkennen ist. Ein hohes Angebot in Kombination mit einem hohen Ange-



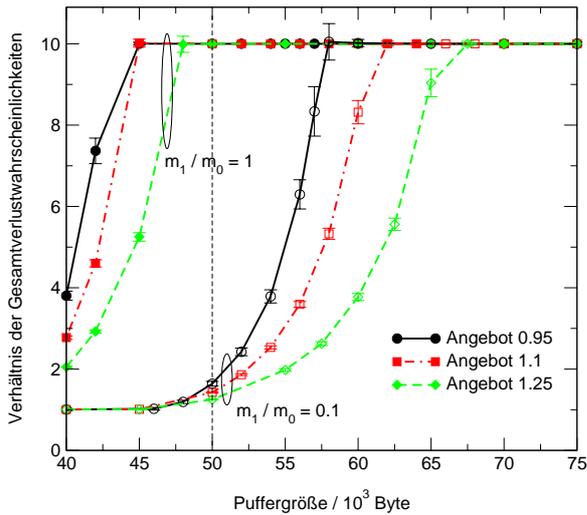
**Bild 6.22:** Verhältnis der Gesamtverlustwahrscheinlichkeiten für verschiedene Lastsituationen



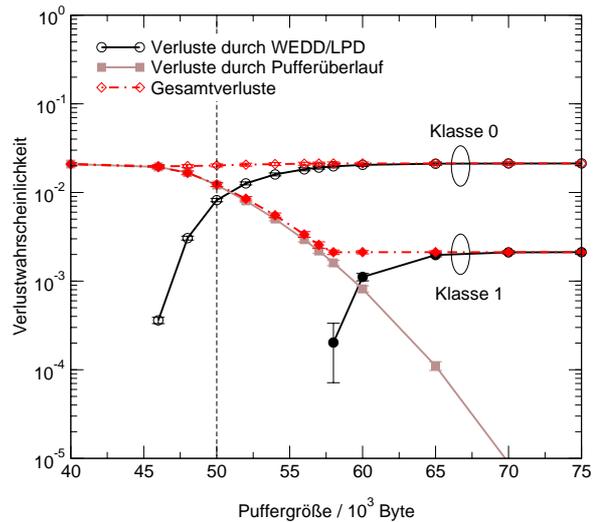
**Bild 6.23:** Verhältnis der Gesamtverlustwahrscheinlichkeiten für verschiedene Werte von  $A_1$  bei  $A_0 = 0.8 = \text{const.}$

botsanteil von Klasse 0 bewirkt, dass deutlich mehr Pufferkapazität als  $S_{min}$  erforderlich ist, damit das Verhältnis der Gesamtverlustwahrscheinlichkeiten den durch  $w_0/w_1$  eingestellten Wert von 10 erreicht. Es liegt somit die Schlussfolgerung nahe, dass es in erster Linie auf das Angebot in der Klasse mit der größten maximalen Verzögerung ankommt. Dies wird durch die in Bild 6.23 dargestellten Ergebnisse bestätigt. In der zugrunde liegenden Untersuchung wurde nur  $A_1$  variiert, während das Angebot in Klasse 0 festgehalten wurde. Ab dem Wert  $A_1 = 0.2$ , was einem Gesamtangebot von 1 entspricht, sind kaum mehr Veränderungen im Verlauf zu erkennen. Vor allem bleibt der für das Erreichen des gewünschten Verlustverhältnisses erforderliche Wert der Puffergröße bei Variation von  $A_1$  unverändert.

Eine Möglichkeit, bei gegebener Puffergröße eine Verbesserung herbeizuführen, ist die bereits in Abschnitt 5.3.3 angedeutete Option, die Verluste durch Pufferüberlauf in den Zählervariablen  $M_i$ , die für die Abschätzung der Verwerfungshäufigkeiten eingesetzt werden, mit zu berücksichtigen. Dies bedeutet anschaulich, dass Verluste am Puffereingang in der Klasse mit geringerem Verwerfungsgewicht  $w_i$  dadurch ausgeglichen werden, dass die Priorität beim Scheduling im Überlastmodus zusätzlich erhöht wird. Die Ergebnisse in Bild 6.24 machen deutlich, dass diese Option tatsächlich in der Lage ist, den für das Erreichen des eingestellten Verhältnisses der Gesamtverlustwahrscheinlichkeiten notwendigen Speicherplatz signifikant zu reduzieren. Bei großem Angebot in Klasse 0 und einer Puffergröße, die den Wert von  $S_{min}$  nur wenig übersteigt, kann aber auch dieses Verfahren das gewünschte Verhältnis nicht herstellen. Dies liegt daran, dass in diesem Fall selbst eine statische Priorisierung von Verkehr der Klasse 1 und damit ein völliger Verzicht auf das Verwerfen von Paketen dieser Klasse am Pufferausgang (Bild 6.25) nicht in der Lage ist, die Verluste am Puffereingang auszugleichen.



**Bild 6.24:** Verhältnis der Gesamtverlustwahrscheinlichkeiten bei Berücksichtigung von Pufferüberläufen in der Abschätzung



**Bild 6.25:** Verlustwahrscheinlichkeiten bei erweiterter Verlusthäufigkeitsabschätzung für  $A = 0.95$  und  $m_1/m_0 = 0.1$

Insgesamt zeigt sich, dass eine gewisse Überdimensionierung des Puffers erforderlich ist, um eine korrekte Funktion von WEDD zu gewährleisten. Allerdings orientiert sich diese bei sinnvollen Werten der maximalen Verzögerung an einem eher geringen Wert  $S_{min}$ . Eine Puffergröße von  $1.5 S_{min}$  ist – zumindest in den hier untersuchten Szenarien – bis in einen hohen Bereich des Angebots ausreichend. Eine weitere Verbesserung ist nur möglich, wenn zusätzlich ein differenzierender Puffermanagement-Mechanismus am Eingang eingesetzt wird. Dabei genügt ein einfacher Mechanismus wie PBS, da ein proportionales Verlustverhältnis durch die Berücksichtigung aller Verluste bei der Verlusthäufigkeitsabschätzung durch WEDD/LPD erreicht werden kann. Sinnvoller erscheint es jedoch, wenn bei einer gegebenen, im Gegensatz zu den WEDD-Parametern in der Praxis nicht einstellbaren Puffergröße, der zulässige Wertebereich der maximalen Verzögerung eingeschränkt wird.

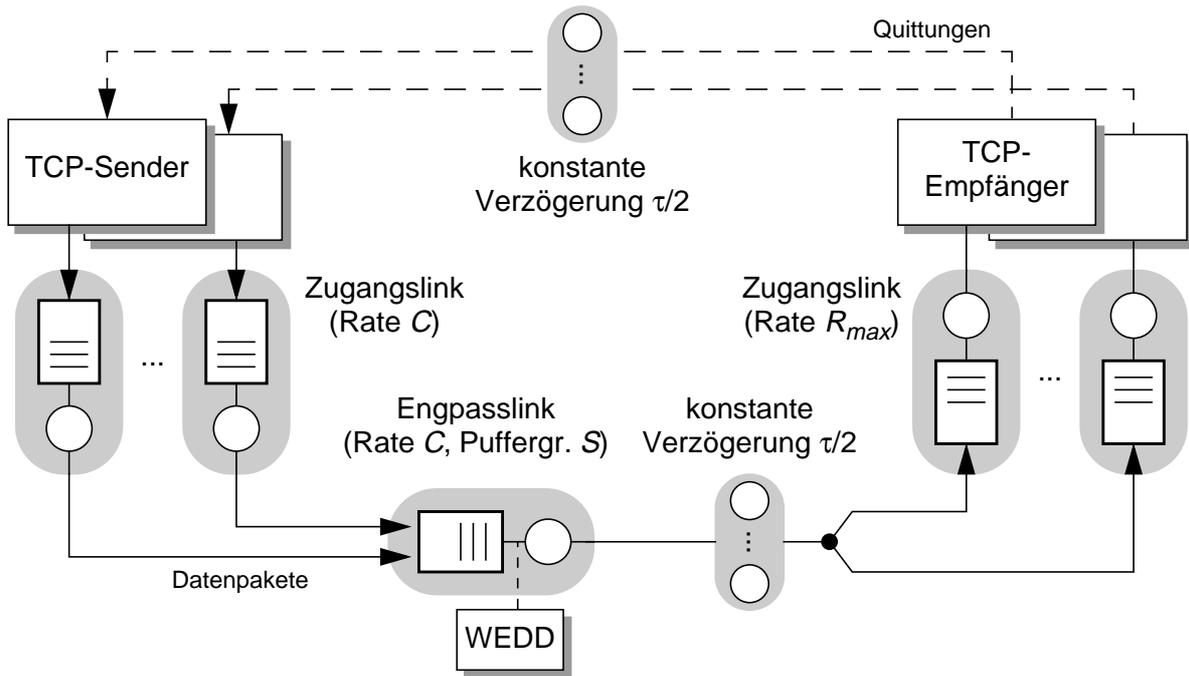
## 6.2 Differenzierung von elastischem Verkehr

Wie in Kapitel 5 beschrieben, ist WEDD nicht nur geeignet, um Echtzeitverkehr entsprechend seiner Verzögerungsanforderungen unterschiedlich zu behandeln, sondern es ermöglicht auch eine Differenzierung von elastischem, d. h. TCP-basiertem Verkehr. In diesem Abschnitt werden Untersuchungen vorgenommen, die nur Verkehr auf der Basis von TCP berücksichtigen und auf dem nachfolgend beschriebenen Modell beruhen. Ergebnisse werden für den Fall ungesättigter TCP-Quellen sowie für Szenarien mit dynamischem TCP-Verkehr präsentiert.

### 6.2.1 Modell

Das Systemmodell für die Untersuchungen mit elastischem Verkehr ist in Bild 6.26 dargestellt. Je nach TCP-Modell (siehe Abschnitt 4.2.2) wird von einer statischen oder dynamischen Anzahl von TCP-Verbindungen ausgegangen, die im Modell jeweils durch eine Sender- und eine Empfängerkomponente realisiert werden. Es wird angenommen, dass jeder TCP-Sender und -Empfänger durch je einen Zugangslink an das Netz angeschlossen ist, der durch eine Bedieneinheit mit konstanter Bedienrate und FIFO-Bearbeitungsdisziplin sowie eine DT-Warteschlange modelliert wird. Das Netz selbst wird im Wesentlichen durch einen Engpasslink auf dem Hinkanal (TCP-Datenpakete vom Sender zum Empfänger) repräsentiert. Dieser wird im Modell durch eine Bedieneinheit mit konstanter Bedienrate  $C$  und einen Puffer mit Gesamtkapazität  $S$  wiedergegeben. Bei Pufferung und Bedienung werden hier mehrere Klassen unterschieden, denen jeweils eine logische Warteschlange zugeordnet ist. Die Abarbeitung erfolgt mit Hilfe der WEDD-Strategie. Hinzu kommen noch jeweils als *Infinite Server* modellierte konstante Verzögerungskomponenten auf dem Hinkanal (Datenpakete vom Sender zum Empfänger) und dem Rückkanal (TCP-Quittungen vom Empfänger zum Sender), mit denen insbesondere Signallaufzeiten, aber auch Verzögerungen in den Endgeräten berücksichtigt werden.

In den nachfolgenden Untersuchungen werden sowohl die Bedienrate  $C$  des Engpasslinks als auch die dort vorhandene Puffergröße  $S$  variiert. Die Bedienrate auf dem empfangsseitigen Zugangslink wird mit  $R_{max}$  bezeichnet und stellt einen Parameter dar, dessen Wert i. Allg. deutlich kleiner ist als die Rate auf dem Engpasslink. Für den sendeseitigen Zugangslink wird hingegen angenommen, dass die Rate der auf dem Engpasslink entspricht. Damit ist die Beschränkung der Bandbreite auf der Sendeseite gegenüber der auf der Empfangsseite vernachlässigbar, wodurch reale Szenarien, bei denen ein Server mit breitbandiger Netzanbindung als Sender und ein Client mit relativ schmalbandigem Netzzugang als Empfänger fungieren, angemessen wiedergegeben werden. Ein komplettes Weglassen der Komponente für den sendeseitigen Zugang könnte allerdings zu ungewollten Effekten führen, da die Wahrscheinlichkeit für den Verlust unmittelbar aufeinander folgend gesendeter Pakete deutlich ansteigen würde. Für beide Zugangslinks wird eine konstante Puffergröße von  $S_S = S_E = 100\,000$  Byte



**Bild 6.26:** Systemmodell für die Untersuchungen mit elastischem Verkehr

angenommen. Ein weiterer Parameter ist die Verzögerung durch die beiden konstanten Verzögerungsglieder, deren Summe mit  $\tau$  bezeichnet wird.

Wie bei den meisten Untersuchungen für Echtzeitverkehr wird wieder von einer Konfiguration mit zwei Klassen ausgegangen, wobei angenommen wird, dass alle Pakete einer TCP-Verbindung der gleichen Dienstgüteklasse zugeordnet sind. Während die Werte von  $\delta_i$  und  $w_i$  in den Studien verändert werden, sind für die anderen WEDD-Parameter sowie für die Systemgrößen  $S$ ,  $R_{max}$  und  $\tau$  in Tabelle 6.3 Standardwerte angegeben, die verwendet werden, solange in den folgenden Abschnitten keine anders lautenden Angaben gemacht werden. Der große Wert der Puffergröße  $S$  bedeutet, dass keine Verluste am Puffereingang auftreten.

Für die TCP-Parameter und -Optionen werden Werte entsprechend Tabelle 4.1 zugrunde gelegt. Das bedeutet insbesondere, dass als Version der Überlaststeuerung standardmäßig „New Reno“ gewählt wird und auf der Empfängerseite kein *Delayed ACK* eingestellt ist. Die Größe des Puffers im TCP-Empfänger beträgt  $S_{TCP} = 32\,000$  Byte. Damit ist gewährleistet, dass im Fall von  $R_{max} = 2$  Mbit/s und  $\tau = 100$  ms – diese Werte werden als Standardeinstellungen in den meisten der nachfolgenden Untersuchungen verwendet – und bei  $C > R_{max}$  das Bandbreite-Verzögerungs-Produkt im unbelasteten System kleiner ist als die Empfangspuffergröße:

$$R_{max} \cdot \left( \tau + 2 \cdot \frac{L_{MSS}}{C} + \frac{L_{MSS}}{R_{max}} \right) < S_{TCP} \quad (6.9)$$

**Tabelle 6.3:** Standardwerte für die Parameter bei Untersuchungen mit TCP-Verkehr

Parameter	Standardwert
$\varepsilon_i$	$10 \cdot L_{MSS}/C$
$\beta$	0
$\gamma$	0
LPD	ja
$S$	$10^6$ Byte
$R_{max}$	2 Mbit/s
$\tau$	100 ms

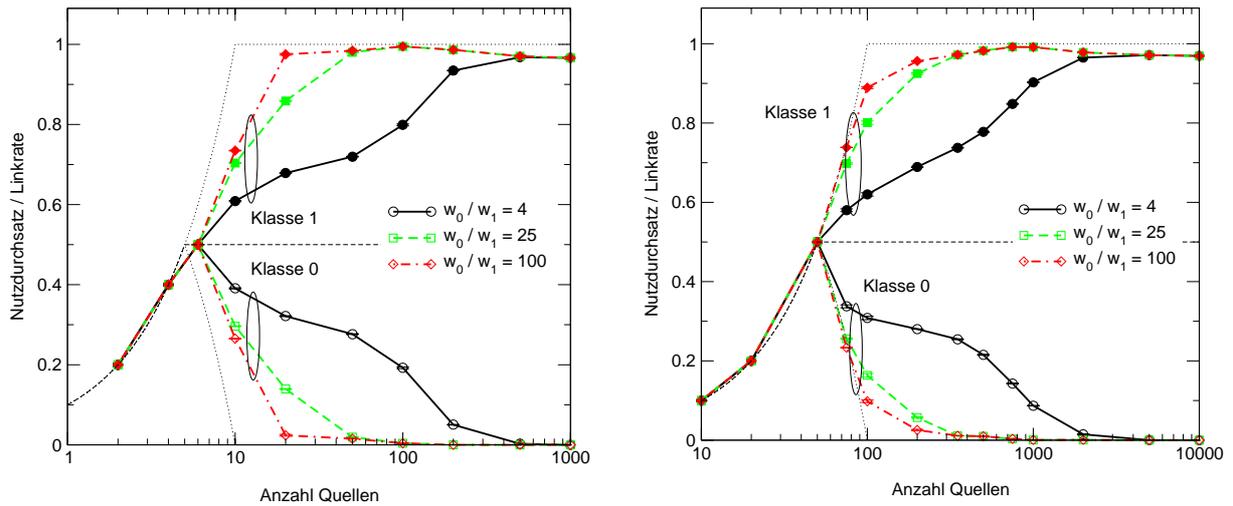
Somit wird verhindert, dass im (fast) leeren System, d. h. in dem Fall, dass keine Warteverzögerung auf dem Engpasslink auftritt, durch die beschränkte Empfangspuffergröße eine Begrenzung des Durchsatzes auf einen Wert kleiner als  $R_{max}$  verursacht wird.

## 6.2.2 Untersuchungen mit ungesättigten Quellen

Eine Untersuchung des vorgestellten Modells soll zunächst auf der Basis ungesättigter TCP-Quellen (siehe Abschnitt 4.2.2.2) erfolgen, die einen guten Einblick in das Langzeitverhalten von elastischem Internetverkehr und in die Auswirkungen von TCP geben. Damit kommt zu den im letzten Abschnitt genannten Systemparametern die Anzahl von ungesättigten Quellen (jeweils bestehend aus TCP-Sender und -Empfänger) als weiterer Parameter hinzu.

Die Untersuchungen erfolgen wie in Abschnitt 6.1 simulativ unter Verwendung des in [35, 116, 180] beschriebenen TCP-Simulationsmoduls. Zwar dienen ungesättigte Quellen häufig als Grundlage für eine analytische Betrachtung von TCP-Verkehr [2, 7, 22, 54, 157, 169, 165, 190, 194, 212, 278]. Allerdings gehen die Analysemodelle meist von weiter gehenden Annahmen aus, wie z. B. unabhängigen Paketverlusten, deren Wahrscheinlichkeit vorab bekannt ist [190, 212, 278], oder bestimmten Puffermanagementstrategien wie RED [2, 157]. Viele analytische Ansätze basieren auf einer Abbildung des Systems auf eine mehrdimensionale Markoffkette und einer Berechnung der Zustandswahrscheinlichkeiten [22, 157, 165, 194]. Die Anzahl der Zustände wächst dabei stark mit der Quellenanzahl, sodass eine praktische Anwendung oft nur für sehr wenige Quellen möglich ist. Außerdem können bei einer Analyse häufig bestimmte Protokolleigenschaften von TCP nicht berücksichtigt werden.

Dennoch können analytische Lösungen bereits brauchbare Näherungen für den Durchsatz liefern und wichtige Zusammenhänge zwischen System- und Leistungskenngrößen verdeutlichen. Dies gilt insbesondere für den in Anhang B.1.2 untersuchten Zusammenhang zwischen Verlustwahrscheinlichkeit  $p$ , mittlerer RTT  $\overline{T}_{RTT}$  und maximaler Segmentgröße  $L_{MSS}$  einerseits und mittlerer Senderate  $m_S$  andererseits [190, 212]:



**Bild 6.26:** Normierter Nutzdurchsatz in Abhängigkeit von der Gesamtzahl der Quellen für  $n_0 = n_1$  sowie  $C = 10$  Mbit/s (links) bzw.  $C = 100$  Mbit/s (rechts)

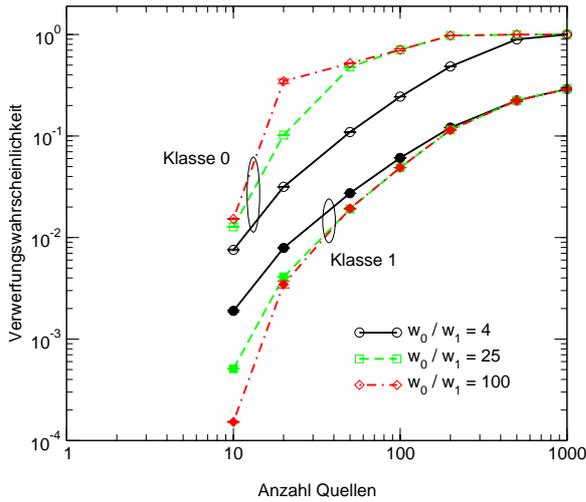
$$m_S \sim \frac{L_{MSS}}{T_{RTT} \cdot \sqrt{P}} \quad (6.10)$$

Darüber hinaus werden in Anhang B.1 Ergebnisse von Simulationen für ein System ohne Differenzierung präsentiert, auf die im Folgenden an einigen Stellen verwiesen wird. Für den undifferenzierten Fall sind außerdem in der Literatur zahlreiche Simulationsstudien mit ungesättigten TCP-Quellen zu finden [97, 106, 147, 230], die aber meist auf anderen Modellen oder anderen Parameterwerten beruhen oder die Ergebnisse in anderer Weise darstellen (z. B. mit Hilfe von Traces).

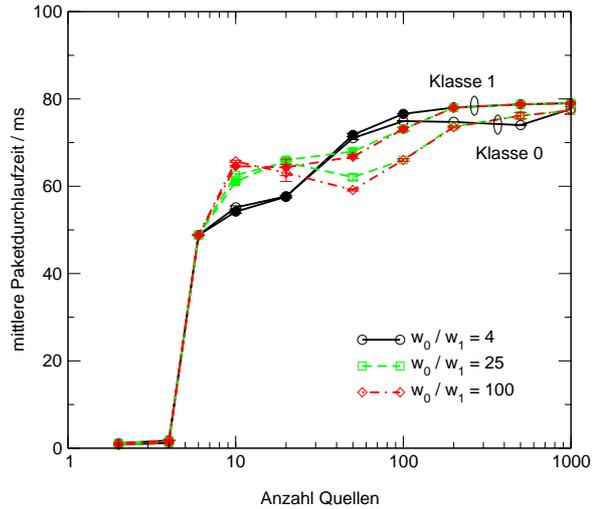
### 6.2.2.1 Einfluss der Quellenanzahl

Eine erste Simulationsreihe dient dazu, die Möglichkeiten der WEDD-Differenzierung bei Anwendung auf TCP-Verkehr mit unterschiedlicher Anzahl von Quellen zu demonstrieren. Dabei wird die Anzahl  $n_i$  von ungesättigten Quellen in Klasse  $i$  (mit  $i \in \{0, 1\}$ ) variiert, und zwar so, dass entweder das Verhältnis  $n_1/n_0$  oder die Gesamtzahl  $n = n_0 + n_1$  von Quellen konstant bleibt. Die bei WEDD eingestellte maximale Verzögerung  $\delta_i$  wird bei den Untersuchungen in diesem Abschnitt in beiden Klassen auf einen Wert von  $10^5$  Byte/ $C$  gesetzt, was im Fall von  $C = 10$  Mbit/s  $\delta_i = 80$  ms, bei  $C = 100$  Mbit/s  $\delta_i = 8$  ms entspricht. Eine Differenzierung findet somit allein aufgrund der unterschiedlichen Verwerfungsgewichte  $w_i$  statt.

Bild 6.26 zeigt den auf die Linkrate  $C$  normierten Summennutzdurchsatz  $n_i \cdot g_i$  in den beiden Klassen in Abhängigkeit von  $n$  bei gleicher Quellenzahl  $n_0 = n_1 = n/2$  in beiden Klassen für verschiedene Werte von  $w_0/w_1$ . Gestrichelt bzw. punktiert eingezeichnet sind außerdem die Kurvenverläufe, die sich im Idealfall bei Gleichbehandlung sowie bei statischer Priorisierung von Klasse 1 in Kombination mit einer Verdrängung von Paketen der Klasse 0 ergeben



**Bild 6.27:** Verwerfungswahrscheinlichkeit für  $n_0 = n_1$  und  $C = 10$  Mbit/s



**Bild 6.28:** Mittlere Paketdurchlaufzeit für  $n_0 = n_1$  und  $C = 10$  Mbit/s

würden. Bei dieser extremsten Form der Priorisierung wird allgemein in Klasse 1 bzw. 0 ein mittlerer Nutzdurchsatz pro Quelle von

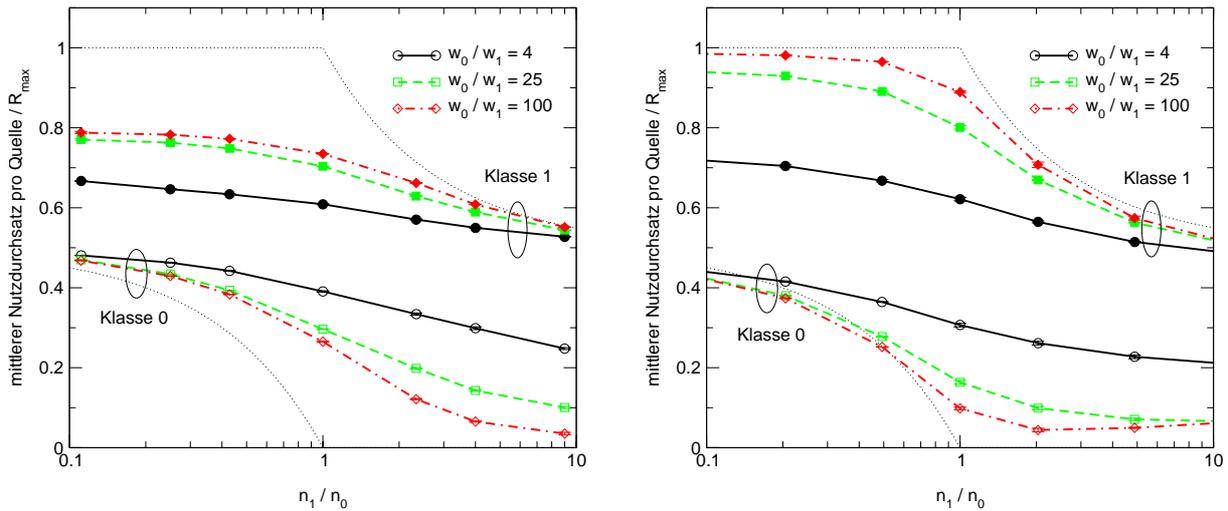
$$g_1 = \min\left(\frac{C}{n_1}, R_{max}\right)$$

$$g_0 = \min\left(\frac{C - n_1 \cdot g_1}{n_2}, R_{max}\right) = \min\left(\max\left(0, \frac{C - n_1 \cdot R_{max}}{n_2}\right), R_{max}\right) \quad (6.11)$$

erzielt, d. h. für  $n \geq 2 \cdot C/R_{max}$  erhält Klasse 0 keinen Anteil mehr an der Linkbandbreite.

Bei WEDD ergeben sich wie erwartet Kurven, die zwischen denen für Gleichbehandlung und statische Priorisierung liegen. Jedoch stellt sich offensichtlich nicht wie gewünscht ein konstantes Verhältnis  $g_1/g_0$  entsprechend  $\sqrt{w_0/w_1}$  ein (siehe Gleichung 5.2). Lediglich im Fall  $w_0/w_1 = 4$  wird das Verhältnis  $g_1/g_0 = 2$  im Bereich einer mäßig hohen Quellenanzahl näherungsweise erreicht. Für eine zunehmende Anzahl von Quellen erfolgt hingegen immer mehr eine Angleichung an die statische Priorisierung. Dies bedeutet, dass es für große Werte von  $n$  zu einem Aushungern von Klasse 0 kommt. Das Verhältnis  $w_0/w_1$  bestimmt dabei, wie schnell der Durchsatz in Klasse 0 mit zunehmenden Werten von  $n$  abnimmt.

Wie es zu dem Aushungern von Klasse 0 kommt, lässt sich zumindest teilweise mit Hilfe der in Bild 6.27 dargestellten Verwerfungswahrscheinlichkeiten verdeutlichen. Während für kleine Werte von  $n$  die Kurven der Verwerfungswahrscheinlichkeiten  $p_i$  in den beiden Klassen parallel verlaufen und das Verhältnis  $p_0/p_1$  dem über  $w_0/w_1$  eingestellten Wert entspricht, muss sich  $p_0$  für zunehmende Quellenzahl (und damit gemäß Anhang B.1.1 insgesamt zunehmende Verluste) der Obergrenze 1 annähern, damit das Verhältnis der Verwerfungswahrscheinlichkeiten dem eingestellten Wert so nahe wie möglich kommt. Damit wird gemäß Anhang B.1.2 der Zusammenhang zwischen Verwerfungswahrscheinlichkeit und Nutzdurchsatz in die-



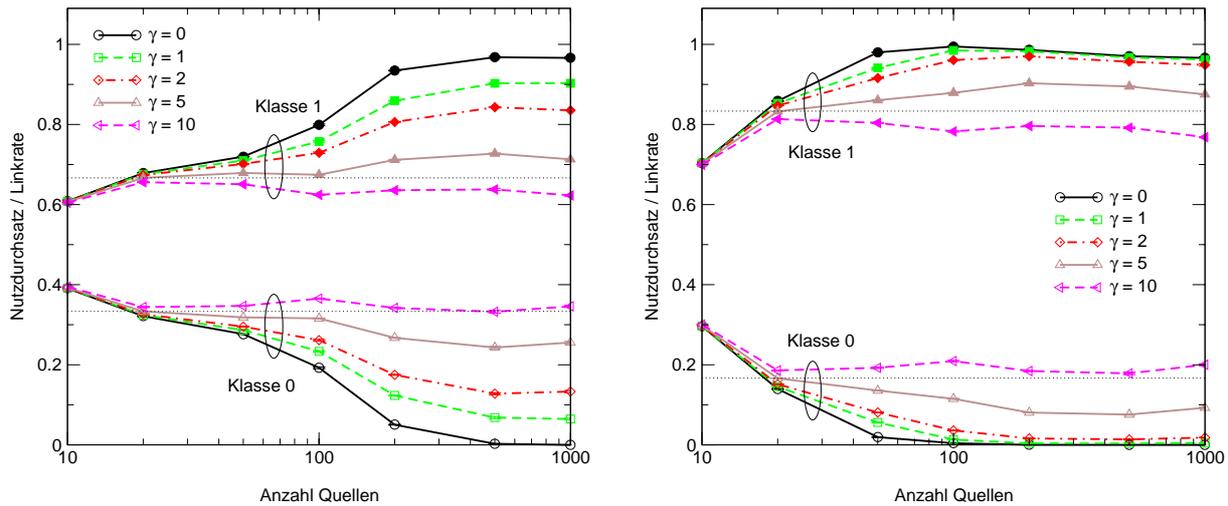
**Bild 6.29:** Mittlerer Nutzdurchsatz pro Quelle für  $C = 10$  Mbit/s und  $n = 10$  (links) bzw.  $C = 100$  Mbit/s und  $n = 100$  (rechts)

sem Bereich nicht mehr von  $1/\sqrt{p_0}$ , sondern vom Faktor  $(1 - p_0)$  dominiert. Dadurch kommt es mit zunehmender Verwerfungswahrscheinlichkeit zu einem steilen Abfall von  $g_0$ .

Vor allem bei  $C = 10$  Mbit/ zeigt sich außerdem, dass im Bereich geringer Quellenanzahl auch für einen hohen Wert von  $w_0/w_1$  die Kurve für statische Priorisierung nicht ganz erreicht wird. Dies liegt daran, dass identische Werte für die maximale Verzögerung in beiden Klassen gewählt wurden, sodass es zu einer differenzierten Behandlung erst kommt, wenn ein Paket Gefahr läuft, seine Frist nicht einzuhalten. Dies bewirkt, dass die mittlere Paketdurchlaufzeit in beiden Klassen hohe Werte annimmt (Bild 6.28), was wiederum zu einer Erhöhung der mittleren RTT und damit zu einer Verringerung des Nutzdurchsatzes auch im verlustfreien Fall führt. In Bild 6.28 kann man erkennen, dass die mittlere Paketdurchlaufzeit in Klasse 1 für größere Werte von  $n$  sogar über der von Klasse 0 liegt. Dies liegt daran, dass angesichts der extrem hohen Verwerfungswahrscheinlichkeit in Klasse 0 nur einige wenige Pakete aus dieser Klasse zur mittleren Durchlaufzeit beitragen, die zu einem Zeitpunkt im Knoten angekommen sind, an dem die Warteschlangenlänge gerade kleiner ist als  $(\delta_i - \varepsilon_i) \cdot C$  und damit geringer als ihr Mittelwert.

Anstelle der Gesamtzahl von Quellen wird nun das Verhältnis  $n_1/n_0$  von Quellen in den beiden Klassen variiert, wobei die Gesamtzahl  $n = n_0 + n_1$  konstant gehalten wird. Für  $n = 10$  im Fall von  $C = 10$  Mbit/s bzw.  $n = 100$  bei einer Linkrate von 100 Mbit/s ist in Bild 6.29 der auf die Zugangsrate  $R_{max}$  bezogene mittlere Nutzdurchsatz  $g_i$  pro Quelle<sup>2</sup> über  $n_1/n_0$  aufgetragen. Außerdem ist wieder der Verlauf für  $g_i$  eingezeichnet, der sich gemäß Gleichung (6.11) als Extremfall bei statischer Priorisierung und Verdrängung ergeben würde. Zu sehen ist, dass zwar wie im Fall der Ergebnisse in Bild 6.26 kein konstantes Verhältnis  $g_1/g_0$  gemäß

<sup>2</sup> Diese Größe kann in Analogie zu den in Abschnitt 4.3.2.2 beschriebenen Maßen für dynamischen TCP-Verkehr auch als *Fun Factor* bei ungesättigten TCP-Quellen betrachtet werden.



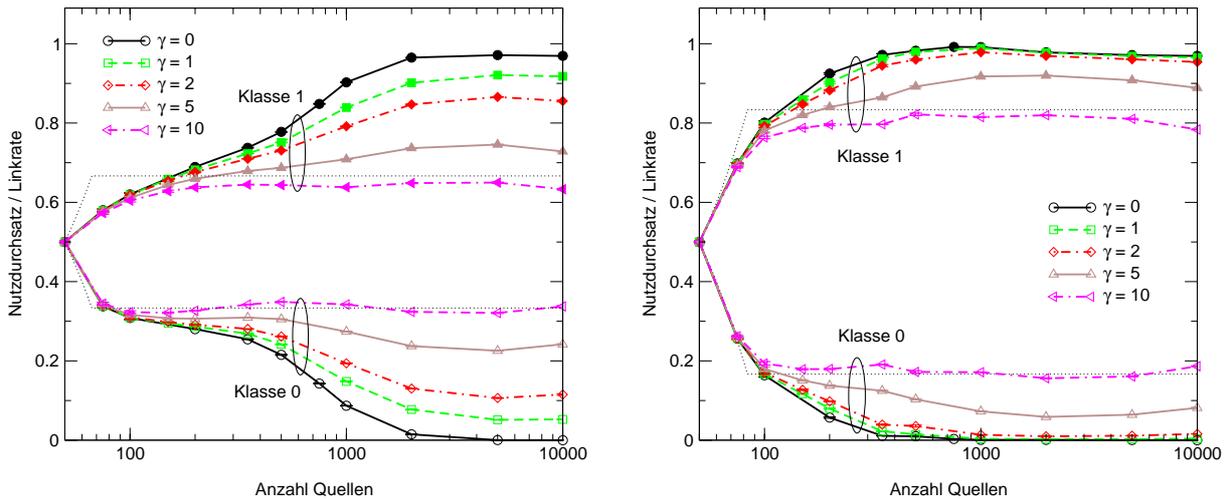
**Bild 6.30:** Normierter Nutzdurchsatz bei Anwendung von CSA für  $C = 10$  Mbit/s und  $w_0/w_{1_1} = 4$  (links) bzw.  $w_0/w_{1_1} = 25$  (rechts)

Gleichung (5.2) erzielt werden kann, dass sich aber zumindest für  $w_0/w_1 = 4$  die Änderungen des Durchsatzverhältnisses bei variiertem  $n_1/n_0$  in Grenzen halten. Auch bei den anderen Werten für  $w_0/w_1$  sind über weite Bereiche die in Abschnitt 5.1 genannten qualitativen Anforderungen an den Grad der Differenzierung erfüllt, wenn sich auch für einen zunehmenden Anteil von Quellen der Klasse 1 der oben beschriebene Effekt des Aushungerns der niederpriorien Klasse andeutet.

### 6.2.2.2 Auswirkung von CSA

Im letzten Abschnitt wurde gezeigt, dass für eine große Zahl von Quellen der Klasse 1 in der nieder priorisierten Klasse 0 der erzielbare Durchsatz verschwindet. Um dieses Aushungern von Klasse 0 zu vermeiden, wurde in Abschnitt 5.3.4 der CSA-Mechanismus als Erweiterung von WEDD vorgeschlagen. Während in den Untersuchungen für Echtzeitverkehr (siehe Abschnitt 6.1.3.4) die Vorteile dieser Erweiterung noch weitgehend im Dunkeln blieben, zeigt sich nun der Nutzen dieses Verfahrens.

Dazu wird wieder das Zwei-Klassen-WEDD-System mit gleichen Werten  $\delta_i = 80$  ms (bei  $C = 10$  Mbit/s) bzw.  $\delta_i = 8$  ms (bei  $C = 100$  Mbit/s) in beiden Klassen und identischer Anzahl von Quellen pro Klasse ( $n_0 = n_1$ ) bei variabler Gesamtzahl  $n$  von Quellen betrachtet. Aus Bild 6.30 wird deutlich, dass sowohl bei  $w_0/w_1 = 4$  als auch bei  $w_0/w_1 = 25$  der auf die Linkrate von  $C = 10$  Mbit/s normierte mittlere Summennutzdurchsatz in den beiden Klassen stark vom CSA-Parameter  $\gamma$  abhängt. Mit zunehmendem Wert von  $\gamma$  wird die Verwerfungswahrscheinlichkeit in Klasse 0 zu Lasten von Klasse 1 reduziert, sodass auch bei sehr hoher Gesamtzahl  $n$  noch Pakete aus Klasse 0 den Knoten passieren können. Mit Hilfe von CSA ist es sogar möglich, ein näherungsweise konstantes Durchsatzverhältnis zu erzielen, dessen Wert



**Bild 6.31:** Normierter Nutzdurchsatz bei Anwendung von CSA für  $C = 100$  Mbit/s und  $w_0/w_{1_1} = 4$  (links) bzw.  $w_0/w_{1_1} = 25$  (rechts)

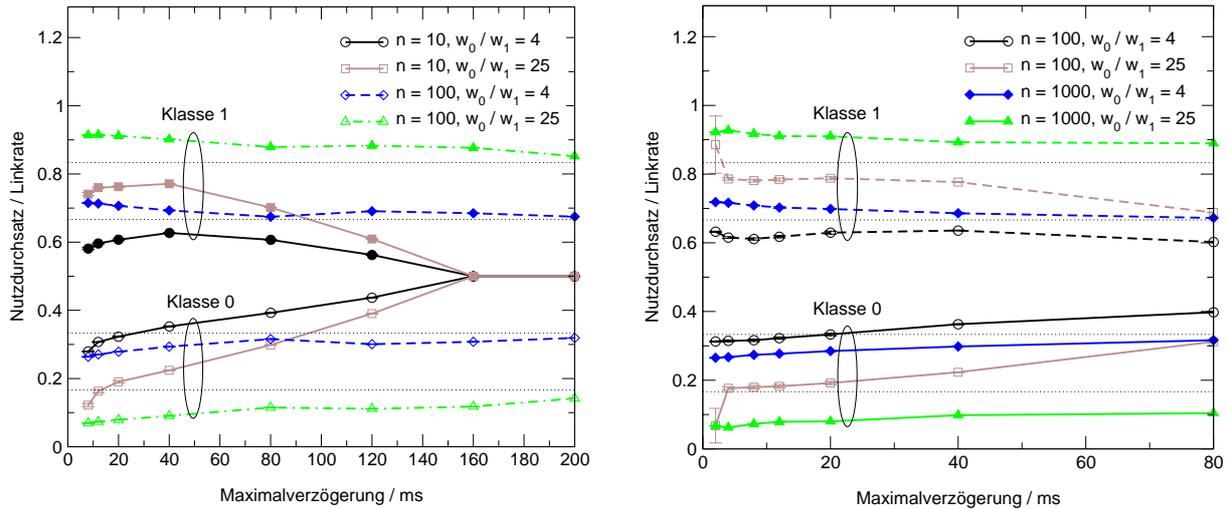
gemäß Gleichung (5.2) über  $\sqrt{w_0/w_1}$  direkt eingestellt werden kann (siehe gepunktete Linien in Bild 6.30). Relativ gut wird dies für  $5 \leq \gamma \leq 10$  erreicht.

Die guten Resultate mit CSA bestätigen sich auch für  $C = 100$  Mbit/s (Bild 6.31). Insbesondere bei  $\gamma = 10$  sind hier lediglich für eine relativ kleine Anzahl von Quellen nennenswerte Abweichungen von den Idealwerten zu beobachten. Dies liegt teilweise an dem in Anhang B.1.1 beschriebenen Effekt, dass bei  $C = 100$  Mbit/s im Bereich  $50 < n < 100$  der Gesamtdurchsatz um einige Prozent unter der Linkrate bleibt. Zu beachten ist hier außerdem, dass die Differenzierung nie größer sein kann als bei der schon in Abschnitt 6.2.2.1 als Referenz herangezogenen statischen Priorisierung mit Pufferverdrängung und somit bei kleinerer Quellenzahl das Durchsatzverhältnis zwangsläufig unter dem Wert  $\sqrt{w_0/w_1}$  bleibt.

### 6.2.2.3 Einfluss der eingestellten maximalen Verzögerung

Die bisherigen Untersuchungen mit ungesättigten TCP-Quellen gingen von einer WEDD-Konfiguration aus, bei der die Maximalverzögerung  $\delta_i$  einen festen, in beiden Klassen gleichen Wert von 80 ms bei einer Linkrate  $C = 10$  Mbit/s bzw. 8 ms bei  $C = 100$  Mbit/s hatte. Für  $n_0 = n_1$  und  $\gamma = 5$  sowie unterschiedliche Werte von  $C$ ,  $n$  und  $w_0/w_1$  wird nun der Einfluss von  $\delta_i$  untersucht.

Zunächst wird die Abhängigkeit vom Absolutwert von  $\delta_i$  betrachtet, wobei weiterhin  $\delta_0 = \delta_1$  angenommen wird. In Bild 6.32 ist dazu wieder der auf die Linkrate normierte Summennutzdurchsatz in beiden Klassen aufgetragen, für den sich idealerweise nach Gleichung (5.2) der jeweilige durch eine punktierte Linie gekennzeichnete Wert ergeben müsste. Die Ergebnisse zeigen, dass diese Werte teilweise mit guter Näherung erreicht werden und vor allem bei vielen Quellen eine geringe Abhängigkeit von  $\delta_i$  besteht. Allerdings ist zu erkennen, dass bei geringer Quellenzahl der Grad der Differenzierung mit zunehmender Maximalverzö-



**Bild 6.32:** Normierter Summennutzdurchsatz in Abhängigkeit von  $\delta_i$  für  $\delta_0 = \delta_1$  und  $C = 10$  Mbit/s (links) bzw.  $C = 100$  Mbit/s (rechts)

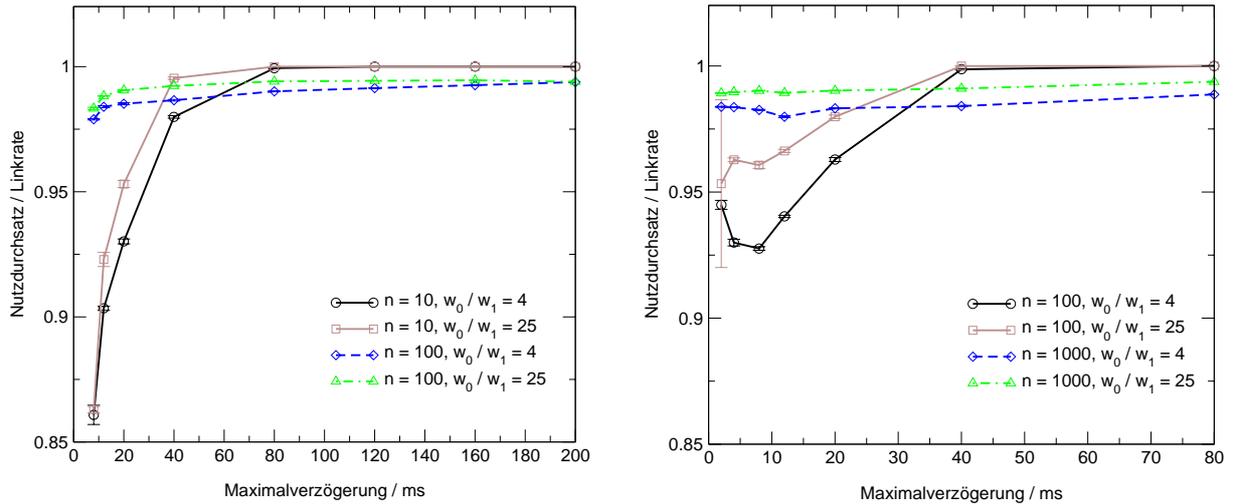
gerung abnimmt. Besonders ausgeprägt ist dieser Effekt bei einer Linkrate von 10 Mbit/s, wo ab einem gewissen Wert von  $\delta_i$  die Differenzierung völlig verschwindet. Das Verhalten ist damit zu erklären, dass in diesem Fall die TCP-Senderate nicht über Verluste, sondern allein über die mittlere RTT  $\overline{T_{RTT,i}}$  geregelt wird. Da bei  $\delta_0 = \delta_1$  die mittlere Paketdurchlaufzeit und damit die mittlere RTT in beiden Klassen nahezu gleich ist (vgl. Bild 6.28), gilt mit  $\overline{T_{RTT,i}} \approx \tau + \delta_i$  näherungsweise, dass für

$$\frac{S_{TCP}}{\tau + \delta_i} \leq \frac{C}{n} \quad (6.12)$$

keine Verluste mehr auftreten und damit alle TCP-Verbindungen im Mittel einen gleich großen Anteil an der Linkbandbreite erhalten. Bei  $C = 10$  Mbit/s und  $n = 10$  folgt aus Gleichung (6.12), dass für  $\delta_i \geq 156$  ms die Verluste verschwinden. Der Effekt ist direkt mit dem in Anhang B.1.1 erläuterten Verhalten bei FIFO-DT und zunehmender Puffergröße zu vergleichen, wobei dort die Maximalverzögerung durch  $S/C$  gegeben ist.

Abgesehen von der reduzierten Differenzierung bei zu großen Werten von  $\delta_i$  gibt es für eine zu kleine Maximalverzögerung Probleme. Zwar sind kaum Auswirkungen auf das Durchsatzverhältnis vorhanden, doch kommt es zu einer signifikanten Reduktion des Gesamtnutzdurchsatzes (Bild 6.33). Dies gilt vor allem für eine relativ geringe Anzahl von Quellen und insbesondere für  $C = 10$  Mbit/s. Auch dieser Effekt findet wieder seine Entsprechung im undifferenzierten Fall mit FIFO-DT-Disziplin. Dort zeigt sich das gleiche Verhalten bei geringer Puffergröße (siehe Anhang B.1.1).

Um das Problem der verschwindenden Differenzierung bei kleiner Quellenzahl zu beheben, bietet es sich an, nicht nur unterschiedliche Verwerfungsgewichte  $w_i$ , sondern auch verschiedene Werte für die Maximalverzögerung  $\delta_i$  in beiden Klassen zu wählen. Dies bringt aller-



**Bild 6.33:** Normierter Gesamtnutzdurchsatz in Abhängigkeit von  $\delta_i$  für  $\delta_0 = \delta_1$  und  $C = 10$  Mbit/s (links) bzw.  $C = 100$  Mbit/s (rechts)

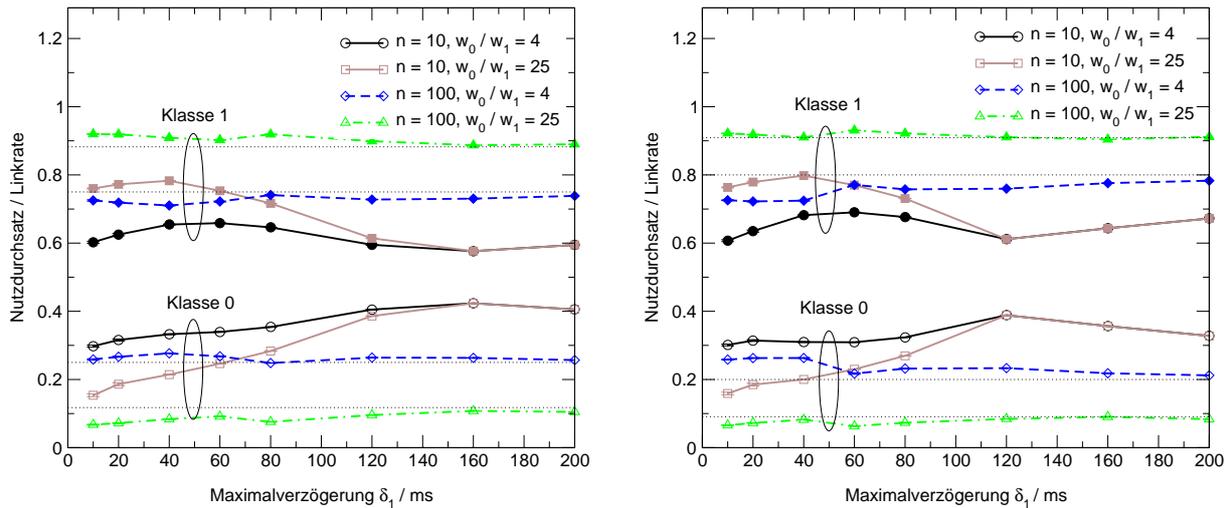
dings die Schwierigkeit mit sich, das resultierende Durchsatzverhältnis  $g_1/g_0$  vorherzusehen, da nun Gleichung (5.2) nicht mehr angewandt werden kann. In die Abschätzung von  $g_1/g_0$  muss nun nicht nur  $w_0/w_1$ , sondern auch das Verhältnis der mittleren RTTs eingehen. Letztere hängen allerdings nicht nur von  $\delta_i$ , sondern auch von der absoluten Verzögerung  $\tau$  ab. Vernachlässigt man den Einfluss von  $\tau$ , kommt man zu der folgenden groben Näherung:

$$\frac{g_i}{g_j} \approx \frac{\delta_j}{\delta_i} \cdot \sqrt{\frac{w_j}{w_i}} \quad (6.13)$$

In der nachfolgenden Studie werden nun unterschiedliche Werte für das Verhältnis  $\delta_0/\delta_1$  gewählt und dabei die Absolutwerte der Maximalverzögerung gleichmäßig variiert. Die Ergebnisse, die in Bild 6.34 nur für eine Linkrate von 10 Mbit/s dargestellt sind, verdeutlichen, dass nun auch für  $n = 10$  über den ganzen untersuchten Wertebereich von  $\delta_i$  hinweg eine Differenzierung der Dienstgüteklassen im Hinblick auf den Nutzdurchsatz stattfindet, wenn auch das angestrebte Verhältnis nicht ganz erreicht wird. Für  $n = 100$  wird der sich aus Gleichung (6.13) ergebende Wert (punktierter Linie) sogar gut getroffen, sodass diese Näherung durchaus einen brauchbaren Anhaltspunkt für das Durchsatzverhältnis liefert.

#### 6.2.2.4 Vergleich mit anderen Verfahren

Als Abschluss der Untersuchungen mit ungesättigten TCP-Quellen werden nun noch einige Vergleiche mit anderen Verfahren vorgenommen, die sich prinzipiell zur differenzierten Behandlung von TCP-Verkehr eignen. Zunächst wird hierbei die statische Priorisierung als einfach zu realisierende Lösung betrachtet. Anschließend wird WRED untersucht, das wie WEDD Unterstützung für eine proportionale Differenzierung bietet. Bei allen Studien wird von einer identischen Anzahl von Quellen in beiden Klassen ausgegangen ( $n_0 = n_1$ ).



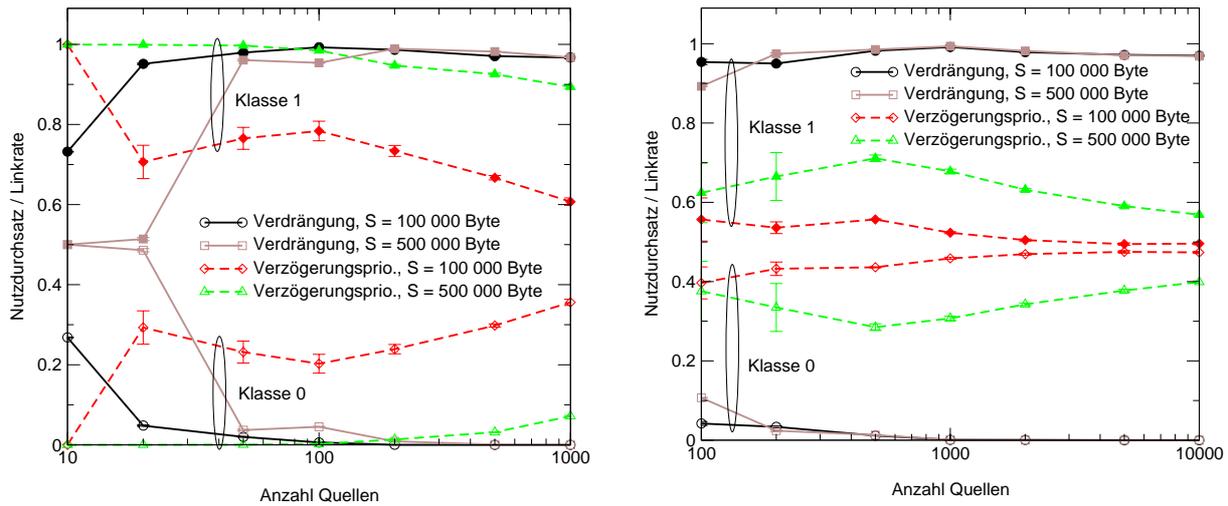
**Bild 6.34:** Normierter Summennutzdurchsatz bei konstantem Verhältnis  $\delta_0/\delta_1 = 1.5$  (links) bzw.  $\delta_0/\delta_1 = 2$  (rechts) für  $C = 10$  Mbit/s

### Statische Priorisierung

Eine statische Priorisierung kann sich auf eine der beiden Ressourcen Bandbreite und Speicherplatz beziehen. Im einen Fall handelt es sich um statische Verzögerungsprioritäten, im anderen Fall stellt die Verdrängung von niederpriorien Paketen das entsprechende Puffermanagementverfahren dar. In Kombination bewirken beide, wie schon in den vorangegangenen Abschnitten mehrfach erwähnt, ein völliges Aushungern von Klassen mit geringer Priorität. Hier kommen nun die beiden Mechanismen jeweils allein zur Anwendung, wobei statische Verzögerungsprioritäten mit DT-Puffermanagement und der Verdrängungsmechanismus mit FIFO-Scheduling kombiniert werden.

Aus den Ergebnissen in Bild 6.35 ist abzulesen, dass die Auswirkungen der Mechanismen in der genannten Anwendungsform stark von der Größe  $S$  des Puffers sowie von der Linkrate  $C$  abhängen. So zeigt bei einer Linkrate von 10 Mbit/s der Verdrängungsmechanismus bei einer Puffergröße von 500 000 Byte keine Wirkung, wenn nur wenige TCP-Verbindungen um die Ressourcen konkurrieren, während in allen anderen Fällen eine fast vollständige Verdrängung des geringer priorisierten Verkehrs stattfindet. Der Grund für das Verschwinden der Differenzierung liegt – wie schon in Anhang B.1.1.2 beobachtet – darin, dass es bei großem Puffer nicht zu Verlusten kommt, da eine Begrenzung der TCP-Senderate aufgrund der in diesem Fall möglichen großen RTT erfolgt.

Ein noch uneinheitlicheres Verhalten zeigt die statische Verzögerungspriorisierung. Hier findet nur für  $C = 10$  Mbit/s und  $S = 500\,000$  Byte über den gesamten Wertebereich von  $n$  hinweg eine vollständige Inanspruchnahme des Links durch den höher priorisierten Verkehr statt. In allen anderen Fällen ist der Grad der Differenzierung deutlich geringer, da es dort neben der Verzögerung auch zu Verlusten kommt, die bei dieser Strategie alle Klassen gleichermaßen



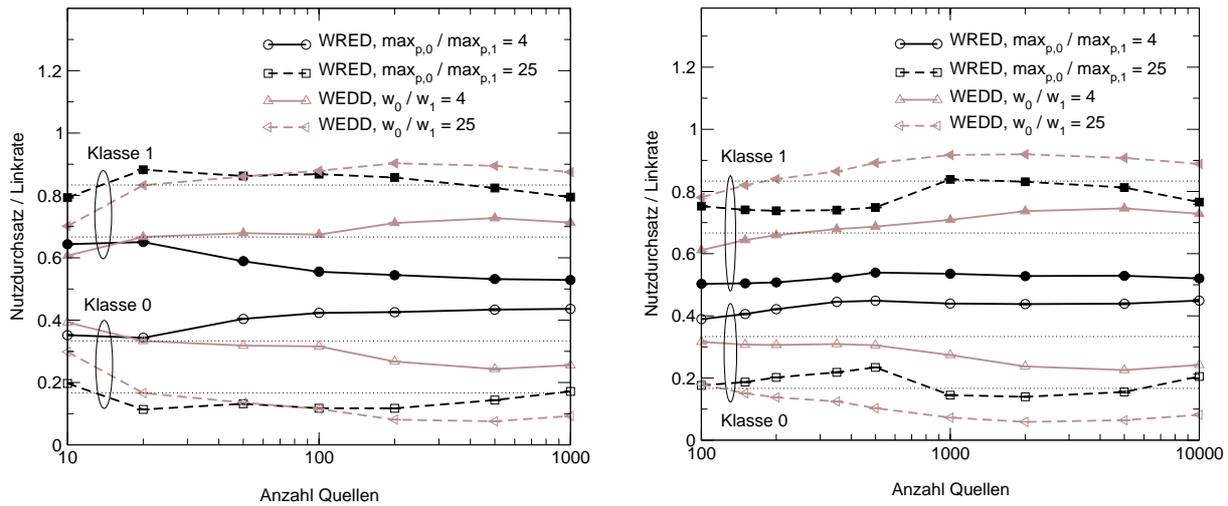
**Bild 6.35:** Normierter Summennutzdurchsatz bei statischer Verzögerungspriorität sowie bei Verdrängung für  $C = 10$  Mbit/s (links) bzw.  $C = 100$  Mbit/s (rechts)

treffen. Dies gilt besonders für  $C = 100$  Mbit/s und  $S = 100\,000$  Byte, wo sich der Vorteil der geringeren Verzögerung kaum auf die von  $\tau$  dominierte mittlere RTT auswirkt und somit der Unterschied zwischen den Klassen fast vollständig verschwindet. Eine begrenzte Differenzierung entspricht zwar durchaus den Anforderungen, die auch für WEDD aufgestellt wurden. Allerdings kann bei statischer Verzögerungspriorisierung das Durchsatzverhältnis bestenfalls über die Puffergröße näherungsweise eingestellt werden. Außerdem zeigt sich eine Abhängigkeit von der Anzahl der Quellen, die auf die mit steigenden Werten von  $n$  zunehmende Verlustwahrscheinlichkeit zurückzuführen ist.

### WRED

Die Untersuchung von WRED, das mit der FIFO-Abarbeitungsdisziplin kombiniert wird, geht von einer Puffergröße von  $S = 100\,000$  Byte sowie den in beiden Klassen identischen WRED-Schwellwerten  $min_{th} = 25\,000$  Byte und  $max_{th} = 75\,000$  Byte aus. Für den Autoregressionsfaktor  $w_q$  zur Berechnung der geglätteten Warteschlangenlänge wird ein Wert von 0.002 gewählt. Eine differenzierte Behandlung ankommender Pakete erfolgt lediglich über unterschiedliche Werte von  $max_{p,i}$ , was gemäß den Ausführungen in Abschnitt 5.4.2.2 eine proportionale Verlustdifferenzierung ermöglicht. Bei Anwendung der bekannten Näherung für den Zusammenhang zwischen mittlerem Nutzdurchsatz und Verlustwahrscheinlichkeit [212] ist damit auch eine proportionale Durchsatzdifferenzierung gemäß Gleichung (5.17) verbunden. In der vorliegenden Studie ist in allen Fällen  $max_{p,1} = 0.01$ , während für  $max_{p,0}$  die Werte 0.04 und 0.25 gewählt werden. Dies entspricht einem Verhältnis der WEDD-Gewichtungsfaktoren von  $w_0/w_1 = 4$  bzw.  $w_0/w_1 = 25$  und sollte gemäß Gleichung (5.17) auf ein Durchsatzverhältnis  $g_0/g_1 = 2$  bzw.  $g_0/g_1 = 5$  führen.

In Bild 6.36 ist wieder der in den beiden Klassen erzielte Summennutzdurchsatz in normierter Darstellung über der Anzahl von TCP-Quellen aufgetragen. Neben den Ergebnissen für



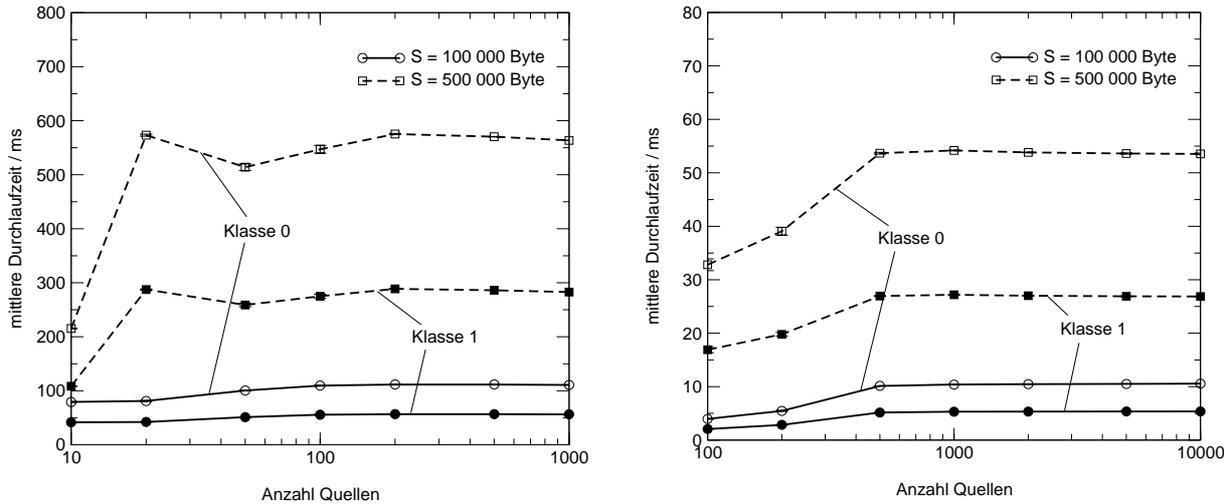
**Bild 6.36:** Normierter Summennutzdurchsatz bei WRED sowie bei WEDD für  $C = 10$  Mbit/s (links) bzw.  $C = 100$  Mbit/s (rechts)

WRED und WEDD sind die oben erwähnten Idealwerte eingezeichnet, die sich unter der vereinfachenden Annahme  $g_0 + g_1 = C$  aus Gleichung (5.17) für WRED bzw. Gleichung (5.2) für WEDD ergeben. Die Resultate von WEDD basieren auf einer Konfiguration mit gleichen Werten  $\delta_i = 80$  ms (bei  $C = 10$  Mbit/s) bzw.  $\delta_i = 8$  ms (bei  $C = 100$  Mbit/s) der Maximalverzögerung und einer Anwendung der CSA-Erweiterung mit  $\gamma = 5$ . Es kann festgestellt werden, dass die Leistung von WRED im Großen und Ganzen mit der von WEDD vergleichbar ist. Bei genauerer Betrachtung fällt allerdings auf, dass WRED im Fall  $\max_{p,0}/\max_{p,1} = 4$  deutlich zu geringe Unterschiede zwischen den Klassen liefert. Andererseits weicht das Durchsatzverhältnis bei WEDD bei einer großen Anzahl von Quellen nach oben vom Idealwert ab.

Insgesamt kann die Untersuchung als Indiz gewertet werden, dass WEDD für eine proportionale Differenzierung von TCP-Verkehr ebenso geeignet ist wie WRED. Dies ist vor allem vor dem Hintergrund zu sehen, dass WRED speziell auf TCP-Verkehr zugeschnitten ist, während WEDD als Universallösung zur Differenzierung von Echtzeit- und TCP-Verkehr antritt. Darüber hinaus gilt es zu beachten, dass WRED noch ein gewisses Potenzial für Verbesserungen durch Anpassung seiner zahlreichen Parameter besitzt, die in dieser Untersuchung zwar in Anlehnung an die Empfehlungen in [109] auf prinzipiell sinnvolle Werte gesetzt, jedoch nicht weiter untersucht wurden.

## WTP

In einem letzten Vergleich wird WEDD dem in [88] als Verfahren zur proportionalen Differenzierung vorgeschlagenen WTP gegenübergestellt. Die WTP-Differenzierungsparameter  $s_i$  bewirken zunächst unterschiedliche mittlere Verzögerungen, die jeweils proportional zum Kehrwert von  $s_i$  sind. Dies wird durch Simulationen für  $s_1/s_0 = 2$  mit verschiedenen Puffergrößen und Linkraten, deren Ergebnisse für die mittlere Paketdurchlaufzeit in Bild 6.37 über der Anzahl von Quellen aufgetragen sind, bestätigt. Die Absolutwerte sind dabei allerdings



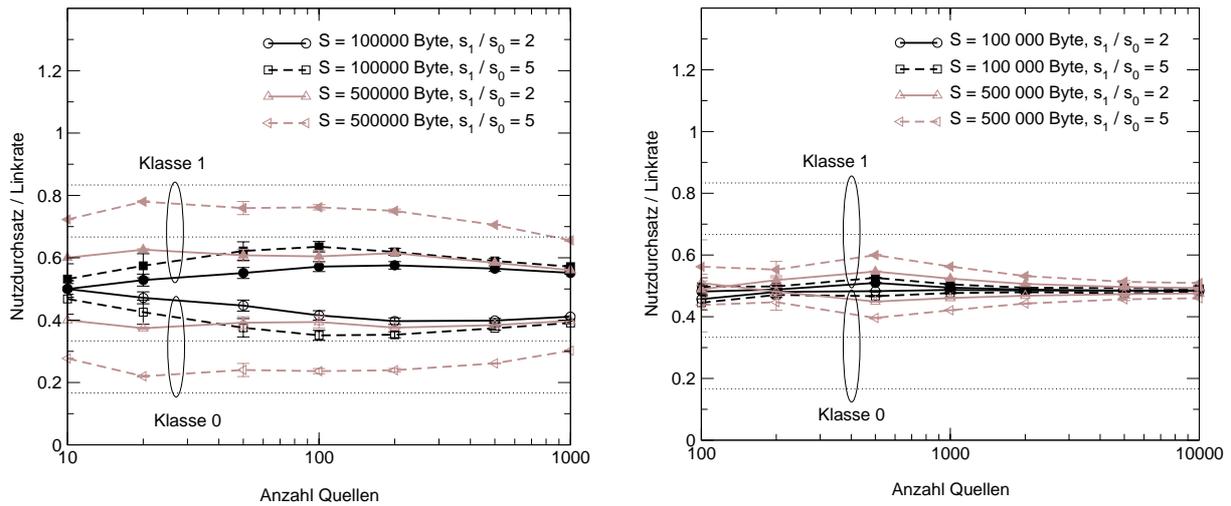
**Bild 6.37:** Mittlere Paketdurchlaufzeit bei WTP für  $s_1/s_0 = 2$  und  $C = 10$  Mbit/s (links) bzw.  $C = 100$  Mbit/s (rechts)

sehr hoch, insbesondere in Klasse 0. Dort wird der Wert der Maximalverzögerung  $S/C$  bei FIFO-Abarbeitung deutlich überschritten.

Der Einfluss der WTP-Differenzierungsparameter  $s_i$  auf den mittleren Nutzdurchsatz wird durch Gleichung (5.15) beschrieben. Dabei tritt allerdings das Problem auf, dass eine Abhängigkeit von der konstanten Verzögerung  $\tau$  besteht. Nur wenn  $\tau$  gegenüber der mittleren Verzögerung im Puffer vernachlässigt wird, folgt das Durchsatzverhältnis  $g_1/g_0$  dem Verhältnis der Differenzierungsparameter  $s_1/s_0$ . Wie in Bild 6.38 zu sehen ist, gilt dies bei den untersuchten Szenarien jedoch nur im Fall einer Linkrate von 10 Mbit/s und eines großen Puffers annähernd. Bei einem kleinen Puffer hingegen wird die mittlere RTT von  $\tau$  dominiert, sodass die Differenzierung geringer ausfällt, als das Verhältnis  $s_1/s_0$  vorgibt. Besonders stark macht sich dies bei  $C = 100$  Mbit/s bemerkbar. Hier wären sehr große Puffer notwendig, um Wartezeiten zu erreichen, die wesentlich größer sind als  $\tau$ . Allerdings haben Untersuchungen mit unbegrenzter Pufferkapazität, deren Ergebnisse hier nicht dargestellt sind, auch dort große Abweichungen von  $s_1/s_0$  und zudem – wie auch in Bild 6.38 zu sehen – eine starke Abhängigkeit von der Anzahl von Quellen ergeben.

### 6.2.3 Untersuchungen mit dynamischem TCP-Verkehr

Die Untersuchungen mit ungesättigten TCP-Quellen in Abschnitt 6.2.2 spiegeln das Verhalten im Fall von lang andauernden Transfers wieder, bei dem die TCP-Überlastregelung für ein dynamisches Gleichgewicht unter denjenigen Verbindungen sorgt, die im Netz statistisch gesehen die gleiche Behandlung erfahren. Da aber in IP-Netzen WWW-Verkehr, verbunden mit der Übertragung eher kurzer Datenbursts, dominiert, werden in diesem Abschnitt nun Studien mit dem in Abschnitt 4.2.2.3 vorgestellten Modell für aggregierten dynamischen TCP-Verkehr präsentiert, bei dem Anforderungen zur Übertragung von Bursts in zufälligen Abständen  $T_B$  ein-



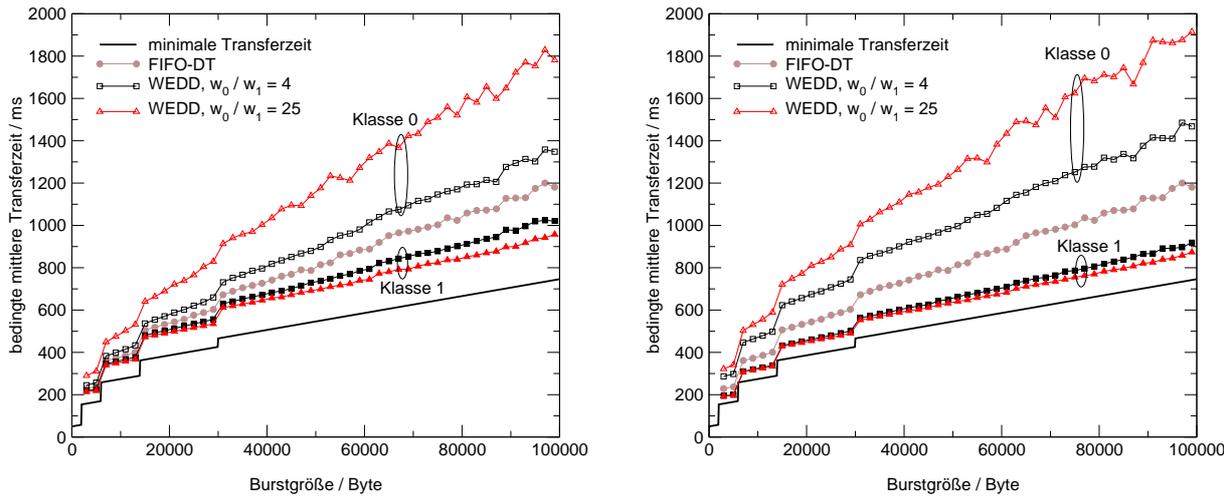
**Bild 6.38:** Normierter Summennutzdurchsatz bei WTP für  $C = 10$  Mbit/s (links) bzw.  $C = 100$  Mbit/s (rechts)

treffen. Die Ankünfte werden dabei durch einen Markoffprozess beschrieben. Die Burstgröße  $B$  folgt bei den folgenden Untersuchungen einer Pareto-Verteilung mit Formparameter  $\alpha = 1.6$  und Mittelwert  $E[B] = b = 10\,000$  Byte, was zu selbstähnlichem Verkehr mit Hurst-Parameter  $H = 0.7$  führt. Im Gegensatz zum Modell mit ungesättigten Quellen dient als Lastparameter nicht die Anzahl von Quellen, die sich hier dynamisch ändert, sondern das Angebot  $A$ , das sich gemäß Gleichung (4.21) aus den Mittelwerten von  $T_B$  und  $B$  berechnet. Allerdings wird zusätzlich die Anzahl der insgesamt aktiven Verbindungen auf einen Maximalwert  $n_{max}$  begrenzt, der hier auf 5 000 gesetzt wird.

Wie in den bisherigen Untersuchungen mit TCP-Verkehr wird von einer WEDD-Konfiguration mit zwei Klassen und den Parametern nach Tabelle 6.3 ausgegangen. Die Verteilung der Burstgröße ist in beiden Klassen identisch unter Verwendung der o. g. Parameterwerte. Außerdem wird in allen Untersuchungen, in denen keine anders lautenden Angaben gemacht werden, angenommen, dass die Angebotsanteile  $A_i$  in beiden Klassen gleich sind, d. h.  $A_0 = A_1 = A/2$ .

Mit dem Übergang zu einem Verkehrsmodell mit dynamischen TCP-Verbindungen ist auch eine Änderung im Hinblick auf die betrachteten Leistungsmaße verbunden. Anstelle des mittleren Nutzdurchsatzes sind jetzt die in Abschnitt 4.3.2.2 vorgestellten Metriken, insbesondere Mittelwert und Verteilung der Transferzeit von Bursts sowie des *Fun Factor* in verschiedenen Formen relevant.

Die folgenden Untersuchungen werden simulativ durchgeführt, wobei die in Anhang A analytisch abgeleiteten Ausdrücke für die minimale Transferzeit als Referenz herangezogen werden. Diese ergibt sich als Dauer für eine Burstübertragung im Idealfall, wobei hier die Sicht des Clients, der im vorliegenden Modell durch den Empfänger repräsentiert wird (vgl. Bild 6.26), ausschlaggebend ist. Das bedeutet, dass in diesem Idealfall der Zugangslink auf der Empfangs-



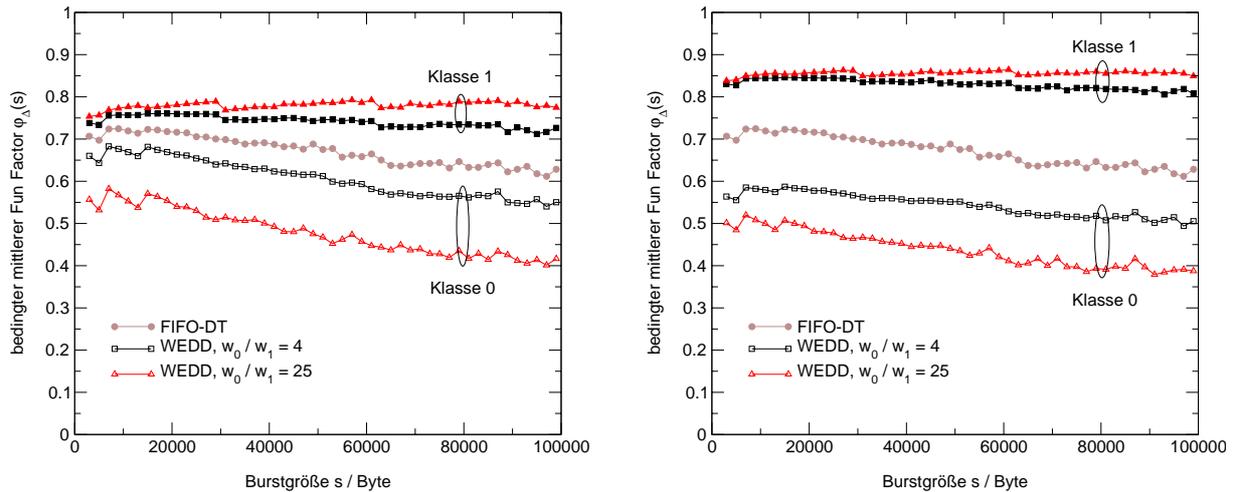
**Bild 6.39:** Bedingte mittlere Transferzeit  $t(s)$  in Abhängigkeit von der Burstgröße  $s$  für  $\delta_0 = \delta_1 = 80$  ms (links) bzw.  $\delta_0 = 160$  ms,  $\delta_1 = 80$  ms (rechts)

seite mit Rate  $R_{max}$  sowie die absolute Verzögerung  $\tau$  als gegeben angesehen werden, jedoch weder der Engpasslink noch der senderseitige Zuganglink vorhanden sind. Die minimale Transferzeit wird insbesondere auch während der Simulation für jeden Transfer berechnet, um als Bezugsgröße für die Berechnung des *Fun Factor* verwendet werden zu können. Daneben bauen die nachfolgenden Ergebnisse teilweise auf Erkenntnissen auf, die im Fall ohne Differenzierung aus Simulationen gewonnen wurden und in Anhang B.2 dargestellt sind. Dort werden auch Vergleiche mit analytischen Ergebnissen für ein *Processor-Sharing*-Modell vorgenommen, das häufig als Grundlage für die Untersuchung von dynamischem TCP-Verkehr verwendet wird [24, 62, 178, 234]. Derartige Lösungen können allerdings ebenso wie andere aus der Literatur bekannte analytische Ansätze für dynamischen TCP-Verkehr [56, 59, 129, 186, 248, 255] nicht ohne weiteres auf ein System mit Differenzierung übertragen werden.

### 6.2.3.1 Detaillierte Betrachtung eines einzelnen Lastpunktes

Die Bewertung der von WEDD erzielten Differenzierung erfolgt in diesem Abschnitt durch Betrachtung von bedingten Mittelwerten bei gegebener Burstgröße und Verteilungen bei einem Gesamtangebot von  $A = 0.9$  auf einem Link mit Rate  $C = 10$  Mbit/s. Hinsichtlich der WEDD-Parameter werden sowohl Konfigurationen mit gleichen Maximalverzögerungen ( $\delta_0 = \delta_1 = 80$  ms) als auch solche mit unterschiedlichen Werten ( $\delta_0 = 160$  ms,  $\delta_1 = 80$  ms) untersucht. Außerdem wird der Einfluss der Gewichtungparameter  $w_i$  ermittelt, für deren Verhältnis die Werte  $w_0/w_1 = 4$  oder  $w_0/w_1 = 25$  gewählt werden.

Einen ersten Einblick in die Auswirkungen von WEDD und seiner Differenzierungsparameter gibt die bedingte mittlere Transferzeit  $t(s)$  bei Auftragung über der Burstgröße  $s$  (Bild 6.39). Neben den Ergebnissen für WEDD mit  $w_0/w_1 = 4$  und  $w_0/w_1 = 25$  sind in Bild 6.39 die minimale Transferzeit  $T_{min}(s)$  nach Gleichung (A.12) sowie das Resultat bei undifferenzierter

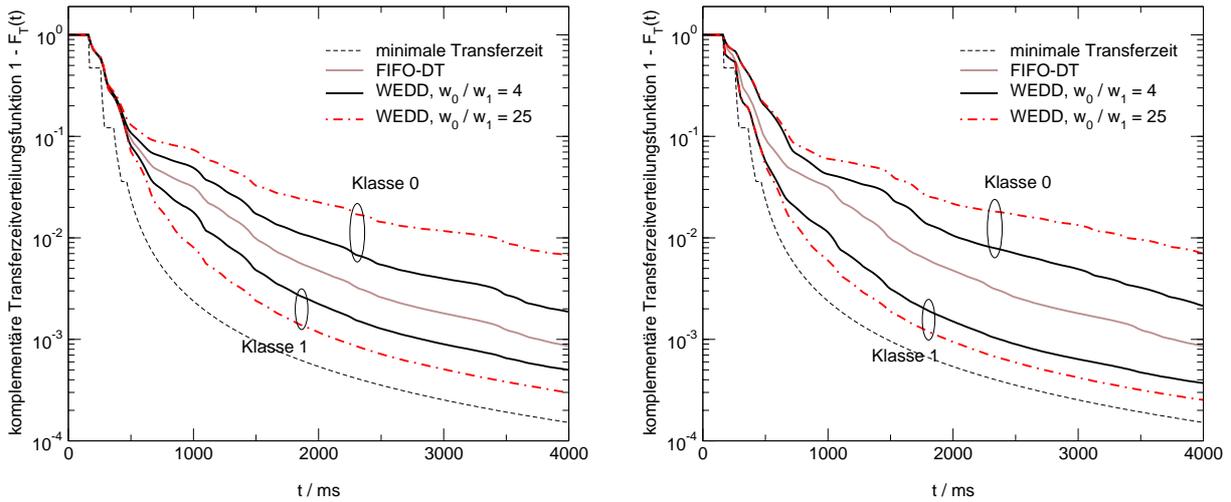


**Bild 6.40:** Bedingter mittlerer *Fun Factor*  $\varphi_{\Delta}(s)$  in Abhängigkeit von der Burstgröße  $s$  für  $\delta_0 = \delta_1 = 80$  ms (links) bzw.  $\delta_0 = 160$  ms,  $\delta_1 = 80$  ms (rechts)

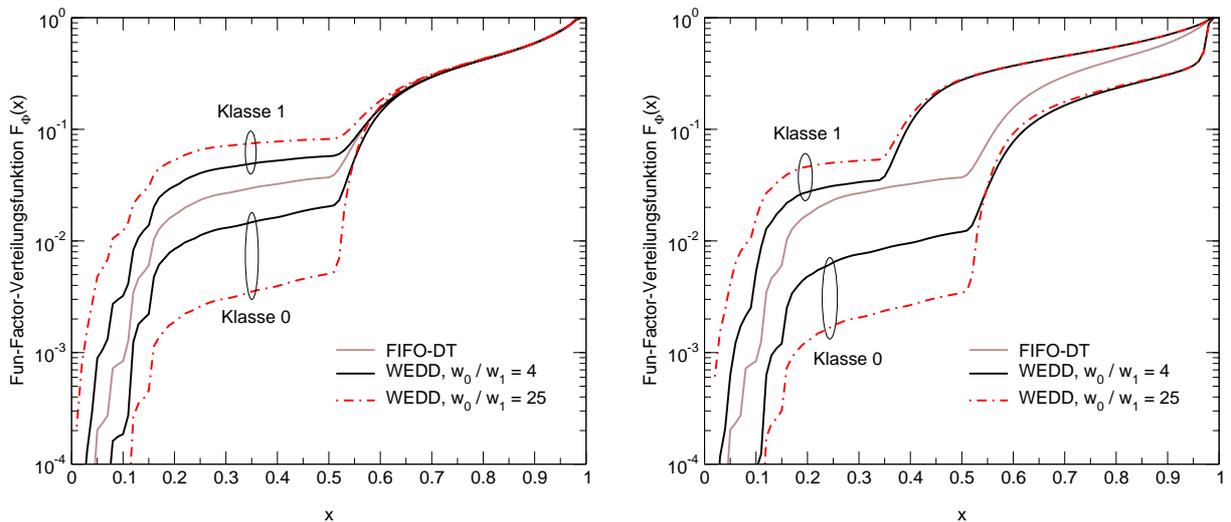
FIFO-DT-Abarbeitung mit einer Puffergröße von  $S = 100\,000$  Byte (vgl. Anhang B.2.1.1) dargestellt. Zu erkennen ist zunächst, dass sowohl  $w_0/w_1$  als auch der Unterschied der Maximalverzögerungen  $\delta_i$  zu einer Differenzierung beitragen, die für Bursts beliebiger Größe auftritt. Bei  $w_0/w_1 = 25$  muss allerdings für die Besserstellung von Klasse 1 ein vergleichsweise hoher Preis in Klasse 0 gezahlt werden, indem sich dort die mittlere Transferzeit stark erhöht. Die Verbesserung muss dabei ohnehin relativ gesehen gering ausfallen, da der Unterschied zwischen  $t(s)$  im undifferenzierten Fall und  $T_{min}(s)$  nicht allzu groß ist.

Die relative Verbesserung lässt sich noch besser mit Hilfe des bedingten mittleren *Fun Factor*  $\varphi_{\Delta}(s)$  beurteilen (Bild 6.40). Dieser entspricht nach Gleichung (4.30) dem Verhältnis von  $T_{min}(s)$  und bedingter mittlerer Transferzeit. Aus der Darstellung erkennt man zunächst die geringe Abhängigkeit von  $\varphi_{\Delta}(s)$  von der Burstgröße. Lediglich bei Klasse 0 kann eine leichte Benachteiligung von langen Bursts beobachtet werden. Darüber hinaus ist abzulesen, dass sich der mittlere *Fun Factor* von etwa 0.7 im Fall ohne Differenzierung durch die Bevorzugung bei der Verwerfung auf etwa 0.8 in Klasse 1 anheben lässt. Unterschiedliche Werte von  $\delta_i$  bringen eine weitere Steigerung auf einen Wert von knapp 0.9.

Im Gegensatz zum Nutzdurchsatz ist es beim dynamischem TCP-Modell nicht ohne weiteres möglich, die für die genannten Leistungsmaße resultierenden Werte aus den Differenzierungsparametern abzuleiten. Dies gilt auch im Hinblick auf die Verteilungen der Transferzeit  $F_T(t)$  (in Bild 6.41 in komplementärer Form dargestellt) sowie des *Fun Factor*  $F_{\Phi}(x)$  (Bild 6.42). Um eine Bewertung anhand von  $F_T(t)$  durchzuführen, ist es wieder erforderlich, die sich im Idealfall (d. h. wenn der Zugang mit Rate  $R_{max}$  den einzigen Engpass darstellt) ergebende Verteilungsfunktion zu berücksichtigen, deren Herleitung in Anhang A.4.2 zu finden ist. Dass selbst bei  $w_0/w_1 = 25$  und  $\delta_0/\delta_1 = 2$  die Wahrscheinlichkeit für Transferzeiten von einigen Sekunden nicht verschwindet, liegt also nicht am WEDD-Verfahren, sondern daran, dass durch



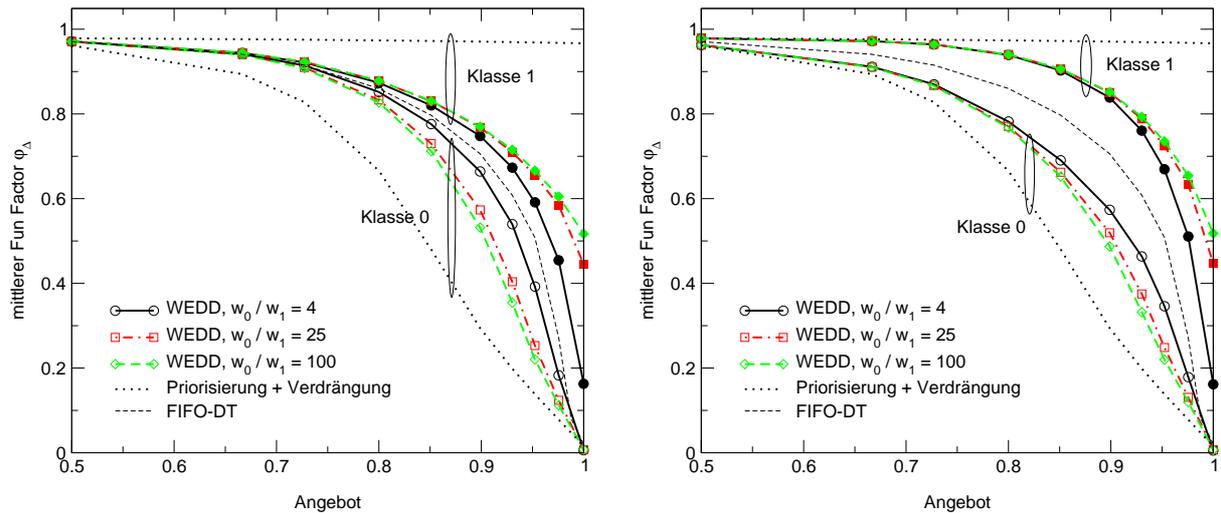
**Bild 6.41:** Komplementäre Verteilungsfunktion der Transferzeit  $T$  bei  $\delta_0 = \delta_1 = 80$  ms (links) bzw.  $\delta_0 = 160$  ms,  $\delta_1 = 80$  ms (rechts)



**Bild 6.42:** Verteilungsfunktion des *Fun Factor*  $\Phi$  bei  $\delta_0 = \delta_1 = 80$  ms (links) bzw.  $\delta_0 = 160$  ms,  $\delta_1 = 80$  ms (rechts)

die Pareto-Verteilung der Burstgröße auch sehr große Werte von  $B$  nicht unwahrscheinlich sind. Bild 6.41 zeigt außerdem, dass sich Unterschiede der Gewichtungsfaktoren bei kleinen Transferzeiten kaum bemerkbar machen.

Die Betrachtung von  $F_\Phi(x)$  anstelle der Transferzeitverteilung bringt wieder den Vorteil, dass der Vergleich mit dem Idealfall bereits implizit erfolgt, indem gemäß Gleichungen (4.26) und (4.32) für jeden Burst dessen gemessene Übertragungszeit auf die minimale Transferzeit normiert wird (mit anschließender Kehrwertbildung). Bei dieser Darstellung fällt auf, dass auch sehr kleine Werte von  $\Phi$  noch mit nicht zu vernachlässigender Wahrscheinlichkeit auftreten. Dies ist vor allem auf die Erhöhung der Transferdauer durch Paketwiederholungen zurückzuführen. Dabei wird allerdings der Einfluss kleiner Bursts, bei denen diese Erhöhung infolge



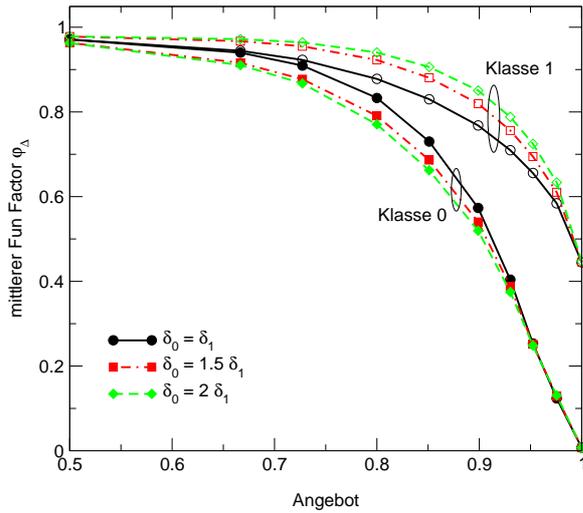
**Bild 6.43:** Mittlerer *Fun Factor*  $\varphi_\Delta$  in Abhängigkeit vom Angebot für  $\delta_0 = \delta_1 = 80$  ms (links) bzw.  $\delta_0 = 160$  ms,  $\delta_1 = 80$  ms (rechts)

von Timeouts häufig besonders stark ausfällt, überbetont (siehe entsprechende Anmerkungen hierzu in Anhang B.2.1.2). Die Differenzierung durch die unterschiedlichen Verwerfungsge-  
wichte bewirkt somit gerade im Bereich kleiner Werte von  $\Phi$  eine deutlich reduzierte Auf-  
trittswahrscheinlichkeit in Klasse 1, da Paketwiederholungen seltener erforderlich werden. Die  
Verbesserung wird durch verschiedene Werte von  $\delta_i$  verstärkt, wobei in diesem Fall auch eine  
Differenzierung im Bereich von  $\Phi > 0.6$  ermöglicht wird.

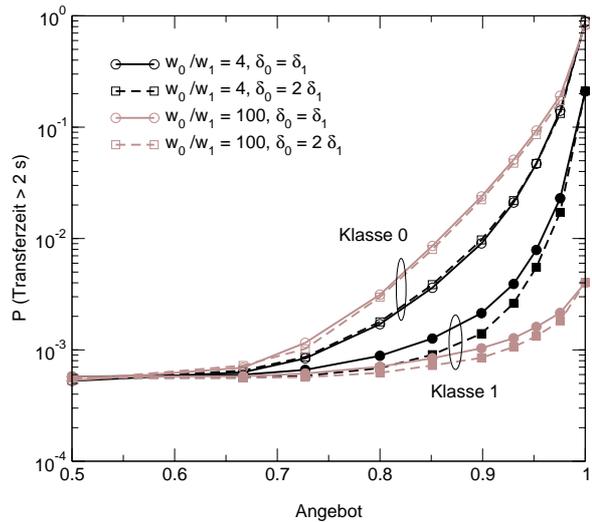
### 6.2.3.2 Betrachtung unter variablem Angebot

Die bisherigen Untersuchungen mit dynamischem TCP-Verkehr beschränkten sich auf die  
Betrachtung eines einzelnen Lastpunkts. Nun soll das Gesamtangebot  $A$  variiert werden. Als  
Leistungsmaß wird vorrangig der mittlere *Fun Factor*  $\varphi_\Delta$  herangezogen, der gemäß Gleichung  
(4.28) durch Mittelung der auf den jeweiligen Minimalwert  $T_{min}$  normierten Transferzeiten  
von Bursts und anschließende Kehrwertbildung (um einen Wert zwischen 0 und 1 zu erhalten)  
entsteht. Die Ergebnisse sind über dem Angebot aufgetragen, wobei für  $A \rightarrow 1$  die Begren-  
zung der Anzahl aktiver Verbindungen auf  $n_{max}$  greift.

Für die beiden schon im letzten Abschnitt untersuchten WEDD-Konfigurationen mit  
 $\delta_0 = \delta_1 = 80$  ms bzw.  $\delta_0 = 160$  ms,  $\delta_1 = 80$  ms sowie für unterschiedliche Werte von  $w_0/w_1$   
sind in Bild 6.43 die Ergebnisse für  $\varphi_\Delta$  dargestellt. Darüber hinaus sind als Anhaltspunkte die  
Resultate bei Verwendung von FIFO-DT anstelle von WEDD sowie bei statischer Priori-  
sierung in Kombination mit verdrängender Pufferstrategie eingezeichnet, wobei dort jeweils von  
einer Puffergröße von  $S = 100\,000$  Byte ausgegangen wird. Bei der extremen Form der Piori-  
sierung wirkt sich eine Erhöhung des Angebots auf die höher priorisierte Klasse 1 kaum aus.  
Dafür geht dort der *Fun Factor* in Klasse 0 schon bei mäßigem Angebot deutlich zurück. Bei  
WEDD hingegen ist festzustellen, dass sich eine Unterscheidung hinsichtlich der Verwer-



**Bild 6.44:** Mittlerer *Fun Factor*  $\varphi_{\Delta}$  für  $w_0/w_1 = 25$  und  $\delta_1 = 80$  ms



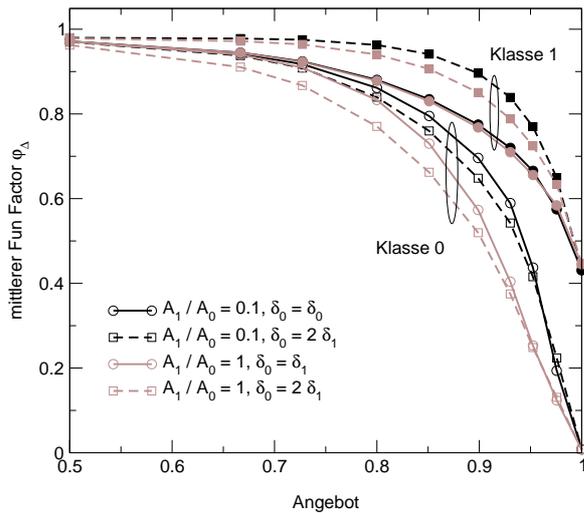
**Bild 6.45:** Transferzeit-Überschreitungswahrscheinlichkeit für  $\delta_1 = 80$  ms

fungsgewichte erst bei höherem Angebot auswirkt. Selbst mit einem Verhältnis von  $w_0/w_1 = 100$  kommt der Grad der Differenzierung nicht in die Nähe von dem bei statischer Priorisierung. Daraus kann gefolgert werden, dass mit gleichen Werten für die eingestellte Maximalverzögerung in beiden Klassen das Ausmaß der Besserstellung von Klasse 1 nicht beliebig eingestellt werden kann. Betrachtet man hingegen die Ergebnisse für  $\delta_0 = 2 \cdot \delta_1$  in Bild 6.43, so sieht man, dass sich nun auch im Bereich mittleren Angebots unterschiedliche Kurven für Klasse 0 und Klasse 1 ergeben. Für die höher priorisierte Klasse bedeutet dies, dass der Rückgang von  $\varphi_{\Delta}$  erst bei höherer Last erfolgt. Im Angebotsbereich  $A > 0.9$  andererseits sind kaum Veränderungen gegenüber der Konfiguration mit  $\delta_0 = \delta_1$  zu beobachten, d. h. dort ist ausschließlich das Verhältnis  $w_0/w_1$  ausschlaggebend. Besonders macht sich  $w_0/w_1$  bei der Annäherung an  $A = 1$  bemerkbar.<sup>3</sup>

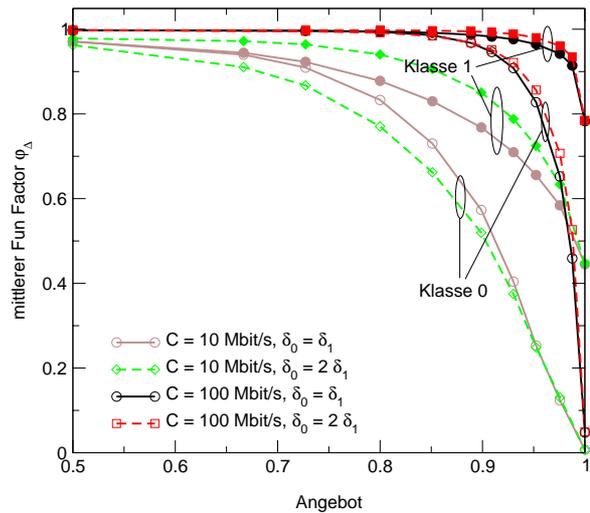
Den Einflussbereich der unterschiedlichen Werte für  $\delta_i$  macht auch Bild 6.44 deutlich, wo für  $w_0/w_1 = 25$  drei verschiedene Werte von  $\delta_0$  bei konstanter Maximalverzögerung  $\delta_1 = 80$  ms in Klasse 1 verglichen werden. Eine Vergrößerung von  $\delta_0$  wirkt sich – im Gegensatz zu einer Erhöhung von  $w_0/w_1$  wie in Bild 6.43 – am stärksten im Bereich  $0.7 < A < 0.9$  aus.

Während bei der Betrachtung von  $\varphi_{\Delta}$  alle Bursts gleichermaßen zum resultierenden Wert beitragen und damit eine verlängerte Transferdauer von kurzen Bursts im Vergleich zu ihrer tatsächlichen Bedeutung etwas überbetont wird, erlaubt die Darstellung in Bild 6.45 eine Konzentration auf längere Transfers. Zu sehen ist dort die Wahrscheinlichkeit dafür, dass eine Burstübertragung länger dauert als 2 s. In diesem Fall zeigt sich, dass das Verhältnis der Verwerfungsgewichte bereits bei mittlerem Angebot einen signifikanten Einfluss hat .

<sup>3</sup> Dieser Angebotswert bedeutet hier aufgrund der auf  $n_{max}$  begrenzten Anzahl von TCP-Verbindungen keine Stabilitätsgrenze. Daher kann auch bei  $A = 1$  in Klasse 1 noch ein Wert  $\varphi_{\Delta} > 0$  erzielt werden, der jedoch von  $n_{max}$  abhängt.



**Bild 6.46:** Mittlerer *Fun Factor*  $\varphi_{\Delta}$  für unterschiedliche Verkehrszusammensetzungen bei  $w_0/w_1 = 25$ ,  $\delta_1 = 80$  ms



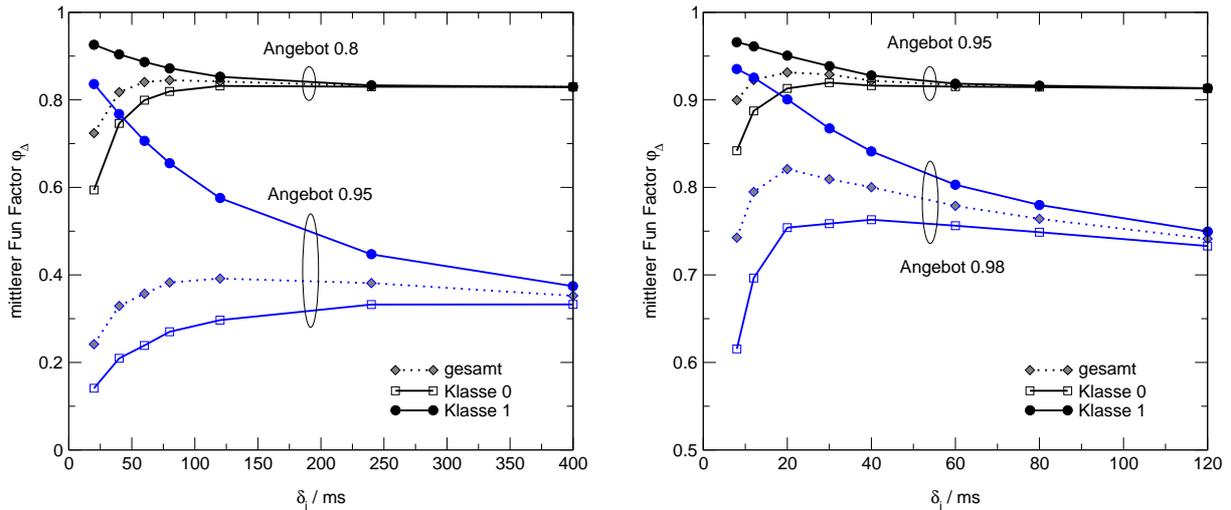
**Bild 6.47:** Einfluss der Linkrate auf  $\varphi_{\Delta}$  bei  $w_0/w_1 = 25$ ,  $\delta_1 = 80$  ms ( $C = 10$  Mbit/s) bzw.  $\delta_1 = 8$  ms ( $C = 100$  Mbit/s)

Um der WEDD-Untersuchung mit dynamischem TCP-Verkehr eine größere Aussagekraft zu verleihen, findet nun noch eine Betrachtung mit verändertem Angebotsverhältnis  $A_1/A_0$  (Bild 6.46) sowie einer höheren Linkrate  $C$  (Bild 6.47) statt. Ein reduzierter Anteil von Verkehr der Klasse 1 resultiert wie erwartet in einer Vergrößerung von  $\varphi_{\Delta}$  in allen Klassen. Darüber hinaus ist in Bild 6.46 zu erkennen, dass die Verbesserung für Klasse 1 durch unterschiedliche Werte der Maximalverzögerungen noch stärker ausfällt als bei  $A_0 = A_1$ . Eine Anhebung sämtlicher Kurven ergibt sich auch, wenn statt  $C = 10$  Mbit/s eine Linkrate von 100 Mbit/s bei gleichem Angebot und identischen Angebotsanteilen in beiden Klassen betrachtet wird. Diese Erhöhung ist allgemein mit dem Effekt des Bündelungsgewinns zu erklären (siehe auch Abschnitt B.2.2). Eine Differenzierung ist hier erst bei einem Angebot von über 0.85 erforderlich. Diese lässt sich dann mit WEDD aber auch ebenso wie bei einer kleineren Linkrate bewerkstelligen. Die Auswirkung von  $\delta_0/\delta_1$  ist hier aber offensichtlich geringer.

### 6.2.3.3 Einfluss der Maximalverzögerung

Unter den WEDD-Differenzierungsparametern hat neben den Gewichtungsfaktoren  $w_i$  auch die über  $\delta_i$  eingestellte Maximalverzögerung in den einzelnen Klassen einen großen Einfluss darauf, wie sehr sich die Leistung in der besser gestellten Klasse 1 von der in Klasse 0 abhebt. In diesem Abschnitt werden dazu nicht nur das Verhältnis, sondern auch die absoluten Werte von  $\delta_i$  über einen größeren Bereich variiert.

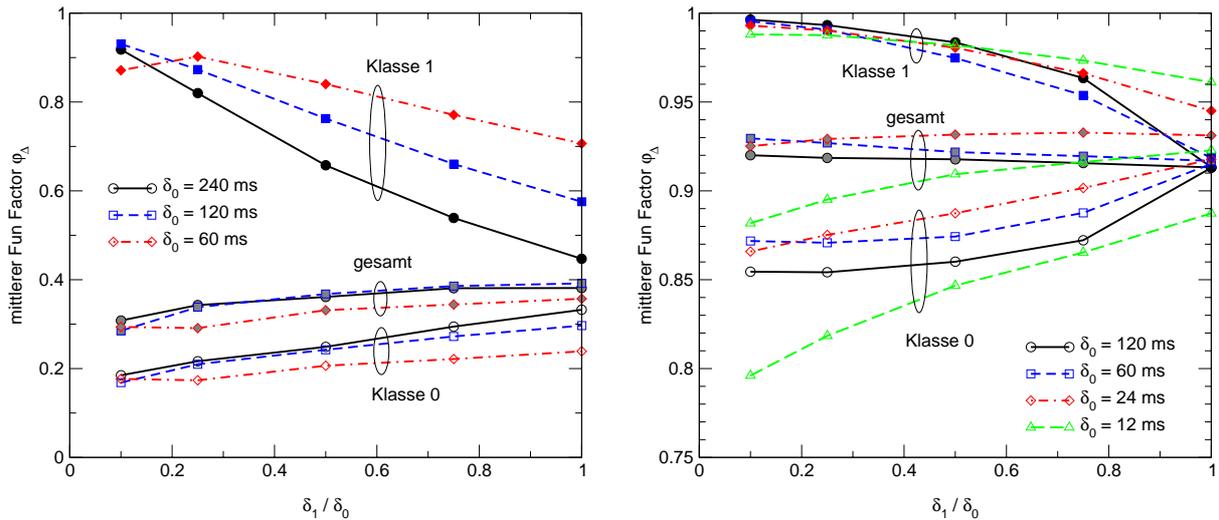
Zunächst wird von gleichen Werten der Maximalverzögerung in beiden Klassen ausgegangen. Die sich in Abhängigkeit von  $\delta_i$  ergebenden Werte des mittleren *Fun Factor*  $\varphi_{\Delta}$  sind für  $w_0/w_1 = 25$  und verschiedene Angebotswerte bei Linkraten von 10 Mbit/s und 100 Mbit/s in



**Bild 6.48:** Mittlerer *Fun Factor*  $\varphi_{\Delta}$  in Abhängigkeit von  $\delta_i$  bei  $\delta_0 = \delta_1$  und  $w_0/w_1 = 25$  für  $C = 10$  Mbit/s (links) bzw.  $C = 100$  Mbit/s (rechts)

Bild 6.48 dargestellt. Neben den Werten für  $\varphi_{\Delta}$  in den einzelnen Klassen ist auch der für den Gesamtverkehr resultierende Wert dieses Leistungsmaßes eingezeichnet. Für diesen ergibt sich – wie auch schon in Abschnitt 6.2.3.3 im Zusammenhang mit dem Nutzdurchsatz im Fall von ungesättigten TCP-Quellen beobachtet wurde – ein Abfall, wenn  $\delta_i$  zu klein ist. Bei einer Linkrate von 10 Mbit/s tritt dieser Abfall etwa für  $\delta_i < 60$  ms auf. Bei  $C = 100$  Mbit/s ist ein Maximum für  $\varphi_{\Delta}$  etwa bei  $\delta_i = 20$  ms zu erkennen, d. h. hier zeigt sich vor allem bei hohem Angebot auch für zunehmende Werte von  $\delta_i$  jenseits von  $\delta_i = 20$  ms ein zurückgehender mittlerer *Fun Factor* in Bezug auf den Gesamtverkehr. Ein anderes Bild ergibt sich, wenn man den Unterschied zwischen den Kurven für Klasse 0 und 1 betrachtet. Dieser nimmt in allen untersuchten Konfigurationen mit abnehmenden Werten für  $\delta_i$  zu. Umgekehrt ergibt sich bei großer Maximalverzögerung nur noch bei sehr hohem Angebot eine leichte Differenzierung der beiden Klassen. Fasst man die genannten Effekte zusammen, so kommt man zu dem Ergebnis, dass bei  $C = 10$  Mbit/s der Bereich  $50 \text{ ms} < \delta_i < 150 \text{ ms}$  am besten ist, während sich im Fall  $C = 100$  Mbit/s Werte aus dem Bereich  $10 \text{ ms} < \delta_i < 30 \text{ ms}$  anbieten.

Neben dem Absolutwert der Maximalverzögerung ist aber auch, wie bereits festgestellt wurde, das Verhältnis von  $\delta_0$  und  $\delta_1$  von Bedeutung. Dieser Einfluss kann mit Hilfe von Bild 6.49 verdeutlicht werden, wo für  $w_0/w_1 = 25$  und verschiedene Werte von  $\delta_0$  der mittlere *Fun Factor*  $\varphi_{\Delta}$  über  $\delta_1/\delta_0$ , d. h. bei variabler Maximalverzögerung in Klasse 1, aufgetragen ist. Im Fall einer Linkrate von 10 Mbit/s erkennt man, dass ein größerer Unterschied zwischen den Maximalverzögerungen, hier ausgedrückt durch einen kleineren Wert von  $\delta_1/\delta_0$ , zu einer starken Verbesserung in Klasse 1 bei geringer Verschlechterung in Klasse 0 führt. Der Einfluss von  $\delta_0$  nimmt bei kleiner werdendem Verhältnis  $\delta_1/\delta_0$  ab, allerdings nur sehr allmählich. Darüber hinaus ist zu erkennen, dass unabhängig von  $\delta_0$  der auf den Gesamtverkehr bezogene mittlere *Fun Factor* leicht zurückgeht, was den schon bei Bild 6.48 beobachteten Effekt bestä-



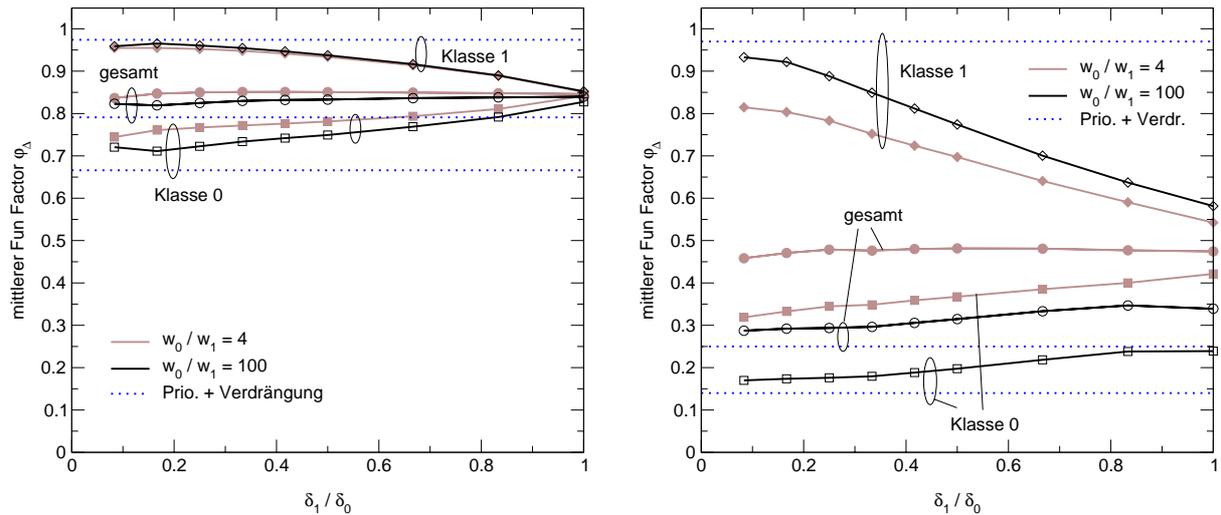
**Bild 6.49:** Mittlerer *Fun Factor*  $\varphi_{\Delta}$  über dem Verhältnis der Maximalverzögerungen bei  $w_0/w_1 = 25$ ,  $A = 0.95$  sowie  $C = 10$  Mbit/s (links) bzw.  $C = 100$  Mbit/s (rechts)

tigt.<sup>4</sup> Wird allerdings bei gleichem Angebot  $A = 0.95$  der Link bei einer Rate von 100 Mbit/s betrachtet, ändert sich am mittleren *Fun Factor* für den Gesamtverkehr nur dann merklich etwas, wenn  $\delta_0$  bereits sehr klein ist. In diesem Szenario zeigt sich deutlich der Vorteil einer unterschiedlichen Wahl von  $\delta_0$  und  $\delta_1$ , da diese eine Differenzierung teilweise erst möglich macht. Da  $\varphi_{\Delta}$  im Fall von  $C = 100$  Mbit/s insgesamt recht hoch ist, genügt bereits ein Wert von  $\delta_1/\delta_0 = 0.75$ , um den mittleren *Fun Factor* in Klasse 1 nahe an 1 heranzuführen.

In Bild 6.50 ist für  $C = 10$  Mbit/s,  $\delta_0 = 120$  ms sowie verschiedene Angebotswerte nun neben  $\delta_1/\delta_0$  auch noch die Auswirkung des Verhältnisses  $w_0/w_1$  der Gewichtungparameter zu sehen. Gleichzeitig sind dort auch die Ergebnisse im Fall einer statischen Priorisierung in Kombination mit einer Verdrängung von Paketen der Klasse 0 aus dem Puffer als Extremfall der Priorisierung eingezeichnet. Bei einem Angebot von  $A = 0.8$  ist die Differenzierung über  $w_0/w_1$  nahezu wirkungslos, während hier ein größerer Unterschied der Maximalverzögerungen annähernd eine Besserstellung von Klasse 1 wie bei statischer Priorisierung herbeiführen kann. Bei einem Angebot von  $A = 0.95$  macht sich  $w_0/w_1$  nun stärker bemerkbar. Allerdings bleibt auch hier der Einfluss von  $\delta_1/\delta_0$  insgesamt mindestens ebenso stark. Darüber hinaus erkennt man, dass der auf den Gesamtverkehr bezogene mittlere *Fun Factor* mit steigendem  $w_0/w_1$  signifikant abnimmt, während mit einem zunehmenden Unterschied in den Maximalverzögerungen – wie schon in Bild 6.49 beobachtet, nur ein leichter Rückgang verbunden ist.

Als Schlussfolgerung aus den präsentierten Untersuchungen kann man ableiten, dass eine Differenzierung hinsichtlich der Verzögerung zusätzlich zur Unterscheidung im Hinblick auf die Verwerfungswahrscheinlichkeiten in vielen Szenarien eine deutliche Besserstellung des höher priorisierten Verkehrs erst ermöglicht und damit den notwendigen Spielraum bei der Einstel-

<sup>4</sup> Man beachte den veränderten Maßstab auf der y-Achse im Diagramm für  $C = 100$  Mbit/s in Bild 6.49.



**Bild 6.50:** Mittlerer *Fun Factor*  $\varphi_{\Delta}$  über  $\delta_1/\delta_0$  bei konstanter Maximalverzögerung  $\delta_0 = 120$  ms in Klasse 0 für  $C = 10$  Mbit/s sowie  $A = 0.8$  (links) bzw.  $A = 0.95$  (rechts)

lung des Grads der Differenzierung schafft. Allerdings muss bedacht werden, dass eine damit verbundene Abkehr von der FIFO-Abarbeitung nur dann empfehlenswert ist, wenn die bereits in Abschnitt 5.1 formulierte Randbedingung gilt, dass alle Pakete einer TCP-Verbindung zur selben Klasse gehören. Andernfalls würde die bevorzugte Bedienung von Paketen in einzelnen Klassen zu Paketüberholungen führen, was bekanntermaßen negative Auswirkungen auf die Leistungsfähigkeit hat [28].

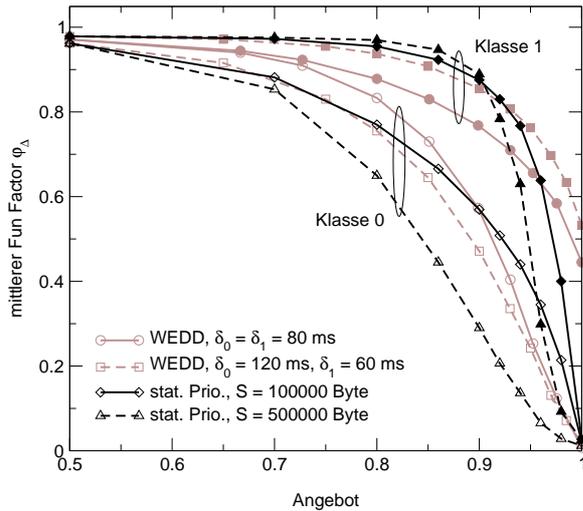
### 6.2.3.4 Vergleich mit anderen Verfahren

Wie im Fall der Untersuchungen mit ungesättigten TCP-Quellen bildet auch hier ein Vergleich mit anderen Verfahren zur Differenzierung den Abschluss der Studien dieses Abschnitts. Als Vergleichskriterium wird wie in den Untersuchungen der letzten Abschnitte der mittlere *Fun Factor*  $\varphi_{\Delta}$  herangezogen. In allen Studien wird nur eine Linkrate von  $C = 10$  Mbit/s betrachtet. Auf der Seite von WEDD wird  $w_0/w_1 = 25$  als Verhältnis der Verwerfungsgewichte gewählt. Neben einem Szenario mit  $\delta_0 = \delta_1 = 80$  ms wird auch der Fall unterschiedlicher Maximalverzögerungen ( $\delta_0 = 120$  ms,  $\delta_1 = 60$  ms) berücksichtigt.

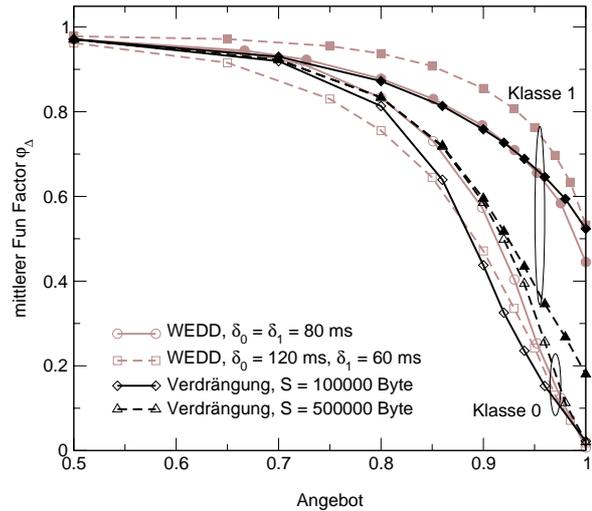
### Statische Priorisierung

Zunächst werden für einen Vergleich wieder die statische Verzögerungspriorisierung in Kombination mit DT am Puffereingang sowie der Verdrängungsmechanismus bei der Vergabe von Speicherplatz in Verbindung mit FIFO-Scheduling herangezogen.

Der Vergleich mit statischer Verzögerungspriorisierung zeigt, dass diese nur im Bereich um  $A = 0.8$  eine noch deutlichere Verbesserung für Klasse 1 gegenüber dem Fall mit WEDD bringt, wenn  $\delta_0 = 120$  ms und  $\delta_1 = 60$  ms gewählt werden (Bild 6.51). Für ein Angebot von mehr als 0.9 hingegen geht die Leistung in Klasse 1 weitaus stärker zurück als bei WEDD.



**Bild 6.51:** Vergleich von WEDD mit statischer Verzögerungspriorisierung



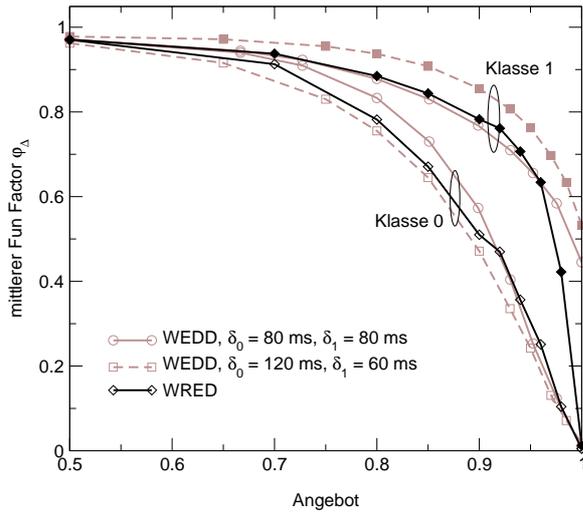
**Bild 6.52:** Vergleich von WEDD mit verdrängendem Puffermanagement

Dies liegt daran, dass in diesem Angebotsbereich die Verluste, die bei DT-Pufferdisziplin beide Klassen gleichermaßen treffen, stark zunehmen. Im Fall eines großen Puffers liegt bei statischer Priorisierung der mittlere *Fun Factor* sogar in beiden Klassen unter dem bei WEDD. Dies hängt mit dem in Anhang B.1.1 unter anderen Randbedingungen beobachteten Effekt zusammen, dass mit der Kapazität des Puffers auch die Blindlast zunimmt.

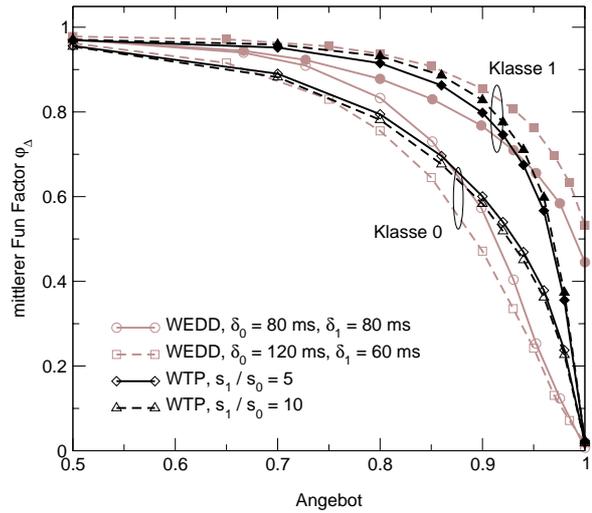
In der Gegenüberstellung mit dem Verdrängungsmechanismus liefert WEDD mit  $\delta_0 = \delta_1$  ähnliche Ergebnisse für  $\phi_\Delta$  wie der Verdrängungsmechanismus bei einer Puffergröße von  $S = 100\,000$  Byte. Eine so deutliche Besserstellung von Klasse 1 wie bei WEDD mit  $\delta_0 = 2 \cdot \delta_1$  ist hingegen mit dem Verdrängungsmechanismus unter den gegebenen Randbedingungen nicht möglich. Im Gegenteil, bei großem Puffer verschwindet dort die Differenzierung fast vollständig und ist nur noch für sehr hohes Angebot sichtbar. Dies unterstreicht die Aussage, die schon in Abschnitt 6.2.2.4 im Fall ungesättigter TCP-Quellen gemacht werden konnte, dass die Auswirkung der statischen Priorisierung stark von der Puffergröße abhängt.

## WRED

Beim Vergleich mit WRED wird wie in Abschnitt 6.2.2.4 von den Parametern  $S = 100\,000$  Byte,  $\min_{th,i} = 25\,000$  Byte,  $\max_{th,i} = 75\,000$  Byte und  $w_q = 0.002$  ausgegangen. Entsprechend dem Verhältnis  $w_0/w_1 = 25$  im Fall von WEDD wird durch unterschiedliche Werte  $\max_{p,0} = 0.25$  und  $\max_{p,1} = 0.01$  eine Differenzierung in Bezug auf die Verwerfungswahrscheinlichkeit vorgenommen. Bild 6.53 lässt eine große Ähnlichkeit zu den Ergebnissen mit WEDD bei identischen Werten für die Maximalverzögerung in beiden Klassen erkennen. Da aber WRED keine Priorisierung im Hinblick auf die Abarbeitungsreihenfolge vornimmt, ist auch hier der Grad der durch WRED erreichten Priorisierung geringer als bei WEDD mit  $\delta_0 = 2 \cdot \delta_1$ . Außerdem bricht die Leistung in Klasse 1 bei sehr hoher Last im Gegensatz zu



**Bild 6.53:** Vergleich von WEDD mit WRED ( $max_{p,0} = 0.25$ ,  $max_{p,1} = 0.01$ )



**Bild 6.54:** Vergleich von WEDD mit WTP ( $S = 100\,000$  Byte)

WEDD ein. Dies liegt daran, dass es in diesem Bereich häufig dazu kommt, dass die gemittelte Warteschlangenlänge  $\bar{Q}$  die obere Schwelle  $max_{th,i}$  überschreitet, die in beiden Klassen gleich ist. In diesem Fall wird das ankommende Paket verworfen, unabhängig davon, zu welcher Klasse es gehört.

### WTP

Bei WTP, das hier mit zwei unterschiedlichen Wertepaaren für die Differenzierungsparameter  $s_i$  bei einer Puffergröße von  $S = 100\,000$  Byte untersucht wird (Bild 6.54), gelten ähnliche Aussagen wie bei statischer Verzögerungspriorisierung. Das bedeutet vor allem, dass bei hoher Last der Unterschied zwischen den Klassen allmählich verschwindet, sodass  $\phi_\Delta$  in beiden Klassen gegen 0 geht. Im Lastbereich  $A < 0.9$  ist wie bei WEDD mit  $\delta_0 = 2 \cdot \delta_1$  eine Besserstellung von Klasse 1 gewährleistet. In diesem Bereich ist auch ein leichter Einfluss von  $s_1/s_0$  bei WTP zu erkennen, der aber deutlich geringer ausfällt als der von  $\delta_i$  bei WEDD.

## 6.3 Integration von elastischem und unelastischem Verkehr

Die bisherigen Untersuchungen haben gezeigt, dass WEDD zur Differenzierung von Verkehr mit unterschiedlichen Anforderungen, aber gleicher Charakteristik verwendet werden kann. Nun werde ein integriertes Szenario betrachtet, bei dem sowohl eine Trennung von Echtzeit- und TCP-Verkehr als auch eine Unterscheidung von Klassen innerhalb der beiden Verkehrstypen durchgeführt wird.

### 6.3.1 Szenario

Insgesamt werden beim Zugang zu einem Engpasslink mit Rate  $C$  vier Klassen ( $i \in \{0, 1, 2, 3\}$ ) unterschieden, denen unterschiedliche WEDD-Parameter gemäß Tabelle 6.4 zugewiesen sind. Die Werte ohne Klammer bei  $\delta_i$  und  $\varepsilon_i$  beziehen sich auf eine Linkrate von 10 Mbit/s, während in Klammer die Werte für  $C = 100$  Mbit/s angegeben sind. Für die anderen Parameter von WEDD werden die Standardwerte aus Tabelle 6.3 übernommen. Das bedeutet insbesondere, dass verspätete Pakete verworfen werden und dass der CSA-Mechanismus zunächst nicht angewandt wird.

Echtzeitverkehr wird wie in Abschnitt 6.1 durch ein  $M/G/\infty$ -Burst-Modell repräsentiert, wobei die Parameterkombination  $V_A$  nach Tabelle 6.1 verwendet wird. Das Angebot an Echtzeitverkehr in Klasse  $i$  wird mit  $A_i^{(RT)}$  bezeichnet und ist nur in den Klassen 2 und 3 von null verschieden (siehe Tabelle 6.4). Für elastischen Verkehr gilt das geschlossene Modell aus Abschnitt 6.2, d. h. neben den beiden Bediensystemen zur Beschreibung des Netzzugangs (mit Rate  $C$  bzw.  $R_{max} = 2$  Mbit/s) enthält das Modell im Hin- und Rückkanal jeweils eine Komponente zur konstanten Verzögerung mit einer Gesamtlaufzeit von  $\tau = 100$  ms (vgl. Bild 6.26). Ein Teil der TCP-Verbindungen wird als dynamisch angenommen, wobei wieder das in Abschnitt 6.2.3 eingesetzte Modell zur Beschreibung von Ankunft und Größe der per TCP zu übertragenden Bursts zur Anwendung kommt. Das Angebot durch diesen dynamischen TCP-Verkehr in Klasse  $i$  werde mit  $A_i^{(TCP)}$  bezeichnet. Nur in den Klassen 0 und 1 ist derartige Verkehr zu finden (vgl. Tabelle 6.4). Das Gesamtangebot  $A$  aus Echtzeit- und dynamischem TCP-Verkehr setzt sich somit wie folgt zusammen:

$$A = \sum_{i=0}^3 (A_i^{(TCP)} + A_i^{(RT)}) = A_0^{(TCP)} + A_1^{(TCP)} + A_2^{(RT)} + A_3^{(RT)} \quad (6.14)$$

Daneben gibt es noch statischen TCP-Verkehr, modelliert mit Hilfe der bereits in Abschnitt 6.2.2 isoliert betrachteten ungesättigten Quellen. Diese können nicht über ein Angebot charakterisiert werden. Statt dessen spielt hier die Anzahl  $n_i$  von Quellen in Klasse  $i$  eine Rolle, deren Summe die Gesamtzahl  $n$  ergibt. Wie der dynamische TCP-Verkehr sind auch die ungesättigten Quellen den Klassen 0 und 1 zugeordnet (vgl. Tabelle 6.4).

**Tabelle 6.4:** Klassenspezifische Parameter für integrierte Untersuchung

Parameter	Klasse 0	Klasse 1	Klasse 2	Klasse 3
$\delta_i$	120 ms (24 ms)	60 ms (12 ms)	20 ms (4 ms)	20 ms (4 ms)
$\varepsilon_i$	10 ms (2 ms)	10 ms (2 ms)	10 ms (2 ms)	10 ms (2 ms)
$w_i$	100	4	10	1
$A_i^{(TCP)}/A$	0.6	0.2	0	0
$A_i^{(RT)}/A$	0	0	0.1	0.1
$n_i/n$	0.75	0.25	0	0

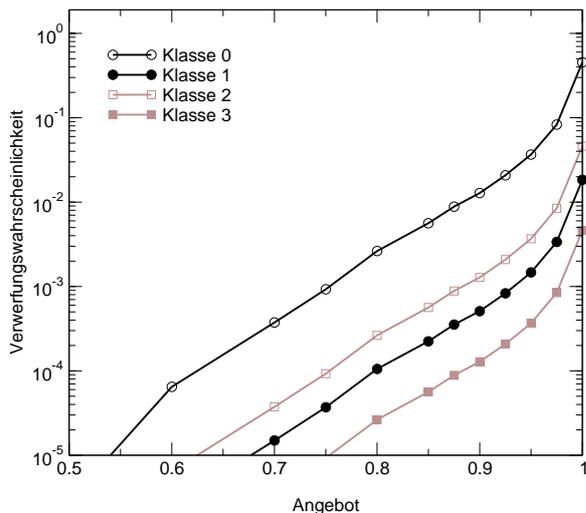
Aus den Werten für die WEDD-Parameter nach Tabelle 6.4 ergibt sich neben den höheren zeitlichen Anforderungen in den Klassen 2 und 3 die Besserstellung der Klassen 1 und 3 gegenüber den Klassen 0 bzw. 2, d. h. eine Priorisierung innerhalb des TCP- bzw. des Echtzeitverkehrs mit Hilfe der Verwerfungsgewichte. Im Fall von TCP-Verkehr fließt außerdem die aus den Untersuchungen in Abschnitt 6.2.3 gewonnene Erkenntnis ein, dass es zusätzlich zu der Differenzierung mit Hilfe unterschiedlicher Verwerfungsgewichte (hier  $w_0$  und  $w_1$ ) sinnvoll ist, auch für die Maximalverzögerungen (hier  $\delta_0$  und  $\delta_1$ ) unterschiedliche Werte zu wählen.

### 6.3.2 Ergebnisse

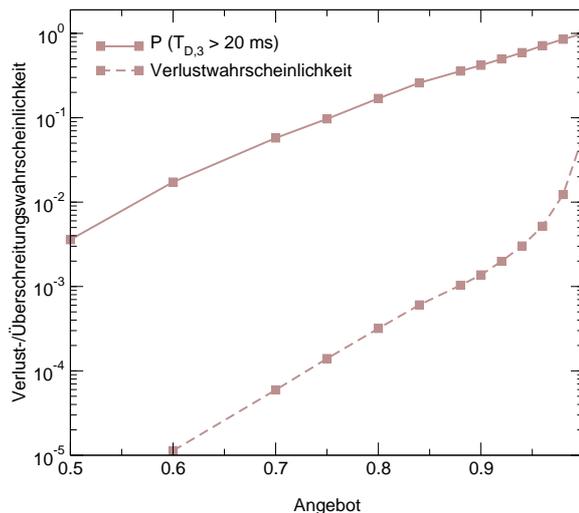
Um die auftretenden Effekte besser identifizieren zu können, wird bei den Untersuchungen schrittweise vorgegangen. Zunächst wird der Fall betrachtet, dass es keine ungesättigten TCP-Quellen gibt ( $n = 0$ ). Anschließend wird gezeigt, mit welchen Auswirkungen das zusätzliche Vorhandensein solcher Quellen verbunden ist.

#### 6.3.2.1 Untersuchungen ohne ungesättigte TCP-Quellen

In den folgenden Studien wird das Gesamtangebot  $A$  nach Gleichung (6.14) verändert, während die Angebotsanteile der einzelnen Verkehre am Gesamtangebot konstant gehalten werden (vgl. Tabelle 6.4). Zunächst werde die Verwerfungswahrscheinlichkeit betrachtet, die sich bei  $C = 10$  Mbit/s in den einzelnen Klassen in Abhängigkeit von  $A$  ergibt (Bild 6.55). Wenn auch in Bild 6.55 nur die Absolutwerte  $p_i$  dargestellt sind, so erkennt man doch am parallelen Verlauf der Kurven, dass über den ganzen Angebotsbereich hinweg ein konstantes Verhältnis  $p_i/p_j$  entsprechend dem Verhältnis der Gewichtungsfaktoren  $w_i/w_j$  aufrechterhalten werden kann. Dies bedeutet insbesondere, dass die Wahrscheinlichkeit des Überschreitens der Maximalverzögerung von 20 ms (mit anschließendem Verwerfen) in Klasse 3, also der besseren der beiden Klassen für Echtzeitverkehr, um den Faktor 100 geringer ist als die Verwerfungswahrscheinlichkeit in Klasse 0, der schlechteren der beiden Klassen für elastischen Verkehr.



**Bild 6.55:** Verwerfungswahrscheinlichkeit bei WEDD

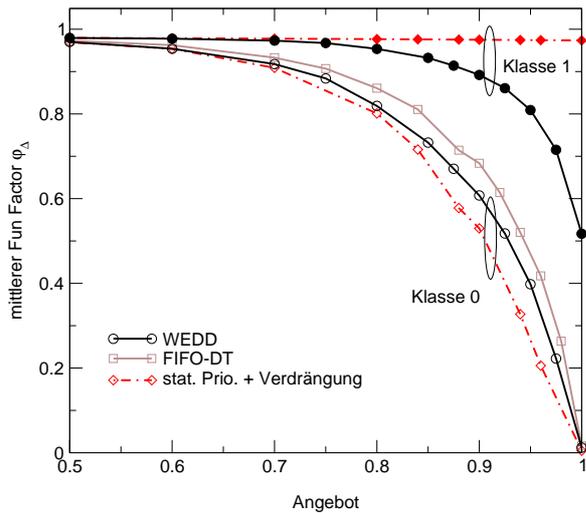


**Bild 6.56:** Verlust- und Überschreitungswahrscheinlichkeit bei FIFO-DT

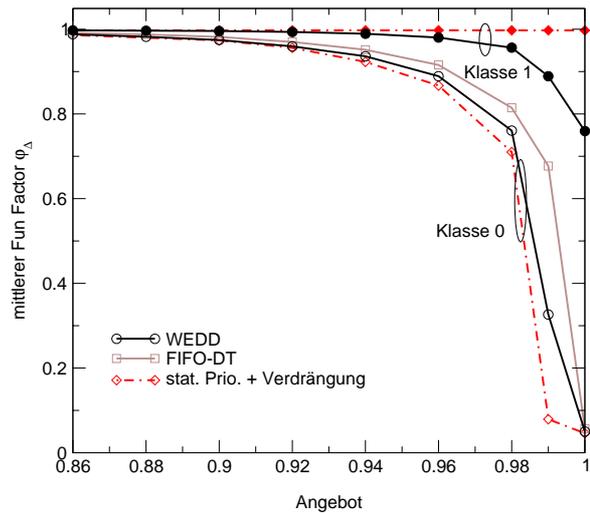
Die Ergebnisse gewinnen an Aussagekraft, wenn sie im Vergleich zu den sich ergebenden Werten im Falle eines Systems ohne differenzierende Behandlung von Paketen gesehen werden. Zu diesem Zweck sind in Bild 6.56 für Klasse 3 die Ergebnisse für die Verlustwahrscheinlichkeit sowie die Wahrscheinlichkeit, dass die Durchlaufzeit  $T_{D,3}$  mehr als 20 ms beträgt, bei FIFO-DT-Abarbeitung<sup>5</sup> aufgetragen. Zu sehen ist, dass dort bereits die Verlustwahrscheinlichkeit um etwa eine Größenordnung höher ausfällt als die Verwerfungswahrscheinlichkeit  $p_3$  im Falle von WEDD. Von den Paketen, die bei FIFO-DT nicht verloren gehen, erfährt ein großer Teil (für  $A > 0.8$  mehr als 10%) eine Verzögerung, die über 20 ms liegt. Dies würde eine Echtzeitkommunikation zumindest erschweren. Bei WEDD hingegen liegt in Klasse 3 die Verwerfungswahrscheinlichkeit selbst bei  $A = 1$  noch unterhalb von 1%. Bei geringerem Angebot ist  $p_3$  so klein, dass – je nach Anwendung – nahezu keine Auswirkungen auf die wahrnehmbare Qualität zu erwarten sind. In Klasse 2 sind die Verwerfungswahrscheinlichkeiten bei WEDD zwar um den Faktor 10 höher, liegen aber immer noch in einem akzeptablen Bereich.

Zur Betrachtung der vom TCP-Verkehr erfahrenen Leistung kann wieder der mittlere *Fun Factor*  $\phi_\Delta$  herangezogen werden. In Bild 6.57 sind neben den in den Klassen 0 und 1 erzielten Ergebnissen im Fall von WEDD auch die Werte bei FIFO-DT sowie bei statischer Priorisierung in Kombination mit Verdrängung am Puffereingang aufgetragen. Bei letztgenannter Form der Differenzierung, die als Extremfall angesehen werden kann, hat die Klasse mit höherem Index auch höhere Priorität. Bei WEDD zeigt sich, dass das Gewichtsverhältnis  $w_0/w_1 = 25$  zusammen mit den unterschiedlichen Maximalverzögerungen in den Klassen 0 und 1 eine angemessene Besserstellung von Klasse 1 gegenüber Klasse 0 über den gesamten Angebotsbe-

<sup>5</sup> Als Puffergröße wird in den Untersuchungen mit FIFO-DT sowie bei statischer Priorisierung  $S = 4/3 \cdot S_{min}$  angenommen, wobei sich  $S_{min}$  nach Gleichung (6.8) aus der größten Maximalverzögerung bei WEDD ergibt. Im Fall  $C = 10$  Mbit/s bedeutet dies  $S = 200\,000$  Byte, während bei  $C = 100$  MBit/s  $S = 400\,000$  Byte resultiert.



**Bild 6.57:** Mittlerer *Fun Factor*  $\varphi_{\Delta}$  bei  $C = 10$  Mbit/s



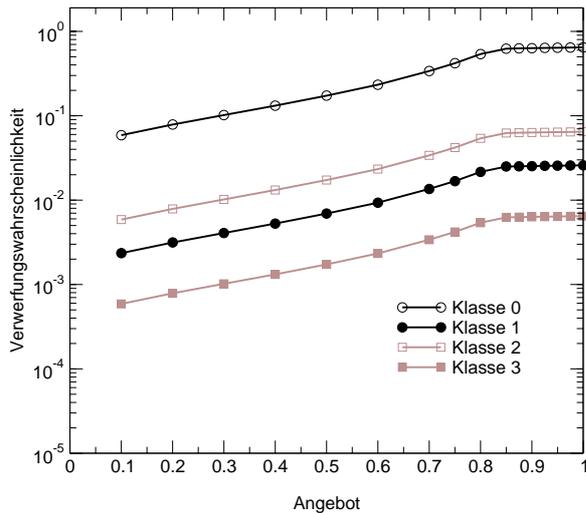
**Bild 6.58:** Mittlerer *Fun Factor*  $\varphi_{\Delta}$  bei  $C = 100$  Mbit/s

reich hinweg bewirkt. Dadurch werden die aus den Untersuchungen in Abschnitt 6.2.3 gewonnenen Erfahrungen bestätigt. Im Gegensatz zur extremen Priorisierung ist dennoch in Klasse 1 ein Rückgang von  $\varphi_{\Delta}$  auf einen Wert von 0.5 zu beobachten, wenn das Angebot gegen 1 geht. Allerdings steht dem nicht nur die bessere Leistung in Klasse 0 entgegen. Die Leistungseinbußen in Klasse 1 können auch nützlich sein, da auf diese Weise auch Nutzer mit höher priorisiertem Verkehr die Überlast zu spüren bekommen, wenn auch nur in schwachem Maße. Die Reaktion darauf könnte sein, dass diese Nutzer – wie ohnehin Nutzer in der niedrigsten Klasse – den erzeugten Verkehr drosseln (z. B. durch weniger Seitenabrufe), was dann wiederum zur Auflösung der Überlastsituation beiträgt.

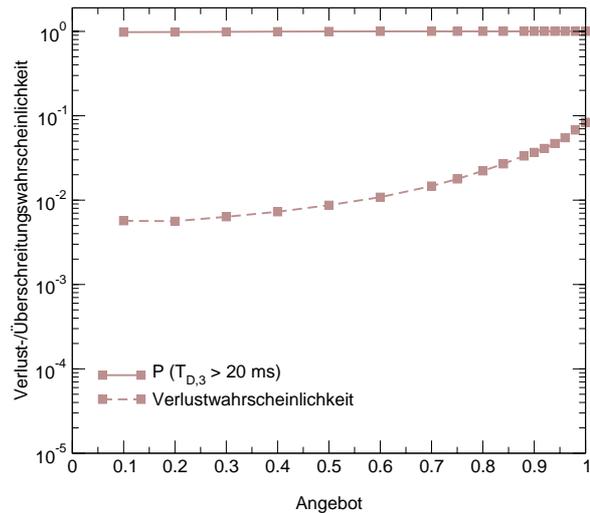
Ähnliche Ergebnisse liegen vor, wenn der Fall  $C = 100$  Mbit/s anstelle von  $C = 10$  Mbit/s betrachtet wird (Bild 6.58). Dabei ist jedoch der veränderte Maßstab auf der x-Achse zu beachten. Dies bedeutet, dass der Rückgang des mittleren *Fun Factor* aufgrund des Bündelungsgewinns erst bei einem höheren Angebot auftritt. Ein weiterer Unterschied liegt darin, dass  $\varphi_{\Delta}$  bei Annäherung an  $A = 1$  nur auf einen Wert von etwa 0.8 abfällt.

### 6.3.2.2 Untersuchungen mit ungesättigten TCP-Quellen

Nun wird der Fall betrachtet, dass zum Verkehr in den Klassen 0 und 1 außer dem dynamischen TCP-Verkehr auch noch ungesättigte TCP-Quellen beitragen. Die Gesamtzahl  $n$  dieser Quellen beträgt 20. Diese teilen sich, wie in Tabelle 6.4 beschrieben, auf die Klassen 0 und 1 auf. Die Anwesenheit ungesättigter Quellen bewirkt zunächst eine deutliche Erhöhung der Verwerfungswahrscheinlichkeiten infolge des WEDD/LPD-Mechanismus, wie in Bild 6.59 zu sehen ist. Dort wird wie im vorigen Abschnitt das Angebot  $A$ , das durch Echtzeitverkehr sowie dynamischen TCP-Verkehr zustande kommt, variiert, während  $n$  konstant bleibt. Auffallend ist in Bild 6.59 neben der generellen Erhöhung von  $p_i$  im Vergleich zum Fall  $n = 0$ ,



**Bild 6.59:** Verwerfungswahrscheinlichkeit bei WEDD im Fall mit ungesättigten TCP-Quellen



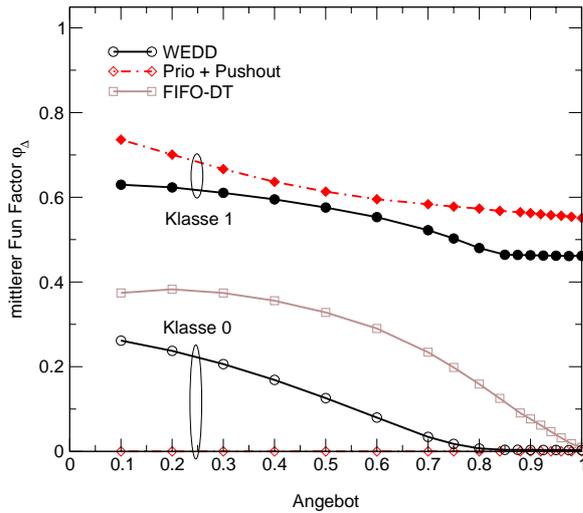
**Bild 6.60:** Verlust- und Überschreitungswahrscheinlichkeit bei FIFO-DT im Fall mit ungesättigten TCP-Quellen

das die Verwerfungswahrscheinlichkeiten ab einem Angebot von etwa 0.8 konstant bleiben. Dies liegt daran, dass ab diesem Punkt bereits die Begrenzung der Gesamtzahl dynamischer TCP-Verbindungen auf  $n_{max} = 5\,000$  greift. Dies wiederum hat seine Ursache darin, dass die für die TCP-Verbindungen in Klasse 0 zur Verfügung stehende Rate nicht nur durch die in  $A$  enthaltenen Verkehrsanteile der anderen Klassen, sondern zusätzlich durch die  $n_1 = 5$  statischen TCP-Verbindungen in Klasse 1 reduziert wird. Der mittlere Durchsatz dieser 5 TCP-Verbindungen wird dabei durch die Verwerfungswahrscheinlichkeit in Klasse 1 bestimmt, die um den Faktor  $w_0/w_1 = 25$  geringer ist als die in Klasse 0 und daher selbst im Fall  $p_0 = 1$  höchstens 0.04 betragen würde. Mit Hilfe des in [212] gefundenen Zusammenhangs kann näherungsweise der mittlere Durchsatz  $g_1$  pro Quelle aus  $p_1$  berechnet werden. Aus  $p_1 \approx 0.04$  folgt dann mit  $\overline{T_{RTT}} \approx \tau$  ein Nutzdurchsatz von  $g_1 \approx 392$  kbit/s. Damit lässt sich aus

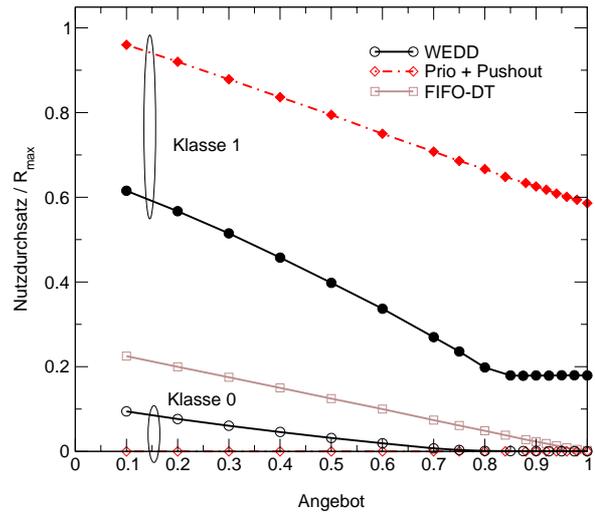
$$A_s = 1 - \frac{n_1 \cdot g_1}{C} \tag{6.15}$$

das Angebot errechnen, bei dem in Klasse 0 die Sättigungsgrenze erreicht wird. Mit den oben genannten Werten kommt man dann auf  $A_s = 0.804$ , was in guter Übereinstimmung mit Bild 6.59 steht.

Der Vergleich mit einem FIFO-DT-System fällt hier noch drastischer aus. Dort erfahren nahezu alle Pakete aus Klasse 3, die nicht schon am Eingang verworfen werden, eine Verzögerung, die größer ist als 20 ms (Bild 6.60). Anwendungen, bei denen eine größere Verzögerung nicht toleriert werden kann, können somit bei Anwesenheit von ungesättigten TCP-Quellen nicht genutzt werden.



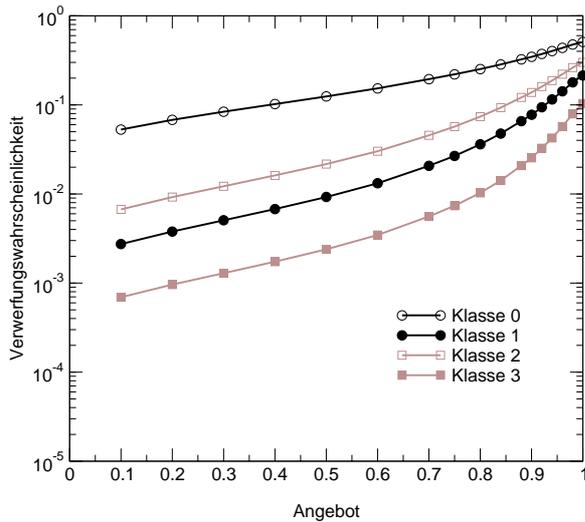
**Bild 6.61:** Mittlerer *Fun Factor*  $\varphi_{\Delta}$  für dynamischen TCP-Verkehr im Fall mit ungesättigten TCP-Quellen bei  $C = 10$  Mbit/s



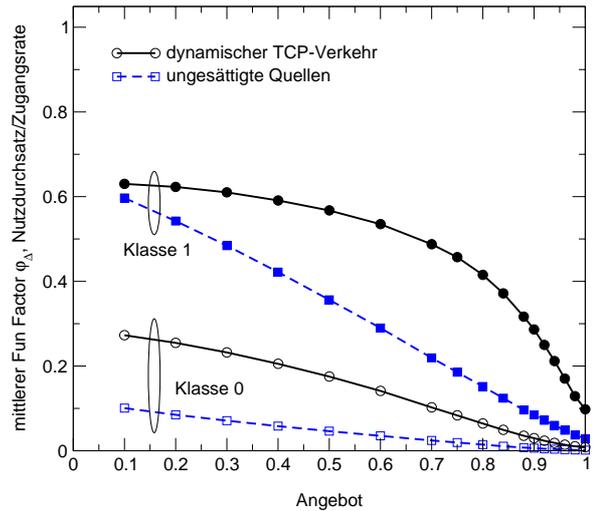
**Bild 6.62:** Normierter Nutzdurchsatz pro Verbindung bei ungesättigten TCP-Quellen bei  $C = 10$  Mbit/s

Eine deutliche Veränderung gegenüber dem Fall mit  $n = 0$  zeigt sich auch beim Blick auf den mittleren *Fun Factor*  $\varphi_{\Delta}$ , den der dynamische TCP-Verkehr erfährt (Bild 6.61). Dieser ist nun in beiden Klassen wesentlich geringer als bei der Untersuchung ohne ungesättigte TCP-Quellen. Schon bei geringem Angebot erreicht  $\varphi_{\Delta}$  im Fall von WEDD nur Werte von knapp über 0.6 in Klasse 1 und etwa 0.25 in Klasse 0. Während in Klasse 1 mit steigendem Angebot nur ein leichter Rückgang auf einen Wert von ca. 0.46 zu verzeichnen ist, erfolgt in Klasse 0 ein näherungsweise linearer Abfall bis zum Sättigungspunkt (Angebot  $A_s$ ), an dem der mittlere *Fun Factor* nur noch unmerklich über 0 liegt. Das beschriebene Verhalten erweist sich sowohl im Vergleich mit FIFO-DT als auch mit der statischen Priorisierung als vorteilhaft. Denn im Gegensatz zu einem FIFO-DT-System ist in Klasse 1 mit wachsendem Angebot kein drastisches Einbrechen der Leistung verbunden. In Klasse 0 ist bei WEDD die erfahrene Leistung zwar gering; es kommt jedoch nicht wie bei extremer Priorisierung zu einem völligen Aushungern dieser Klasse, verursacht durch die ungesättigten Quellen in Klasse 1.

Betrachtet man den auf die Zugangsrate  $R_{max}$  normierten mittleren Nutzdurchsatz  $g_i$  pro Verbindung für die ungesättigten TCP-Quellen selbst (Bild 6.62), so ergibt sich ein ähnliches Bild wie beim dynamischen TCP-Verkehr. In Klasse 1 ist allerdings der Unterschied zwischen den Werten bei WEDD und denen bei statischer Priorisierung größer. Jedoch fällt die Verbesserung bei WEDD im Vergleich zum Fall ohne Differenzierung immer noch deutlich aus. Außerdem macht es gerade der bei WEDD geringere Nutzdurchsatz für diesen Verkehr möglich, dass in Klasse 0 die Leistung nicht völlig einbricht. Dies zeigt sich auch in Bild 6.62 – wie schon im Fall des dynamischen Verkehrs – durch einen erst bei etwa  $A = 0.8$  verschwindenden mittleren Nutzdurchsatz  $g_0$ .



**Bild 6.63:** Verwerfungswahrscheinlichkeit bei WEDD mit CSA



**Bild 6.64:** Mittlerer *Fun Factor* bzw. normierter Nutzdurchsatz bei WEDD mit CSA

Eine Option, die bei den Untersuchungen mit ungesättigten Quellen in Abschnitt 6.2.2 erfolgreich eingesetzt wurde, ist die CSA-Erweiterung von WEDD, die das Ziel hat, ein völliges Aushungern in der Klasse mit dem höchsten Verwerfungsgewicht  $w_i$  zu verhindern. Wird diese Option mit  $\gamma = 5$  auch im vorliegenden Szenario verwendet, ergeben sich zunächst die in Bild 6.63 dargestellten Verwerfungswahrscheinlichkeiten. Dabei ist zu sehen, dass sich die Kurven wie erwartet mit zunehmendem Angebot annähern. Vor allem für den Echtzeitverkehr in Klasse 3 bedeutet dies eine deutliche Verschlechterung gegenüber dem Fall ohne CSA.

Positive Auswirkungen von CSA werden erst beim Blick auf die TCP-Leistungsmetriken in den Klassen 0 und 1 deutlich. Sowohl für den mittleren *Fun Factor*  $\phi_A$  als auch für den auf die Zugangsrate bezogenen mittleren Nutzdurchsatz in Klasse 0 ergeben sich nun bis zu einem Angebot von 1 nicht verschwindende Werte. Dafür kommt es nun jedoch bei einem Angebot  $A > 0.8$  zu einem Leistungsabfall in Klasse 1, der sich für dynamischen TCP-Verkehr ebenso wie für ungesättigte TCP-Quellen bemerkbar macht. Somit ist es eher fraglich, ob angesichts dieser Nachteile die Verbesserungen in Klasse 0 die Verwendung von CSA noch rechtfertigen.

# Kapitel 7

## Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurde ein Verfahren zur relativen Differenzierung von Verkehrsströmen innerhalb von IP-Netzknoten vorgeschlagen und untersucht. Zunächst wurden dazu im zweiten Kapitel wichtige Protokolle und Anwendungen in IP-Netzen vorgestellt und klassifiziert. Die Anwendungen unterscheiden sich nicht nur in Bezug auf das Volumen und die Charakteristik des von ihnen erzeugten Verkehrs, sondern stellen auch teilweise sehr unterschiedliche Anforderungen an die Übertragung von Paketen in einem IP-Netz. Insbesondere kann dabei eine Unterteilung erfolgen in Verfahren mit Echtzeitanforderungen und solche, bei denen die gesicherte Übertragung im Vordergrund steht.

Die unterschiedlichen Anforderungen können mit Hilfe des Begriffes der Dienstgüte ausgedrückt werden, der in Kapitel 3 vor dem Hintergrund von IP-Netzen eingehend behandelt wurde. Dort wurden Klassifizierungsmerkmale von Dienstgüte ebenso beschrieben wie die verschiedenen Verkehrsmanagementfunktionen, die innerhalb eines Netzes zur Bereitstellung von Dienstgüte eingesetzt werden können. Darüber hinaus wurde ein Überblick über Ansätze und Architekturen zur Dienstgüteunterstützung in IP-Netzen gegeben. Dabei wurden auch die Vorteile der DiffServ-Architektur deutlich, die keine Reservierung von Netzressourcen für einzelne Verkehrsflüsse, sondern eine Differenzierung aggregierter Verkehrsströme vorsieht.

Eine mögliche Form der Umsetzung der DiffServ-Rahmenarchitektur ist die Bereitstellung relativer Garantien zwischen einzelnen Klassen unter Verzicht auf Verkehrsmanagementfunktionen wie Quellflusskontrolle und Verbindungsannahmesteuerung. Stattdessen müssen geeignete Scheduling- und Puffermanagementmechanismen gefunden werden, die Verkehrsströme in den IP-Netzknoten beim Zugriff auf die dort vorhandenen Ressourcen Bandbreite und Speicherplatz unterschiedlich behandeln. Aufbauend auf Kapitel 4, wo unter anderem ein Überblick über die in IP-Netzen relevanten Leistungsmetriken gegeben wurde, erfolgte im fünften Kapitel die Vorstellung eines neuen Verfahrens, das eine solche relative Differenzierung mit Hilfe eines integrierten Scheduling und Puffermanagements ermöglicht. Dieses als WEDD

bezeichnete Verfahren stellt eine Erweiterung des bekannten EDD-Scheduling-Mechanismus dar, bei der jedoch nicht nur eine Unterscheidung hinsichtlich der maximalen Verzögerung von IP-Paketen einer Klasse erfolgt, sondern auch eine Differenzierung bzgl. der Wahrscheinlichkeit einer Überschreitung des vorgegebenen Maximalwerts. Der Grad dieser Differenzierung, in diesem Fall also das Verhältnis der Überschreitungswahrscheinlichkeiten, wird durch die Angabe von Gewichtungsparemtern für die einzelnen Klassen eingestellt, womit das Verfahren dem aus der Literatur bekannten Modell der proportionalen Differenzierung folgt.

In Kapitel 5 wurde außerdem gezeigt, wie das WEDD-Verfahren realisiert werden kann. Dabei spielt die Abschätzung der Häufigkeit von in der Vergangenheit aufgetretenen Überschreitungseignissen in den einzelnen Klassen eine wichtige Rolle. In diesem Zusammenhang wurden mehrere Lösungsalternativen vorgeschlagen, die mit Hilfe verschiedener Zähler arbeiten. Darüber hinaus wurden einige Realisierungsoptionen von WEDD vorgeschlagen. Dazu gehören insbesondere der LPD-Mechanismus, der dafür sorgt, dass Pakete, bei denen die Maximalverzögerung überschritten wurde, bereits im Netzknoten verworfen werden, sowie die CSA-Erweiterung, die ein Aushungern von Klassen mit geringer Priorität verhindert. Schließlich wurden in Kapitel 5 auch Aspekte der Implementierung sowie des Einsatzes in einem Netzzenario betrachtet.

Ein Schwerpunkt dieser Arbeit stellt die Leistungsuntersuchung dar. Hierzu wurden zunächst in Kapitel 4 die Vor- und Nachteile der verschiedenen zur Verfügung stehenden Untersuchungsmethoden – insbesondere von Analyse und Simulation – bei einer Anwendung im Kontext von IP-Netzen gegeneinander abgewogen. Außerdem wurde in diesem Kapitel ein Überblick über relevante Verkehrsmodelle für IP-Verkehr gegeben, wobei verschiedene Modellierungsebenen unterschieden wurden. Dabei wurden vor allem auch Modelle betrachtet, in denen die Überlaststeuerungsmechanismen des Transportprotokolls TCP berücksichtigt werden. Diese bewirken eine Anpassung des erzeugten Verkehrs an den aktuellen Netzzustand, die sich mit einfachen Paket- oder Burstebenenmodellen nicht nachbilden lässt. Außer der Transportprotokollebene selbst kommt auch der Modellierung der Verkehrslast, die durch Anwendungen oberhalb von TCP erzeugt wird, eine große Bedeutung zu. Hier gilt es, Modelle zu finden, die einerseits repräsentativ und andererseits möglichst einfach sind, damit die Haupteffekte zur Geltung kommen. In Kapitel 4 wurden neben dem häufig verwendeten Modell mit ungesättigten TCP-Quellen auch Modelle für dynamischen TCP-Verkehr mit eher kurzen TCP-Verbindungen, wie sie bei WWW-Verkehr typisch sind, betrachtet.

Neben der Verkehrsmodellierung spielen auch die für die Leistungsuntersuchung verwendeten Metriken eine wichtige Rolle. Um aussagekräftige Ergebnisse zu erhalten, müssen wie bei der Verkehrsmodellierung verschiedene Ebenen unterschieden werden. In Kapitel 4 wurden daher neben den Paketebenenmaßen auch Metriken vorgestellt, die sich auf die Transportprotokollebene beziehen. Dazu gehört z. B. der für die Bewertung der Leistung bei dynamischem TCP-

Verkehr relevante *Fun Factor*, bei dem die Dauer der Übertragung eines Datenbursts mittels TCP in Relation zu einer analytisch bestimmbar minimalen Transferzeit (siehe Anhang A) gesetzt wird.

Die Ergebnisse der simulativen Leistungsuntersuchung eines IP-Netzknotens, in dem der in Kapitel 5 vorgeschlagene WEDD-Mechanismus zur Differenzierung von Verkehrsströmen eingesetzt wird, wurden im sechsten Kapitel präsentiert. Dabei wurde nur der Zugang zu einem Ausgangslink des Netzknotens betrachtet, der einen Engpass darstellt. Zunächst wurde angenommen, dass nur Echtzeitverkehr vorhanden ist, der mit Hilfe eines Burstebenenmodells charakterisiert wurde (Abschnitt 6.1). Für verschiedene Verkehrsparameter konnte nachgewiesen werden, dass WEDD seinen Anforderungen gerecht wird und eine Verzögerungsdifferenzierung unter Einhaltung des vorgegebenen Verhältnisses der Fristüberschreitungswahrscheinlichkeiten erreichen kann. Dies gilt für das Langzeitverhalten ebenso wie für das Verhalten innerhalb kurzer Zeitintervalle. Im Weiteren wurden der Einfluss verschiedener Realisierungsoptionen und Parameter von WEDD untersucht. Unter anderem konnte gezeigt werden, dass das Verwerfen verspäteter Pakete eine deutliche Leistungssteigerung zur Folge hat. Schließlich wurde in Abschnitt 6.1 der im Hinblick auf eine Realisierung ebenfalls wichtigen Frage nach der benötigten Puffergröße nachgegangen. Hier wurde zwar festgestellt, dass ein Verwerfen von Paketen am Pufferausgang eine gewisse Überdimensionierung des Puffers erfordert. Allerdings bewegt sich die zusätzlich erforderliche Kapazität selbst bei ungünstigen Lastbedingungen in einem vertretbaren Rahmen.

Im Weiteren wurden in Abschnitt 6.2 Untersuchungen von WEDD mit TCP-Verkehr durchgeführt, wobei entweder von ungesättigten TCP-Quellen oder dynamischem Verkehr mit kurzen TCP-Verbindungen ausgegangen wurde. In beiden Fällen konnte dabei Bezug auf die in Anhang B beschriebenen Studien für einen Netzknoten ohne Differenzierung genommen werden, die bereits wesentliche bei TCP-Verkehr auftretende Effekte aufzeigen. Darüber hinaus konnte festgestellt werden, dass WEDD im Fall ungesättigter Quellen mit guter Näherung eine proportionale Differenzierung hinsichtlich des Nutzdurchsatzes ermöglicht, wenn mit Hilfe der CSA-Erweiterung das Aushungern von nieder priorisierten Klassen vermieden wird. Bei dynamischem Verkehr hingegen kann zwar der Grad der Differenzierung nicht mehr direkt über die WEDD-Parameter eingestellt werden, da der Zusammenhang zwischen Verzögerungen und Verlusten einerseits und relevanten Maßen wie dem mittleren *Fun Factor* andererseits sehr komplex ist. Jedoch konnte durch zahlreiche Studien der prinzipielle Einfluss dieser Parameter auf die Unterschiede zwischen den Klassen ermittelt werden. Dabei wurde herausgefunden, dass für eine signifikante Differenzierung unterschiedliche Werte für die Maximalverzögerungen in den einzelnen Klassen eingestellt werden müssen. Sowohl für den Fall ungesättigter TCP-Quellen als auch für dynamischen TCP-Verkehr wurden Vergleiche mit etablierten Verfahren vorgenommen. Dabei konnte nachgewiesen werden, dass WEDD mindestens

ebenso gut für die Differenzierung von elastischem Verkehr geeignet ist wie jene Verfahren, die teilweise speziell auf einen Einsatz bei TCP-Verkehr ausgerichtet sind.

Ein großer Vorteil von WEDD besteht darin, dass es sich um ein integriertes Verfahren handelt, das eine Differenzierung sowohl im Hinblick auf Verzögerungen als auch auf Verluste ermöglicht. Dies wurde in Abschnitt 6.3 am Beispiel eines integrierten Szenarios mit jeweils zwei Klassen für Echtzeit- und TCP-Verkehr demonstriert. Dabei war einerseits eine deutliche Verbesserung – vor allem für Echtzeitverkehr – im Vergleich zum undifferenzierten Fall zu erkennen. Andererseits konnte gezeigt werden, dass bei entsprechender Einstellung der Parameter ein komplettes Aushungern von niederprioren Klassen wie im Fall einer statischen Priorisierung verhindert werden kann.

Die präsentierten Studien konnten trotz des großen Umfangs nur einen kleinen Teil des riesigen Parameterraums abdecken. In weiteren Untersuchungen könnten vor allem Verkehrsmodelle und deren Parameter sowie die Verkehrszusammensetzung variiert werden. Als Beispiele sind Modelle insbesondere für Echtzeitverkehr, die auf bestimmte Anwendungen zugeschnitten sind, oder Szenarien mit heterogenem Verkehr in jeder Klasse zu nennen.

Darüber hinaus kann die Betrachtung eines Netzszenarios mit mehreren Knoten, in denen ein differenzierendes Scheduling- und Puffermanagementverfahren eingesetzt wird, zusätzliche Erkenntnisse bringen. Dabei stellt sich zum einen die Frage, wie sich Parametereinstellungen auf die Ende-zu-Ende-Dienstgüte auswirken. Zum anderen spielt das Zusammenwirken mit Verkehrslenkungsverfahren eine Rolle, bei denen Dienstgüteanforderungen berücksichtigt werden. Schließlich ist im Rahmen eines Netzszenarios auch die Betrachtung von hybriden Ansätzen im Sinne von Abschnitt 3.3.4 denkbar, bei denen in verschiedenen Bereichen unterschiedliche Architekturen zur Dienstgüteunterstützung zur Anwendung kommen.

Die Grundidee von WEDD, nämlich eine Aufteilung des Gesamtverkehrs in Klassen mit unterschiedlichen Verzögerungsanforderungen und gleichzeitig eine proportionale Differenzierung der Wahrscheinlichkeiten einer Überschreitung der maximalen Verzögerung vorzunehmen, ist prinzipiell auch auf andere Anwendungsgebiete übertragbar, bei denen es um die Zuteilung knapper Ressourcen geht. Als Beispiele können hier der Zugriff auf Rechnerressourcen in einem WWW-Server oder auch die Realisierung von Überlastabwehrmechanismen bei der Bearbeitung von Steuermeldungen in einem Vermittlungsknoten genannt werden.

# Literaturverzeichnis

- [1] E. AARSTAD, L. BURGSTÄHLER, M. LORANG: “Description of the flow-to-VC mapping control module in DIANA’s RSVP over ATM architecture.” *Proceedings of the QoS Summit ‘99*, Paris, Nov. 1999.
- [2] A. ABOUZEID, S. ROY: “Analytic understanding of RED gateways with multiple competing TCP flows.” *Proceedings of IEEE GLOBECOM 2000*, San Francisco, Nov. 2000.
- [3] R. ADDIE, P. MANNERSALO, I. NORROS: “Performance formulae for queues with Gaussian input.” *Proceedings of the 16th International Teletraffic Congress (ITC 16)*, Edinburgh, June 1999, pp. 1169-1178.
- [4] M. ALLMAN, A. FALK: “On the effective evaluation of TCP.” *Computer Communications Review*, Vol. 29, No. 5, Oct. 1999, pp. 59-70.
- [5] M. ALLMAN, V. PAXSON, W. STEVENS: *TCP congestion control*, RFC 2581, IETF, April 1999.
- [6] J. ALMEIDA, M. DABU, A. MANIKUTTY, P. CAO: “Providing differentiated levels of service in web content hosting.” *Proceedings of the 1998 SIGMETRICS Workshop on Internet Server Performance (WISP ‘98)*, Madison, June 1998.
- [7] E. ALTMAN, K. AVRACHENKOV, C. BARAKAT: “A stochastic model of TCP/IP with stationary random losses.” *Proceedings of ACM SIGCOMM 2000*, Stockholm, Aug. 2000, pp. 231-242.
- [8] D. ANICK, D. MITRA, M. SONDHI: “Stochastic theory of a data-handling system with multiple sources.” *Bell System Technical Journal*, Vol. 61, No. 8, Oct. 1982, pp. 1871-1894.
- [9] G. APOSTOLOPOULOS, D. WILLIAMS, S. KAMAT, R. GUÉRIN, A. ORDA, T. PRZYGIENDA: *QoS routing mechanisms and OSPF extensions*, IETF, RFC 2676, Aug. 1999.
- [10] C. M. ARAS, J. F. KUROSE, D. S. REEVES, H. SCHULZRINNE: “Real-Time communication in packet-switched networks.” *Proceedings of the IEEE*, Vol. 82, No. 1, Jan. 1994, pp. 122-139.
- [11] G. ARMITAGE, K. M. ADAMS: “Packet reassembly during cell loss.” *IEEE Network*, Vol. 7, No. 5, Sep. 1993, pp. 26-34.
- [12] A. ARVIDSSON, P. KARLSSON: “On traffic models for TCP/IP.” *Proceedings of the International Teletraffic Congress (ITC 16)*, Edinburgh, June 1999, pp. 457-466.

- [13] S. ATHURALIYA, S. H. LOW, V. H. LI, Q. YIN: “REM: active queue management.” *IEEE Network*, Vol. 15, No. 3, May. 2001, pp. 48-53.
- [14] ATM FORUM: *LAN emulation over ATM version 1.0*, af-lane-0021.000, The ATM Forum, Jan. 1995.
- [15] ATM FORUM: *Private network-network interface specification version 1.0 (PNNI 1.0)*, af-pnni-0055.000, The ATM Forum, March 1996.
- [16] ATM FORUM: *Traffic management specification version 4.1*, af-tm-0121.000, The ATM Forum, March 1999.
- [17] ATM FORUM: *Multi-protocol over ATM version 1.1*, af-mpoa-0114.000, The ATM Forum, May 1999.
- [18] G. ASH: *Dynamic routing in telecommunications networks*, McGraw-Hill, New York, 1998.
- [19] D. O. AWDUCHE, L. BERGER, D.-H. GAN, T. LI, V. SRINIVASAN, G. SWALLOW: *RSVP-TE: extensions to RSVP for LSP tunnels*, Internet Draft draft-ietf-mpls-rsvp-lsp-tunnel-08.txt, IETF, Feb. 2001, work in progress.
- [20] S. BAJAJ, L. BRESLAU, S. SHENKER: “Uniform versus priority dropping for layered video.” *Proceedings of ACM SIGCOMM '98*, Vancouver, Sep. 1998, pp. 131-143.
- [21] A. BANERJEE, J. DRAKE, J. P. LANG, B. TURNER, K. KOMPPELLA, Y. REKHTER: “Generalized multiprotocol label switching: An overview of routing and management enhancements.” *IEEE Communications Magazine*, Vol. 39, No. 1, Jan. 2001, pp. 144-150.
- [22] C. BARAKAT, E. ALTMAN: “A Markovian model for TCP analysis in a differentiated services network.” *Proceedings of the 1st International Workshop on Quality of Future Internet Services (QofIS 2000)*, Berlin, Sep. 2000, pp. 55-67.
- [23] P. BARFORD, M. CROVELLA: “Generating representative web workloads for networks and server performance evaluation.” *Proceedings of ACM SIGMETRICS '98*, Madison, June 1998, pp. 151-160.
- [24] S. BEN FREDJ, T. BONALD, A. PROUTIERE, G. REGNIE, J. W. ROBERTS: “Statistical bandwidth sharing: a study of congestion at flow level.” *Proceedings of ACM SIGCOMM 2001*, San Diego, CA, Aug. 2001.
- [25] J. C. R. BENNETT, H. ZHANG: “WF<sup>2</sup>Q: worst-case fair weighted fair queueing.” *Proceedings of IEEE INFOCOM '96*, San Francisco, March 1996, pp. 120-128.
- [26] J. C. R. BENNETT, H. ZHANG: “Hierarchical packet fair queueing algorithms.” *IEEE/ACM Transactions on Networking*, Vol. 5, No. 5, Oct. 1997, pp. 675-689.
- [27] S. BLAKE, D. BLACK, M. CARLSON, E. DAVIES, Z. WANG, W. WEISS: *An architecture for differentiated services*, RFC 2475, IETF, Dec. 1998.

- [28] E. BLANTON, M. ALLMAN: “On making TCP more robust to packet reordering.” *Computer Communication Review*, Vol. 32, No. 1, Jan. 2002, pp. 20 -30.
- [29] S. BODAMER: *Charging in multi-service networks*, Internal Report No. 29, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1998.
- [30] S. BODAMER: “A scheduling mechanism for relative delay differentiation.” *Proceedings of the IEEE Conference on High Performance Switching and Routing (ATM 2000)*, Heidelberg, June 2000, pp. 357-364.
- [31] S. BODAMER, J. CHARZINSKI: “Evaluation of effective bandwidth schemes for self-similar traffic.” *Proceedings of the 13th ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management*, Monterey, CA, Sep. 2000, pp. 21-1-21-10.
- [32] S. BODAMER, J. CHARZINSKI, K. DOLZER: „Dienstgütemetriken für elastischen Internetverkehr.“ *PIK*, Band 25, Nr. 2, April-Juni 2002, S. 82-89.
- [33] S. BODAMER, K. DOLZER: *QoS metrics for elastic IP traffic*, Internal Report No. 37, IND, University of Stuttgart, June 2001.
- [34] S. BODAMER, K. DOLZER, C. GAUGER: *IND simulation library 2.3 user guide – Part I: introduction*, Institute of Communication Networks and Computer Engineering, University of Stuttgart, April 2002.
- [35] S. BODAMER, M. LORANG: *IND TCP Library 1.1 user guide*, Institute of Communication Networks and Computer Engineering, University of Stuttgart, March 2002.
- [36] S. BODAMER, T. RENGER, G. RÖSSLER: “Traffic management in an ATM multi-service switch for workgroups.” *Proceedings of the IFIP 4th International Conference on Broadband Communications (BC '98)*, Stuttgart, April 1998, pp. 341-355.
- [37] U. BODIN, A. JONSSON, O. SCHELEN: “On creating proportional loss-rate differentiation: predictability and performance.” *Proceedings of the 9th International Workshop on Quality of Service (IWQoS 2001)*, Karlsruhe, June 2001, pp. 372-386.
- [38] U. BODIN, O. SCHELEN, S. PINK: “Load-tolerant differentiation with active queue management.” *Computer Communication Review*, Vol. 30, No. 3, July 2000, pp. 4-15.
- [39] R. BOHN, H.-W. BRAUN, K. C. CLAFFY, S. WOLFF: “Mitigating the coming Internet crunch: multiple service levels via precedence.” *Journal of High Speed Networks*, Vol. 3, No. 4, 1994.
- [40] T. BONALD, A. PROUTIERE, G. RÉGNIÉ, J. W. ROBERTS: “Insensitivity results in statistical bandwidth sharing.” *Proceedings of the 17th International Teletraffic Congress (ITC 17)*, Salvador da Bahia, Brazil, Dec. 2001, pp. 125-136.
- [41] T. BONALD, J. ROBERTS: “Performance modeling of elastic traffic in overload.” *Proceedings of ACM SIGMETRICS 2001*, Cambridge, MA, June 2001, pp. 342-343.

- [42] S. C. BORST, D. MITRA: “Virtual partitioning for robust resource sharing: computational techniques for heterogeneous traffic.” *IEEE Journal on Selected Areas in Communications*, Vol. 16, No. 5, June 1998, pp. 668-678.
- [43] D. BOTVICH, Y. CHEN, T. CURRAN, B. KERSWELL, J. MCGIBNEY, D. MORRIS: “On charging for Internet services provided over an ATM network.” *Proceedings of the IEEE ATM '97 Workshop*, Lisboa, May 1997, pp. 669-679.
- [44] A. BOUCH, M. A. SASSE: “Network quality of service: what do users need?” *Proceedings of the 4th International Distributed Conference (IDC '99)*, Madrid, Sep. 1999, pp. 78-90.
- [45] R. BRADEN: *Requirements for Internet hosts – communication layers*, RFC 1122, IETF, Oct. 1989.
- [46] B. BRADEN, D. CLARK, J. CROWCROFT, B. DAVIE, S. DEERING, D. ESTRIN, S. FLOYD, V. JACOBSON, G. MINSHALL, C. PARTRIDGE, L. PETERSON, K. RAMAKRISHNAN, S. SHENKER, J. WROCLAWSKI, L. ZHANG: *Recommendations on queue management and congestion avoidance in the Internet*, RFC 2309, IETF, April 1998.
- [47] R. BRADEN, D. CLARK, S. SHENKER: *Integrated services in the Internet architecture: an overview*, RFC 1633, IETF, July 1994.
- [48] R. BRADEN, L. ZHANG, S. BERSON, S. HERZOG, S. JAMIN: *Resource ReSerVation Protocol (RSVP)*, RFC 2205, IETF, Sep. 1997.
- [49] L. BRESLAU, D. ESTRIN, K. FALL, S. FLOYD, J. HEIDEMANN, A. HELMY, P. HUANG, S. MCCANNE, K. VARADHAN, Y. XU, H. YU: “Advances in network simulation.” *IEEE Computer*, Vol. 33, No. 5, May 2000, pp. 59-67.
- [50] L. BRESLAU, S. SHENKER: “Best-effort versus reservations: a simple comparative analysis.” *Proceedings of ACM SIGCOMM '98*, Vancouver, Sep. 1998, pp. 3-16.
- [51] F. BRICHET, M. MANDJES, M.F. SANCHEZ-CANABATE: “Admission control in multiservice networks.” *Proceedings of the COST 257 Mid-Term Seminar*, Vilamoura, Jan. 1999.
- [52] F. BRICHET, J. ROBERTS, A. SIMONIAN, D. VEITCH: “Heavy traffic analysis of a storage model with long range dependent on/off sources.” *Queueing Systems*, Vol. 23, Nos. 1-4, 1996, pp. 197-215.
- [53] U. BRIEM: *Verbindungslose Datenkommunikation über ATM-Weitverkehrsnetze: Architekturen, Protokolle und Verkehrsleistung*, Dissertation, Universität Stuttgart, 1998.
- [54] P. BROWN: “Resource sharing of TCP connections with different round trip times.” *Proceedings of IEEE INFOCOM 2000*, Tel Aviv, March 2000.
- [55] L. BURGSTAHLER: *Definition und Vergleich verschiedener Aggregationsverfahren*, Interner Bericht Nr. 32, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, März 2000.

- [56] N. CARDWELL, S. SAVAGE, T. ANDERSON: “Modeling TCP latency.” *Proceedings of IEEE INFOCOM 2000*, Tel Aviv, March 2000, pp. 1742-1751.
- [57] G. CARLE, M. SMIRNOV, T. ZSEBY: “Charging and accounting for IP multicast integrated services over ATM.” *Proceedings of the 4th International Symposium on Networking (Internetworking '98)*, Ottawa, July 1998.
- [58] T. L. CASAVANT, J. G. KUHL: “A taxonomy of scheduling in general-purpose distributed computing systems.” *IEEE Transactions on Software Engineering*, Vol. 14, No. 2, Feb. 1988, pp. 141-154.
- [59] C. CASETTI, M. MEO: “A new approach to model the stationary behavior of TCP connections.” *Proceedings of IEEE INFOCOM 2000*, Tel Aviv, March 2000, pp. 367-375.
- [60] H. J. CHAO, H. CHENG, Y.-R. JENQ, D. JEONG: “Design of a generalized priority queue manager for ATM switches.” *IEEE Journal on Selected Areas in Communications*, Vol. 15, No. 5, June 1997, pp. 867-880.
- [61] J. CHARZINSKI: *TDMA-Medienzugriffsverfahren im Rückkanal passiver optischer ATM-Zugangsnetze*, Dissertation, Universität Stuttgart, 1999.
- [62] J. CHARZINSKI: “Fun factor dimensioning for elastic traffic.” *Proceedings of the 13th ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management*, Monterey, CA, Sep. 2000, pp. 11.1-11.9.
- [63] J. CHARZINSKI: “HTTP/TCP connection and flow characteristics.” *Performance Evaluation*, Vol. 42, Nos. 2-3, Oct. 2000, pp. 150-162.
- [64] J. CHARZINSKI: “Measured HTTP performance and fun factors.” *Proceedings of the 17th International Teletraffic Congress (ITC 17)*, Salvador da Bahia, Brazil, Dec. 2001, pp. 1063-1074.
- [65] J. CHARZINSKI: *IP based networks and applications*, Manuskript zur Vorlesung an der Universität Stuttgart, 2001.
- [66] J. CHARZINSKI, J. FÄRBER, N. VICARI: „Verkehrsmessungen und Lastmodellierung im Internet.“ *PIK*, Band 25, Nr. 2, S. 64-72, April-Juni 2002.
- [67] S. CHEN, K. NAHRSTEDT: “An overview of quality of service routing for next-generation high-speed networks: problems and solutions.” *IEEE Network*, Vol. 12, No. 6, Dec. 1998, pp. 64-79.
- [68] X. CHEN, D. T. TANTIPRASUT: “Evolution of ATM internetworking.” *Bell Labs Technical Journal*, Vol. 2, No. 2, April 1997, pp. 82-110.
- [69] H.-K. CHOI, J. O. LIMB: “A behavioral model of Web traffic.” *Proceedings of the 7th IEEE International Conference on Network Protocols (ICNP '99)*, Toronto, Oct. 1999.
- [70] D. D. CLARK: “Adding service discrimination to the Internet.” *Telecommunications Policy*, Vol. 20, No. 3, April 1996, pp. 169-181.

- [71] D. D. CLARK, W. FANG: “Explicit allocation of best effort packet delivery service.” *IEEE/ACM Transactions on Networking*, Vol. 6, No. 4, Aug. 1998, pp. 362-373.
- [72] D. D. CLARK, S. SHENKER, L. ZHANG: “Supporting real-time applications in an integrated services packet network: architecture and mechanism.” *Proceedings of ACM SIGCOMM '92*, Baltimore, Aug. 1992, pp. 14-26.
- [73] W. S. CLEVELAND, D. LIN, D. X. SUN: “IP packet generation: statistical models for TCP start times based on connection-rate superposition.” *Proceedings of ACM SIGMETRICS 2000*, Santa Clara, June 2000, pp. 166-177.
- [74] R. COCCHI, D. ESTRIN, S. SHENKER, L. ZHANG: “A study of priority pricing in multiple service class networks.” *Proceedings of ACM SIGCOMM '91*, Zurich, Sep. 1991.
- [75] J. W. COHEN: “The multiple phase service network with generalized processor sharing.” *Acta Informatica*, Vol. 12, 1979, pp. 245-284.
- [76] D. E. COMER: *Internetworking with TCP/IP – Volume I: Principles, protocols, and architecture*, 4th edition, Prentice Hall, Upper Saddle River, 2000.
- [77] R. W. CONWAY, W. L. MAXWELL, L. W. MILLER: *Theory of scheduling*, Addison-Wesley, Reading, MA, 1967.
- [78] C. COURCOUBETIS, V. A. SIRIS: “An Evaluation of pricing schemes that are based on effective usage.” *Proceedings of the IEEE International Conference on Communications (ICC '98)*, Atlanta, GA, June 1998.
- [79] D. R. COX: “Long-range dependence: a review.” *Statistics: an appraisal*, H. A. David, H. T. David (Eds.), Iowa State University Press, Ames, IA, 1984, pp. 55-74.
- [80] E. CRAWLEY, R. NAIR, B. RAJAGOPALAN, H. SANDICK: *A framework for QoS-based routing in the Internet*, IETF, RFC 2386, August 1998.
- [81] M. E. CROVELLA, A. BESTAVROS: “Self-similarity in World Wide Web traffic: evidence and possible causes.” *IEEE/ACM Transactions on Networking*, Vol. 5, No. 6, Dec. 1997, pp. 835-846.
- [82] L. A. DASILVA: “Pricing for QoS-enabled networks: a survey.” *IEEE Communications Surveys*, Vol. 3, No. 2, April 2000.
- [83] S. DEERING, R. HINDEN: *Internet protocol, version 6 (IPv6) specification*, RFC 2460, IETF, Dec. 1998.
- [84] A. DEMERS, S. KESHAV, S. SHENKER: “Analysis and simulation of a fair queueing algorithm.” *Proceedings of ACM SIGCOMM '89*, Austin, TX, Sep. 1989, pp. 1-12.
- [85] A. DETTI, M. LISTANI, S. SALSANO, L. VELTRI: “Supporting RSVP in a differentiated service domain: an architectural framework and a scalability analysis.” *Proceedings of the IEEE International Conference on Communications (ICC '99)*, Vancouver, June 1999.

- [86] K. DOLZER, W. PAYER, M. EBERSPÄCHER: “A simulation study on traffic aggregation in multi-service networks.” *Proceedings of the IEEE Conference on High Performance Switching and Routing (ATM 2000)*, Heidelberg, June 2000, pp. 157-165.
- [87] C. DOVROLIS: *Proportional differentiated services for the Internet*, PhD Thesis, University of Wisconsin-Madison, Dec. 2000.
- [88] C. DOVROLIS, P. RAMANATHAN: “A case for relative differentiated services and the proportional differentiation model.” *IEEE Network*, Vol. 13, No. 5, Sep.-Oct. 1999, pp. 26-34.
- [89] C. DOVROLIS, P. RAMANATHAN: “Proportional differentiated services, Part II: Loss rate differentiation and packet dropping.” *Proceedings of the 8th International Workshop on Quality of Service (IWQoS 2000)*, Pittsburgh, June 2000.
- [90] C. DOVROLIS, D. STILIADIS, P. RAMANATHAN: “Proportional differentiated services: delay differentiation and packet scheduling.” *Proceedings of ACM SIGCOMM '99*, Cambridge, MA, Sep. 1999.
- [91] D. DURHAM, J. BOYLE, R. COHEN, S. HERZOG, R. RAJAN, A. SASTRY: *The COPS (common open policy service) protocol*, RFC 2748, IETF, Jan. 2000.
- [92] R. EDELL, P. P. VARAIYA: “Providing Internet access: what we learn from INDEX.” *IEEE Network*, Vol. 13, No. 5, Sep. 1999, pp. 18-25.
- [93] C. A. ELDERING, M. L. SYLLA, J. A. EISENACH: “Is there a Moore’s law for bandwidth?” *IEEE Communications Magazine*, Vol. 37, No. 10, October 1999, pp. 117-121.
- [94] J. ENSSLE: *Modellierung und Leistungsuntersuchung eines verteilten Video-On-Demand-Systems für MPEG-codierte Videodatenströme mit variabler Bitrate*, Dissertation, Universität Stuttgart, März 1998.
- [95] ETSI: *Considerations on network mechanisms for charging and revenue accounting*, ETSI Technical Committee Network Aspects (NA), Sophia Antipolis, France, TR 101619, 1998.
- [96] M. FALKNER, M. DEVETSIKIOTIS, I. LAMBADARIS: “An overview of pricing concepts for broadband IP networks.” *IEEE Communications Surveys*, Vol. 3, No. 2, April 2000.
- [97] K. FALL, S. FLOYD: “Simulation-based comparisons of Tahoe, Reno, and SACK TCP.” *Computer Communication Review*, Vol. 26, No. 3, July 1996, pp. 5-21.
- [98] A. FELDMANN: “Characteristics of TCP connection arrivals.” *Self-similar network traffic and performance evaluation*, K. Park, W. Willinger (Eds.), Wiley, New York, 2000, pp. 367-399.
- [99] A. FELDMANN, A. C. GILBERT, W. WILLINGER: “Data networks as cascades: Investigating the multifractal nature of Internet WAN traffic.” *Proceedings of ACM SIGCOMM '98*, Vancouver, Sep. 1998, pp. 42-55.

- [100] W. FENG, D. D. KANDLUR, D. SAHA, K.G. SHIN: “Adaptive packet marking for providing differentiated services in the Internet.” *Proceedings of the 7th IEEE International Conference on Network Protocols (ICNP '98)*, Austin, Oct. 1998.
- [101] W. FENG, D. D. KANDLUR, D. SAHA, K. G. SHIN: *BLUE: a new class of active queue management algorithms*, Technical Report CSE-TR-387-99, University of Michigan, Ann Arbor, MI, Jan. 1999.
- [102] W. FENG, D. D. KANDLUR, D. SAHA, K. G. SHIN: “A self-configuring RED gateway.” *Proceedings of IEEE INFOCOM '99*, New York, March 1999, pp. 1320-1328.
- [103] P. FERGUSON, G. HUSTON: *Quality of service: Delivering QoS on the Internet and in Corporate Networks*, Wiley, New York, 1998.
- [104] R. FIELDING, J. GETTYS, J. MOGUL, H. FRYSTYK, L. MASINTER, P. LEACH, T. BERNERS-LEE: *Hypertext transfer protocol – HTTP/1.1*, RFC 2616, IETF, June 1999.
- [105] S. FLOYD: *Comments on measurement-based admissions control for controlled-load service*, Technical Report, Lawrence Berkeley National Laboratory, Berkeley, CA, July 1996.
- [106] S. FLOYD, K. FALL: “Promoting the use of end-to-end congestion control in the Internet.” *IEEE/ACM Transactions on Networking*, Vol. 7, No. 4, Aug. 1999, pp. 458-472.
- [107] S. FLOYD, T. HENDERSON: *The NewReno modification to TCP's fast recovery algorithm*, RFC 2582, IETF, April 1999.
- [108] S. FLOYD, V. JACOBSON: “Traffic phase effects in packet-switched gateways.” *Computer Communication Review*, Vol. 21, No. 4, April 1991, pp. 26-42.
- [109] S. FLOYD, V. JACOBSON: “Random early detection gateways for congestion avoidance.” *IEEE/ACM Transactions on Networking*, Vol. 1, No. 4, Aug. 1993, pp. 397-413.
- [110] S. FLOYD, V. JACOBSON: “Link-sharing and resource management models for packet networks.” *IEEE/ACM Transactions on Networking*, Vol. 3, No. 4, Aug. 1995, pp. 365-386.
- [111] H. J. FOWLER, W. E. LELAND: “Local area network traffic characteristics, with implications for broadband network congestion management.” *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 7, Sep. 1991, pp. 1139-1149.
- [112] V. S. FROST, B. MELAMED: “Traffic modeling for telecommunications networks.” *IEEE Communications Magazine*, March 1994, pp. 70-81.
- [113] L. GEORGIADIS, R. GUÉRIN, A. PAREKH: “Optimal multiplexing on a single link: delay and buffer requirements.” *IEEE Transactions on Information Theory*, Vol. 43, No. 5, Sep. 1997, pp. 1518-1535.
- [114] R. J. GIBBENS, P. J. HUNT: “Effective bandwidths for the multi-type UAS channel.” *Queueing Systems*, Vol. 9, No. 1-2, 1991, pp. 17-28.

- [115] S. J. GOLESTANI: “A self-clocked fair queueing scheme for broadband applications.” *Proceedings of IEEE INFOCOM '94*, Toronto, June 1994, pp. 636-646.
- [116] C. GRÉVENT: *Simulative Leistungsuntersuchung von TCP über ATM*, Diplomarbeit Nr. 1524, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, April 1997.
- [117] R. GUÉRIN, H. AHMADI, M. NAGSHINEH: “Equivalent capacity and its application to bandwidth allocation in high-speed networks.” *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 7, September 1991, pp. 968-981.
- [118] R. GUÉRIN, S. KAMAT, V. PERIS, R. RAJAN: “Scalable QoS provision through buffer management.” *Proceedings of ACM SIGCOMM '98*, Vancouver, Sep. 1998, pp. 29-40.
- [119] R. GUÉRIN, V. PERIS: “Quality-of-service in packet networks: basic mechanisms and directions.” *Computer Networks*, Vol. 31, No. 3, Feb. 1999, pp. 169-189.
- [120] A. GUPTA, D. O. STAHL, A. B. WHINSTON: “Priority pricing of integrated services networks.” *Internet Economics*, L. MCKNIGHT, J. BAILEY (EDS.), MIT Press, Cambridge, MA, 1998.
- [121] M. HANDLEY, H. SCHULZRINNE, E. SCHOOLER, J. ROSENBERG: *SIP: Session Initiation Protocol*, RFC 2543, IETF, March 1999.
- [122] R. HARRIS, S. KÖHLER: *Possibilities for QoS in existing Internet routing protocols*, COST 257, Document 257TD(99)22, 1999.
- [123] C. HEDRICK: *Routing information protocol*, IETF, RFC 1058, June 1988.
- [124] X. HEI, D. H. K. TSANG: “The EDF scheduling with multiple loss requirements support.” *Proceedings of the 9th IFIP Working Conference on Performance Modelling and Evaluation of ATM & IP Networks*, Budapest, June 2001.
- [125] J. HEIDEMANN, K. OBRACZKA, J. TOUCH: “Modeling the performance of HTTP over several transport protocols.” *IEEE/ACM Transactions on Networking*, Vol. 5, No. 5, Oct. 1997, pp. 616-630.
- [126] J. HEINANEN, F. BAKER, W. WEISS, J. WROCLAWSKI: *Assured forwarding PHB group*, RFC 2597, IETF, June 1999.
- [127] J. HEINANEN, R. GUÉRIN: *A single rate three color marker*, RFC 2697, IETF, Sep. 1999.
- [128] J. HEINANEN, R. GUÉRIN: *A two rate three color marker*, RFC 2698, IETF, Sep. 1999.
- [129] D. P. HEYMAN, T. V. LAKSHMAN, A. L. NEIDHARDT: “A new method for analysing feedback-based protocols with applications to engineering web traffic over the Internet.” *Proceedings of ACM SIGMETRICS '97*, Seattle, June 1997, pp. 24-38.
- [130] J. HUI: “Resource allocation for broadband networks.” *IEEE Journal on Selected Areas in Communications*, Vol. 6, No. 9, Dec. 1988, pp. 1598-1608.

- [131] P. HURLEY, J.-Y. LE BOUDEC, P. THIRAN, M. KARA: “ABE: providing a low-delay service within best effort.” *IEEE Network*, Vol. 15, No. 3, May 2001, pp. 60-69.
- [132] G. HUSTON: *Internet performance survival guide: QoS strategies for multiservice networks*, John Wiley & Sons, New York, 2000.
- [133] ISO: *Information technology – Open systems interconnection – Basic reference model: The basic model*, ISO/IEC 7498-1, International Organisation for Standardization, 1994.
- [134] ITG: *Begriffe der Nachrichtenverkehrstheorie*, ITG-Empfehlung 5.2-03, VDE-Verlag, Berlin, Okt. 1997.
- [135] ITU-T: *One-way transmission time*, Recommendation G.114, International Telecommunication Union, March 1993.
- [136] ITU-T: *Packet-based multimedia communications systems*, Recommendation H.323, International Telecommunication Union, Nov. 2000.
- [137] ITU-T: *General aspects of quality of service and network performance in digital networks, including ISDN*, Recommendation I.350, International Telecommunication Union, March 1993.
- [138] ITU-T: *Traffic control and congestion control in B-ISDN*, Recommendation I.371, International Telecommunication Union, Aug. 1996.
- [139] ITU-T: *Methods for subjective determination of transmission quality*, Recommendation P.800, International Telecommunication Union, Aug. 1996.
- [140] ITU-T: *General quality of service parameters for communication via public data networks*, Recommendation X.140, International Telecommunication Union, Sep. 1992.
- [141] J. JACKSON: *Scheduling a production line to minimize maximum tardiness*, Research Report No. 43, Management Science Research Project, University of California, Los Angeles, 1955.
- [142] V. JACOBSON: “Congestion avoidance and control.” *Proceedings of ACM SIGCOMM '88*, Stanford, CA, Aug. 1988.
- [143] V. JACOBSON, K. NICHOLS, K. PODURI: *An expedited forwarding PHB*, RFC 2598, IETF, June 1999.
- [144] R. K. JAIN, D.-M. W. CHIU, W. R. HAW: *A quantitative measure of fairness and discrimination for resource allocation in shared computer systems*, Technical Report DEC-TR-301, DEC Eastern Research Lab, Hudson, MA, Sep. 1984.
- [145] S. JAMIN, P. B. DANZIG, S. J. SHENKER, L. ZHANG: “A measurement-based admission control algorithm for integrated services packet networks.” *Proceedings of ACM SIGCOMM '95*, Cambridge, MA, Aug. 1995, pp. 2-13.
- [146] P. R. JELENKOVIC, A. A. LAZAR: “Multiplexing on-off sources with subexponential on periods: Part I.” *Proceedings of IEEE INFOCOM '97*, Kobe, April 1997, pp. 187-195.

- [147] Y. JOO, V. RIBEIRO, A. FELDMANN, A. C. GILBERT, W. WILLINGER: “TCP/IP traffic dynamics and network performance: a lesson in workload modeling, flow control, and trace-driven simulations.” *Computer Communications Review*, Vol. 31, No. 2, April 2001.
- [148] S. KALYANASUNDARAM, E. K. P. CHONG, N. B. SHROFF: “Admission control schemes to provide class-level QoS in multiservice networks.” *Computer Networks*, Vol. 35, Nos. 2-3, Feb. 2001, pp. 307-326.
- [149] B. KANTOR, P. LAPSLEY: *Network news transfer protocol*, RFC 977, IETF, Feb. 1986.
- [150] P. KARN, C. PARTRIDGE: “Improving round-trip time estimates in reliable transport protocols.” *ACM Transactions on Computer Systems*, Vol. 9, No. 4, Nov. 1991, pp. 364-373.
- [151] M. KARSTEN, J. SCHMITT, L. WOLF, R. STEINMETZ: “Abrechnungsverfahren für paketvermittelte diensteintegrierende Kommunikationsnetze.” *PIK*, Bd. 21, Nr. 4, Okt. 1998, S. 211-218.
- [152] M. KATEVENIS, S. SIDIROPOULOS, C. COURCOUBETIS: “Weighted round-robin cell multiplexing in a general-purpose ATM switch chip.” *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 8, Oct. 1991, pp. 1265-1279.
- [153] F. P. KELLY: “Effective bandwidths at multi-class queues.” *Queueing Systems*, Vol. 9, Nos. 1-2, 1991, pp. 5-16.
- [154] F.P. KELLY: “Tariffs and effective bandwidths in multiservice networks.” *Proceedings of the 14th International Teletraffic Congress (ITC 14)*, Antibes Juan-les-Pins, June 1994, pp. 401-410.
- [155] L. KLEINROCK: *Queueing systems – Volume II: computer applications*, John Wiley & Sons, New York, NY, 1976.
- [156] J. KLENSIN: *Simple mail transfer protocol*, RFC 2821, IETF, April 2001.
- [157] S. KÖHLER, M. MENTH, N. VICARI: *Analytic performance evaluation of the RED algorithm for QoS in TCP/IP networks*, Technical Report No. 259, University of Würzburg, Institute of Computer Science, Würzburg, March 2000.
- [158] T. KOSTAS, M. S. BORELLA, I. SIDHU, G. M. SCHUSTER, J. GRABIEC, J. MAHLER: “Real-time voice over packet switched networks.” *IEEE Network*, Vol. 12, No. 1, Jan. 1998, pp. 18-27.
- [159] C. KRASIC, K. LI, J. WALPOLE: “The case for streaming multimedia with TCP.” *Proceedings of the 8th International Workshop on Interactive Distributed Multimedia Systems (IDMS 2001)*, Lancaster, UK, Sep. 2001, pp. 213-218.
- [160] H. KRÖNER: *Verkehrssteuerung in ATM-Netzen: Verfahren und verkehrstheoretische Analysen zur Zellpriorisierung und Verbindungsannahme*, Dissertation, Universität Stuttgart, 1994.

- [161] H. KRÖNER, G. HÉBUTERNE, P. BOYER, A. GRAVEY: “Priority management in ATM switching nodes.” *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 3, April 1991, pp. 418-427.
- [162] P. J. KÜHN: “Reminder on queueing theory for ATM networks.” *Telecommunication Systems*, Vol. 5, Nos. 1-3, May 1996, pp. 1-24.
- [163] P.J. KÜHN: *Technische Informatik II*, Manuskript zur Vorlesung, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 2001.
- [164] V. KUMAR: *MBone: Interactive multimedia on the Internet*, New Riders, Indianapolis, 1995.
- [165] A. KUMAR: “Comparative performance analysis of versions of TCP in a local network with a lossy link.” *IEEE/ACM Transactions on Networking*, Vol. 6, No. 4, Aug. 1998, pp. 485-498.
- [166] K. R. R. KUMAR, A. L. ANANDA, L. JACOB: “TCP-friendly traffic conditioning in Diff-Serv networks: a memory-based approach.” *Computer Networks*, Vol. 38, No. 6, April 2002, pp. 731-743.
- [167] V. P. KUMAR, T. V. LAKSHMAN, D. STILIADIS: “Beyond best effort: router architectures for the differentiated services of tomorrow’s Internet.” *IEEE Communications Magazine*, Vol. 36, No. 5, May 1998, pp. 152-164.
- [168] M.A. LABRADOR, S. BANERJEE: “Enhancing application throughput by selective packet dropping.” *Proceedings of the IEEE International Conference on Communications (ICC '99)*, Vancouver, June 1999, pp. 1217-1222.
- [169] T. V. LAKSHMAN, U. MADHOW: “The performance of TCP/IP for networks with high bandwidth-delay products and random loss.” *IEEE/ACM Transactions on Networking*, Vol. 5, No. 3, June 1997, pp. 336-350.
- [170] M. LAUBACH, J. HALPERN: *Classical IP and ARP over ATM*, RFC 2225, IETF, April 1998.
- [171] A. L. LAW, W. D. KELTON: *Simulation modelling and analysis*, McGraw-Hill, New York, 1991.
- [172] W. E. LELAND, M. S. TAQQU, W. WILLINGER, D. V. WILSON: “On the self-similar nature of Ethernet traffic (extended version).” *IEEE/ACM Transactions on Networking*, Vol. 2, No. 1, Feb. 1994, pp. 1-15.
- [173] H. LE POCHER, V. C. M. LEUNG, D. W. GILLIES: “An efficient ATM voice service with flexible jitter and delay guarantees.” *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 1, Jan. 1999, pp. 51-62.
- [174] J.-S. LI, H.-C. LAI: “Providing proportional differentiated services using PLQ.” *Proceedings of IEEE GLOBECOM 2001*, San Antonio, Nov. 2001, pp. 2280-2284.

- [175] J. LIEBEHERR, N. CHRISTIN: “JoBS: joint buffer management and scheduling for differentiated services.” *Proceedings of the 9th International Workshop on Quality of Service (IWQoS 2001)*, Karlsruhe, June 2001, pp. 404-418.
- [176] D. LIN, R. MORRIS: “Dynamics of random early detection.” *Proceedings of ACM SIGCOMM '97*, Cannes, Sep. 1997.
- [177] K. LINDBERGER: “Cost based charging principles in ATM networks.” *Proceedings of the 15th International Teletraffic Congress (ITC 15)*, Washington, DC, June 1997, pp. 771-780.
- [178] K. LINDBERGER: “Balancing quality of service, pricing and utilisation in multiservice networks with stream and elastic traffic.” *Proceedings of the 16th International Teletraffic Congress (ITC 16)*, Edinburgh, June 1999, pp. 1127-1136.
- [179] C. L. LIU, J. W. LAYLAND: “Scheduling algorithms for multiprogramming in a hard-real-time environment.” *Journal of the ACM*, Vol. 20, No. 1, Jan. 1973, pp. 46-61.
- [180] M. LORANG: *Skalierbares Verkehrsmanagement für diensteintegrierende IP-Netze mit virtuellen Verbindungen und verbindungslosem Routen von Datagrammen*, Monographie, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 2002.
- [181] J. K. MACKIE-MASON, H. R. VARIAN: “Pricing congestible network resources.” *IEEE Journal on Selected Areas in Communications*, Vol. 13, No. 7, Sep. 1995, pp. 1141-1149.
- [182] J. K. MACKIE-MASON, H. R. VARIAN: “Pricing the Internet.” *Public access to the Internet*, B. Kahin, J. Keller (Eds.), MIT Press, Cambridge, MA, 1995.
- [183] B. A. MAH: “An empirical model of HTTP network traffic.” *Proceedings of IEEE INFOCOM '97*, Kobe, April 1997, pp. 592-600.
- [184] G. MALKIN: *RIP version 2 - carrying additional information*, IETF, RFC 1723, Nov. 1994.
- [185] B. B. MANDELBROT, J. W. VAN NESS: “Fractional Brownian motions, fractional noises and applications.” *SIAM Review*, Vol. 10, No. 4, Oct. 1968, pp. 422-437.
- [186] M. MARSAN, C. CASETTI, R. GAETA, M. MEO: “Performance analysis of TCP connections sharing a congested Internet link.” *Performance Evaluation*, Vol. 42, Nos. 2-3, Oct. 2000, pp. 109-127.
- [187] D. MARTOS: *Performance evaluation of the class based queueing hierarchical scheduling algorithm*, Diploma Thesis No. 1638, Institute of Communication Networks and Computer Engineering, University of Stuttgart, Jan. 2000.
- [188] L. MASSOULIÉ, J. ROBERTS: “Arguments in favour of admission control for TCP flows.” *Proceedings of the 16th International Teletraffic Congress (ITC 16)*, Edinburgh, June 1999, pp. 33-44.

- [189] M. MATHIS, J. MAHDAVI, S. FLOYD, A. ROMANOV: *TCP selective acknowledgment options*, RFC 2018, IETF, Oct. 1996.
- [190] M. MATHIS, J. SEMKE, J. MAHDAVI, T. OTT: “The macroscopic behavior of the TCP congestion avoidance algorithm.” *Computer Communication Review*, Vol. 27, No. 3, July 1997, pp. 67-82.
- [191] S. MCCREARY, K. CLAFFY: “Trends in wide area IP traffic patterns: a view from Ames Internet eXchange.” *Proceedings of the 13th ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management*, Monterey, CA, Sep. 2000, pp. 1.1-1.12.
- [192] N. MCKEOWN: *Scheduling algorithms for input-queued cell switches*, PhD Thesis, University of California at Berkeley, 1995.
- [193] L. MCKNIGHT, J. BAILEY: *Internet Economics*, MIT Press, Cambridge, 1998.
- [194] A. MISRA, T. J. OTT: “The window distribution of idealized TCP congestion avoidance with variable packet loss.” *Proceedings of IEEE INFOCOM '99*, New York, March 1999, pp. 1564-1572.
- [195] D. MITRA, M. I. REIMAN, J. WANG: “Robust dynamic admission control for unified cell and call QoS in statistical multiplexing.” *IEEE Journal on Selected Areas in Communications*, Vol. 16, No. 5, June 1998, pp. 692-707.
- [196] C. MOIGNARD: *Erweiterung eines Simulationsprogramms zur Leistungsuntersuchung von Internet-Telefonie*, Studienarbeit Nr. 1629, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, Nov. 1999.
- [197] S. B. MOON, J. KUROSE, D. TOWSLEY: “Packet audio playout adjustment: performance bounds and algorithms.” *ACM/Springer Multimedia Systems*, Vol. 6, No. 1, Jan. 1998, pp. 17-28.
- [198] Y. MORET, S. FDIDA: “A proportional queue control mechanism to provide differentiated services.” *Proceedings of the 13th International Symposium on Computer and Information Systems (ISCIS '98)*, Ankara, Turkey, Oct. 1998, pp. 17-24.
- [199] J. MOY: *OSPF Version 2*, IETF, RFC 2328, April 1998.
- [200] R. NAGARAJAN, J. KUROSE, D. TOWSLEY: “Finite-horizon statistical quality-of-service measures for high-speed networks.” *Journal of High Speed Networks*, Vol. 3, No. 4, 1994, pp. 344-351.
- [201] J. NAGLE: *Congestion control in IP/TCP internetworks*, RFC 896, IETF, Jan. 1984.
- [202] T. NANDAGOPAL, N. VENKITARAMAN, R. SIVAKUMAR, V. BHARGHAVAN: “Delay differentiation and adaptation in core stateless networks.” *Proceedings of IEEE INFOCOM 2000*, Tel Aviv, March 2000.
- [203] T.D. NEAME, M. ZUKERMAN, R.G. ADDIE: “A practical approach for multimedia traffic modeling.” *Proceedings of the 5th International Conference on Broadband Communications (BC '99)*, Hong Kong, Nov. 1999, pp. 73-82.

- [204] K. NICHOLS, S. BLAKE, F. BAKER, D. BLACK: *Definition of the differentiated services field (DS field) in the IPv4 and IPv6 headers*, RFC 2474, IETF, Dec. 1998.
- [205] K. NICHOLS, V. JACOBSON, L. ZHANG: *A two-bit differentiated services architecture for the Internet*, RFC 2638, IETF, July 1999.
- [206] I. NORROS: “On the use of fractional Brownian motion in the theory of connectionless networks.” *IEEE Journal on Selected Areas in Communications*, Vol. 13, No. 6, Aug. 1995, pp. 953-962.
- [207] A. ODLYZKO: “Paris Metro Pricing: the minimalist differentiated services solution.” *Proceedings of the 7th International Workshop on Quality of Service (IWQoS '99)*, London, June 1999.
- [208] J. OIKARINEN, D. REED: *Internet relay chat protocol*, RFC 1459, IETF, May 1993.
- [209] P. OLIVIER, N. BENAMEUR: “Flow level IP traffic characterization.” *Proceedings of the 17th International Teletraffic Congress (ITC 17)*, Salvador da Bahia, Brazil, Dec. 2001, pp. 25-36.
- [210] OMG: *The common object request broker: architecture and specification*, Revision 2.6, Object Management Group, Dec. 2001.
- [211] T. J. OTT, T. V. LAKSHMAN, L. H. WONG: “SRED: stabilized RED.” *Proceedings of IEEE INFOCOM '99*, New York, March 1999, pp. 1346-1355.
- [212] J. PADHYE, V. FIROIU, D. F. TOWSLEY, J. F. KUROSE: “Modeling TCP Reno performance: a simple model and its empirical validation.” *IEEE/ACM Transactions on Networking*, Vol. 8, No. 2, April 2000, pp. 133-145.
- [213] J. PADHYE, S. FLOYD: “On inferring TCP behavior.” *Proceedings of ACM SIGCOMM 2001*, San Diego, CA, Aug. 2001.
- [214] R. PAN, B. PRABHAKAR, K. PSOUNIS: “CHOKe: a stateless active queue management scheme for approximating fair bandwidth allocation.” *Proceedings of IEEE INFOCOM 2000*, Tel Aviv, April 2000, pp. 942-951.
- [215] P. PAN, H. SCHULZRINNE: “YESSIR: A simple reservation mechanism for the Internet.” *Computer Communications Review*, Vol. 29, No. 2, April 1999, pp. 89-101.
- [216] S. S. PANWAR, D. TOWSLEY, J. K. WOLF: “Optimal scheduling policies for a class of queues with customer deadlines to the beginning of service.” *Journal of the ACM*, Vol. 35, No. 4, Oct. 1988, pp. 832-844.
- [217] A. K. PAREKH, R. G. GALLAGER: “A generalized processor sharing approach to flow control in integrated services networks: the single-node case.” *IEEE/ACM Transactions on Networking*, Vol. 1, No. 3, June 1993, pp. 344-357.
- [218] K. PARK, G. KIM, M. CROVELLA: “On the relationship between file sizes, transport protocols, and self-similar network traffic.” *Proceedings of the 4th IEEE International Conference on Network Protocols (ICNP '96)*, Columbus, OH, Oct. 1996, pp. 171-180.

- [219] K. PARK, W. WILLINGER (EDS.): *Self-similar network traffic and performance evaluation*, Wiley, New York, 2000.
- [220] V. PAXSON: “Fast, approximate synthesis of fractional Gaussian noise for generating self-similar network traffic.” *Computer Communications Review*, Vol. 27, No. 4, Oct. 1997, pp. 1-14.
- [221] V. PAXSON, M. ALLMAN, S. DAWSON, W. FENNER, J. GRINER, I. HEAVENS, K. LAHEY, J. SEMKE, B. VOLZ: *Known TCP implementation problems*, RFC 2525, IETF, March 1999.
- [222] V. PAXSON, S. FLOYD: “Wide area traffic: the failure of Poisson modeling.” *IEEE/ACM Transactions on Networking*, Vol. 3, No. 3, June 1995, pp. 226-244.
- [223] J. E. PITKOW: “Summary of WWW characterizations.” *World Wide Web*, Vol. 2, Nos. 1-2, Jan. 1999, pp. 3-13.
- [224] J. PITTS, J. SCHORMANS: *Introduction to ATM: design and performance*, Wiley, New York, 1996.
- [225] J. POSTEL: *User datagram protocol*, RFC 768, IETF, Aug. 1980.
- [226] J. POSTEL: *Internet protocol*, RFC 791, IETF, Sep. 1981.
- [227] J. POSTEL: *Transmission control protocol*, RFC 793, IETF, Sep. 1981.
- [228] J. POSTEL, J. REYNOLDS: *Telnet protocol specification*, RFC 854, IETF, May 1983.
- [229] J. POSTEL, J. REYNOLDS: *File transfer protocol (FTP)*, RFC 959, IETF, Oct. 1985.
- [230] L. QIU, Y. ZHANG, S. KESHAV: “Understanding the performance of many TCP flows.” *Computer Networks*, Vol. 37, No. 5, Nov. 2001, pp. 277-306.
- [231] R. RAJAN, D. VERMA, S. KAMAT, E. FELSTAIN, S. HERZOG: “A policy framework for integrated and differentiated services in the Internet.” *IEEE Network*, Vol. 13, No. 5, Sep.-Oct. 1999, pp. 36-41.
- [232] R. RAMJEE, J. KUROSE, D. TOWSLEY, H. SCHULZRINNE: “Adaptive playout mechanisms for packetized audio applications in wide-area networks.” *Proceedings of IEEE INFOCOM '94*, Toronto, June 1994, pp. 680-688.
- [233] E.P. RATHGEB: *Verkehrsflüsse in ATM-Netzen – Modellierung und Analyse von Verkehrsquellen und Quellflusskontrollverfahren*, Dissertation, Universität Stuttgart, 1991.
- [234] A. RIEDL, T. BAUSCHERT, M. PERSKE, A. PROBST: “Investigation of the M/G/R processor sharing model for dimensioning of IP access networks with elastic traffic.” *Proceedings of the 1st Polish-German Teletraffic Symposium (PGTS 2000)*, Dresden, Sep. 2000, pp. 97-106.
- [235] J. ROBERTS: “Quality of service guarantees and charging in multiservice networks.” *IEICE Transactions on Communications*, Vol. E81-B, No. 5, May 1998, pp. 824-831.

- [236] J. ROBERTS, U. MOCCI, J. VIRTAMO (EDS.): *Broadband network teletraffic: Final Report of Action COST 242*, Springer, 1996.
- [237] A. RODRIGUEZ, J. GATRELL, J. KARAS, R. PESCHKE: *TCP/IP tutorial and technical overview*, IBM Redbook, IBM Corporation, International Technical Support Organization, Aug. 2001.
- [238] A. ROMANOW, S. FLOYD: “Dynamics of TCP traffic over ATM networks.” *IEEE Journal on Selected Areas in Communications*, Vol. 13, No. 4, May 1995, pp. 633-641.
- [239] M. ROSE: *Post office protocol – version 3*, RFC 1081, IETF, Nov. 1988.
- [240] O. ROSE: *Traffic modeling of variable bit rate MPEG video and its impacts on ATM networks*, Dissertation, University of Würzburg, 1997.
- [241] E. ROSEN, A. VISWANATHAN, R. CALLON: *Multiprotocol label switching architecture*, RFC 3031, IETF, Jan. 2001.
- [242] J. RUUTU, K. KILKKI: “Simple integrated media access – a comprehensive service for future Internet.” *Proceedings of the IFIP Conference on Performance of Information and Communications Systems (PICS '98)*, Lund, May 1998.
- [243] M. SANTIFALLER: *TCP/IP und ONC/NFS in Theorie und Praxis*, 2. Auflage, Addison-Wesley, Reading, MA, 1993.
- [244] K. SAUER: *Integration von Sprach- und Datenkommunikation in Lokalen Netzen*, Dissertation, Universität Stuttgart, 1990.
- [245] R. W. SCHEIFLER: *X window system protocol, version 11*, RFC 1013, IETF, June 1987.
- [246] T. SCHMIDT: *Untersuchung von Adaptionsalgorithmen verteilter Multimedia-Anwendungen*, Dissertation (eingereicht), Universität Stuttgart, 2001.
- [247] H. SCHULZRINNE, S. CASNER, R. FREDERICK, V. JACOBSON: *RTP: a transport protocol for real-time applications*, RFC 1889, IETF, Jan. 1996.
- [248] H.-P. SCHWEFEL: “Behavior of TCP-like elastic traffic at a buffered bottleneck router.” *Proceedings of IEEE INFOCOM 2001*, Anchorage, AK, April 2001, pp. 1698-1705.
- [249] N. SEMRET: *Market mechanisms for network resource sharing*, PhD Thesis, Columbia University, New York, April 1999.
- [250] S. SHENKER, C. PARTRIDGE, R. GUÉRIN: *Specification of guaranteed quality of service*, RFC 2212, IETF, Sep. 1997.
- [251] S. SHEPLER, B. CALLAGHAN, D. ROBINSON, R. THURLOW, C. BEAME, M. EISLER, D. NOVECK: *NFS version 4 protocol*, RFC 3010, IETF, Dec. 2000.
- [252] K. SHIOMOTO, N. YAMANAKA, T. TAKAHASHI: “Overview of measurement-based connection admission control methods in ATM networks.” *IEEE Communications Surveys*, Vol. 2, No. 1, Jan. 1999, pp. 2-13.

- [253] M. SHREEDHAR, G. VARGHESE: “Efficient fair queuing using deficit round-robin.” *IEEE/ACM Transactions on Networking*, Vol. 4, No. 3, June 1996, pp. 375-385.
- [254] R. SIGLE, T. RENGER: “Hierarchical scheduling strategies for wireless ATM MAC protocols.” *Proceedings of the 3rd Workshop on Personal Wireless Communications (PWC '98)*, Tokyo, April 1998, pp. 175-182.
- [255] B. SIKDAR, S. KALYANARAMAN, K. S. VASTOLA: “An integrated model for the latency and steady-state throughput of TCP connections.” *Performance Evaluation*, Vol. 46, Nos. 2-3, Oct. 2001, pp. 139-154.
- [256] K. SOLLINS: *The TFTP protocol (revision 2)*, RFC 1350, IETF, July 1992.
- [257] J. SPÄTH: *Entwurf und Bewertung von Verfahren zur Verkehrslenkung in WDM-Transportnetzen*, Dissertation (eingereicht), Universität Stuttgart, 2001.
- [258] G. D. STAMOULIS, M. E. ANAGNOSTOU, A. D. GEORGANTAS: “Traffic source models for ATM networks: a survey.” *Computer Communications*, Vol. 17, No. 6, June 1994, pp. 428-438.
- [259] W. R. STEVENS: *TCP/IP illustrated – Volume 1: The protocols*, Addison-Wesley, Reading, MA, 1996.
- [260] D. STILIADIS, A. VARMA: “Latency-rate servers: a general model for analysis of traffic scheduling algorithms.” *IEEE/ACM Transactions on Networking*, Vol. 6, No. 5, Oct. 1998, pp. 611-624.
- [261] B. STILLER, G. FANKHAUSER, B. PLATTNER, N. WEILER: “Charging and accounting for integrated services – state of the art, problems, and trends.” *Proceedings of the Internet Summit (INET '98)*, Geneva, July 1998.
- [262] M. STÜMPFLE: *Planung und Optimierung von prioritätsbasierten Steuergerätenetzen für Fahrzeuge*, Dissertation, Universität Stuttgart, 1999.
- [263] B. TEITELBAUM, S. HARES, L. DUNN, R. NEILSON, V. NARAYAN, F. REICHMEYER: “Internet2 QBone: building a testbed for differentiated services.” *IEEE Network*, Vol. 13, No. 5, Sep. 1999, pp. 8-16.
- [264] T. THEIMER: *Vergleichende Untersuchungen an ATM-Koppelnetzstrukturen*, Dissertation, Universität Stuttgart, 1994.
- [265] R. THIERAUF: *An introductory approach to operations research*, Wiley, New York, 1978.
- [266] K. THOMPSON, G. J. MILLER, R. WILDER: “Wide-area Internet traffic patterns and characteristics.” *IEEE Network*, Vol. 11, Nov. 1997, pp. 10-23.
- [267] B. TSYBAKOV, N. D. GEORGANAS: “On self-similar traffic in ATM queues: definitions, overflow probability bound, and cell delay distribution.” *IEEE/ACM Transactions on Networking*, Vol. 5, No. 3, June 1997, pp. 397-409.

- [268] D. VEITCH, P. ABRYS: “A wavelet-based joint estimator of the parameters of long-range dependence.” *IEEE Transactions on Information Theory*, Vol. 45, No. 3, April 1999, pp. 878-897.
- [269] N. VICARI, S. KÖHLER, J. CHARZINSKI: “The dependence of Internet user characteristics on access speed.” *Proceedings of the 25th IEEE Conference on Local Computer Networks (LCN 2000)*, Tampa, FL, Nov. 2000.
- [270] A. VISWANATHAN, N. FELDMAN, Z. WANG, R. CALLON: “Evolution of multiprotocol label switching.” *IEEE Communications Magazine*, Vol. 36, No. 5, May 1998, pp. 165-173.
- [271] Z. WANG: *User-share differentiation (USD): scalable bandwidth allocation for differentiated services*, Internet Draft <draft-wang-diff-serv-usd-00.txt> (work in progress), IETF, Nov. 1997.
- [272] Z. WANG: *Internet QoS: Architecture and mechanisms for quality of service*, Morgan Kaufmann Publishers, San Francisco, 2001.
- [273] P. P. WHITE: “RSVP and integrated services in the Internet: a tutorial.” *IEEE Communications Magazine*, Vol. 35, No. 5, May 1997, pp. 100-106.
- [274] W. WILLINGER, M.S. TAQQU, R. SHERMAN, D.V. WILSON: “Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level.” *IEEE/ACM Transactions on Networking*, Vol. 5, No. 1, Feb. 1997, pp. 71-86.
- [275] D. E. WREGE, J. LIEBEHERR: “A near-optimal packet scheduler for QoS networks.” *Proceedings of IEEE INFOCOM '97*, Kobe, April 1997, pp. 576-583.
- [276] J. WROCLAWSKI: *Specification of the controlled-load network element service*, RFC 2211, IETF, Sep. 1997.
- [277] Y. XIE, T. YANG: “Cell discarding policies supporting multiple delay and loss requirements in ATM networks.” *Proceedings of IEEE GLOBECOM '97*, Phoenix, Nov. 1997, pp. 1075-1080.
- [278] I. YEOM, A. L. N. REDDY: “Modeling TCP behavior in a differentiated services network.” *IEEE/ACM Transactions on Networking*, Vol. 9, No. 1, Feb. 2001, pp. 31-46.
- [279] H. ZHANG: “Service disciplines for guaranteed performance service in packet-switching networks.” *Proceedings of the IEEE*, Vol. 83, No. 10, Oct. 1995.
- [280] L. ZHANG: “Virtual clock: a new traffic control algorithm for packet switching networks.” *Proceedings of ACM SIGCOMM '90*, Philadelphia, Sep. 1990, pp. 19-29.
- [281] H. ZHANG, D. FERRARI: “Rate-controlled static priority queueing.” *Proceedings of IEEE INFOCOM '93*, San Francisco, April 1993, pp. 227-236.
- [282] H. ZHANG, D. FERRARI: “Rate-controlled service disciplines.” *Journal of High Speed Networks*, Vol. 3, No. 4, March 1994.

# Anhang A

## Analyse eines TCP-Einzeltransfers

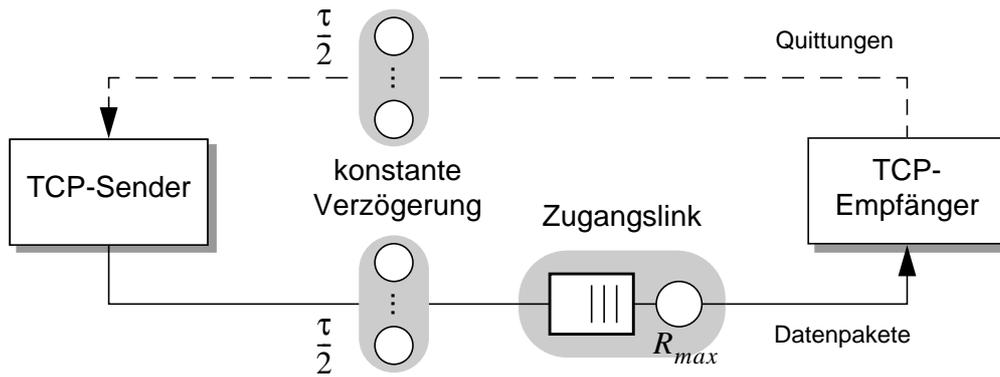
Um den Einfluss von Parametern und Verfahren innerhalb eines IP-Netzes auf die Dienstgüte für TCP-basierte Anwendungen beurteilen zu können, ist die Kenntnis der maximalen Leistungsfähigkeit unter der Voraussetzung eines unbelasteten Systems als Referenzgröße von großer Bedeutung. In Anlehnung an das in Kapitel 4 vorgestellte dynamische TCP-Verkehrsmo- dell sowie die dafür definierten Leistungskenngrößen wird dazu die Dauer  $T_{min}(s)$  eines TCP- Transfers über ein aus Nutzersicht ideales IP-Netz betrachtet.

### A.1 Modell

Das „ideale Netz“, das als Grundlage für die nachfolgende Analyse dient, enthält nur solche Komponenten, die als unveränderlich angesehen werden. Dazu gehört eine Beschränkung der Zugangsbandbreite – bei einem üblichen Szenario bedingt durch die Anschlussbandbreite des Empfängers – sowie eine konstante Verzögerung zwischen Sender und Empfänger, die im Wesentlichen auf die Signallaufzeit (*propagation delay*) zurückzuführen ist.

Als Modell kann daher die in Bild A.1 dargestellte Anordnung angenommen werden. Der Netzzugang wird dabei durch ein Bediensystem mit Bedienrate  $R_{max}$ , die konstante Verzöge- rung durch jeweils einen *Infinite Server* mit Verzögerung  $\tau/2$  auf dem Hin- und dem Rückka- nal modelliert. Vereinfachend wird die Länge des TCP/IP-Paketkopfs vernachlässigt, sodass eine Verarbeitungszeit im Bediensystem nur für die Nutzinformation anfällt und folglich auf dem Rückkanal für die Quittungen außer der konstanten Verzögerung keine weitere, durch den Netzzugang bedingte Verzögerung auftritt.

Für die Bestimmung der minimalen Dauer eines TCP-Transfers werde zunächst von einem festen Transfervolumen (Burstgröße)  $s$  ausgegangen. Der Overhead durch Steuerinformatio- nen im Paketkopf werde nicht berücksichtigt, sodass die Länge eines Pakets als identisch mit



**Bild A.1:** Modell zur Betrachtung der minimalen Dauer eines TCP-Transfers

der Größe des darin übertragenen Segments angenommen wird. Die maximale Paketlänge entspricht daher der maximalen Segmentgröße MSS und werde mit  $L_{MSS}$  bezeichnet, woraus sich

$$h = \frac{L_{MSS}}{R_{max}} \quad (\text{A.1})$$

als Übertragungsdauer für ein Paket mit maximaler Größe auf dem Zugangslink ergibt. Ferner werden folgende Annahmen bezüglich der TCP-Parameter getroffen:

- Das *Congestion Window* (CWnd) beim Sender werde zu Beginn der Übertragung mit dem Wert  $w \cdot L_{MSS}$  initialisiert ( $w > 0$ ), sodass zunächst  $w$  Pakete geschickt werden können, ohne auf eine Empfangsbestätigung zu warten.
- Der TCP-Empfangspuffer sei in seiner Größe begrenzt. Die Puffergröße sei dabei einerseits kleiner als die Kapazität des Puffers vor dem Zugangslink und andererseits größer als das Bandbreite-Verzögerungsprodukt  $R_{max} \cdot (\tau + h)$ . In diesem Fall haben die genannten Puffergrößen keinen Einfluss auf das Systemverhalten, d. h. es treten insbesondere keine Verluste auf. Gleiches kann auch durch eine Begrenzung des CWnd beim Sender mit entsprechender Wahl des Wertebereichs erreicht werden.
- Der Initialwert der *Slow-Start*-Schwelle sei größer gleich dem Minimum von Empfangspufferkapazität und Maximalwert für das CWnd. Da aufgrund der obigen Annahme keine Verluste auftreten, ändert sich der Schwellwert während der Übertragung nicht. Somit kann unter der genannten Bedingung ein Einfluss des *Slow-Start*-Schwellwerts ausgeschlossen werden.
- Eine Beeinflussung durch den Nagle-Algorithmus wird nicht betrachtet.

Im folgenden Abschnitt werde zunächst angenommen, dass die *Delayed-ACK*-Option beim Empfänger nicht aktiviert sei. Die durch diese Option verursachten Effekte werden dann in Abschnitt A.3 berücksichtigt.

## A.2 Minimale Transferzeit ohne Delayed-ACK-Option

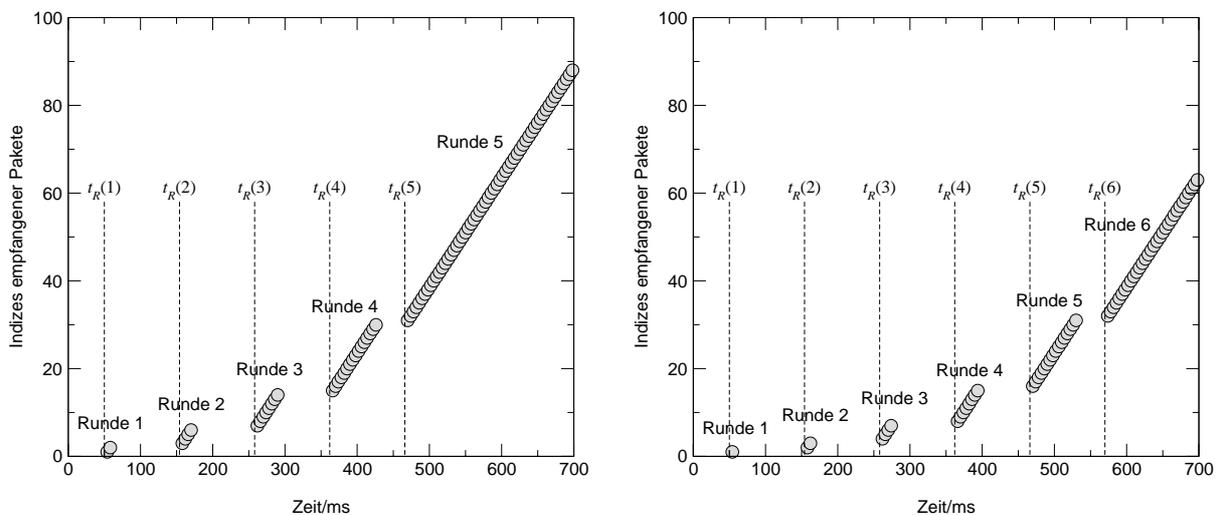
Das Übertragungsverhalten wird zunächst durch den beim Sender implementierten *Slow-Start*-Mechanismus bestimmt. Die Übertragung kann in dieser Phase in einzelne Runden unterteilt werden. Dies wird z. B. aus Bild A.2, das die Indizes gesendeter Pakete für die Parameter  $\tau = 100$  ms und  $h = 4$  ms darstellt, ersichtlich. Innerhalb einer Runde  $i$  ( $i = 1, 2, \dots$ ) können  $N(i)$  Pakete der Größe  $L_{MSS}$  verschickt werden.  $N(i) \cdot L_{MSS}$  entspricht dabei der Größe des CWnd zu Beginn von Runde  $i$ . Nachdem das letzte Paket einer Runde gesendet worden ist, muss gewartet werden, bis die Empfangsbestätigung für das erste Paket aus der Runde eingetroffen ist. Daraufhin kann die nächste Runde mit dem Senden neuer Pakete beginnen. Die einzelnen Runden beginnen im Abstand der minimalen Zykluszeit (*round trip time*, RTT)  $\tau + h$ .

In der *Slow-Start*-Phase wird das Empfangsfenster für jedes quittierte Paket um  $L_{MSS}$  erhöht, d. h. es können nach dem Empfang einer Quittung zwei neue Pakete gesendet werden. Dies bewirkt einen exponentiellen Anstieg der Anzahl pro Runde gesendeter Pakete:

$$N(i) = w \cdot 2^{i-1} \tag{A.2}$$

Das typische Verhalten mit der Möglichkeit der Unterteilung in Runden endet, wenn  $N(i)$  so groß geworden ist, dass das letzte Paket sich noch im Bediensystem befindet, wenn die Quittung für das erste Paket der Runde beim Sender eingetroffen ist. Damit werden bereits wieder neue Pakete verschickt, die sich dann in der Warteschlange dahinter einreihen müssen. Ab dieser Runde treffen alle Pakete beim Empfänger im Abstand ihrer Bedienzeit auf dem Zugangslink ein. Der Index dieser Runde kann aus

$$\kappa = \min(i | N(i) \cdot h > \tau + h) \tag{A.3}$$



**Bild A.2:** Zeitlicher Verlauf der Übertragung aus Sicht des Empfängers, kein *Delayed ACK*,  $w = 2$  (links) bzw.  $w = 1$  (rechts)

bestimmt werden und ergibt sich unter Verwendung von (A.2) zu

$$\kappa = \left\lceil \text{ld} \left( \frac{1}{w} \cdot \left( \frac{\tau}{h} + 1 \right) \right) \right\rceil + 2 \quad (\text{A.4})$$

Allerdings kann die Übertragung schon vor Erreichen der Runde  $\kappa$  beendet sein, wenn das Transfervolumen  $s$  so klein ist, dass gilt:

$$s < \sum_{i=1}^{\kappa-1} N(i) \cdot L_{MSS} \quad (\text{A.5})$$

Allgemein werde der Index der letzten Runde mit  $I(s)$  bezeichnet. Um  $I(s)$  angeben zu können, werde zunächst die Funktion

$$J(i) = 1 + w \cdot \sum_{k=1}^{i-1} N(k) = 1 + w \cdot (2^{i-1} - 1) \quad (\text{A.6})$$

definiert, die den Index des ersten Pakets in Runde  $i$  liefert. Als Wertebereich für den Paketindex gilt  $J(i) \in \{1, 2, \dots, j_{max}\}$ , wobei

$$j_{max} = \left\lceil \frac{s}{L_{MSS}} \right\rceil \quad (\text{A.7})$$

den Index des letzten Pakets des Transfers repräsentiert. Umgekehrt kann damit  $I(s)$  aus

$$I(s) = \min(\max(i | J(i) \leq j_{max}), \kappa) \quad (\text{A.8})$$

ermittelt werden. Daraus folgt nach Auflösung von Gleichung (A.6) und Einsetzen von (A.4):

$$I(s) = \min \left( \left\lceil \text{ld} \left( \frac{1}{w} \cdot \left( \left\lceil \frac{s}{L_{MSS}} \right\rceil - 1 \right) + 1 \right) \right\rceil + 1, \left\lceil \text{ld} \left( \frac{1}{w} \cdot \left( \frac{\tau}{h} + 1 \right) \right) \right\rceil + 2 \right) \quad (\text{A.9})$$

Wenn nun angenommen wird, dass die Übertragung zum Zeitpunkt  $t = 0$  beginnt, gilt für den Zeitpunkt, an dem die ersten Daten aus Runde  $i$  beim Empfänger eintreffen:

$$t_R(i) = \frac{\tau}{2} + (\tau + h) \cdot (i - 1) \text{ für } i \in \{1, 2, \dots, \kappa\} \quad (\text{A.10})$$

Aus Empfängersicht beginnt die letzte Runde also zum Zeitpunkt  $t_R(I(j_{max}))$ . Hierzu muss nun noch die Bedienzeit für die Pakete in der letzten Runde addiert werden, sodass sich schließlich für die Gesamtdauer des Transfers folgender Ausdruck ergibt:

$$T_{min}(s) = t_R(I(s)) + \frac{s - (J(I(s)) - 1) \cdot L_{MSS}}{R_{max}} \quad (\text{A.11})$$

Nach Einsetzen der in (A.6) und (A.10) definierten Ausdrücke und Zusammenfassen von Termen folgt daraus schließlich die Transferdauer

$$T_{min}(s) = \frac{s}{R_{max}} + \frac{\tau}{2} + (\tau + h) \cdot (I(s) - 1) - h \cdot w \cdot (2^{I(s)-1} - 1) \quad (\text{A.12})$$

mit  $I(s)$  gemäß Gleichung (A.9). Der Anteil

$$T_{Offset}(s) = T_{min}(s) - \frac{s}{R_{max}} \quad (\text{A.13})$$

kann dabei als Offset betrachtet werden, der auf den *Slow-Start*-Mechanismus zurückzuführen ist und der sich auch als zusätzliche Verzögerung im Vergleich zur Transferdauer in einem idealisierten Flüssigkeits-Fluss-Modell ohne die Berücksichtigung konstanter Verzögerungskomponenten ergibt.

Im Fall  $w = 2$ , der in realen TCP-Implementierungen am häufigsten vorzufindenden ist [213], vereinfacht sich der Ausdruck nach Gleichung (A.9):

$$I(s) = \min \left( \left\lfloor \text{ld} \left( \left\lceil \frac{s}{L_{MSS}} \right\rceil + 1 \right) \right\rfloor, \left\lfloor \text{ld} \left( \frac{\tau}{h} + 1 \right) \right\rfloor + 1 \right) \quad (\text{A.14})$$

Interessant ist auch eine Betrachtung des Grenzwertes für  $R_{max} \rightarrow \infty$ , da dieser Wert eine untere Schranke für  $T_{min}(s)$  darstellt. Dies bedeutet gleichzeitig  $h \rightarrow 0$  und  $\kappa \rightarrow \infty$ , was

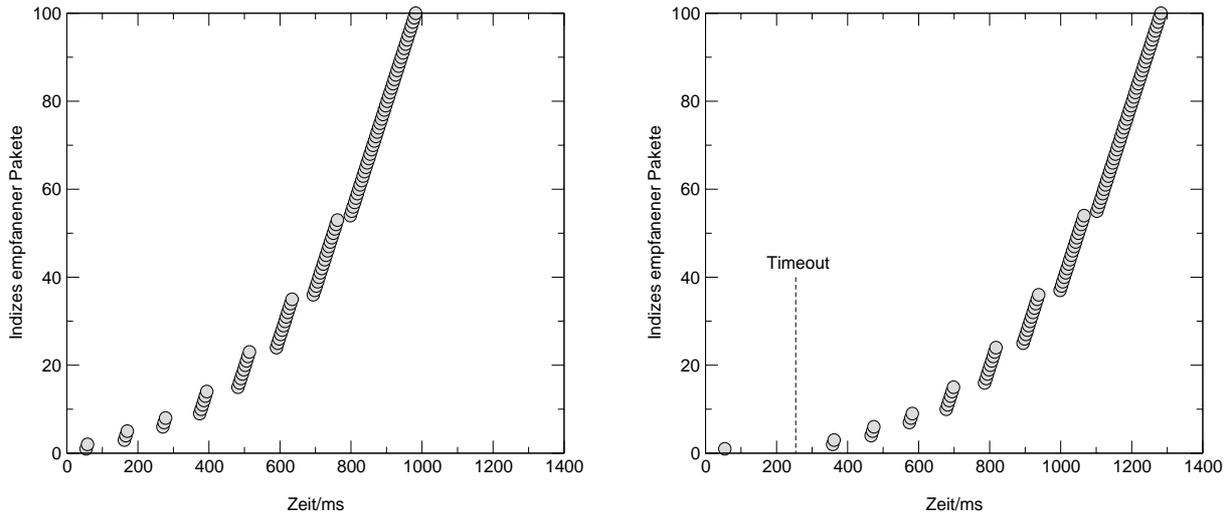
$$T_{min}(s) \rightarrow \frac{\tau}{2} + \tau \cdot \left\lfloor \text{ld} \left( \frac{1}{w} \cdot \left( \left\lceil \frac{s}{L_{MSS}} \right\rceil - 1 \right) + 1 \right) \right\rfloor \quad (\text{A.15})$$

als Grenzwert für die Transferdauer zur Folge hat.

### A.3 Minimale Transferzeit mit Delayed-ACK-Option

Die *Delayed-ACK*-Option auf der TCP-Empfängerseite bewirkt, dass nur für jedes zweite empfangene Paket eine Quittung gesendet wird. Dadurch wächst das CWnd, das mit jeder erhaltenen Quittung um  $L_{MSS}$  erhöht wird, nicht so stark wie im Fall ohne *Delayed ACK* und somit ist die Anzahl der pro Runde gesendeten Pakete geringer (siehe Bild A.3 für  $\tau = 100$  ms,  $h = 4$  ms). Die daraus resultierende Erhöhung der Transferdauer kann entweder approximativ (Abschnitt A.3.1) oder exakt (Abschnitt A.3.2) bestimmt werden. Die Herleitung soll dabei allgemein gehalten werden, indem davon ausgegangen wird, dass jedes  $q$ -te Paket bestätigt wird, sodass sich das üblicherweise implementierte *Delayed ACK* als Spezialfall  $q = 2$  ergibt.

Im Fall von  $w < q$  wird für die Pakete der ersten Runde erst nach Ablauf einer Zeitüberwachung beim Empfänger eine Bestätigung geschickt (Bild A.3, rechtes Diagramm). Dieser Fall,



**Bild A.3:** Zeitlicher Verlauf der Übertragung aus Sicht des Empfängers, *Delayed ACK* mit  $q = 2$ ,  $w = 2$  (links) bzw.  $w = 1$  (rechts)

der eher selten auftritt, da in den meisten TCP-Implementierungen  $q \leq 2$  und  $w = 2$  eingestellt ist [213], wird in Abschnitt A.3.3 behandelt, während für die Analyse in den Abschnitten A.3.1 und A.3.2  $w \geq q$  angenommen wird.

### A.3.1 Approximative Analyse

Dadurch, dass bei *Delayed ACK* nur noch jedes  $q$ -te Paket sofort bestätigt wird, erhöht sich das CWnd beim Sender zwar weiterhin exponentiell, der Anstieg erfolgt aber langsamer als bei  $q = 1$ . In Beiträgen, die sich mit der Leistung von TCP bei *Delayed ACK* beschäftigen (z. B. [56, 255]), wird dies meist durch die vereinfachende Annahme berücksichtigt, dass die Größe des CWnd am Ende von Runde  $i$  stets mit der Anzahl  $N(i)$  der in Runde  $i$  gesendeten Pakete identisch ist, deren Verlauf dann durch die Funktion  $\tilde{N}(i)$  mit reellem Wertebereich angenähert wird:

$$N(i) \approx \tilde{N}(i) = w \cdot \mu^{i-1} \tag{A.16}$$

Dabei leitet sich die Grundzahl, die dem Faktor entspricht, mit dem sich das CWnd in jeder Runde erhöht, direkt aus dem Parameter  $q$  ab:

$$\mu = 1 + \frac{1}{q} \tag{A.17}$$

Für den Fall  $q = 2$  ergibt sich also ein exponentieller Anstieg mit Mantisse 1,5. In Anlehnung an Gleichung (A.3) berechnet sich der Index  $\kappa$  der Runde, ab der das charakteristische Rundenverhalten spätestens endet, aus

$$\kappa \approx \min(i \mid \tilde{N}(i) \cdot h > \tau + h) \tag{A.18}$$

**Tabelle A.1:** Abhängigkeit charakteristischer Größen vom Rundenindex für  $q = 1$  bzw.  $q = 2$  (Approximation), jeweils für  $w = 2$

$i$	$q = 1$		$q = 2$ (Approximation)	
	$J(i)$	$N(i)$	$\lfloor \tilde{J}(i) \rfloor$	$\lfloor \tilde{N}(i) \rfloor$
1	1	2	1	2
2	3	4	3	3
3	7	8	6	4
4	15	16	10	6
5	31	32	17	10
6	63	64	27	15
7	127	128	42	22
8	255	256	65	34
9	511	512	99	51
10	1023	1024	150	76

und resultiert schließlich in dem Ausdruck

$$\kappa \approx \tilde{\kappa} = \left\lceil \log_{\mu} \left( \frac{1}{w} \cdot \left( \frac{\tau}{h} + 1 \right) \right) \right\rceil + 2 \quad (\text{A.19})$$

Entsprechend den Gleichungen (A.9) bzw. (A.6) ergeben sich nach dieser Approximation der Index des ersten Pakets in Runde  $i$  zu

$$J(i) \approx \tilde{J}(i) = 1 + w \cdot \sum_{k=1}^{i-1} \tilde{N}(k) = 1 + w \cdot \frac{\mu^{i-1} - 1}{\mu - 1} \quad (\text{A.20})$$

und damit der Index der letzten Runde zu

$$I(s) \approx \tilde{I}(s) = \min \left( \left\lceil \log_{\mu} \left( \frac{1}{w} \cdot \left( \left\lceil \frac{s}{L_{MSS}} \right\rceil - 1 \right) + 1 \right) \right\rceil + 1, \tilde{\kappa} \right) \quad (\text{A.21})$$

In Tabelle A.1 sind für den Fall  $w = 2$  die Werte von  $\lfloor \tilde{N}(i) \rfloor$  und  $\lfloor \tilde{J}(i) \rfloor$  bei Verwendung von *Delayed ACK* mit  $q = 2$  im Vergleich zu denen von  $N(i)$  und  $J(i)$  bei sofortiger Quittierung dargestellt. Der wesentlich langsamere Anstieg der Anzahl gesendeter Pakete bei *Delayed ACK* ist insbesondere für höhere Rundenindizes deutlich zu erkennen.

Die Gleichungen (A.10) und (A.11) können nun unverändert aus der Analyse für  $q = 1$  übernommen werden. Das Einsetzen der Ausdrücke nach (A.19) und (A.20) in (A.11) liefert schließlich

$$T_{min}(s) \approx \tilde{T}_{min}(s) = \frac{s}{R_{max}} + \frac{\tau}{2} + (\tau + h) \cdot (\tilde{I}(s) - 1) - h \cdot w \cdot \frac{\mu^{i-1} - 1}{\mu - 1} \quad (\text{A.22})$$

mit  $\tilde{I}(s)$  nach Gleichung (A.21) und  $\gamma$  nach Gleichung (A.17) als Approximation für die Transferdauer.

### A.3.2 Exakte Analyse

Die Größe des CWnd am Ende von Runde  $i$  (d. h. nach dem Eintreffen der letzten Bestätigung für ein Paket aus Runde  $i - 1$ ) werde mit  $W(i)$  bezeichnet, wobei  $W(1) = w$ . Ferner repräsentieren  $Z(i)$  den Index des letzten in Runde  $i$  gesendeten Pakets,  $A(i)$  den Index des letzten Pakets aus Runde  $i$ , das vom Empfänger sofort quittiert wird, und  $N_A(i)$  die Gesamtzahl der in den Runden 1 bis  $i$  gesendeten Quittungen. Dabei gelten die folgenden Zusammenhänge:

$$N_A(i) = \left\lfloor \frac{Z(i)}{q} \right\rfloor \quad (\text{A.23})$$

$$A(i) = N_A(i) \cdot q \quad (\text{A.24})$$

$$Z(i) = \begin{cases} W(1) & \text{für } i = 1 \\ A(i-1) + W(i) & \text{für } i > 1 \end{cases} \quad (\text{A.25})$$

$$W(i) = w + \begin{cases} 0 & \text{für } i = 1 \\ N_A(i-1) & \text{für } i > 1 \end{cases} \quad (\text{A.26})$$

Aus diesen Gleichungen kann die Rekursionsfunktion

$$Z(i) = w + \begin{cases} 0 & \text{für } i = 1 \\ (q+1) \cdot \left\lfloor \frac{Z(i-1)}{q} \right\rfloor & \text{für } i > 1 \end{cases} \quad (\text{A.27})$$

abgeleitet werden, woraus ebenfalls rekursive Ausdrücke für die bereits in Abschnitt A.2 eingeführten Größen

$$J(i) = \begin{cases} 1 & \text{für } i = 1 \\ Z(i-1) + 1 & \text{für } i > 1 \end{cases} \\ = \begin{cases} 1 & \text{für } i = 1 \\ 1 + w + (q+1) \cdot \left\lfloor \frac{J(i-1) - 1}{q} \right\rfloor & \text{für } i > 1 \end{cases} \quad (\text{A.28})$$

**Tabelle A.2:** Abhängigkeit charakteristischer Größen vom Rundenindex für  $q = 2$ ,  $w = 2$

$i$	$J(i)$	$N(i)$	$Z(i)$	$A(i)$	$N_A(i)$	$W(i)$
1	1	2	2	2	1	2
2	3	3	5	4	2	3
3	6	3	8	8	4	4
4	9	6	14	14	7	6
5	15	9	23	22	11	9
6	24	12	35	34	17	13
7	36	18	53	52	26	19
8	54	27	80	80	40	28
9	81	42	122	122	61	42
10	123	63	185	184	92	63

und

$$N(i) = \begin{cases} w & \text{für } i = 1 \\ Z(i) - Z(i-1) & \text{für } i > 1 \end{cases} = J(i+1) - J(i) \quad (\text{A.29})$$

folgen, die jeweils für Runde  $i$  den Index des ersten Pakets sowie die Anzahl der in dieser Runde gesendeten Pakete angeben. Für den wichtigen Spezialfall  $q = 2$ ,  $w = 2$  sind in Tabelle A.2 die Werte der genannten Größen in Abhängigkeit vom Rundenindex aufgelistet, die mit den in [125] angegebenen Ergebnissen übereinstimmen. Daraus wird u. a. ersichtlich, dass die Werte für  $N(i)$  und  $W(i)$ , die bei der approximativen Analyse in Abschnitt A.3.1 als gleich angenommen wurden, für manche Rundenindizes um eins (allgemein um bis zu  $q - 1$ ) differieren. Darüber hinaus zeigt sich, dass der Anstieg der Werte noch langsamer erfolgt als bei approximativen Analyse angenommen.

Für  $\kappa$  gilt hier Gleichung (A.3) entsprechend, d. h. mit modifiziertem  $N(i)$  gemäß (A.29). Da sich  $J(i)$  gegenüber dem Fall ohne *Delayed ACK* geändert hat, muss auch die Funktion, die den Index der letzten Runde liefert, angepasst werden, wobei nun kein geschlossener Ausdruck mehr angegeben werden kann, sondern auf die Definitionsgleichung (A.8) zurückgegangen werden muss.

Bei der Berechnung des Empfangszeitpunkts  $t_R(i)$  für das erste Paket in Runde  $i$  kann Gleichung (A.10) weitgehend aus der Analyse für den Fall ohne *Delayed ACK* übernommen werden. Allerdings muss nun berücksichtigt werden, dass sich die jeweils auf das erste Paket einer Runde bezogene Zykluszeit in den Runden, in denen dieses nicht sofort quittiert wird, um  $h$

erhöht. Dies wirkt sich in Form eines um  $h$  verschobenen Zeitpunkts für den Empfangsbeginn des ersten Pakets der nächsten Runde aus:

$$t_R(i) = \frac{\tau}{2} + (\tau + h) \cdot (i - 1) + h \cdot \sum_{k=1}^{i-1} \left( \left\lceil \frac{J(k)}{q} \right\rceil - \left\lfloor \frac{J(k)}{q} \right\rfloor \right) \quad (\text{A.30})$$

Um schließlich die Transferdauer zu berechnen, kann Gleichung (A.11) unverändert übernommen werden, so dass sich nach Einsetzen der oben abgeleiteten Größen und Zusammenfassen von Termen folgender Ausdruck ergibt:

$$T_{min}(s) = \frac{s}{R_{max}} + \frac{\tau}{2} + (\tau + h) \cdot (I(s) - 1) - h \cdot \left( J(I(s)) - 1 - \sum_{k=1}^{I(s)-1} \left( \left\lceil \frac{J(k)}{q} \right\rceil - \left\lfloor \frac{J(k)}{q} \right\rfloor \right) \right) \quad (\text{A.31})$$

mit

$$I(s) = \min \left( \max \left( i \mid J(i) \leq \left\lceil \frac{s}{L_{MSS}} \right\rceil \right), \min(i \mid N(i) \cdot h > \tau + h) \right) \quad (\text{A.32})$$

Die Spezialisierung auf  $q = 2$ ,  $w = 2$  bringt hier keine wesentliche Vereinfachung der Ausdrücke, weshalb auf eine separate Darstellung der Ergebnisse für diesen Fall verzichtet wird. Im Übrigen wird deutlich, dass auch  $q = 1$  als Spezialfall in Gleichung (A.31) abgedeckt ist.

### A.3.3 Berücksichtigung der Zeitüberwachung beim Empfänger

Trifft beim Empfänger ein Paket ein, das nicht sofort bestätigt wird, wird ein Timer gesetzt, der nach der Dauer  $T_{DelACK}$  abläuft und dann das Senden einer Quittung für alle bis dahin empfangenen Pakete bewirkt. Im gegebenen Modell kann ein solcher Timeout nur am Beginn der Übertragung auftreten<sup>1</sup>, und zwar nur falls  $w < q$ . Vereinfachend werde angenommen, dass der Initialwert für den Wiederholungstimer beim Sender größer ist als  $T_{DelACK}$ , sodass keine Beeinflussung durch wiederholte Pakete – die beim Empfänger immer sofort quittiert werden müssen – entsteht.

Im Fall  $w < q$  können sogar – in Abhängigkeit von der Burstgröße  $s$  sowie von  $q - w$  – mehrere Runden erforderlich sein, bevor entweder die letzte Runde stattfindet oder eine Runde, in der eine Bestätigung ohne Timeout gesendet werden kann. Die Anzahl dieser „Timeout-Run-

<sup>1</sup> Während im vorliegenden Modell davon ausgegangen wird, dass der *Timer* jedes Mal zurückgesetzt wird, wenn eine Bestätigung gesendet wird, existieren auch Implementierungen, die auf ein solches Rücksetzen verzichten [125].

den“ werde mit  $v$  bezeichnet. Um  $v$  zu bestimmen, werde zunächst der Index  $Z'(i)$  des letzten in der Timeout-Runde  $i$  gesendeten Pakets betrachtet:

$$Z'(i) = \sum_{k=1}^i (w+k-1) = i \cdot w + \frac{1}{2} \cdot i \cdot (i-1) \quad (\text{A.33})$$

Dabei muss berücksichtigt werden, dass das CWnd sich mit jeder gesendeten Quittung und damit nach jeder Timeout-Runde um eins erhöht. Aus den Randbedingungen  $Z'(v) < j_{max}$  (für  $v \geq 1$ ) und  $0 \leq v \leq q - w$  ergibt sich dann folgender Ausdruck für  $v$ :

$$\begin{aligned} v &= \max(i \mid ((Z'(i) < j_{max}) \cap (1 \leq i \leq q - w)) \cup (i = 0)) \\ &= \min\left(\left\lceil \frac{1}{2} \cdot \left(1 - 2w + \sqrt{(2w-1)^2 + 8 \left\lceil \frac{s}{L_{MSS}} \right\rceil}\right) \right\rceil - 1, \max(q - w, 0)\right) \end{aligned} \quad (\text{A.34})$$

Jede Timeout-Runde trägt  $\tau + h + T_{DelACK}$  zur Gesamtverzögerung bei. An die Timeout-Runden schließt sich definitionsgemäß mindestens eine weitere Runde an. Die Dauer für die Übertragung in diesen „normalen“ Runden kann mit Hilfe der für den Fall  $w \geq q$  abgeleiteten Gleichungen (A.31) (exakte Analyse) bzw. (A.22) (approximative Analyse) berechnet werden. Dazu gilt es allerdings zu berücksichtigen, dass beim Eintritt in den „Normalmodus“ das CWnd bereits den Wert

$$w' = w + v \quad (\text{A.35})$$

hat und dass schon einige Pakete in den Timeout-Runden übertragen worden sind und daher gemäß Gleichung (A.33) nur noch ein Restvolumen

$$s' = s - \left(v \cdot w + \frac{1}{2} \cdot v \cdot (v-1)\right) \cdot L_{MSS} \quad (\text{A.36})$$

als Burstgröße in die Berechnung eingeht. Insgesamt ergibt sich also eine gegenüber dem Fall  $w \geq q$  modifizierte Transferdauer

$$T_{min}'(s) = v \cdot (\tau + h + T_{DelACK}) + T_{min}(s = s', q = 1, w = w') \quad (\text{A.37})$$

Von besonderer Relevanz innerhalb des Parameterraums  $w < q$  ist der Spezialfall  $q = 2$ ,  $w = 1$ . Hierfür vereinfachen sich die oben abgeleiteten Größen erheblich, sodass

$$T_{min}'(s) = \begin{cases} \frac{s}{R_{max}} + \frac{\tau}{2} & \text{für } s \leq L_{MSS} \\ \tau + h + T_{DelACK} + T_{min}(s = s - L_{MSS}, q = 1, w = 2) & \text{für } s > L_{MSS} \end{cases} \quad (\text{A.38})$$

als Ausdruck für die Transferdauer resultiert.

## A.4 Stochastische Betrachtung mit zufälliger Burstgröße

Die für die unterschiedlichen Parameter abgeleiteten Ausdrücke gemäß Gleichungen (A.12), (A.31) und (A.38) können als untere Grenze der Transferdauer über ein IP-Netz bei gegebener Burstgröße  $s$  interpretiert werden. In einer stochastischen Betrachtung ist die Burstgröße jedoch nicht fest vorgegeben, sondern kann durch eine Zufallsvariable  $B$  mit Verteilungsfunktion  $F_B(s) = P(B \leq s)$  und Mittelwert  $b = E[B]$  beschrieben werden. Dadurch ergibt sich auch der Minimalwert der Transferdauer als Zufallsvariable  $T_{min}(B)$ , deren bedingter Erwartungswert  $E[T_{min}(B)|B = s] = T_{min}(s)$  den oben genannten Gleichungen folgt.

### A.4.1 Erwartungswert

Neben dem bedingten Mittelwert ist auch der Erwartungswert  $\overline{T_{min}}$  von  $T_{min}(B)$  interessant, der bei gegebener Verteilungsfunktion der Burstgröße aus  $T_{min}(s)$  berechnet werden kann:

$$\overline{T_{min}} = E[T_{min}(B)] = \int_0^{\infty} E[T_{min}(B)|B = s] \cdot dF_B = \int_0^{\infty} T_{min}(s) \cdot dF_B \quad (\text{A.39})$$

Für die weitere Berechnung, die nur für  $q \leq w$  gezeigt wird, erweist sich die Einführung zweier Abkürzungen als günstig:

$$\Omega(i) = (\tau + h) \cdot (i - 1) - h \cdot (J(i) - 1) + h \cdot \sum_{k=1}^{i-1} \left( \left\lceil \frac{J(k)}{q} \right\rceil - \left\lfloor \frac{J(k)}{q} \right\rfloor \right) \quad (\text{A.40})$$

$$\Lambda(i) = (J(i) - 1) \cdot L_{MSS} \quad (\text{A.41})$$

Mit Hilfe von  $\Omega(i)$  ( $i = 1, 2, \dots$ ) kann die Transferzeit nach Gleichung (A.12) (für  $q = 1$ ) bzw. Gleichung (A.31) (für  $q > 1$ ) wie folgt ausgedrückt werden:

$$T_{min}(s) = \frac{s}{R_{max}} + \frac{\tau}{2} + \Omega(I(s)) \quad (\text{A.42})$$

$\Lambda(i)$  ( $i = 1, 2, \dots$ ) dient dazu, die Werte von  $s$  zu beschreiben, an denen  $T_{min}(s)$  einen Sprung aufweist, weil eine zusätzliche Runde (mit Index  $i$ ) erforderlich wird. Darüber hinaus wird noch  $\kappa$  nach Gleichung (A.4) (für  $q = 1$ ) bzw. (A.3) unter Verwendung von (A.29) (für  $q > 1$ ) benötigt. Damit kann dann die mittlere Minimaltransferdauer bestimmt werden:

$$\begin{aligned}
\overline{T}_{min} &= \frac{b}{R_{max}} + \frac{\tau}{2} + \int_0^{\infty} \Omega(I(s)) \cdot \frac{dF_B(s)}{ds} \cdot ds \\
&= \frac{b}{R_{max}} + \frac{\tau}{2} + \sum_{i=1}^{\kappa-1} \Omega(i) \cdot \int_{\Lambda(i)}^{\Lambda(i+1)} \frac{dF_B(s)}{ds} \cdot ds + \Omega(\kappa) \cdot \int_{\Lambda(\kappa)}^{\infty} \frac{dF_B(s)}{ds} \cdot ds \\
&= \frac{b}{R_{max}} + \frac{\tau}{2} + \sum_{i=1}^{\kappa-1} (\Omega(i) \cdot (F_B(\Lambda(i+1)) - F_B(\Lambda(i)))) + \Omega(\kappa) \cdot (1 - F_B(\Lambda(\kappa))) \\
&= \frac{b}{R_{max}} + \frac{\tau}{2} + \Omega(\kappa) - \sum_{i=2}^{\kappa} ((\Omega(i) - \Omega(i-1)) \cdot F_B(\Lambda(i))) \tag{A.43}
\end{aligned}$$

Im Fall ohne *Delayed ACK* ( $q = 1$ ) ergeben sich für  $\Omega(i)$  und  $\Lambda(i)$  die Ausdrücke

$$\Omega(i) = (\tau + h) \cdot (i - 1) - h \cdot w \cdot (2^{i-1} - 1) \tag{A.44}$$

$$\Lambda(i) = w \cdot (2^{i-1} - 1) \cdot L_{MSS} \tag{A.45}$$

Damit kann Gleichung (A.43) für diesen Fall weiter vereinfacht werden:

$$\overline{T}_{min} = \frac{b}{R_{max}} + \frac{\tau}{2} + \Omega(\kappa) - \sum_{i=2}^{\kappa} ((\tau + h - h \cdot w \cdot 2^{i-2}) \cdot F_B(\Lambda(i))) \tag{A.46}$$

Es wird deutlich, dass der Erwartungswert der minimalen Transferdauer von der Verteilung der Burstgröße und nicht nur von deren Mittelwert abhängt. Dies ist insbesondere von Interesse, da dieser Erwartungswert der minimalen Transferdauer die Untergrenze für die – z. B. vom Angebot auf einem zentralen Engpasslink abhängige – mittlere Transferdauer darstellt.

## A.4.2 Verteilungsfunktion

Wenn die Verteilungsfunktion der Burstgröße bekannt ist, kann schließlich auch die Verteilungsfunktion  $F_{T_{min}}(t)$  der minimalen Transferdauer angegeben werden. Aus der bereits beobachteten Tatsache, dass  $T_{min}(s)$  sprunghaft ansteigt, wenn eine zusätzliche Runde für die Übertragung erforderlich wird, folgt im Fall  $q \leq w$  eine abschnittsweise definierte Verteilungsfunktion:

$$F_{T_{min}}(t) = P(T_{min}(s) \leq t) = \sum_{i=1}^{\kappa-1} \Psi_i(t) + \hat{\Psi}(t) \tag{A.47}$$

mit

$$\Psi_i(t) = \begin{cases} F_S(\Lambda(i) + (t - t_R(i)) \cdot R_{max}) & t_R(i) \leq t < t_R(i) + N(i) \cdot h \\ F_S(\Lambda(i + 1)) & t_R(i) + N(i) \cdot h \leq t < t_R(i + 1) \\ 0 & \text{sonst} \end{cases} \quad (\text{A.48})$$

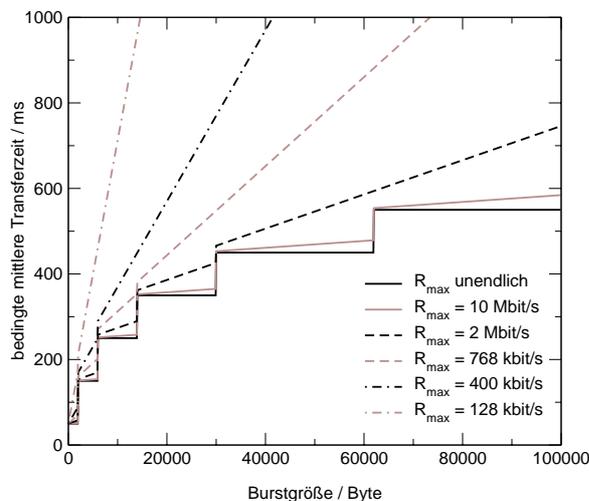
$$\hat{\Psi}(t) = \begin{cases} F_S(\Lambda(\kappa) + (t - t_R(\kappa)) \cdot R_{max}) & t \geq t_R(\kappa) \\ 0 & \text{sonst} \end{cases} \quad (\text{A.49})$$

Für  $q = 1$  gelten dabei für  $N(i)$ ,  $J(i)$ ,  $t_R(i)$  und  $\Lambda(i)$  die in den Abschnitten A.2 bzw. A.4.1 abgeleiteten Ausdrücke gemäß Gleichungen (A.2), (A.6), (A.10) bzw. (A.45).

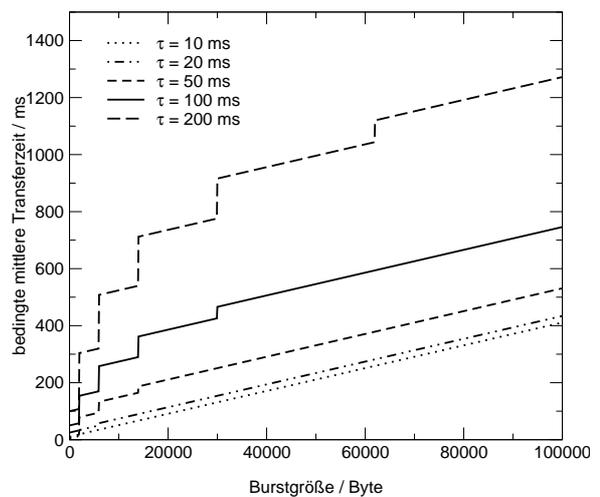
## A.5 Numerische Auswertung

Abschließend sollen durch numerische Auswertung der oben hergeleiteten Gleichungen die Abhängigkeiten von den gegebenen Randbedingungen des Systems veranschaulicht werden. In allen Fällen wird  $L_{MSS} = 1\,000$  Byte angenommen.

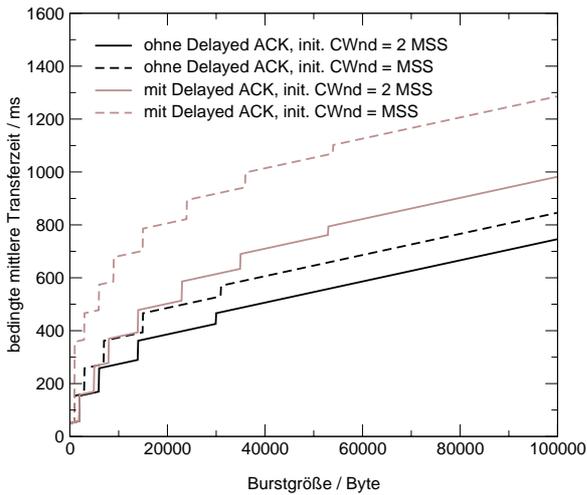
Für die minimale Transferdauer bei gegebener Burstgröße sind in erster Linie die Zugangsrate  $R_{max}$  sowie der Absolutanteil der Verzögerung  $\tau$  wichtige Einflussgrößen. In Bild A.4 ist daher  $T_{min}(s)$  in für verschiedene Zugangsraten bei konstanter absoluter Verzögerung  $\tau = 100$  ms dargestellt, während Bild A.5 die minimale Transferdauer für unterschiedliche Werte von  $\tau$  bei gleicher Zugangsrate  $R_{max} = 2$  Mbit/s zeigt. In beiden Diagrammen ist der treppenartige Verlauf von  $T_{min}(s)$  über der Burstgröße, der durch das beschriebene Rundenverhalten von TCP ausgelöst wird, deutlich zu erkennen. Die Höhe der „Treppenstufe“ wird dabei von  $\tau$  bestimmt. Zwischen den Sprungstellen steigt  $T_{min}(s)$  linear an, wobei die Steigung proportional zu  $R_{max}$  ist. Dies bedeutet, dass für kleinere Zugangsraten der treppenartige



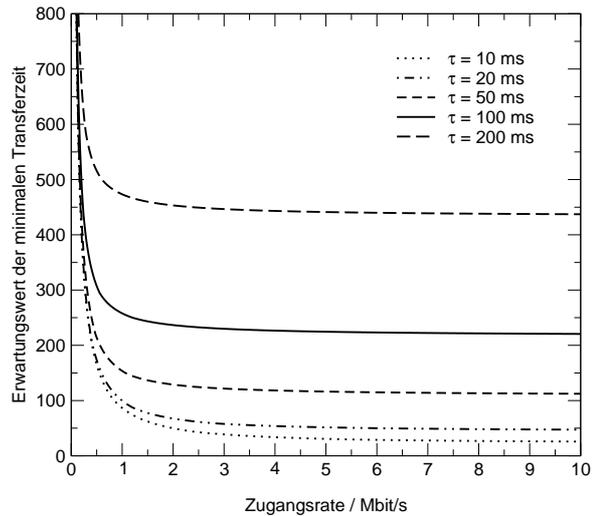
**Bild A.4:** Minimale Transferzeit für verschiedene Zugangsraten ( $\tau = 100$  ms)



**Bild A.5:** Minimale Transferzeit in Abhängigkeit von  $\tau$  ( $R_{max} = 2$  Mbit/s)



**Bild A.6:** Minimale Transferzeit für unterschiedliche TCP-Parameter

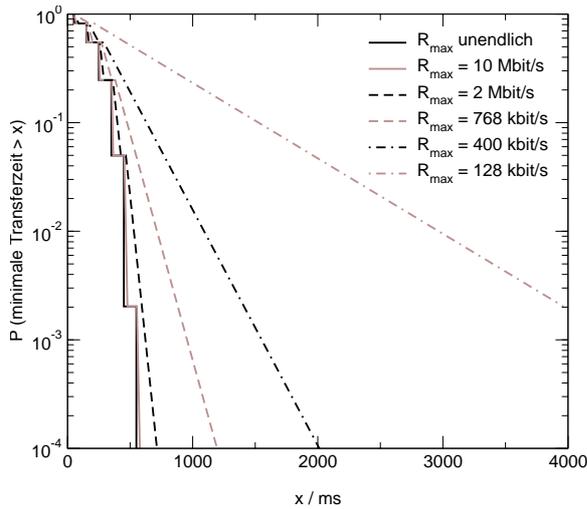


**Bild A.7:** Gemittelte minimale Transferzeit für neg.-exp. verteilte Burstgröße

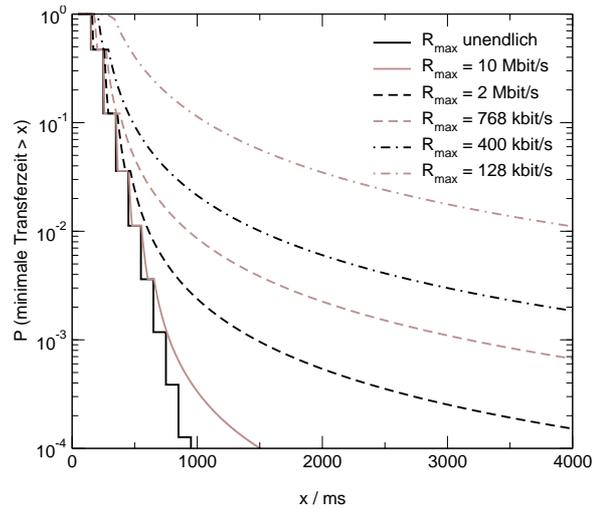
Verlauf kaum noch wahrnehmbar ist und außerdem die Zahl der Sprungstellen – entsprechend dem Wert von  $\kappa$  nach Gleichung (A.4) – abnimmt. Für größere Zugangsraten hingegen zeigt sich eine deutliche Verschiebung der Kurve für die Transferdauer im Vergleich zu der Ursprungsgeraden mit gleicher Steigung, die sich bei Vernachlässigung des TCP-Verhaltens ergeben würde.

Neben den Systemgrößen  $R_{max}$  und  $\tau$  hängt die Transferdauer von den eingestellten TCP-Parametern ab. Bild A.6 veranschaulicht dies am Beispiel von vier typischen Kombinationen von  $q$  und  $w$  für  $R_{max} = 2$  Mbit/s und  $\tau = 100$  ms. Im Fall  $q = 1$  (also bei ausgeschaltetem *Delayed ACK*) bewirkt ein gegenüber dem Standardwert  $w = 2$  reduziertes initiales CWnd ( $w = 1$ ) nur eine geringe Erhöhung der minimalen Transferdauer. Insbesondere wird deutlich, dass die Zahl der „Treppenstufen“ nicht von  $w$  abhängt, während die Verwendung von *Delayed ACK* (also  $q = 2$ ) die Zahl der Sprungstellen erhöht und damit die Transferdauer verlängert. Schließlich ist Bild A.6 auch der Spezialfall  $w = 1$  und  $q = 2$  dargestellt. In diesem Fall wird für das erste Paket erst nach Ablauf eines Timers beim Empfänger eine Bestätigung gesendet. Dies bewirkt näherungsweise eine Erhöhung der Transferdauer um  $T_{DelACK} + \tau + h$  (hier  $T_{DelACK} = 200$  ms) gegenüber dem Fall  $q = 2, w = 2$ .

Als Leistungsmaß bei dynamischem TCP-Verkehr wird häufig die mittlere Transferzeit verwendet. Bei Vergleichen eignet sich als Referenz der in Abschnitt A.4.1 abgeleitete Erwartungswert der minimalen Transferzeit bei gegebener Burstgrößenverteilung. Für den Fall einer negativ-exponentiell verteilten Burstgröße sind diese Erwartungswerte in Bild A.7 in Abhängigkeit von  $R_{max}$  und  $\tau$  dargestellt. Zunächst wird deutlich, dass eine Erhöhung der Zugangsrate nur in einem gewissen Bereich eine signifikante Reduzierung der mittleren minimalen Transferdauer bewirkt. Für weiter zunehmende Werte von  $R_{max}$  nähern sich die Kurven einem Wert, der von  $\tau$  abhängt.



**Bild A.8:** CCDF der minimalen Transferzeit für neg.-exp. verteilte Burstgröße



**Bild A.9:** CCDF der minimalen Transferzeit für Pareto-vertelile Burstgröße ( $\alpha = 1.6$ )

Während der Verlauf der minimalen Transferdauer über der Burstgröße grundsätzlich unabhängig von der Burstgrößenverteilung ist, hat bei einer stochastischen Betrachtung der minimalen Transferzeit die Burstgrößenverteilung durchaus einen Einfluss. Dies zeigt sich auch in Bild A.8 und Bild A.9, wo die komplementäre Verteilungsfunktion (*complementary cumulative distribution function*, CCDF) der minimalen Transferzeit bei negativ-exponentieller bzw. Pareto-verteilter Burstgröße (im zweiten Fall mit Formparameter  $\alpha = 1.6$ ) mit Mittelwert  $b = 10000$  Byte dargestellt ist. Auch hier zeigt sich insbesondere bei großen Zugangsraten wieder ein treppenartiger Verlauf, der im Fall einer Pareto-verteilter Burstgröße ausgeprägter ist. Die Kurven in Bild A.8 und Bild A.9 können als Untergrenze für die komplementäre Transferzeitverteilung über ein beliebig belastetes Netz angesehen werden und können daher als Anhaltspunkt für die Beurteilung eines Systems (z. B. eines differenzierenden IP-Knotens wie in Kapitel 6) unter realen Randbedingungen herangezogen werden.

## Anhang B

# TCP-Simulationen für einen IP-Netzknoten ohne Dienstgütedifferenzierung

In den nachfolgenden Untersuchungen werden einige grundlegende Effekte aufgezeigt, die entstehen, wenn mehrere TCP-Verbindungen über einen gemeinsamen Engpasslink geführt werden. Hierzu wird von dem in Abschnitt 6.2.1 beschriebenen Modell eines Ausgangsmoduls in einem IP-Netzknoten ausgegangen, das den Zugang zu einem Engpasslink ermöglicht. An die Stelle des WEDD-Verfahrens tritt dabei die FIFO-Abarbeitungsdisziplin mit DT-Puffermanagement, sodass hier ein Szenario ohne Dienstgütedifferenzierung vorliegt. Die in den Abschnitten 4.2.2.1 und 6.2.1 genannten Standardwerte der Systemparameter (insbesondere „New Reno“ als TCP-Version,  $L_{MSS} = 1000$  Byte,  $S_{TCP} = 32\,000$  Byte,  $R_{max} = 2$  Mbit/s und  $\tau = 100$  ms) werden auch in den nachfolgenden Untersuchungen verwendet, sofern nicht andere Angaben gemacht werden.

Die Studien werden alle simulativ mit Hilfe der in [34] beschriebenen Simulationsumgebung und auf der Basis eines darauf aufsetzenden TCP-Moduls durchgeführt [35, 116, 180]. Zunächst folgen Untersuchungen, in denen ungesättigte TCP-Quellen (siehe Abschnitt 4.2.2.2) verwendet werden. Anschließend wird dynamischer TCP-Verkehr untersucht, bei dem Ankunftsverhalten und Volumen von Datenbursts, die jeweils innerhalb einer TCP-Verbindung übertragen werden, mit Hilfe eines M/G/ $\infty$ -Modells gemäß Abschnitt 4.2.2.3 beschrieben werden.

## B.1 Untersuchungen mit ungesättigten Quellen

Ungesättigte TCP-Quellen geben zum einen das Verhalten von TCP-Verbindungen wieder, bei denen ein großes Datenvolumen übertragen wird. Zum anderen lässt sich mit ihnen zeigen, wie sich die durch die TCP-Überlaststeuerung verursachte Rückkopplung des aktuellen Netzzustands auf relevante Leistungsmaße auswirkt.

### B.1.1 Einfluss von Verkehrs- und Systemparametern

Zu den Parametern, deren Einfluss in diesem Abschnitt untersucht wird, zählen einerseits die während einer Simulation konstante Anzahl  $n$  von ungesättigten Quellen als Verkehrsparameter und andererseits als Systemparameter Puffergröße  $S$  und Bedienrate  $C$  auf dem Engpasslink sowie die konstante Verzögerung  $\tau$ , die im Wesentlichen als Signallaufzeit auf dem Hin- und dem Rückkanal interpretiert werden kann. Als Leistungsmaß wird in erster Linie der Summennutzdurchsatz aller TCP-Verbindungen verwendet, der auf die Linkrate normiert wird.

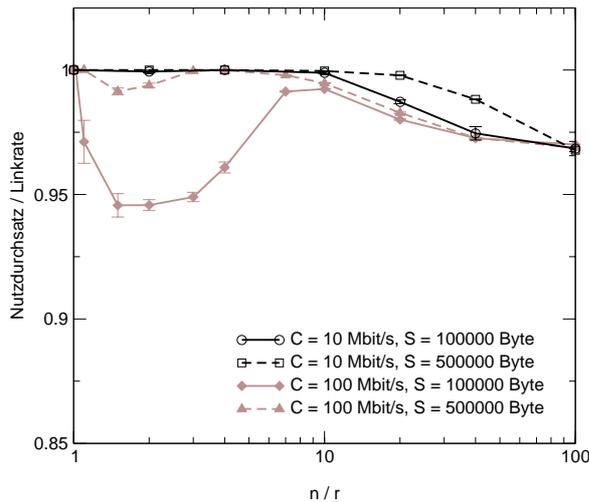
#### B.1.1.1 Einfluss der Anzahl von Quellen

Bei der Untersuchung der Auswirkungen der Quellenanzahl  $n$  kommt dem Ratenverhältnis

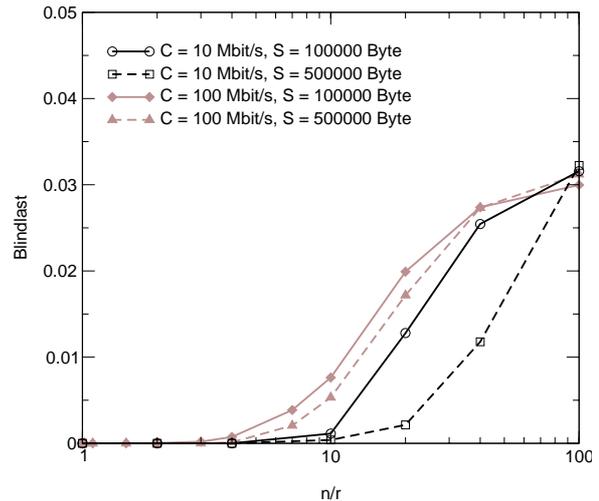
$$r = \frac{C}{R_{max}} \quad (\text{B.1})$$

eine besondere Bedeutung zu, weil erst ab  $n > r$  der zentrale Link einen Engpass darstellt. Daher sind in den Diagrammen dieses Abschnitts die Ergebnisse nicht über der absoluten Anzahl von Quellen, sondern über dem Verhältnis  $n/r$  aufgetragen.

In Bild B.1 sind die Ergebnisse für den Summennutzdurchsatz in Abhängigkeit von der Quellenzahl für unterschiedliche Werte von Linkrate  $C$  und Puffergröße  $S$  dargestellt. Bei  $C = 10$  Mbit/s ergibt sich ab einem bestimmten Wert von  $n/r$  ein kontinuierlicher Rückgang des Nutzdurchsatzes mit steigender Quellenzahl. Allerdings wird selbst bei sehr vielen Quellen ein Wert von 95% der Linkrate nicht unterschritten. Ein ähnlicher Rückgang ist auch bei  $C = 100$  Mbit/s zu verzeichnen, jedoch erst bei einer größeren Anzahl von Quellen. Zu erklären ist dieser Rückgang mit der in Bild B.2 aufgetragenen Blindlast, die sich als die Differenz der mittleren Linkauslastung (die gleichbedeutend mit dem normierten Bruttodurchsatz ist) und des normierten Nutzdurchsatzes ergibt. Da die Auslastung im genannten Bereich immer 1 ist, korrespondiert die Reduktion des Nutzdurchsatzes direkt mit der Zunahme der Blindlast. Diese ist auf unnötig wiederholte Segmente zurückzuführen. Zu solchen unnötigen Wiederholungen kommt es vor allem im Zusammenhang mit Timeouts (siehe Abschnitt 4.2.2.1), deren Häufigkeit mit steigender Quellenzahl aufgrund der zunehmenden Verlustwahrscheinlichkeit (siehe Bild B.3) und des abnehmenden Mittelwerts des CWnd steigt.



**Bild B.1:** Summennutzdurchsatz in Abhängigkeit von der Quellenanzahl



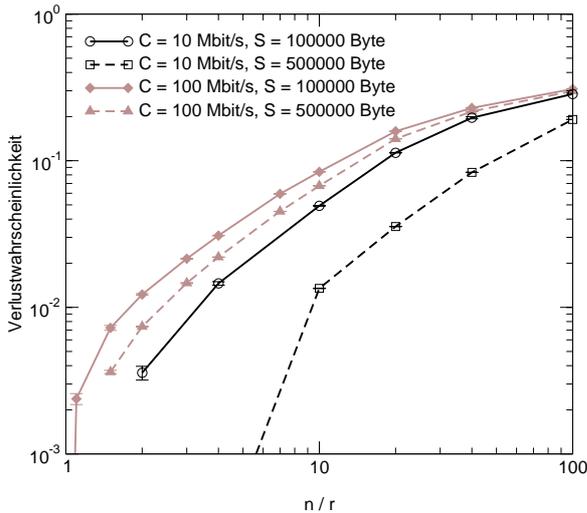
**Bild B.2:** Blindlast in Abhängigkeit von der Quellenanzahl

Außer der Abnahme des Summennutzdurchsatzes mit steigender Quellenanzahl ist in Bild B.1 jedoch noch ein weiterer Effekt zu beobachten. Bei  $C = 100$  Mbit/s bleibt der Nutzdurchsatz um bis zu 5% unter der Linkrate, wenn die Quellenanzahl nur wenig größer ist als  $r$ . Das Einbrechen des Nutzdurchsatzes bei Überschreiten der Grenze  $n = r$  ist um so stärker, je kleiner der Puffer ist. Die Tatsache, dass der Effekt nur bei  $C = 100$  Mbit/s auftritt, deutet darauf hin, dass nicht der Absolutwert von  $S$  entscheidend ist, sondern der Quotient  $S/C$ , der die maximale Paketverzögerung im Puffer wiedergibt. Dass sich ein kleiner Wert von  $S/C$  negativ auf die Leistung auswirkt, liegt in diesem Fall nicht an der Blindlast, die im relevanten Bereich verschwindend gering ist. Der Effekt ist vielmehr darauf zurückzuführen, dass eine Reduktion der Senderate einer Quelle infolge eines Paketverlusts nicht rechtzeitig von den anderen Quellen kompensiert werden kann, sodass es kurzzeitig zu einer vollständigen Entleerung des Puffers kommt. Dies spiegelt sich auch in den Ergebnissen für die mittlere Durchlaufzeit wieder (Bild B.4). In allen Fällen kommt die mittlere Durchlaufzeit ab etwa  $n = 10 \cdot r$  in die Nähe des Maximalwerts (gepunktete Linien in Bild B.4), bleibt aber für eine geringere Quellenzahl deutlich darunter. Für  $C = 100$  Mbit/s,  $S = 100\,000$  Byte ist dieser Maximalwert allerdings am geringsten ( $S/C = 8$  ms), sodass auch die mittlere Paketdurchlaufzeit bei kleinen Werten von  $n$  gegenüber den anderen Fällen am kleinsten ist.

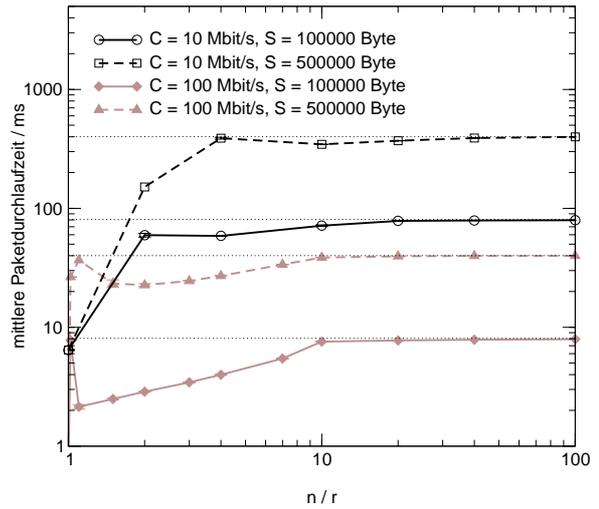
### B.1.1.2 Einfluss der Puffergröße

Die Abhängigkeit von der auf die Linkrate normierten Puffergröße  $S/C$  wird in Bild B.5 weiter verdeutlicht, wo der Summennutzdurchsatz über diesem Quotienten aufgetragen ist. Bei einer geringen Quellenzahl<sup>1</sup>  $n = 2 \cdot r$  bricht sowohl bei  $C = 10$  Mbit/s als auch bei  $C = 100$  Mbit/s der Summennutzdurchsatz mit abnehmender Puffergröße ein. Umgekehrt

<sup>1</sup> Die Wertepaare  $C = 10$  Mbit/s,  $n = 10$  sowie  $C = 100$  Mbit/s,  $n = 100$  entsprechen jeweils  $n/r = 2$ , während  $C = 10$  Mbit/s,  $n = 100$  sowie  $C = 100$  Mbit/s,  $n = 1000$  jeweils auf  $n/r = 20$  führen.



**Bild B.3:** Verlustwahrscheinlichkeit in Abhängigkeit von der Quellenanzahl



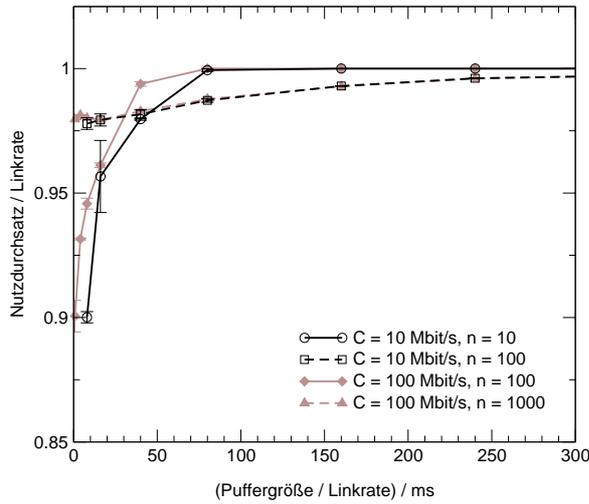
**Bild B.4:** Mittlere Paketdurchlaufzeit in Abhängigkeit von der Quellenanzahl

erreicht der Summennutzdurchsatz für beide Werte von  $C$  bei etwa  $S/C = 80$  ms die Sättigung, d. h. mit einer weiteren Vergrößerung des Puffers lässt sich keine Steigerung mehr erzielen. Dieser Wert entspricht bei  $C = 100$  Mbit/s einer Puffergröße von  $10^6$  Byte, während bei  $C = 10$  Mbit/s eine Kapazität von  $10^5$  Byte ausreicht, um den maximalen Nutzdurchsatz zu erzielen. Für eine größere Anzahl von Quellen ( $n = 20 \cdot r$ ) ergibt sich ein viel flacherer Anstieg des Nutzdurchsatzes mit zunehmender Puffergröße. Dies bedeutet einerseits, dass der Einbruch bei kleinen Werten von  $S/C$  nicht so deutlich ausfällt. Andererseits wird in diesem Fall erst bei sehr großem Puffer die Sättigung erreicht.

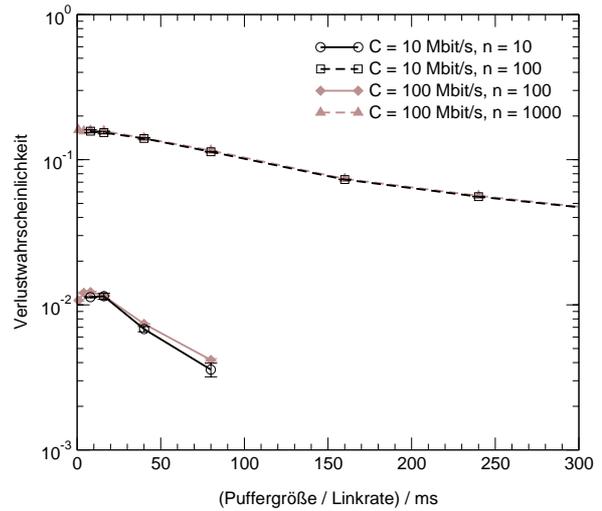
Der Blick auf die Verlustwahrscheinlichkeit macht deutlich, dass diese bei einer Vergrößerung des Puffers zwar zurückgeht (Bild B.6). Jedoch ist der Rückgang weit weniger stark als in Szenarien mit büschelförmigem Verkehr, in denen keine Berücksichtigung von Rückkopplungen stattfindet [31]. Außerdem ist die Reduktion der Verlustwahrscheinlichkeit mit einem Ansteigen der mittleren Durchlaufzeit verbunden (Bild B.7). Dieser Anstieg verläuft näherungsweise linear, wobei die mittlere Durchlaufzeit nur knapp unter der durch  $S/C$  gegebenen Obergrenze (gepunktete Linie) bleibt. Dies deutet darauf hin, dass der Puffer immer fast komplett gefüllt ist (siehe auch Abschnitt B.1.3).

Ein besonderes Verhalten ist in den Fällen  $C = 10$  Mbit/s,  $n = 10$  sowie  $C = 100$  Mbit/s,  $n = 100$  zu beobachten. Dort verschwinden ab einer bestimmten Puffergröße die Verluste völlig; die mittlere Durchlaufzeit erhöht sich mit zunehmender Pufferkapazität nicht weiter. Der Grund hierfür ist, dass in diesem Bereich die beschränkte Größe  $S_{TCP}$  des TCP-Empfangspuffers zum Tragen kommt. Bei homogenem Verkehr ist dies der Fall, wenn gilt:

$$\frac{S_{TCP}}{T_{RTT}} \leq \frac{C}{n} \tag{B.2}$$



**Bild B.5:** Summennutzdurchsatz in Abhängigkeit von der Puffergröße



**Bild B.6:** Verlustwahrscheinlichkeit in Abhängigkeit von der Puffergröße

Dabei bezeichnet  $\overline{T_{RTT}}$  die mittlere RTT, die mit der maximalen RTT

$$T_{RTT, max} = \tau + \frac{S}{C} + 2 \cdot \frac{L_{MSS}}{C} + \frac{L_{MSS}}{R_{max}} \quad (\text{B.3})$$

approximiert werden kann. Diese enthält neben  $\tau$  und der maximalen Wartezeit  $S/C$  die Bedienzeiten auf dem Engpass- und den Zugangslinks. Vernachlässigt man jene, erhält man

$$\overline{T_{RTT}} \approx \tau + \frac{S}{C} \quad (\text{B.4})$$

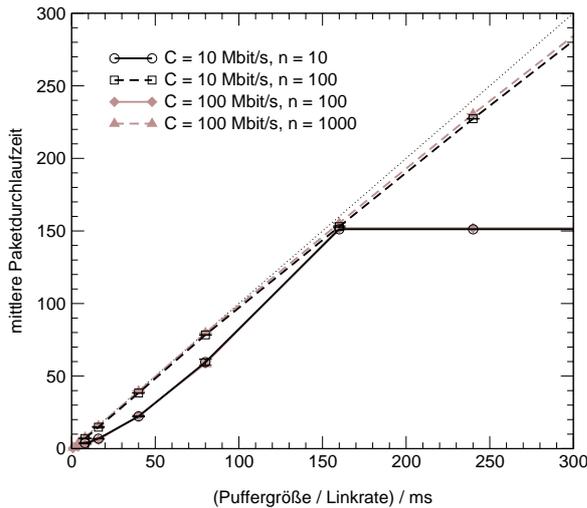
Damit kommt es zu dem beschriebenen Effekt näherungsweise, wenn für die Puffergröße gilt:

$$S \geq n \cdot S_{TCP} - \tau \cdot C \quad (\text{B.5})$$

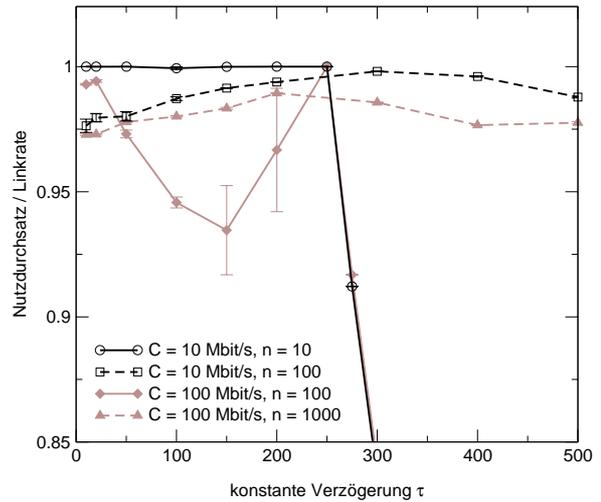
Bei den vorliegenden Werten für die einzelnen Parameter führt dies für  $C/n = 1$  Mbit/s, auf einen Wert von 156 ms (vgl. auch Bild B.7) für  $S/C$ , was einer Puffergröße von 195 000 Byte bei  $C = 10$  Mbit/s bzw. 1 950 000 Byte bei  $C = 100$  Mbit/s entspricht. Außerdem wird aus Gleichung (B.5) klar, dass bei einer Puffergröße von mindestens  $n \cdot S_{TCP}$  die TCP-Regelmechanismen unabhängig von den Werten für  $\tau$  und  $C$  ohne Verluste auskommen.

### B.1.1.3 Einfluss der absoluten Verzögerung

Abschließend wird der Einfluss der konstanten Verzögerung  $\tau$  betrachtet, die im Rahmen der Untersuchungen in Kapitel 6 nicht variiert wird. Der Einfluss von  $\tau$  wird für unterschiedliche Werte von  $C$  und  $n$  untersucht, während für die Puffergröße in allen Fällen ein Wert von  $S = 10^5$  Byte angenommen wird. Der in Bild B.8 dargestellte mittlere Summennutzdurchsatz weist für  $C = 10$  Mbit/s,  $n = 100$  und  $C = 100$  Mbit/s,  $n = 1000$  ein flaches Maximum im



**Bild B.7:** Mittlere Paketdurchlaufzeit in Abhängigkeit von der Puffergröße



**Bild B.8:** Summennutzdurchsatz in Abhängigkeit von  $\tau$

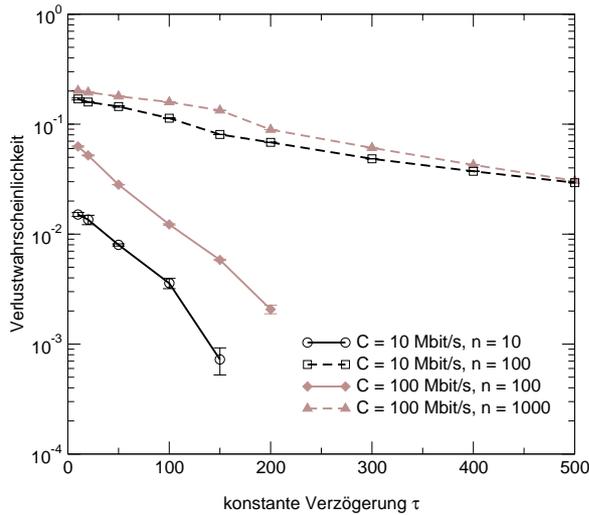
Bereich von  $200 \text{ ms} < \tau < 300 \text{ ms}$  auf. Insgesamt ist der Einfluss auf den Summennutzdurchsatz bei diesen beiden Parameterkombinationen aber eher gering. Beim Blick auf die Paketebenenmaße ist allerdings zu beobachten, dass sowohl die Verlustwahrscheinlichkeit (Bild B.9) als auch die mittlere Paketdurchlaufzeit (Bild B.10) in den beiden genannten Fällen mit steigenden Werten von  $\tau$  zurückgehen. Die Ursache hierfür ist, dass sich die konstante Verzögerung als Bestandteil der mittleren RTT ebenso wie Verluste und Wartezeiten grundsätzlich mindernd auf den mittleren Nutzdurchsatz  $g$  der einzelnen Verbindungen auswirkt (vgl. Abschnitt B.1.2). Da jedoch eine Reduktion von  $g$  gegenüber der Maximalrate  $R_{max}$  aufgrund der Randbedingung  $n \cdot g \leq C$  ohnehin stattfinden muss, bedeutet dies, dass bei größeren Werten von  $\tau$  weniger Verluste und eine geringere Verzögerung im Netzknoten erforderlich sind.

Ein noch stärkerer Rückgang von Verlustwahrscheinlichkeit und mittlerer Paketverzögerung lässt sich in den Fällen  $C = 10 \text{ Mbit/s}$ ,  $n = 10$  und  $C = 100 \text{ Mbit/s}$ ,  $n = 100$  erkennen. Ab einem gewissen Wert von  $\tau$  treten dort überhaupt keine Verluste und Verzögerungen mehr auf. Gleichzeitig ergibt sich an dieser Stelle ein starker Abfall des Summennutzdurchsatzes. Zurückzuführen ist der Einbruch darauf, dass sich wieder die Beschränkung des TCP-Empfangspuffers in Kombination mit einer hohen mittleren RTT, die stets größer gleich  $\tau$  ist, auswirkt. Nach Gleichung (B.2) tritt dies bei homogenem Verkehr näherungsweise dann auf, wenn

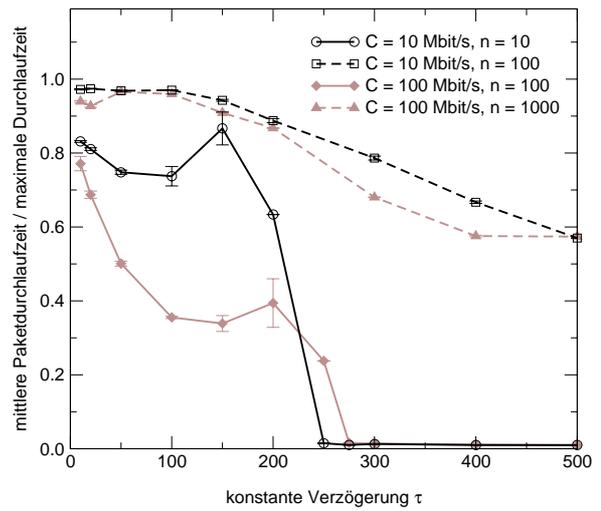
$$\tau \geq \frac{n \cdot S_{TCP}}{C} \tag{B.6}$$

Bei den vorliegenden Werten findet die Reduktion des Summennutzdurchsatzes also für  $\tau \geq 256 \text{ ms}$  statt.

Darüber hinaus ist in Bild B.8 zu erkennen, dass im Fall  $C = 100 \text{ Mbit/s}$ ,  $n = 100$  auch für kleinere Werte von  $\tau$  der Summennutzdurchsatz unter der Linkrate bleibt. Ein ähnlicher Rück-



**Bild B.9:** Verlustwahrscheinlichkeit in Abhängigkeit von  $\tau$



**Bild B.10:** Mittlere Paketdurchlaufzeit in Abhängigkeit von  $\tau$

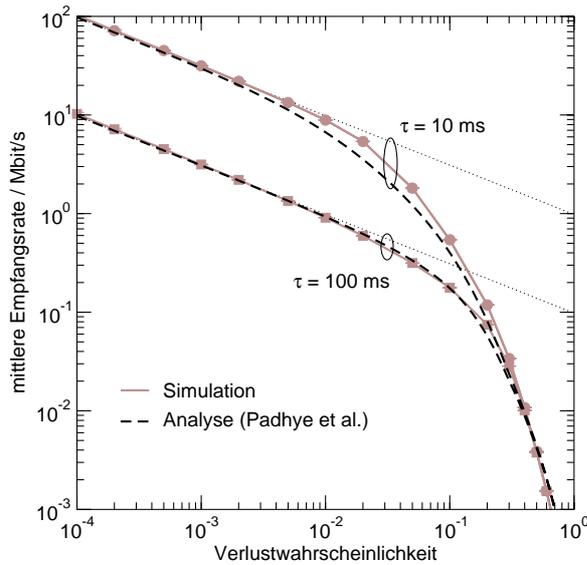
gang wurde bereits in Abschnitt B.1.1.2 für  $\tau = 100$  ms beobachtet und konnte dort auf den geringen Wert von  $S/C$  und die damit verbundene zeitweise Entleerung des Puffers bei relativ geringer Quellenanzahl zurückgeführt werden. Offensichtlich muss der Wert von  $S/C$  in Relation zu  $\tau$  betrachtet werden, um das Phänomen zu erklären, was auch in Übereinstimmung mit den Ergebnissen in [169] steht. Denn mit steigendem Wert von  $\tau$  verstärkt sich der Effekt zunächst, bis schließlich wieder eine Steigerung des Summennutzdurchsatzes bei Annäherung von  $\tau$  an den o. g. Wert erfolgt, ab dem die Verluste verschwinden.

### B.1.2 Vergleich mit analytischen Ergebnissen

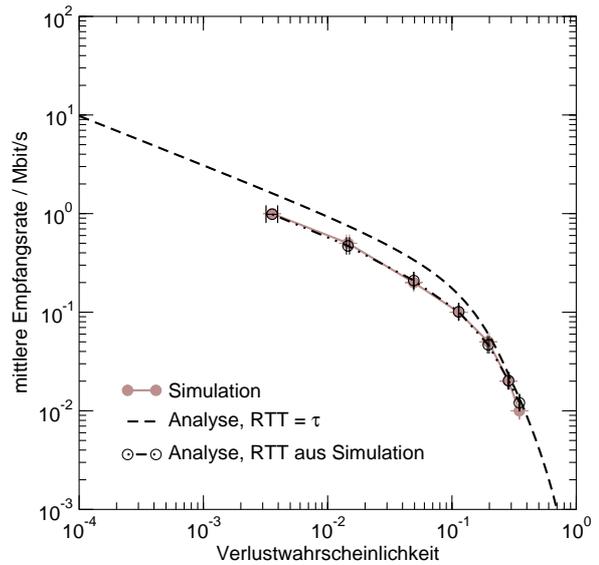
Unter den zahlreichen Forschungsbeiträgen, die sich mit der analytischen Leistungsuntersuchung von TCP-Verkehr auf der Basis ungesättigter Quellen beschäftigen [2, 7, 22, 54, 157, 169, 165, 190, 194, 212], hat die in [212] veröffentlichte Arbeit von Padhye et al., die auf einer früheren Veröffentlichung von Mathis et al. aufbaut [190], herausragende Bedeutung erlangt. Dort wird für „TCP Reno“ unter der Annahme unabhängiger Verluste der Zusammenhang zwischen der Verlustwahrscheinlichkeit  $p$  und der mittleren RTT  $\overline{T_{RTT}}$  einerseits und der mittleren Senderate  $m_S$  andererseits bestimmt. Neben einer unter den gegebenen Randbedingungen exakten Analyse wird in [212] folgende Approximation für  $m_S$  angegeben:

$$m_S \approx \frac{L_{MSS}}{\overline{T_{RTT}} \cdot \sqrt{\frac{2qp}{3}} + T_{RTO} \cdot \min\left(1, 3\sqrt{\frac{3qp}{8}}\right) \cdot p \cdot (1 + 32p^2)} \quad (\text{B.7})$$

Als weitere Parameter neben  $p$  und  $\overline{T_{RTT}}$  sind darin die Anzahl  $q$  der Pakete, die beim Empfänger mit einer (evtl. verzögerten) Quittierung bestätigt werden, sowie die minimale Zeitspanne  $T_{RTO}$ , nach der ein Wiederholungs-Timeout (*retransmission timeout*, RTO) erfolgt,



**Bild B.11:** Unabhängige Verluste:  
Vergleich mit Analyse



**Bild B.12:** Verluste bei FIFO-DT:  
Vergleich mit Analyse ( $\tau = 100$  ms)

enthalten. Da auf *Delayed ACK* hier verzichtet wird, ist  $q = 1$ , während für  $T_{RTO}$  unter den Randbedingungen der nachfolgenden Untersuchungen ein Wert von 200 ms angenommen werden kann. Aus der mittleren Senderate  $m_S$  lässt sich die mittlere Empfangsrate  $m_E$ , die den Bruttodurchsatz wiedergibt und näherungsweise auch dem Nutzdurchsatz  $g$  entspricht (siehe Abschnitt 4.3.2.1), ableiten:

$$g \approx m_E = m_S \cdot (1 - p) \tag{B.8}$$

Im Folgenden wird nun zunächst gezeigt, dass die Näherung auch für die hier zugrunde gelegten TCP-Parameter (z. B. „New Reno“ statt „Reno“) ausreichend genau ist. Dazu werden die analytischen Ergebnisse nach Gleichungen (B.7) und (B.8) mit denen von Simulationen verglichen, bei denen im Modell an die Stelle der Bedieneinheiten für Zugangs- und Engpasslinks (vgl. Bild 6.26) eine Komponente tritt, die eine zufällige Verwerfung mit gegebener Wahrscheinlichkeit  $p$  vornimmt. Damit entspricht in diesem vereinfachten Modell die mittlere RTT der konstanten Verzögerung  $\tau$ . Da die Verluste unabhängig sind, genügt es außerdem, nur eine einzige Verbindung zu betrachten. Die Ergebnisse für die mittlere Empfangsrate in Abhängigkeit von der Verlustwahrscheinlichkeit sind in Bild B.11 dargestellt. Dabei sind nur sehr geringe Unterschiede zwischen Analyse und Simulation zu erkennen. Außerdem ist zu sehen, dass nur für geringe Verlustwahrscheinlichkeiten der häufig angenommene Zusammenhang  $m_E \sim 1/\sqrt{p}$  (gepunktete Linien) gilt. Für  $p > 0.1$  hingegen wirken sich der Faktor  $(1 - p)$  im Zähler sowie der zweite Summand im Nenner stärker auf den Durchsatz aus.

Die TCP-Analyse in [212] geht davon aus, dass Verluste statistisch unabhängig voneinander auftreten. Daher stellt sich die Frage, ob die Näherung nach Gleichung (B.7) auch auf Systeme angewandt werden kann, bei denen diese Voraussetzung nicht gegeben ist, insbesondere auf

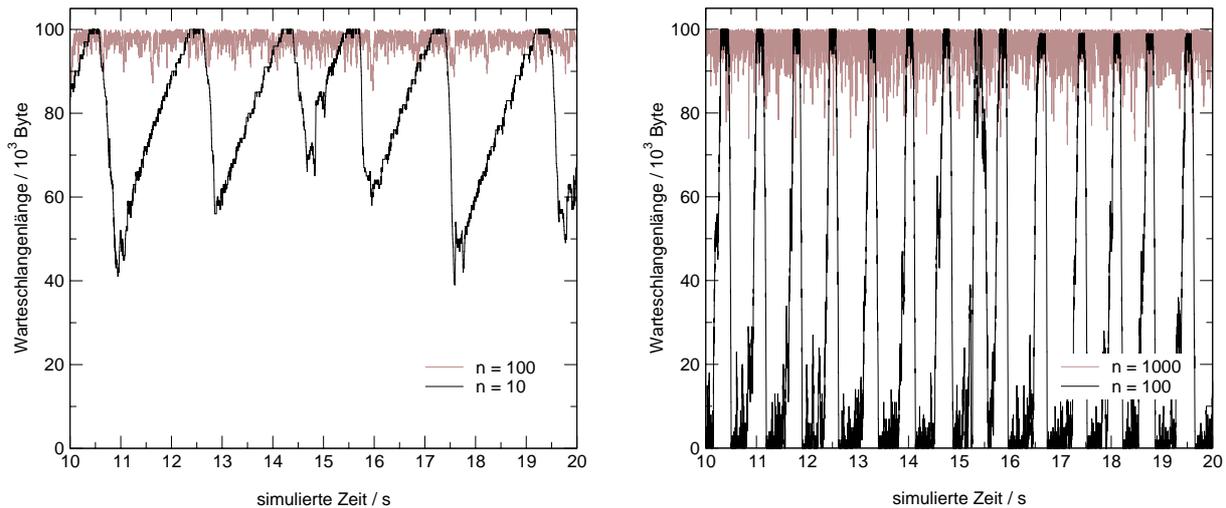
ein FIFO-DT-System mit mehreren ungesättigten TCP-Quellen. Für einen Vergleich von Analyse und Simulation wird auf die Ergebnisse der Untersuchung in Abschnitt B.1.1.1 mit  $C = 10$  Mbit/s und  $S = 10^5$  Byte zurückgegriffen, bei der die Anzahl  $n$  von Quellen variiert wurde. In Bild B.12 sind die sich aus der Simulation ergebenden Werte für die mittlere Empfangsrate (Bruttodurchsatz) pro Quelle über den ebenfalls aus der Simulation resultierenden Verlustwahrscheinlichkeiten (vgl. Bild B.3) aufgetragen. Auf Seiten der Analyse enthält Bild B.12 zunächst die Ergebnisse, zu denen man mit Hilfe von Gleichung (B.7) gelangt, wenn man wie bei Bild B.12 nur die absolute Verzögerung  $\tau = 100$  ms bei der Angabe der mittleren RTT berücksichtigt. Der auf diese Weise berechnete mittlere Durchsatz liegt deutlich über dem, der sich aus der Simulation ergibt. Eine wesentlich genauere Übereinstimmung lässt sich erzielen, wenn für die mittlere RTT in Gleichung (B.7) folgender Wert genommen wird:

$$\overline{T_{RTT}} = \tau + \frac{L_{MSS}}{C} + \frac{L_{MSS}}{R_{max}} + \overline{T_D} \quad (\text{B.9})$$

Dabei setzt sich  $\overline{T_{RTT}}$  einerseits aus den konstanten Werten für die Verzögerung  $\tau$  im Hin- und Rückkanal sowie den Bediendauern eines Pakets auf den Zugangslinks (Sende- und Empfangsseite) und andererseits aus der mittleren Durchlaufzeit  $\overline{T_D}$  durch den Engpassknoten zusammen. Für letztere werden ebenso wie für die Verlustwahrscheinlichkeit die Werte aus der Simulation übernommen (vgl. Bild B.4). Die kaum erkennbaren Unterschiede zwischen dem auf dieser Basis per Gleichung (B.7) berechneten und dem in der Simulation gemessenen Bruttodurchsatz zeigen, dass der Zusammenhang nach Gleichung (B.7) auch für einen Engpasslink mit FIFO-DT, also ein System, bei dem nicht von unabhängigen Verlusten ausgegangen werden kann, mit guter Näherung gilt. Diese Aussage wird durch weitere Vergleiche mit anderen Parameterkonstellationen, deren Ergebnisse hier nicht dargestellt sind, weitgehend bestätigt.

### B.1.3 Betrachtung der Warteschlangenlänge

Ein häufig genutztes Hilfsmittel bei der Untersuchung von TCP-Verkehr sind Darstellungen gemessener Werte wie z. B. der Warteschlangenlänge über der Zeit (Traces). Tatsächlich gibt eine solche Auftragung einen zusätzlichen Einblick in das typische TCP-Verhalten. Traces der Warteschlangenlänge sind für eine Puffergröße von  $S = 10^5$  Byte sowie unterschiedliche Werte der Linkrate  $C$  und Quellenanzahl  $n$  in Bild B.13 zu sehen. Dabei ist in den beiden Fällen  $C = 10$  Mbit/s,  $n = 10$  sowie  $C = 100$  Mbit/s,  $n = 100$  das in der Literatur häufig zitierte periodische Verhalten zu erkennen [108]. Dieses wird dadurch verursacht, dass in den Zeitabschnitten, in denen der Puffer gefüllt ist, mehrere TCP-Verbindungen Verluste erleiden und darauf mit einer Reduktion der Senderate reagieren. Diese Reaktion wirkt sich jedoch um mindestens eine RTT verzögert aus, sodass die Überlastsituation zunächst erhalten bleibt und daher mehrere Verbindungen berührt. Dafür ist die Reaktion dann aber um so drastischer, da gleich eine ganze Reihe von Verbindungen ihre Senderate halbieren. Dadurch fällt die Sum-



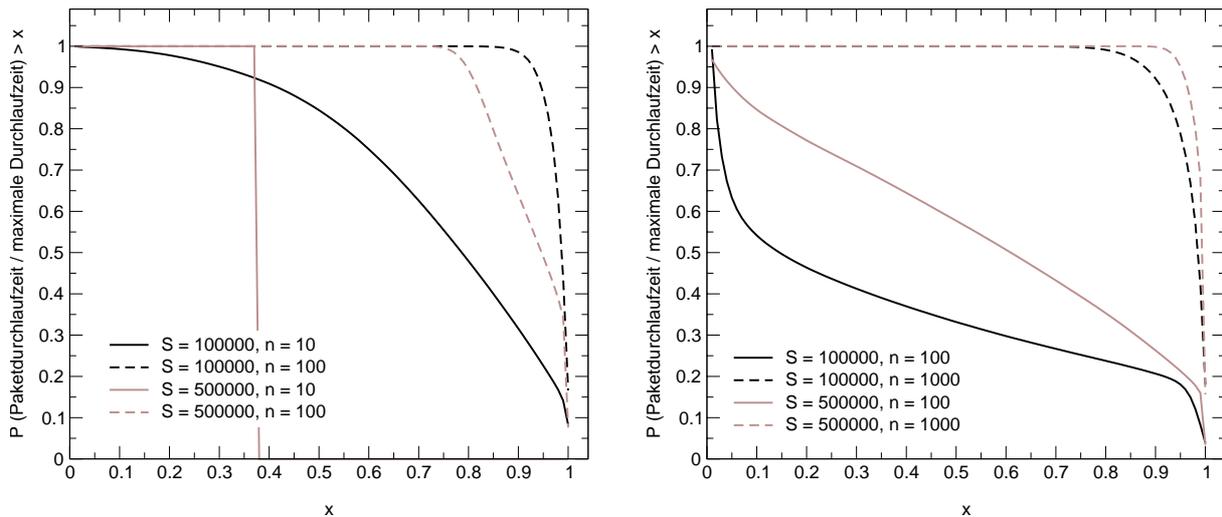
**Bild B.13:** Zeitlicher Verlauf der Warteschlangenlänge bei  $S = 100\,000$  Byte für  $C = 10$  Mbit/s (links) bzw.  $C = 100$  Mbit/s (rechts)

menankunftsrate kurzzeitig unter die Linkrate, was eine Abnahme des Pufferfüllstands nach sich zieht. Der anschließende relativ langsame Anstieg der Warteschlangenlänge, wie er vor allem bei  $C = 10$  Mbit/s,  $n = 10$  zu erkennen ist, hat seine Ursache in dem langsamen Anwachsen des CWnd um  $L_{MSS}$  pro RTT. Bestätigt wird hier auch die Aussage aus Abschnitt B.1.1.1, dass es bei  $C = 100$  Mbit/s,  $n = 10$  häufig zu einer vollständigen Entleerung des Puffers kommt, während dies im Fall  $C = 10$  Mbit/s,  $n = 10$  zumindest im dargestellten Zeitintervall nicht beobachtet werden kann. Als weitere Erkenntnis folgt, dass sich die Periodizität in den beiden Fällen mit relativ vielen Quellen ( $n = 20 \cdot r$ ) nicht mehr bemerkbar macht. Hier bewegt sich die Warteschlangenlänge permanent in der Nähe ihrer Obergrenze.

Außer dem zeitlichen Verlauf der Warteschlangenlänge ist auch die statistische Belegung von großem Interesse. Zwar ist in Bild B.14 die komplementäre Verteilungsfunktion der Paketdurchlaufzeit aufgetragen, doch entspricht diese bei normierter Darstellung weitgehend der CCDF der bei Ankunft angetroffenen Warteschlangenlänge.<sup>2</sup> Der steile Abfall der Kurven für relativ große Werte von  $n$  bei Annäherung an die maximale Durchlaufzeit  $S/C$  zeigt, dass in diesen Fällen ankommende Pakete mit großer Wahrscheinlichkeit einen fast vollständig gefüllten Puffer vorfinden. Dies wirkt sich insbesondere dann negativ aus, wenn neben dem TCP-Verkehr noch Echtzeitverkehr vorhanden ist, der ohne Maßnahmen zur differenzierten Behandlung eine gleich hohe Durchlaufzeit erfährt. Auch bei weniger Quellen ist die Wahrscheinlichkeit nicht besonders hoch, einen wenig gefüllten Puffer vorzufinden. Interessanterweise ist in einigen Fällen ein zumindest phasenweise linearer Abfall der Durchlaufzeit-CCDF<sup>3</sup> zu beobachten, was darauf schließen lässt, dass die in diesem Bereich liegenden Durchlaufzeiten mit gleicher Wahrscheinlichkeit auftreten. Einen Sonderfall bildet die Kombination  $C = 10$  Mbit/s,  $n = 10$ ,  $S = 500\,000$  Byte. Dort wirkt sich die in Abschnitt B.1.1.2

<sup>2</sup> Für einen kleinen Unterschied sorgt die Bedienzeit, die in der Paketdurchlaufzeit enthalten ist.

<sup>3</sup> Man beachte die lineare Darstellung in Bild B.14.



**Bild B.14:** CCDF der Paketdurchlaufzeit für  $C = 10$  Mbit/s (links) bzw.  $C = 100$  Mbit/s (rechts)

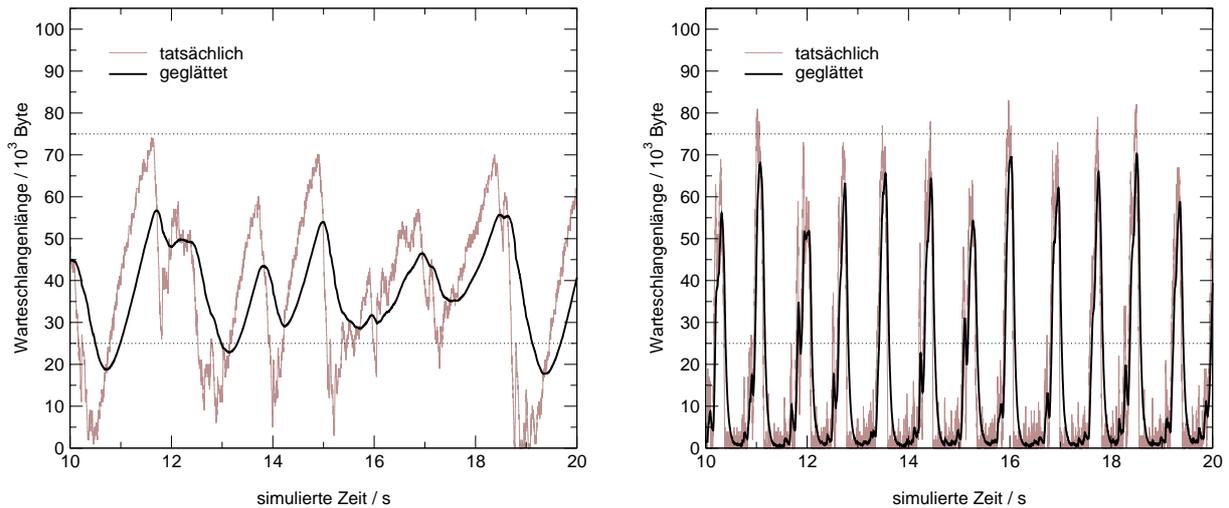
angesprochene Beschränkung des Empfangspuffers aus, sodass die Verzögerung einen nahezu konstanten Wert von etwa 156 ms annimmt.

### B.1.4 Vergleich von DT und RED

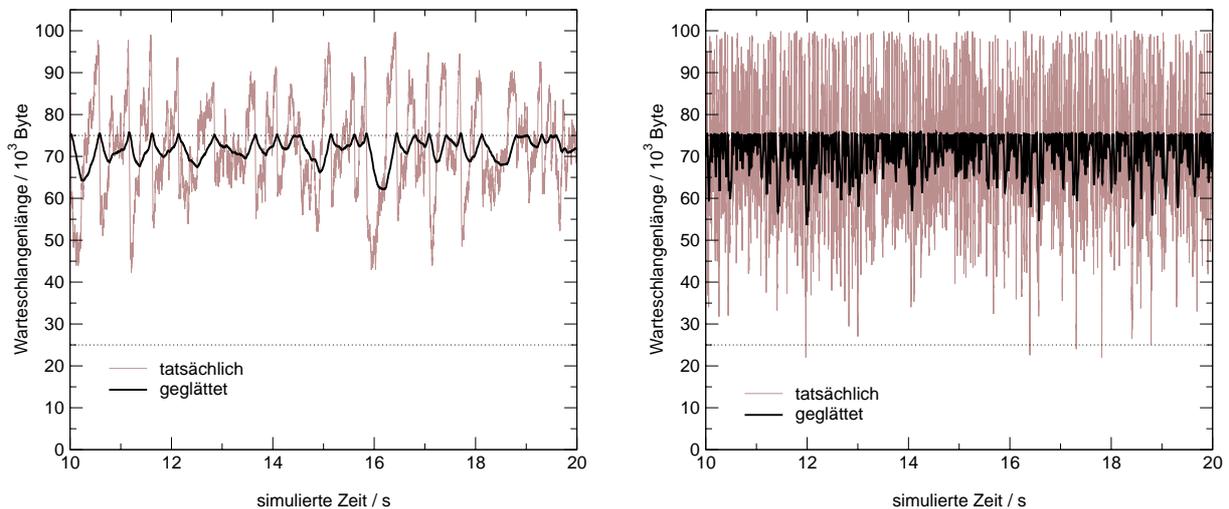
Eine Alternative zu DT als Puffermanagement-Strategie ohne Differenzierung stellt das in Abschnitt 3.2.1.2 beschriebene RED dar [109]. Daher werden im Folgenden die beiden Mechanismen bei einer Puffergröße von  $S = 10^5$  Byte einander gegenübergestellt. Als Parameter bei RED werden in Anlehnung an [109]  $min_{th} = 25\,000$  Byte,  $max_{th} = 75\,000$  Byte,  $max_p = 0.1$  und  $w_q = 0.002$  angenommen.

Für einen Vergleich beider Strategien wird zunächst der bei RED resultierende zeitliche Verlauf der Warteschlangenlänge dargestellt. Bild B.15 zeigt diesen zusammen mit dem Verlauf der geglätteten Warteschlangenlänge  $\bar{Q}$  nach Gleichung (3.5) für  $C = 10$  Mbit/s,  $n = 10$  sowie  $C = 100$  Mbit/s,  $n = 100$  (d. h.  $n = 2 \cdot r$ ). Die bei DT beobachtete Periodizität ist bei  $C = 100$  Mbit/s weiterhin deutlich zu erkennen. Dort erfasst diese auch die geglättete Warteschlangenlänge, die dabei häufig unter  $min_{th}$  fällt. Im Gegensatz dazu treten die nach wie vor großen Schwankungen bei  $C = 10$  Mbit/s,  $n = 10$  nicht mehr in einem klar zu erkennenden periodischen Raster auf. Zu einer Unterschreitung von  $min_{th}$  durch  $\bar{Q}$  kommt es nur selten.

Im Fall einer höheren Quellenzahl  $n = 20 \cdot r$  bewegt sich  $\bar{Q}$  in einem engen Bereich unterhalb von  $max_{th}$  (Bild B.16). In Bezug auf die tatsächliche Warteschlangenlänge ist wie bei DT ein periodisches Verhalten nicht festzustellen. Die Warteschlangenlänge übersteigt jedoch im Gegensatz zu den Fällen in Bild B.15 häufig den Schwellwert  $max_{th}$ . Bei  $C = 100$  Mbit/s,  $n = 1000$  wird bisweilen sogar die gesamte Pufferkapazität ausgenutzt, was darauf hindeutet, dass es zu Verlusten kommt, die nicht durch den RED-Verwerfungsmechanismus erzwungen werden, sondern aufgrund mangelnder Speicherressourcen erfolgen.



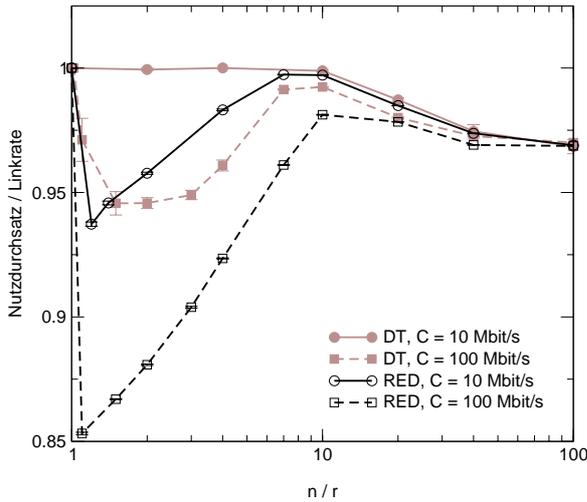
**Bild B.15:** Zeitlicher Verlauf der Warteschlangenlänge bei RED für  $n/r = 2$ , d. h.  $C = 10$  Mbit/s,  $n = 10$  (links) bzw.  $C = 100$  Mbit/s,  $n = 100$  (rechts)



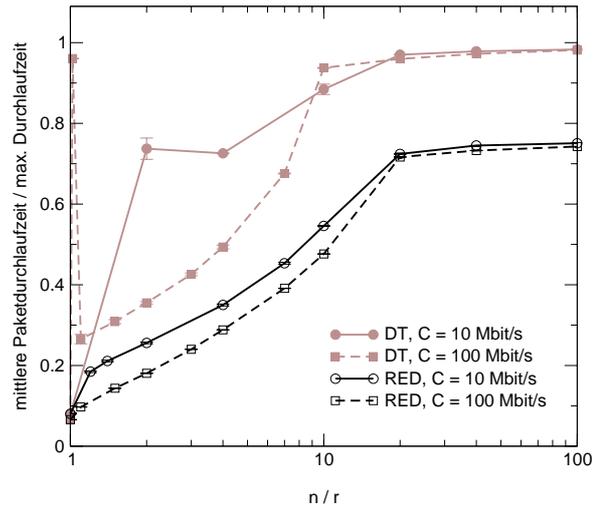
**Bild B.16:** Zeitlicher Verlauf der Warteschlangenlänge bei RED für  $n/r = 20$ , d. h.  $C = 10$  Mbit/s,  $n = 100$  (links) bzw.  $C = 100$  Mbit/s,  $n = 1000$  (rechts)

Im Hinblick auf den Nutzdurchsatz bei variabler Anzahl von Quellen liefert RED für große Werte von  $n$  sehr ähnliche Ergebnisse wie DT, d. h. es ergibt sich ein leichter Rückgang des Summennutzdurchsatzes (Bild B.17). Im Bereich kleiner Werte von  $n/r$  ist hier nicht nur für  $C = 100$  Mbit/s, sondern auch für  $C = 10$  Mbit/s ein Einbruch des Nutzdurchsatzes zu verzeichnen. Bei  $C = 100$  Mbit/s ist die Reduktion zudem stärker ausgeprägt als im Fall von DT. Der Grund für die Reduktion liegt, wie in Abschnitt B.1.1.1 dargelegt, darin, dass es gelegentlich zu einer vollständigen Entleerung des Puffers kommt, was im Fall von RED anhand von Bild B.15 leicht nachvollzogen werden kann.

Wie erwartet ist die mittlere Paketdurchlaufzeit bei RED gegenüber den Werten bei DT deutlich verringert (Bild B.18). Für große Werte von  $n/r$  orientiert sich die auf  $S/C$  normierte mittlere Verzögerung am Verhältnis  $max_{th}/S$ , das hier 0.75 beträgt. Die reduzierte mittlere



**Bild B.17:** Summennutzdurchsatz bei RED und DT



**Bild B.18:** Mittlere Paketdurchlaufzeit bei RED und DT

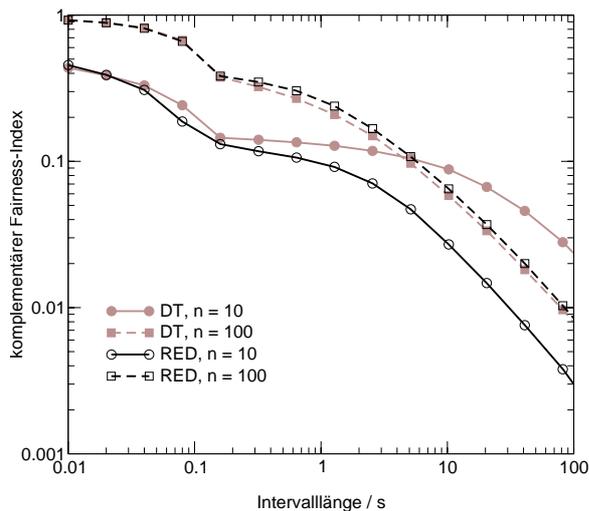
Paketdurchlaufzeit, die eine geringere RTT zur Folge hat, muss bei RED allerdings zwangsläufig durch eine erhöhte Verlustwahrscheinlichkeit erkauft werden, was durch hier nicht wiedergegebene Auswertungen bestätigt wird.

Ein häufig genannter Vorteil von RED stellt die erhöhte Fairness bei der Verteilung der Bandbreiteressourcen an verschiedene TCP-Verbindungen dar. Um dies genauer zu untersuchen, sind in Bild B.19 die Werte des in Abschnitt 4.3.2.1 beschriebenen Fairness-Index  $f_F(\Delta t)$  mit Bezug auf die Intervalllänge  $\Delta t$  in komplementärer Darstellung aufgetragen. Daraus wird ersichtlich, dass eine Fairness-Verbesserung – in Bild B.19 ausgedrückt durch die weiter unten liegende Kurve – unter den betrachteten Randbedingungen (insbesondere homogener Verkehr) nur bei hoher Quellenanzahl verzeichnet werden kann. Die relative Verbesserung macht sich mit zunehmender Intervalllänge stärker bemerkbar. Es ist jedoch fraglich, inwieweit Unfairness bei Werten von  $f_F(\Delta t) > 0.99$  überhaupt noch eine Rolle spielt. In Bild B.20 sind darüber hinaus die Auswirkungen auf die Fairness gezeigt, die sich bei Verwendung unterschiedlicher TCP-Versionen ergeben.<sup>4</sup> Die älteren TCP-Versionen „TCP Reno“ und vor allem „TCP Tahoe“ reduzieren die Fairness bei DT erheblich, während im Fall von RED nur geringe Unterschiede zu „TCP New Reno“ auftreten.

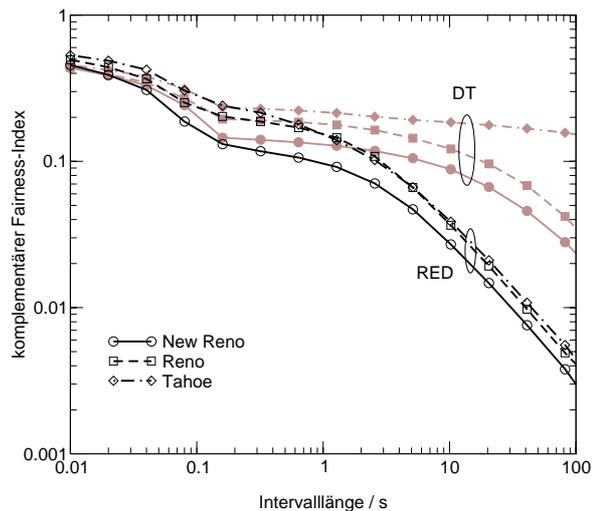
## B.2 Untersuchungen mit dynamischem TCP-Verkehr

Ungesättigte TCP-Quellen, wie sie im letzten Abschnitt untersucht wurden, weisen den Nachteil auf, dass sie das Verhalten des in heutigen IP-Netzen dominierenden WWW-Verkehrs nur unzureichend wiedergeben. Wie bereits in Abschnitt 2.4.1.2 erläutert, wird die Kommunikation im WWW über TCP-Verbindungen realisiert, innerhalb deren einzelne Objekte (Text,

<sup>4</sup> Innerhalb einer Simulation wird immer nur eine Version verwendet.



**Bild B.19:** Fairness-Index für RED und DT ( $C = 10$  Mbit/s, „TCP New Reno“)



**Bild B.20:** Fairness-Index für verschiedene TCP-Versionen ( $C = 10$  Mbit/s,  $n = 10$ )

Grafiken usw.) übertragen werden. Aus Sicht der Verkehrsmodellierung werden mehr oder weniger kurze Datenbursts über TCP vom Sender zum Empfänger übertragen, wobei sich am Beginn eines Bursttransfers die TCP-Verbindung im Initialzustand befindet.<sup>5</sup> Zur Charakterisierung der Verkehrseigenschaften wird gemäß Abschnitt 4.2.2.3 eine  $M/G/\infty$ -Burstmodell verwendet. Dies bedeutet, dass Bursts, die mit Hilfe von TCP übertragen werden sollen, in negativ-exponentiell verteilten Abständen ankommen und eine zufällige Größe  $B$  haben, für die im Nachfolgenden unterschiedliche Verteilungen untersucht werden. Im Gegensatz zu den ungesättigten Quellen wird der Verkehr nicht durch die Anzahl von Quellen, sondern durch das Angebot  $A$  charakterisiert, das sich gemäß Gleichung (4.21) aus den Mittelwerten von Burstgröße und -ankunftsabstand sowie der Linkrate ergibt.

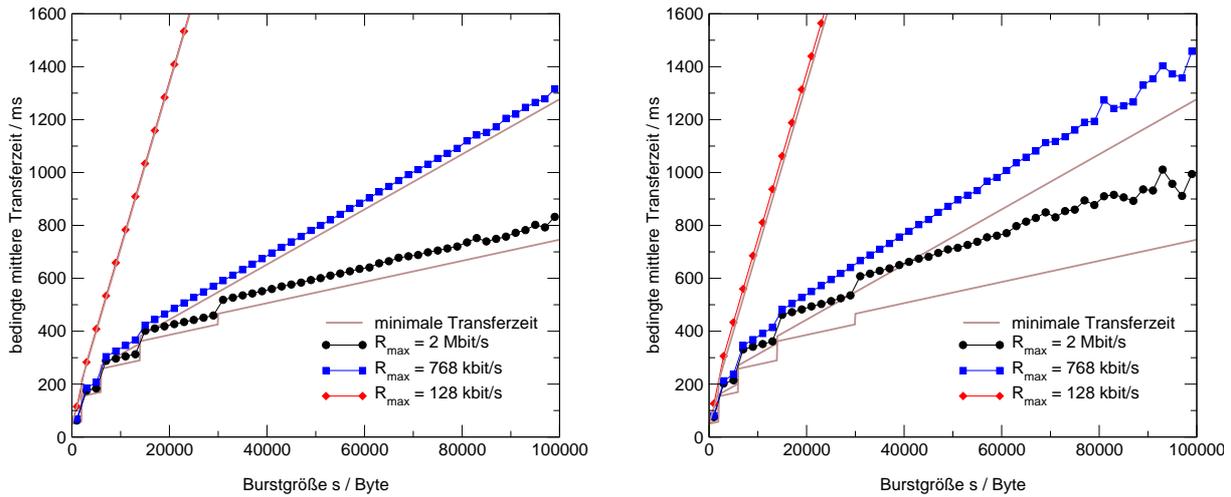
## B.2.1 Detaillierte Betrachtung in einzelnen Lastpunkten

Zunächst werden für  $C = 10$  Mbit/s verschiedene Leistungsmetriken für dynamischen TCP-Verkehr, die in Abschnitt 4.3.2.2 vorgestellt wurden, für ausgewählte Angebotswerte bestimmt. Zu diesen Leistungsmetriken gehören bedingte Mittelwerte, die über der Burstgröße aufgetragen werden, ebenso wie Verteilungen.

### B.2.1.1 Bedingte Mittelwerte

Ein wichtiges Leistungsmaß bei dynamischem TCP-Verkehr stellt die mittlere Transferzeit dar. Wesentliche Effekte treten dabei zu Tage, wenn anstelle des Gesamtmittelwerts die bedingte mittlere Transferzeit  $t(s)$  in Abhängigkeit von der Burstgröße  $s$  betrachtet wird. Diese ist in

<sup>5</sup> Wie in Abschnitt 4.2.2.3 erklärt wurde, spielt es keine wesentliche Rolle, ob es sich dabei um wirklich separate TCP-Verbindungen handelt oder ob eine Wiederverwendung bereits aufgebauter und in der Zwischenzeit wieder zurückgesetzter TCP-Verbindungen erfolgt.

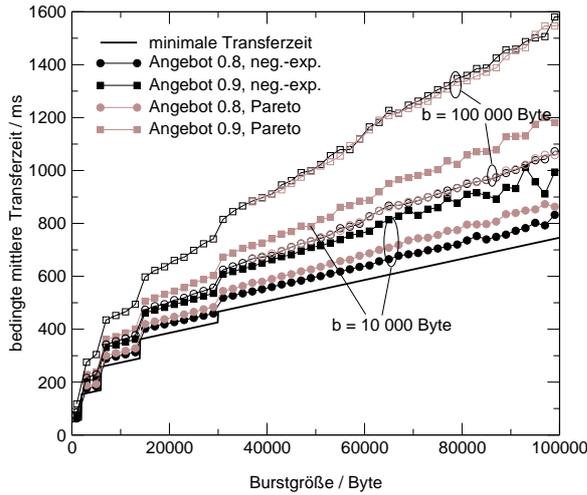


**Bild B.21:** Bedingte mittlere Transferzeit für neg.-exp. verteilte Burstgröße mit Mittelwert  $b = 10\,000$  Byte und  $A = 0.8$  (links) bzw.  $A = 0.9$  (rechts)

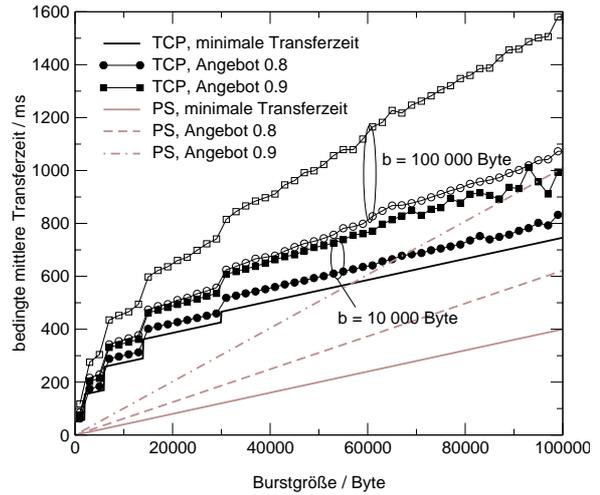
Bild B.21 zusammen mit der in Anhang A nach Gleichung (A.12) berechneten minimalen Transferzeit  $T_{min}(s)$ , die sich ohne den Engpasslink ergeben würde, für unterschiedliche Zugangsraten über der Burstgröße aufgetragen. Da es zunächst um das Aufzeigen grundsätzlicher Effekte geht, wird hier von einer negativ-exponentiell verteilten Burstgröße mit Mittelwert  $b = 10\,000$  Byte ausgegangen. Auffallend ist dabei der treppenförmige Verlauf aller Kurven, der auf das *Slow-Start*-Verhalten der TCP-Verbindungen zurückzuführen ist (siehe Anhang A). Dieser ist um so stärker ausgeprägt, je größer die Zugangsraten sind. Umgekehrt dominiert bei kleiner Zugangsraten, z. B. im Fall  $R_{max} = 128 \text{ bit/s}$ , der Term  $s/R_{max}$  den Verlauf von  $t(s)$  und  $T_{min}(s)$ . Vergleicht man  $t(s)$  mit der unteren Grenze  $T_{min}(s)$ , so stellt man fest, dass beide Kurven prinzipiell den gleichen Verlauf haben. Dies bedeutet, dass auch bei  $t(s)$  zwischen den Sprungstellen ein linearer Anstieg erfolgt. Darüber hinaus fällt auf, dass ein Angebot von 0.8 nur eine geringfügige Erhöhung gegenüber  $T_{min}(s)$  bewirkt. Selbst bei  $A = 0.9$  ist der Unterschied zwischen beiden Kurven noch relativ gering.

Die Steigung von  $t(s)$  und damit der Unterschied zur minimalen Transferzeit hängt nicht nur vom Angebot, sondern in hohem Maße auch von der Verteilung und dem Mittelwert der Burstgröße  $B$  ab (Bild B.22). Vor allem der Übergang zu einer mittleren Burstgröße von  $b = 100\,000$  Byte, verbunden mit einer entsprechenden Reduktion der Burstankunftsrate, führt zu einer deutlichen Zunahme der mittleren Transferzeit für alle Burstgrößen. Ebenso lässt die Verwendung einer Pareto-Verteilung für  $B$  mit  $\alpha = 1.6$  als Formparameter<sup>6</sup> anstelle der negativ-exponentiellen Verteilung die mittlere Transferzeit ansteigen – allerdings nur im Fall  $b = 10\,000$  Byte. Offensichtlich trägt bei gleichem Angebot die vermehrte Anwesenheit von längeren Bursts im System anstelle von kürzeren dazu bei, dass Transfers von Bursts – unabhängig von ihrer Burstgröße – im Mittel länger dauern.

<sup>6</sup> Dadurch entsteht selbstähnlicher Verkehr mit Hurst-Parameter  $H = 0.7$  (siehe Abschnitt 4.2.1.3).



**Bild B.22:** Bedingte mittlere Transferzeit in Abhängigkeit von der Burstgrößenverteilung für  $R_{max} = 2$  Mbit/s



**Bild B.23:** Bedingte mittlere Transferzeit im Vergleich mit PS (neg.-exp. Burstgrößenverteilung,  $R_{max} = 2$  Mbit/s)

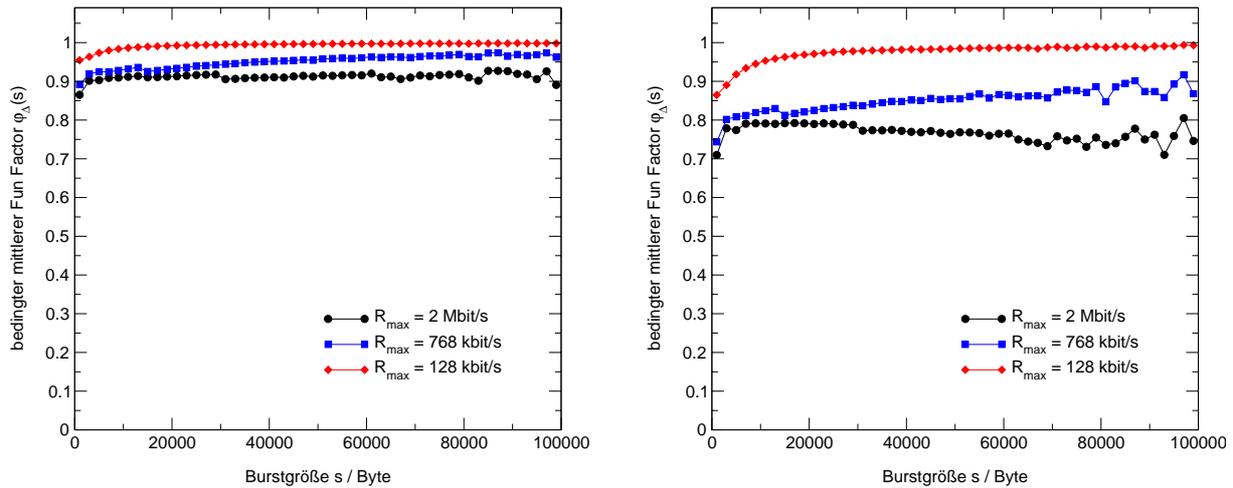
Eine charakteristische Eigenschaft von TCP ist seine Fähigkeit, die Bandbreite auf dem Engpasslink fair unter den aktiven Verbindungen aufzuteilen, wobei im Falle einer endlichen Zugangsrate keine Verbindung mit einer höheren Rate als  $R_{max}$  senden kann. Aufgrund dieser Eigenschaft wird in zahlreichen Forschungsbeiträgen vorgeschlagen, zur Leistungsuntersuchung von dynamischem TCP-Verkehr das aus der Literatur [155] bekannte *Processor-Sharing*-Modell (PS) heranzuziehen [24, 62, 178, 234]. Der vorliegende Fall einer TCP-Verkehrslast auf der Basis des  $M/G/\infty$ -Burstmodells führt auf ein  $M/G/r$ -PS-Modell. Der Parameter  $r$  steht dabei gemäß Gleichung (B.1) für das Verhältnis der Raten auf Engpass- und Zugangslink und gibt die maximale Anzahl gleichzeitig aktiver TCP-Verbindungen an, bei der noch kein Bandbreite-Engpass auftritt. Eine wichtige Eigenschaft des PS-Modells ist, dass die mittlere bedingte Transferzeit proportional zur Burstgröße  $s$  ist:

$$t_{PS}(s) = \frac{s}{R_{max}} \cdot f_{\Delta} \quad (\text{B.10})$$

Der Faktor  $f_{\Delta}$  wird dabei in [177] als Verzögerungsfaktor (*delay factor*) bezeichnet und hängt nur vom Angebot  $A$ , dem Ratenverhältnis  $r$  sowie dessen ganzzahligem Teil  $r_g = \lfloor r \rfloor$  ab [75]:

$$f_{\Delta} = 1 + \frac{1 + (r - r_g) \cdot (1 - A)}{r \cdot (1 - A)} \cdot \frac{\frac{(rA)^{r_g}}{r_g!}}{(1 - A) \cdot \sum_{i=0}^{r_g-1} \frac{(rA)^i}{i!} + \frac{(rA)^{r_g}}{r_g!}} \quad (\text{B.11})$$

Insbesondere besteht keine Abhängigkeit vom Mittelwert und der Verteilung der Burstgröße.



**Bild B.24:** Bedingter mittlerer *Fun Factor*  $\varphi_{\Delta}(s)$  für unterschiedliche Zugangsraten bei neg.-exp. verteilter Burstgröße,  $b = 10\,000$  Byte,  $A = 0.8$  (links) bzw.  $A = 0.9$  (rechts)

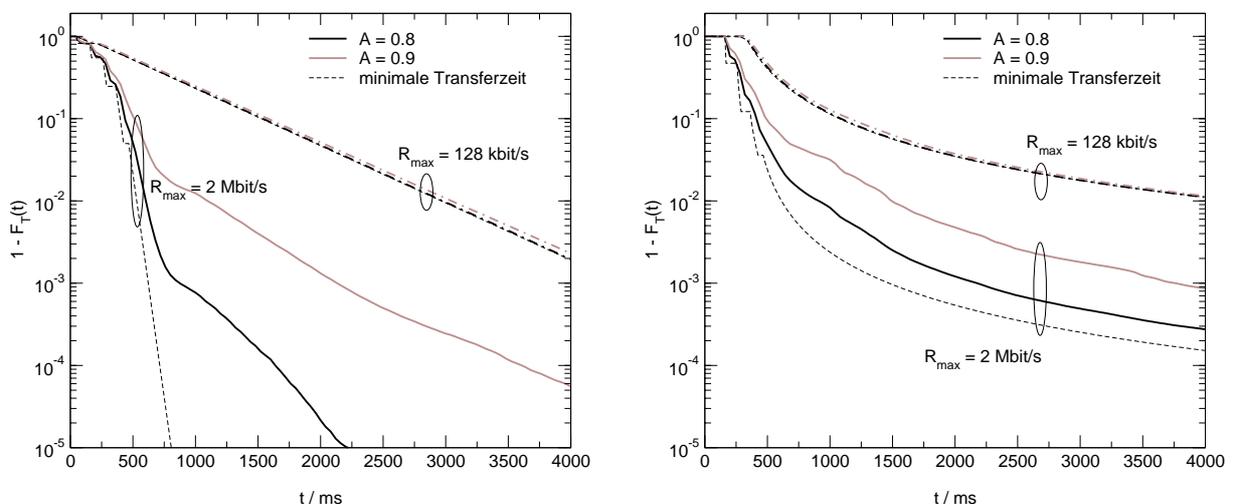
In Bild B.23 werden die Ergebnisse für die bedingte mittlere Transferzeit bei PS den aus der TCP-Simulation erhaltenen Werten gegenübergestellt. Die PS-Approximation liefert offensichtlich deutlich kleinere Werte. Dies ist darauf zurückzuführen, dass im PS-Modell spezielle TCP-Eigenschaften wie das *Slow-Start*-Verhalten, das zu dem treppenförmigen Anstieg von  $t(s)$  führt, nicht berücksichtigt werden. Außerdem kann das PS-Modell die in den Simulationen festgestellte Abhängigkeit von Mittelwert und Verteilung von  $B$  nicht wiedergeben. Vergleicht man die Steigung der Kurven, stellt man eine Übereinstimmung von PS-Approximation und TCP-Simulation für  $b = 100\,000$  Byte fest, während für  $b = 10\,000$  Byte der Anstieg der bedingten mittleren Transferzeit in der TCP-Simulation sehr viel flacher erfolgt.

Die Ergebnisse für die bedingte mittlere Transferzeit müssen immer in Relation zur minimalen Transferzeit gesehen werden. Außerdem hängt der Wertebereich von  $t(s)$  stark von der Zugangsrate  $R_{max}$  ab. Der in Abschnitt 4.3.2.2 vorgestellte bedingte mittlere *Fun Factor*  $\varphi_{\Delta}(s)$  behebt dieses Problem, indem er die Normierung auf die minimale Transferzeit impliziert. Die anschließende Kehrwertbildung bewirkt, dass der Wertebereich zwischen 0 und 1 liegt. Für den Fall einer negativ-exponentiell verteilten Burstgröße mit einem Mittelwert von  $b = 10\,000$  Byte und unterschiedliche Zugangsraten sind die aus der Simulation erhaltenen Werte für  $\varphi_{\Delta}(s)$  in Bild B.24 zu sehen. Selbst bei einem Angebot von 0.9 sinkt  $\varphi_{\Delta}(s)$  in keinem Fall unter einen Wert von 0.7. Dabei ergeben sich für eine geringere Zugangsrate höhere und damit bessere Werte, was zunächst widersprüchlich erscheint. Dies lässt sich dadurch erklären, dass mit einer höheren Zugangsrate auch absolut gesehen höhere Erwartungen (d. h. ein geringerer Minimalwert der Transferzeit) verbunden sind, die bei gleichem Angebot entsprechend schwerer zu erfüllen sind. Andererseits zeigt dies, dass sich durch eine Systemoptimierung an der Engpassstelle (z. B. höhere Linkrate, differenzierte Behandlung) für Nutzer mit einem schmalbandigen Zugang nur geringe Verbesserungen erzielen lassen.

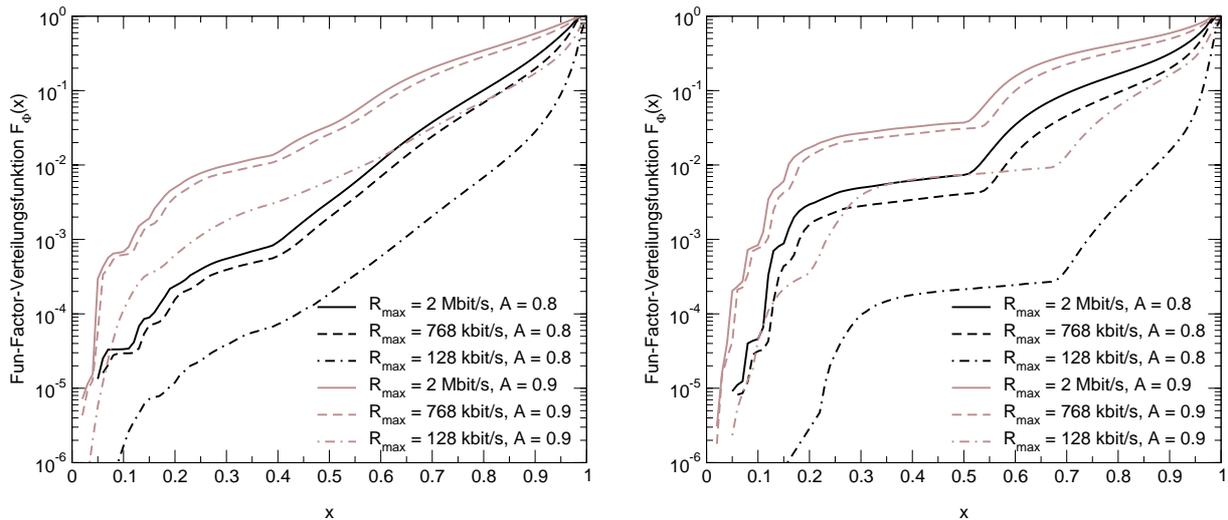
### B.2.1.2 Verteilungen

Ähnlich wie bedingte Mittelwerte geben auch Verteilungen einen tiefen Einblick in das Systemverhalten. Zunächst wird die Transferzeitverteilungsfunktion  $F_T(t)$  betrachtet, die in Bild B.25 in komplementärer Form dargestellt ist. Dabei zeigt sich je nach Burstgrößenverteilung ein deutlich unterschiedlicher Verlauf. Wie erwartet treten bei Pareto-verteilter Burstgröße auch längere Transferzeiten auf. Der Unterschied zur unteren Grenzkurve, die den Idealfall eines leeren Systems repräsentiert und sich gemäß Anhang A.4.2 berechnet, ist im Fall  $R_{max} = 2$  Mbit/s bei Pareto-Verteilung von  $B$  mit  $\alpha = 1.6$  wesentlich kleiner als bei einer negativ-exponentiell verteilten Burstgröße. Bei  $R_{max} = 128$  kbit/s allerdings bewirkt die Last auf dem Engpasslink kaum eine Veränderung gegenüber dem Idealfall, während bei  $R_{max} = 2$  Mbit/s ein erhöhtes Angebot große Transferzeiten sehr viel wahrscheinlicher werden lässt.

Da bei einer Untersuchung auf Basis der Transferzeit-CCDF erneut der Nachteil zum Tragen kommt, dass die sich im Idealfall ergebende Verteilung als Referenz herangezogen werden muss, bietet sich eine Betrachtung der Verteilungsfunktion  $F_\Phi(x)$  des *Fun Factor*  $\Phi$  an. Gemäß Gleichung (4.32) gibt diese die Wahrscheinlichkeit an, dass der Wert des Quotienten  $T_{min}/T$  aus minimaler und gemessener Transferzeit kleiner ist als der Wert auf der x-Achse. Wie die Ergebnisse von Untersuchungen mit negativ-exponentiell sowie mit Pareto-verteilter Burstgröße mit Formparameter  $\alpha = 1.6$  (jeweils mit Mittelwert  $b = 10\,000$  Byte) in Bild B.26 zeigen, ergeben sich auch für das Auftreten für Werte von  $\Phi < 0.5$  noch recht hohe Wahrscheinlichkeiten. So liegt bei  $R_{max} = 2$  Mbit/s und  $A = 0.9$  der Wert von  $F_\Phi(0.5)$  deutlich im Prozentbereich. Selbst sehr kleine Werte ( $\Phi < 0.1$ ) treten noch mit nicht zu vernachlässigender Wahrscheinlichkeit auf. Auffallend ist außerdem der stufenartige Verlauf der Kurven, der darauf schließen lässt, dass  $T_{min}/T$  mit besonders großer Wahrscheinlichkeit Werte aus bestimmten stark eingegrenzten Teilbereichen annimmt. Dies deutet auf Paketwiederholungen



**Bild B.25:** Transferzeit-CCDF für verschiedene Zugangsraten und Angebotswerte bei negativ-exponentiell (links) bzw. Pareto-verteilter Burstgröße (rechts)



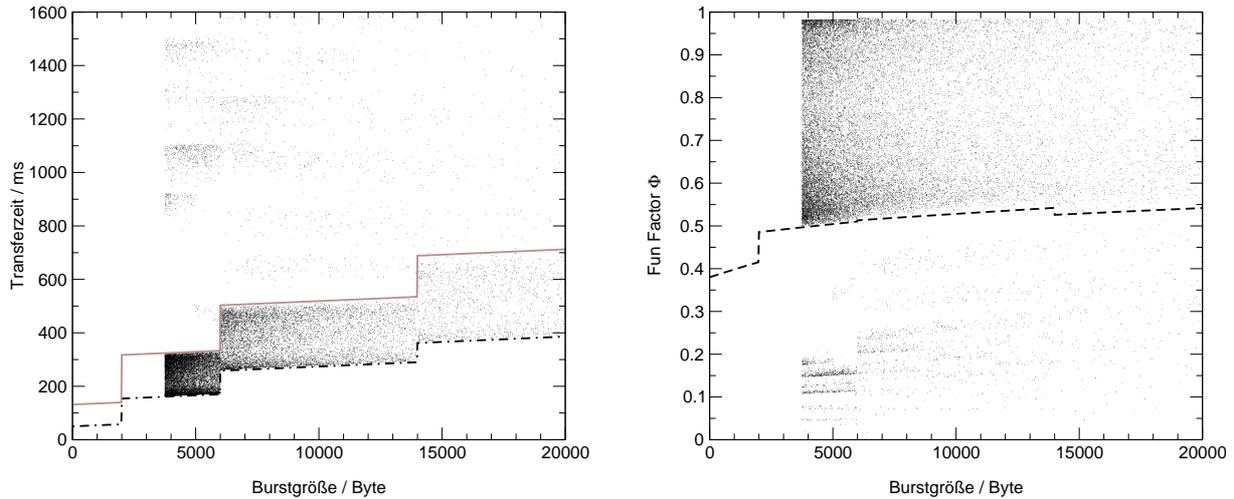
**Bild B.26:** *Fun-Factor*-Verteilungsfunktion für negativ-exponentiell (links) bzw. Pareto-verteilte Burstgröße (rechts) mit Mittelwert  $b = 10\,000$  Byte

insbesondere infolge von Timeouts hin, durch die vor allem bei kurzen Bursts die Transferdauer jeweils um einen näherungsweise konstanten Wert vergrößert wird, der im Verhältnis zu  $T_{\min}$  dominiert. Als Nachteil ist bei dieser Darstellung allerdings zu sehen, dass jeder Burst unabhängig von seiner Größe gleichwertig in die statistische Betrachtung eingeht. Dadurch wird der Einfluss kleiner Bursts, deren Transferzeit zwar deutlich über dem Minimalwert liegt, jedoch trotzdem vom Nutzer nicht unbedingt als besonders störend empfunden wird, überbetont.

### B.2.1.3 Scatter Plots

Eine Möglichkeit, um gleichzeitig einen Einblick in die Verteilung von Transferzeit oder *Fun Factor* ebenso wie in die Abhängigkeit von der Burstgröße zu bekommen, ist die Darstellung der gemessenen Werte in Form eines so genannten *Scatter Plot*. In Bild B.27 sind sämtliche während einer Simulation mit den angegebenen Parametern gemessenen Werte für Transferzeit bzw. *Fun Factor*  $\Phi$  als Einzelpunkte über der jeweiligen Burstgröße aufgetragen. Eine hohe Punktdichte deutet auf eine hohe Wahrscheinlichkeitsdichte für entsprechende Wertepaare aus Burstgröße und Transferzeit bzw. *Fun Factor* hin. Darüber hinaus ist zu beachten, dass für die Burstgröße nur Werte auftreten, die größer als 3 750 Byte sind. Dieser Wert ergibt sich gemäß Gleichung (4.15) als Minimalwert bei der Pareto-Verteilung aus Erwartungswert  $b$  und Formparameter  $\alpha$ .

Im *Scatter Plot* für die Transferzeiten ist neben der unteren Grenze  $T_{\min}(s)$  noch eine weitere Hilfskurve eingezeichnet, welche die maximale Transferzeit all jener Bursts beschreibt, bei denen es nicht zu Paketwiederholungen kommt. Bei diesen Bursts ist die Erhöhung der Transferzeit gegenüber  $T_{\min}(s)$  allein durch die Verzögerung im Engpassknoten sowie die Bedienzeiten auf dem Engpasslink und dem senderseitigen Zugangslink bedingt. Die entsprechende



**Bild B.27:** *Scatter Plots* der Werte von Transferzeit (links) bzw. *Fun Factor*  $\Phi$  (rechts) für Pareto-verteilte Burstgröße,  $b = 10\,000$  Byte,  $R_{max} = 2$  Mbit/s,  $A = 0.9$

Hilfskurve lässt sich also berechnen, indem in Gleichung (A.12) zur Bestimmung von  $T_{min}(s)$  anstelle der absoluten Verzögerung  $\tau$  der Wert

$$\hat{\tau} = \tau + 2 \cdot \frac{L_{MSS}}{C} + \frac{S}{C} \quad (\text{B.12})$$

eingesetzt wird. Da außerdem die zusätzliche Verzögerung  $\hat{\tau} - \tau$  nur im Hinkanal auftritt, während bei der Berechnung von  $T_{min}(s)$  davon ausgegangen wird, dass sich die absolute Verzögerung gleichmäßig auf Hin- und Rückkanal aufteilt, muss zum Ergebnis der Berechnung noch  $(\hat{\tau} - \tau)/2$  hinzuaddiert werden.

Alle Punkte, die außerhalb des durch die beiden Hilfskurven eingegrenzten Bereichs liegen, repräsentieren Burstübertragungen, bei denen es zu Paketwiederholungen kommt. Hierbei können gewisse Häufungen beobachtet werden, die hauptsächlich bei kleinen Transfers auftreten, bei denen Paketverluste häufig durch Timeouts behoben werden. Nach welcher Zeit es dabei zu einem Timeout kommt, hängt davon ab, welches Paket verloren gegangen ist. So erhöht sich z. B. für  $B \leq 4 \cdot L_{MSS}$  die Transferzeit etwa um den initialen Timer-Wert von 1000 ms, falls nur das erste Paket verloren gegangen ist.

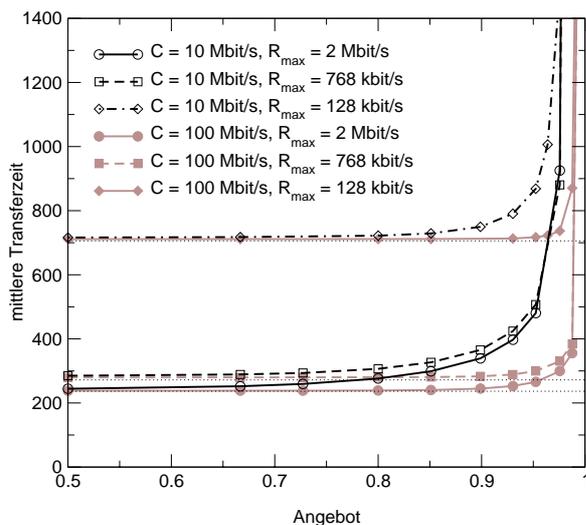
Die Häufungen sind ebenso beim *Scatter Plot* für den *Fun Factor* zu erkennen. Hier ist nun klar zu sehen, dass es hauptsächlich kleine Bursts sind, die für die extrem kleinen Werte von  $\Phi$ , die bereits in Bild B.26 beobachtet wurden, verantwortlich sind. Auch in diesem Diagramm können Burstübertragungen identifiziert werden, die keine Verluste erleiden mussten. Der *Fun Factor* für diese Burstübertragungen liegt oberhalb einer Kurve, die sich aus einer Division der Werte der beiden o. g. Hilfskurven berechnen lässt.

## B.2.2 Betrachtung unter variablem Angebot

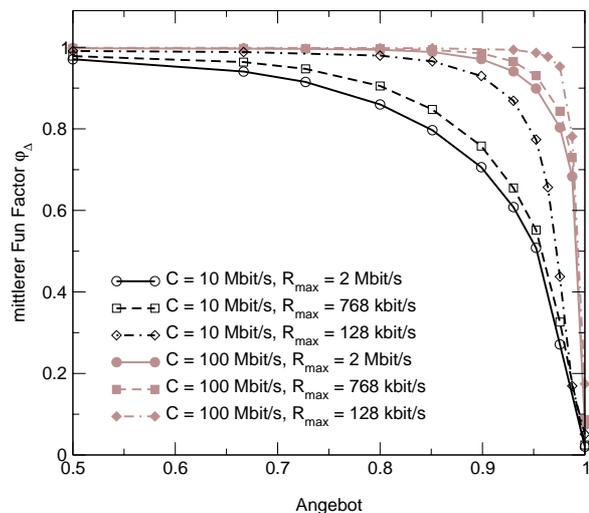
Nachdem der letzte Abschnitt der Untersuchung einzelner Lastpunkte mit Hilfe bedingter Mittelwerte und Verteilungen gewidmet war, werden nun Mittelwerte von Transferzeit und *Fun Factor* in Abhängigkeit vom Angebot betrachtet. Bei sehr hohem Angebot spielt hier außerdem die maximale Anzahl von gleichzeitig aktiven TCP-Verbindungen eine Rolle, die auf einen konstanten Wert von  $n_{max} = 5\,000$  gesetzt wird.

Bild B.28 zeigt zunächst die Ergebnisse für die mittlere Transferzeit bei unterschiedlichen Werten von  $C$  und  $R_{max}$ . Die untere Grenze, die sich ergibt, wenn kein Engpasslink vorhanden ist, kann mit Hilfe von Gleichung (A.46) berechnet werden. Erkennbare Unterschiede zu dieser Kurve können für Angebote im Bereich  $A > 0.8$  festgestellt werden. Für  $A \rightarrow 1$  folgt ein exponentieller Anstieg, wobei aufgrund der o. g. Beschränkung der Anzahl von Verbindungen der resultierende Wert stets endlich bleibt.

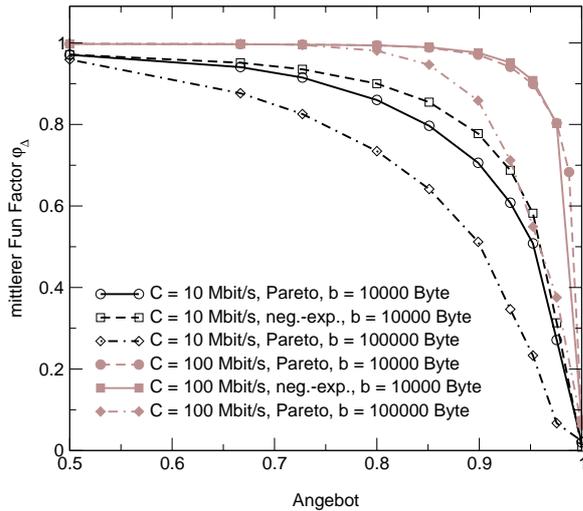
Analog zu den Untersuchungen des vorigen Abschnitts wird auch hier eine alternative Darstellung präsentiert, bei der zum einen die Normierung auf den Minimalwert bereits enthalten ist und die zum anderen einen überschaubaren Wertebereich  $[0, 1]$  aufweist. In Bild B.29 ist daher der mittlere *Fun Factor*  $\varphi_{\Delta}$  nach Gleichung (4.28) zu sehen. Steigende mittlere Transferzeiten korrespondieren mit abnehmenden Werten dieses Leistungsmaßes, wobei für  $A \rightarrow 1$  ein Wert nahe 0 erreicht wird. Wie bereits in Abschnitt B.2.1.1 diskutiert wurde, nimmt der mittlere *Fun Factor* für größere Zugangsraten bei gleichem Angebot kleinere Werte an. Dieser Einfluss kann auch in Bild B.29 beobachtet werden. Offensichtlich erfolgt der Abfall von  $\varphi_{\Delta}$  im Fall von  $C = 100$  Mbit/s erst viel später als bei einer Linkrate von  $C = 10$  Mbit/s. Damit zeigt sich an dieser Stelle wieder der in der Verkehrstheorie häufig zu Tage tretende Effekt des Bündelungsgewinns. Gleichzeitig bedeutet dies aber auch einen wichtigen Unterschied zum Modell



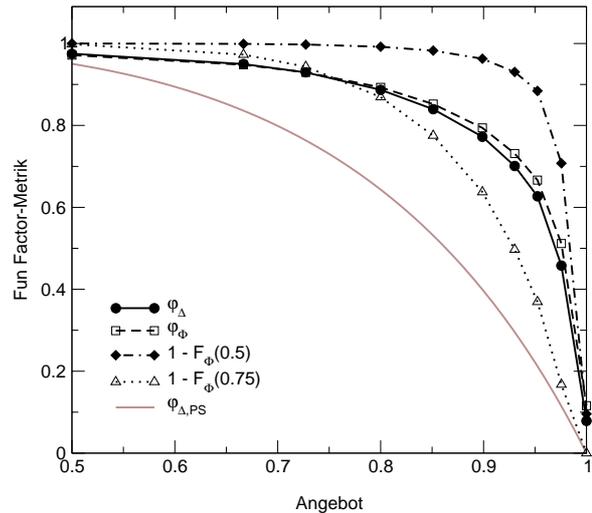
**Bild B.28:** Mittlere Transferzeit (Pareto-vert. Burstgröße,  $b = 10\,000$  Byte)



**Bild B.29:** Mittlerer *Fun Factor*  $\varphi_{\Delta}$  (Pareto-vert. Burstgröße,  $b = 10\,000$  Byte)



**Bild B.30:** Einfluss der Burstgrößenverteilung auf den mittleren *Fun Factor*  $\varphi_{\Delta}$  bei  $R_{max} = 2$  Mbit/s



**Bild B.31:** Vergleich von Leistungsmetriken (Pareto-Burstgrößenverteilung,  $b = 10\,000$  Byte,  $C = 10$  Mbit/s,  $R_{max} = 2$  Mbit/s)

mit ungesättigten Quellen. Denn dort bekommt jede Quelle im Mittel ihren fairen Anteil  $C/n$  an der Linkrate mit lediglich geringen Abweichungen (siehe Abschnitt B.1.1), sodass bei gleichmäßiger Erhöhung von Linkrate  $C$  und Quellenanzahl  $n$  keine Verbesserung erfolgt.

Dass Verteilung und Mittelwert der Burstgröße – im Gegensatz zum PS-Modell – einen nicht zu vernachlässigenden Einfluss haben, zeigt sich auch bei Betrachtung von  $\varphi_{\Delta}$  über dem Angebot (Bild B.30). Insbesondere ein hoher Mittelwert  $b$  wirkt sich über den gesamten Angebotsbereich hinweg leistungsmindernd aus. Dies gilt auch bei einer Linkrate von 100 Mbit/s, während dort der Einfluss der Verteilung geringer ist.

Der mittlere *Fun Factor*  $\varphi_{\Delta}$  ist nur eines von mehreren möglichen Leistungsmaßen für dynamischen TCP-Verkehr, die eine Normierung vornehmen und Werte zwischen 0 und 1 hervorbringen. Alternativen dazu sind in Bild B.31 eingetragen. Der mittlere *Fun Factor*  $\varphi_{\Phi}$ , der sich von  $\varphi_{\Delta}$  in der Art der Mittelwertbildung unterscheidet (siehe Abschnitt 4.3.2.2), liefert nur geringfügig höhere Werte. Ein anderes mögliches Leistungsmaß ist in der Wahrscheinlichkeit dafür gegeben, dass  $\Phi$  eine bestimmte Schwelle unterschreitet, was sich mit Hilfe der komplementären Verteilungsfunktion von  $\Phi$  ausdrücken lässt. Beispielhaft enthält Bild B.31 zwei Kurven, die sich aus der Wahl von 0.5 bzw. 0.75 als Schwellwerte in diesem Sinn ergeben. Darüber hinaus sind in Bild B.31 die Werte des mittleren *Fun Factor*  $\varphi_{\Delta,PS}$  eingezeichnet, die im PS-Modell aus der Beziehung

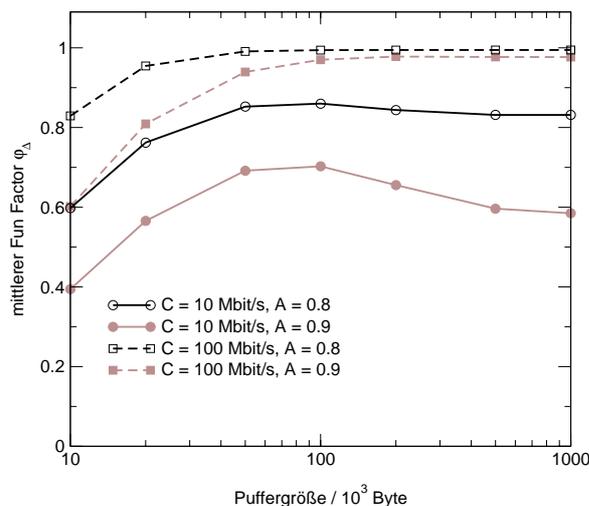
$$\varphi_{\Delta,PS} = \frac{1}{f_{\Delta}} \tag{B.13}$$

resultieren, wobei  $f_{\Delta}$  nach Gleichung (B.11) berechnet werden kann. Dabei ist erneut zu sehen, dass das PS-Modell deutlich geringere Werte liefert als die TCP-Simulation.

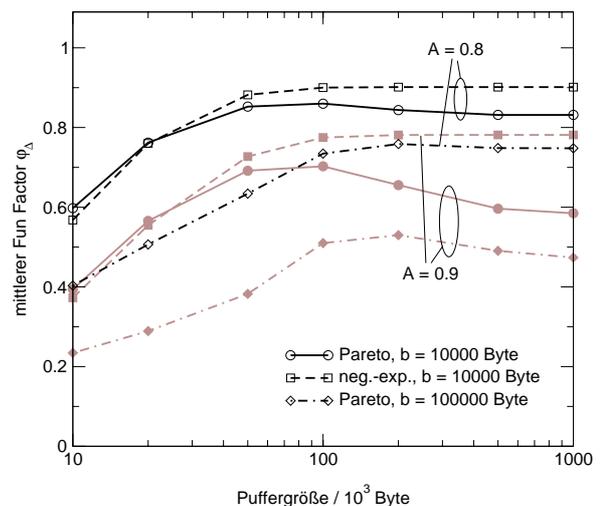
### B.2.3 Einfluss der Puffergröße

Abschließend wird für eine Zugangsrate von  $R_{max} = 2$  Mbit/s der Einfluss der Puffergröße  $S$  auf die Verkehrsleistung bei dynamischem TCP-Verkehr untersucht. Dazu ist in Bild B.32 der mittlere *Fun Factor*  $\varphi_{\Delta}$  für eine Pareto-verteilte Burstgröße mit Formparameter  $\alpha = 1.6$  und Mittelwert  $b = 10\,000$  Byte sowie unterschiedliche Werte von Linkrate und Angebot über der Puffergröße aufgetragen. Wie schon bei den Untersuchungen mit ungesättigten Quellen festgestellt wurde (Abschnitt B.1.1.2), führt ein kleiner Puffer zu Leistungseinbußen. Allerdings ist im Gegensatz zum Szenario mit ungesättigten Quellen weniger die maximale Verzögerung  $S/C$  ausschlaggebend als die Puffergröße selbst. Während für  $C = 100$  Mbit/s mit zunehmender Puffergröße eine Sättigung eintritt, lässt Bild B.32 im Fall  $C = 10$  Mbit/s einen Rückgang von  $\varphi_{\Delta}$  bei großen Werten von  $S$  erkennen. Ein Maximum ergibt sich im Bereich von  $S = 10^5$  Byte. Dies ist gleichzeitig der Wert, ab dem bei  $C = 100$  Mbit/s kein weiterer nennenswerter Anstieg mehr erfolgt. Bild B.33 zeigt, dass das Auftreten eines solchen Maximums stark von der Burstcharakteristik abhängt. Bei einem größeren Erwartungswert der Burstgröße verschiebt sich das Maximum hin zu größeren Werten von  $S$ . Im Fall einer negativ-exponentiell verteilten Burstgröße tritt ein solches gar nicht mehr auf.

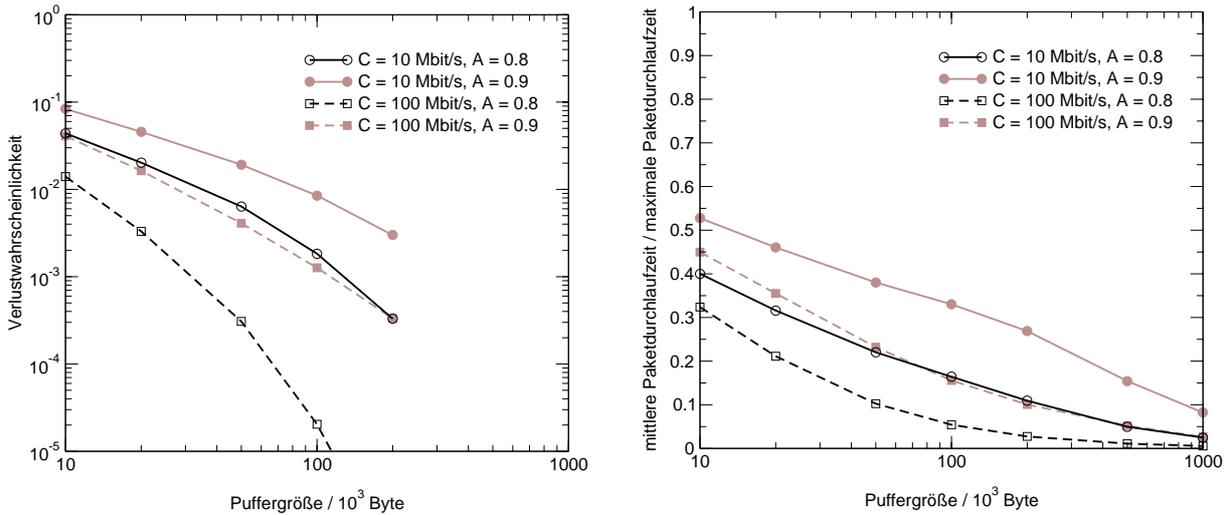
Neben dem Einfluss der Puffergröße auf Leistungsmetriken der Transportprotokollebene ist auch die Betrachtung von Paketebenenmaßen wie Verlustwahrscheinlichkeit und mittlere Paketdurchlaufzeit interessant (Bild B.34). Die Verlustwahrscheinlichkeit nimmt mit zunehmender Puffergröße rasch ab. Ab einer Puffergröße von  $200 \cdot 10^3$  Byte werden überhaupt keine Verluste mehr gemessen. Unterschiede zum Fall mit gesättigten TCP-Quellen ergeben sich vor allem im Hinblick auf die mittlere Paketdurchlaufzeit. Während bei den gesättigten Quellen anhand von Bild B.7 nachgewiesen wurde, dass die mittlere Paketdurchlaufzeit in den meisten Fällen annähernd so groß ist wie der Maximalwert, lässt Bild B.34 wesentlich gerin-



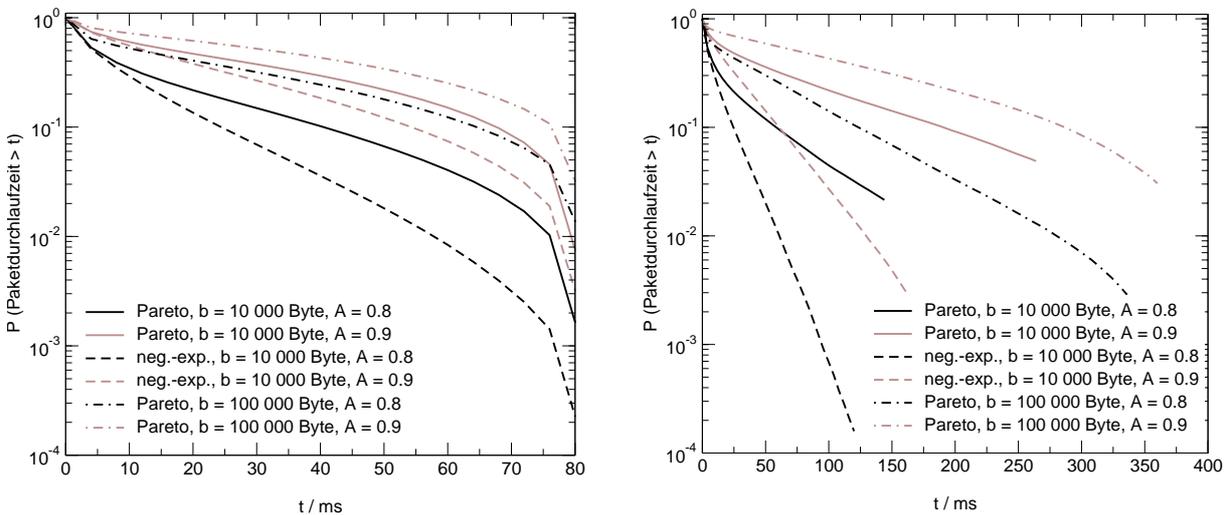
**Bild B.32:** Mittlerer *Fun Factor*  $\varphi_{\Delta}$  für Pareto-verteilte Burstgröße,  $b = 10\,000$  Byte



**Bild B.33:** Mittlerer *Fun Factor*  $\varphi_{\Delta}$  für  $C = 10$  Mbit/s



**Bild B.34:** Verlustwahrscheinlichkeit (links) bzw. mittlere Paketdurchlaufzeit (rechts) für Pareto-verteilte Burstgröße mit Mittelwert  $b = 10\,000$  Byte



**Bild B.35:** Paketdurchlaufzeit-CCDF für  $C = 10$  Mbit/s und  $S = 100\,000$  Byte (links) bzw.  $S = 500\,000$  Byte (rechts)

gere mittlere Durchlaufzeiten für dynamischen TCP-Verkehr erkennen. Dabei ist mit steigender Puffergröße ein starker Rückgang des relativen mittleren Pufferfüllstandes verbunden.

Dies verdeutlicht auch Bild B.35, das die Paketdurchlaufzeit-CCDF für unterschiedlich große Puffer darstellt. Im Gegensatz zum Modell mit ungesättigten TCP-Quellen (vgl. Bild B.14) ist ein näherungsweise exponentieller Abfall festzustellen. Dies bedeutet, dass die Wahrscheinlichkeit, einen nahezu vollen Puffer anzutreffen, insgesamt sehr gering ist und bei hoher Pufferkapazität praktisch vollständig verschwindet.