Connection-oriented and Connection-less Mechanisms of a Scalable Traffic Management for IP Networks

Summary

Resource provisioning in packet switching networks is more flexible and efficient than in circuit switching networks. Whereas circuit switching networks assign channels of fixed bandwidth for the whole lifetime of a connection, ideal packet switching networks have to assign resources only when a packet is being transmitted. For this reason, a first connection-less packet switching network was developed and launched in the late sixties and early seventies. Today's world wide Internet, which is the most important network based on TCP/IP (Transport Control Protocol/Internet Protocol) protocols, has emerged from this ARPANET. At first it was merely used for the exchange of data and messages.

At the same time, connection-oriented packet switching networks have been developed, resulting in such standards and networks as X.25, Frame Relay, ATM (Asynchronous Transfer Mode) and finally MPLS (Multiprotocol Label Switching). Those networks support virtual connections. During set-up, the network determines an appropriate path for a connection and fixes this path by means of switching tables in each node. A unique identifier is assigned to each connection on a per-link basis and included in the packet header before transmission over a link. This enables nodes inside the network to map incoming packets to connections by looking for matching switching table entries and to process and forward packets as agreed upon during connection set-up. In this way, connection oriented packet switching networks – unlike IP networks – do not only avoid making routing decisions for every individual packet but benefit from powerful connection-oriented traffic management functionality as well, thus being prepared to meet the whole spectrum of application QoS (Quality of Service) requirements.

However, the volume of data traffic, i.e. traffic from applications that are insensitive to delay and jitter, has turned out to increase more than telephony, video conferencing and further applications which – because of their strict QoS requirements – benefit most from connection oriented traffic management.

Connection-less packet switching networks are obviously the better solution for data traffic. In such networks, the effort for connection control, a considerable part of which is in fact caused by traffic control to support QoS, does not apply. So, connection-less networks are probably the better starting point for a solution that also includes support for applications with stricter QoS requirements.

A ubiquitous protocol architecture has additional technical and economic advantages. That is also the reason why IP based architectures penetrate domains that have been previously reserved for competing architectures. The most outstanding examples are Voice over IP, which aims at substituting classical circuit switching telephony and introducing multimedia applications, as well as mobile networks beyond 2G (the second generation of public land mobile networks). Convergence of all those types of networks with their broad range of applications based on IP requires increased efforts to realise a consistent traffic control architecture.

In order to support connection-oriented traffic management functions, network nodes need to manage connection states. If such states had to be maintained for individual connections, nodes in wide area networks would probably run out of processing and memory resources.

Therefore, while still being committed to supporting QoS, the Differentiated Services architecture abandons connections. As a result of this, the close relationship between deterministic or statistical QoS parameters as agreed upon in a traffic contract and traffic control functions is released. Instead, a network confines itself to mapping traffic streams to a certain traffic class (behaviour aggregate) and marking them accordingly. The goal is to obtain a particular forwarding behaviour in network nodes. Even if network resources are over-provisioned or adapted to meet a monitored demand, absolute QoS still cannot be guaranteed to a single flow but possibly in terms of a statistical middle-term average of many flows. Hence, due to their connection-less approach, Differentiated Services miss a lot of the opportunities that may be offered by a connection-oriented and at the same time application-oriented traffic management.

For this reason, this thesis combines connection-oriented and connection-less mechanisms to a consistent traffic management concept for a packet switching network. All network nodes support both virtual connections and aggregates as well as connectionless per-hop datagram forwarding. The first part of this thesis, namely chapter 3 and 4, is concerned with an implementation and comprehensive performance evaluation of a traffic management concept for a prototype router. This

concept employs connection aggregation and dynamic bandwidth management to reduce the control overhead induced by signalling and per-connection state while still guaranteeing absolute and reliable QoS. The second part, chapter 5 and 6, studies connection-less traffic management for elastic traffic streams using TCP to adapt their rate to changing traffic conditions. Advanced tools have been developed to ease systematic performance evaluation under varying conditions.

The combination of connection aggregation for applications with strict QoS requirements on the one hand and purely connection-less traffic management for elastic traffic on the other ensures that the requirements both in terms of scalability and in terms of QoS are met. A differentiated treatment of packets based on Differentiated Services is still possible but not considered within this thesis.

A connection-oriented vs. a connection-less traffic management approach has a strong impact on the quality of the available traffic management functions and as a result on the QoS assurances a user may expect of the network. Chapter 2 elaborates further on this relationship with the most important traffic management architectures for IP networks, namely Resource Reservation Protocol (RSVP)/Integrated Services and Differentiated Services for applications with strict QoS requirements as well as TCP based congestion control for elastic data streams. Since the latter is mainly based on distributed algorithms executed in the end points of TCP connections and networks nodes have been involved more and more only recently, it is often also referred to as end-to-end congestion control.

In chapter 3, mechanisms for the multiplexing of connections into aggregates are evaluated and improved to account for practical constraints. The chapter mainly addresses methods for the calculation of the resource requirements of aggregates and dynamic bandwidth management. Deterministic methods based on network calculus are simple but use network resources inefficiently. Methods that exploit statistical multiplexing gain are more efficient in this respect. The large deviation theory has proven to be a powerful tool to compute the effective bandwidth of a connection dependent of the composition of an aggregate. However, whenever a connection is added to or dropped from an aggregate, the degree of statistical multiplexing as represented by the parameters s and (if buffering is taken into account) t in Kelly's expression (3.79) for the effective bandwidth changes thus affecting the effective bandwidth of all connections within the aggregate (Fig. 3.10 , Fig. 3.11, Fig. 3.12, Fig. 3.13 and Fig. 3.14). This is in contrast to the deterministic approach as carried out with the example of an RPQ (Rotating Priority Queues) scheduler, where only the contribution of the connection that is added or dropped has to be considered (section 3.1.5 and 4.3.1). As can be observed in the aforementioned figures, the parameters s and t change the less the bigger the aggregate is. The proposal of a practical algorithm to reduce the computational effort when deploying the

Many Sources Asymptotic in conjunction with periodic on/off sources makes use of this fact (section 3.2.4.4).

Those results as well as the analysis of various models that may serve as a basis for dynamic bandwidth management have influenced the design of the connection-oriented traffic management functionality for the prototype router, the architecture of which is described in chapter 4. This prototype serves as a platform for performance evaluation.

Fig. 4.8 and Fig. 4.9 demonstrate that signalling can control the set-up and modification of an aggregate and the configuration of the scheduler at the entry to the aggregate to ensure the requested QoS. The resource requirements of the aggregate may be calculated based on either deterministic or statistical methods. The comparison of those methods does not only take the efficient use of bandwidth and buffer but also the computational effort into consideration (Fig. 4.13, Fig. 4.14, Fig. 4.15 and Fig. 4.16).

Aggregation reduces the effort needed for managing connection state and for calculating the resource requirements in network nodes being passed by connection aggregates. Moreover, dynamic bandwidth management ensures that a set-up, modification or release of a connection in the aggregate does not necessarily result in signalling activities on the aggregate's level. A simple heuristic algorithm developed for that purpose is sufficient to prove the strengths of the connection oriented, aggregation enabled traffic management.

Chapter 5 provides an overview of congestion control for elastic traffic. Since the majority of elastic applications employs TCP to adjust its rate to varying load conditions within the network, the discussion focuses on TCP and connection-less traffic control functions inside network nodes that are tailored to TCP. Furthermore, the simulation environment and the specific performance metrics used for a systematic performance evaluation are introduced.

Among those metrics, Kelly's utility, which combines throughput and delay based on a model of TCP's congestion avoidance, plays an outstanding role in the performance evaluation as summarised in chapter 6. A remarkable result is that Kelly's utility sharply drops when the offered load approaches 1. This is in contrast to what one might expect of congestion control. Furthermore, adaptive RED (Random Early Detection) or REM (Random Early Marking) in combination with ECN (Explicit Congestion Notification) increase the number of simultaneously active connections and the ratio of packets that are retransmitted due to retransmission timeouts as compared to plain TCP and tail drop. In principle, connection interrupts by impatient users may play a similar role as connection admission control for virtual connections if the network is heavily loaded.

The results as presented in this thesis confirm that both connection-oriented an connection-less network services can be supported efficiently.

In spite of the exponentially increasing traffic demand, increasing transport network capacity raises the question whether QoS issues may be resolved by over-provisioning and/or simple priority mechanisms. The response to this question depends on which types of application will dominate in future broadband networks, which is hard to predict. Therefore, the role of overprovisioning, priority mechanisms and reservation remains open for the time being. At the end, user demand will decide.

Inhaltsverzeichnis

Abkürzungen

AAL	ATM Adaptation Layer
ABR	Available Bit Rate
ABE	Alternative Best Effort
ACK	Acknowledgement, Quittierung bei TCP
AIMD	Additive Increase Multiplicative Decrease
API	Application Programming Interface
ARP	Address Resolution Protocol
ATM	Asynchronous Transfer Mode
ATMARP	ATM Address Resolution Protocol
ATMTCD	ATM Traffic Control Demon
AQM	Active Queue Management
BA	Behaviour Aggregate
BGRP	Border Gateway Reservation Protocol
CAC	Call/Connection Admission Control
CBR	Constant Bit Rate
CE	Congestion Experienced
CLIP	Classical IP over ATM
CS-DMPP	Continuous State Deterministically Modulated Poisson Process
CPU	Central Processing Unit
CW	Congestion Window

CWR	Congestion Window Reduced
DLCI	Data Link Connection Identifier
DS	Differentiated Services
DSD	Duplicate Scheduling with Deadlines
ECN	Explicit Congestion Notification
ECT	ECN-Capable Transport
EDD	Earliest Due Date
EDF	Earliest Deadline First
ELAN	Emulated LAN
EMW	CAC-Verfahren nach Elwalid, Mitra und Wentworth [59]
EOF	End of File
EWMA	Exponentially Weighted Moving Average
FEC	Forwarding Equivalence Class
FIFO	First In First Out
F2VM	Flow-to-VC Mapping Control Module
GCRA	Generic Cell Rate Algorithm
GFR	Guaranteed Frame Rate
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
INT	Interrupt
ISDN	Integrated Services Digital Network
LAN	Local Area Network
LANE	LAN Emulation
LEC	LAN Emulation Client
LIS	Logical IP Subnetwork

LLC	Logical Link Control
LSP	Label Switched Path
MAC	Medium Access Control
MAP	Markovian Arrival Process
MPC	MPOA Client
MPLS	Multiprotocol Label Switching
MPOA	Multiprotocol over ATM
MSA	Many Sources Asymptotic
MSS	Maximum Segment Size
NHRP	Next Hop Resolution Protocol
NHS	Next Hop (Resolution Protocol) Server
NPC	Network Parameter Control
nrt-VBR	Non-Real-Time VBR
OPWA	One Pass with Advertising
PC	Personal Computer
PCR	Peak Cell Rate
PHB	Per Hop Behaviour
RCS	Rate Controlled Service (Discipline)
RED	Random Early Detection
REM	Random Early Marking
RFC	Request for Comments (Empfehlungen der IETF)
RPQ	Rotating Priority Queues
RSVP	Resource Reservation Protocol
RTP	Real-Time Transport Protocol
RTCP	Real-Time Control Protocol
RTT	Round Trip Time

RTO	Retransmission Timeout
rt-VBR	Real-Time VBR
SAAL	Signalling AAL
SACK	Selective Acknowledgement
SCR	Sustainable Cell Rate
SDU	Service Data Unit
SIMA	Simple Integrated Media Access
SLS	Service Level Specification
SRP	Scalable Reservation Protocol
TCS	Traffic Conditioning Specification
TCP	Transmission Control Protocol
TOS	Type of Service
UBR	Unspecified Bit Rate
UDP	User Datagram Protocol
UNI	User Network Interface
UPC	Usage Parameter Control
VBR	Variable Bit Rate
VCI	Virtual Channel Identifier
VSMM	VC Setup and Modification Module
VPI	Virtual Path Identifier
WEDD	Weighted Earliest Due Date
WZ	Working Zone

Formelzeichen

Funktionen und Operatoren

$\lceil x \rceil$	Die kleinste ganze Zahl, die größer oder gleich x ist
$\lfloor x \rfloor$	Die größte ganze Zahl, die kleiner oder gleich x ist
$\langle y \mid x \rangle$	Wert des linearen Funktionals y im Punkt x [100]
$1\{A\}$	Indexfunktion. Die Funktion hat den Wert 1, wenn die Aussage A wahr ist, sonst den Wert 0
$\mathrm{E}\{X\}$	Erwartungswert einer Zufallsgröße X
$\mathrm{E}_s\{X\}$	Erwartungswert einer Zufallsgröße X nach der Transformation ihrer Verteilungsfunktion mit der *Probability Shift* Methode [170]
$(f_1 \oplus f_2)(x)$	Faltungsoperation $\inf\{f_1(x_1)+f_2(x_2) \mid x_1+x_2=x\}$
grad $f(x, y, ...)$	Gradient einer von mehreren Variablen abhängigen Funktion f
$\inf_{t \geq a}\{f(t)\}$	Infimum der Funktionswerte von f im Intervall $[a, \infty)$, $a \in \mathbb{R}$
$\max_{t \geq a}\{f(t)\}$	Maximum der Funktionswerte von f im Intervall $[a, \infty)$, $a \in \mathbb{R}$
$\max\{f(t), g(t)\}$	Funktionswert zum Zeitpunkt t ist $f(t)$, wenn $f(t) \geq g(t)$, sonst $g(t)$
$\min\{f(t), g(t)\}$	Funktionswert zum Zeitpunkt t ist $f(t)$, wenn $f(t) \leq g(t)$, sonst $g(t)$
$\max\{a_i\}$	Maximum der Menge mit den Elementen a_i
$\min\{a_i\}$	Minimum der Menge mit den Elementen a_i
$\mathrm{P}\{A\}$	Wahrscheinlichkeit für Ereignis A
$\mathrm{P}\{A \mid B\}$	Wahrscheinlichkeit von A unter der Bedingung B

$\sup\{a_i\}$	Supremum der Menge mit den Elementen a_i
$\sup_{t \geq a}\{f(t)\}$	Supremum der Funktionswerte von f im Intervall $[a, \infty)$, $a \in \mathbb{R}$
$\mathrm{Var}\{X\}$	Varianz einer Zufallsgröße X
$\mathrm{Var}_s\{X\}$	Varianz einer Zufallsgröße X nach der Transformation ihrer Verteilungsfunktion mit der *Probability Shift* Methode [170]

Formelzeichen

A	Verkehrsangebot
$\boldsymbol{A}$	Matrix mit den Elementen $A_{jr} = 1\{\text{Übertragungsabschnitt } j \in \text{Route der Verbindung } r\}$
$A(t)$	Deterministische obere Schranke für die in einem beliebigen Intervall der Länge t ankommende Datenmenge einer Verbindung
A_A	Verkehrsangebot in einem Aggregat, d. h. der Quotient aus der Summe der mittleren Rate der Verbindungen $\sum_i r_i$ und der für das Aggregat reservierten Rate C_A
$A_1(t), A_i(t), \ldots$	Deterministische obere Schranke für die in einem beliebigen Intervall der Länge t ankommende Datenmenge der Verbindung 1, i, ...
$A_{in}(t)$	Deterministische obere Schranke für die in einem beliebigen Intervall der Länge t am Eingang eines Netzelementes ankommende Datenmenge einer Verbindung
$A_{1,in}(t), A_{i,in}(t)$	Deterministische obere Schranke für die in einem beliebigen Intervall der Länge t am Eingang eines Netzelementes ankommende Datenmenge der Verbindung 1, i
$A_{out}(t)$	Deterministische obere Schranke für die in einem beliebigen Intervall der Länge t ein Netzelement verlassende Datenmenge einer Verbindung
$A_{1,out}(t), A_{i,out}(t)$	Deterministische obere Schranke für die in einem beliebigen Intervall der Länge t ein Netzelement verlassende Datenmenge einer Verbindung 1, i
$A(t_1, t_2)$	Zufallsvariable der im Intervall $[t_1, t_2)$ ankommende Datenmenge eines Zufallsprozesses

$A_N(t_1,t_2)$	Zufallsvariable der im Intervall $[t_1,t_2)$ ankommende Datenmenge eines aus der Überlagerung von N Quellen hervorgehenden Zufallsprozesses
$A^*(t)$	Minimale deterministische obere Schranke (*Minimum Envelope Process* [40]) für die in einem beliebigen Intervall der Länge t ankommende Datenmenge. $A^*(t)$ ist also für alle t kleiner als jede andere deterministische obere Schranke
$A^*(s,t)$	In [40] analog zu $A^*(t)$ für stochastische Ankunftsprozesse definierter *Minimum Envelope Process*
a^*	In [40] als *Minimum Envelope Rate* bezeichneter Grenzwert $\lim_{t\to\infty} A^*(t)/t$ eines Ankunftsprozesses, für den eine deterministische obere Schranke für die in einem Intervall der Länge t Datenmenge angegeben werden kann
$a^*(s)$	In [40] analog zu a^* für stochastische Prozesse definierte *Minimum Envelope Rate*
$a(t)$	Zu diskreten Zeitpunkten t gemessene Zuwächse eines stochastischen Ankunftspozesses
$a_N(t)$	Zu diskreten Zeitpunkten t gemessene Zuwächse eines aus der Überlagerung von N Quellen hervorgehenden stochastischen Ankunftspozesses
$a(s,t)$	Von den Parametern s und t abhängige effektive Bandbreite [114]
B_A	Für ein Verbindungsaggregat vereinbarte Büscheltoleranz, also die Größe des *Bucket* im *Leaky* oder *Token Bucket* genannten Algorithmus zur Quellflußkontrolle
B_T	Büscheldauer, d. h. die Länge des Zeitintervalls, in dem eine Quelle mit der für sie vereinbarten Spitzenrate p sendet
b	Für eine Verbindung vereinbarte Büscheltoleranz
b_i	Für Verbindung i vereinbarte Büscheltoleranz
b_δ	Maximale Abweichung der in einem Intervall ankommenden Datenmenge eines Verkehrsstromes von der aufgrund der vereinbarten mittleren Rate r zu erwartenden, wenn keine Beschränkung der Spitzenrate p vereinbart ist
b_{0i}	Modifizierte Büscheltoleranz einer Verbindung i [128]
C	Kapazität eines Übertragungsabschnittes oder einer Bedieneinheit

$C_1, C_2, ...$	Kapazität des Übertragungsabschnittes 1, 2, ...
C_A	Für ein Aggregat reservierte konstante Rate
$C_{A,1}, C_{A,2}, ...$	Für die Teilaggregate 1, 2, ... reservierten konstanten Raten
$C_A(N)$	Für ein Aggregat aus *n* identischen Verbindungen reservierte Rate
C_j^{GS}	Von der Bedienrate abhängige Komponente der *Latency* einer *Latency Rate* Bedieneinheit nach [176]
C_i^S	Konstante zur Angabe der Abweichung der von einer Bedieneinheit *S* im Intervall $(t_1, t]$ für eine Verbindung *i* geleisteten Arbeit $W_{i,j}^S(t_1, t)$ von der aufgrund der mittleren Bedienrate *R* eigentlich zu erwartenden Bedienarbeit
$C_j(y)$	Kosten für die Nutzung einer Ressource *j*, wenn die Last *y* ist
c	Verkehrsklasse bzw. Verzögerungspriorität einer Verbindung
$\boldsymbol{c}$	Vektor der Kapazitäten aller Übertragungsabschnitte in einem Netz
$c_0, c_1, ...$	Konstanten
$cwnd(t)$	Congestion Window von TCP
$\overline{D_A}$	Mittlere Abweichung eines der effektiven Bandbreite einer Reservierungsanforderung von der geschätzten mittleren effektiven Bandbreite
D_{max}	Maximale Verzögerungszeit von Dateneinheiten
D_{max}^S	Maximale Verzögerungszeit von Dateneinheiten durch eine Bedieneinheit *S*
D_j^{GS}	Von der Bedienrate unabhängige Komponente der *Latency* einer *Latency Rate* Bedieneinheit nach [176]
$D_{i,max}^S$	Maximale Verzögerungszeit von Dateneinheiten von Verbindung *i* durch eine Bedieneinheit *S*
$D_{c,max}^S$	Maximale Verzögerungszeit von Dateneinheiten von Verbindungen der Verkehrsklasse bzw. Verzögerungspriorität durch eine Bedieneinheit *S*
$D_{1,max}, D_{i,max}, ...$	Maximale Verzögerungszeit von Dateneinheiten von Verbindung 1, *i*, ...
$E_L(\Delta t, \epsilon)$	*Local Effective Envelope* [28, 29]

Err	Abweichung eines neuen Meßwertes vom aktuellen *Exponentially Weighted Moving Average*
e_{Di}	Deterministische effektive Bandbreite einer Verbindung i [127]
e_i	Durch ein virtuelles deterministisches Bediensystem modifizierte Spitzenrate einer Verbindung i [59]
$F_X(x)$	Wahrscheinlichkeitsverteilungsfunktion einer Zufallsvariablen X, d. h. $\mathrm{P}\{X \leq x\}$
$F_{X,s}(x)$	Wahrscheinlichkeitsverteilungsfunktion einer Zufallsvariablen X nach der Transformation mit der *Probability Shift* Methode [170]
$f(x)$	Funktion
$f^*(x)$	Young-Fenchel Transformierte der Funktion f [100]
$f_X(x)$	Wahrscheinlichkeitsdichtefunktion einer Zufallsvariablen X
$f_{X,s}(x)$	Wahrscheinlichkeitsdichtefunktion einer Zufallsvariable X nach der Transformation mit der *Probability Shift* Methode [170]
G_A	Granularität, mit der das dynamische Bandbreitemanagement Bandbreite überreserviert
g_i	Gewicht eines Verkehrsstroms in der Definition proportionaler Fairneß
H	Hurst-Parameter
$I(x), I_1(x)$	Die zur Abschätzung der Überschreitungswahrscheinlichkeiten $\mathrm{P}\{X > x\}$ bzw. $\mathrm{P}\{X_1 > x\}$ benötigten *Large Deviation Rate Functions*
$I_G(x)$	*Large Deviation Rate Function* im Theorem von Gärtner und Ellis
i	Ganze Zahl
$K_Ü$	$K_Ü$, die Überreservierung des heuristischen dynamischen Bandbreitemanagements in Vielfachen der Schrittweite G_A
k_i	*Allocation State* der *Working Zone* i im Markoff-Modell des dynamischen Bandbreitemanagement [155]
k_P	Minimalwert der Paretoverteilung

L	(Maximale) Paketlänge
L_i	Maximale Paketlänge einer Verbindung i
l_i	Unterer Randzustand der *Working Zone* i im Markoff-Modell des dynamischen Bandbreitemanagement [155]
$M, M_1, \ldots$	Mengen von Verbindungen
M_c	Menge von Verbindungen der Verkehrsklasse bzw. Verzögerungspriorität c
$M_i(s)$	Momentenerzeugende Funktion der als Zufallsgröße X modellierten Momentanrate einer Verbindung i
$M_X(s)$	Momentenerzeugende Funktion einer Zufallsvariablen X
MSS	Maximum Segment Size
m	Mittlere Rate eines Ankunftsprozesses
m_A	Mittlere Rate eines Verbindungsaggregates
N	Anzahl von Verbindungen
$N_{Ü}$	Anzahl der Überreservierungsfaktoren, die in der Nachbarschaft des aktuellen Wertes $K_{Ü}$ vom dynamischen Bandbreitemanagement jeweils nach unten und oben durchgerechnet werden
N_{WZ}	Anzahl der Zustände, die bei der Berechnung des Prozesses der *Working Zones* des dynamischen Bandbreitemangements berücksichtigt werden
N_j	Anzahl von Verbindungen vom Typ j
N_c	Anzahl von Verkehrsklassen, Verzögerungsprioritäten bzw. Verbindungstypen
N_δ	Von der Überreservierung des heuristischen dynamischen Bandbreitemanagements unabhängiger Faktor in $K_{Ü} N_\delta$, der Anzahl der neuen Meßwerte, nach denen das Gewicht im *Exponentially Weighted Moving Average* eines früheren Meßwertes auf ϵ_C gesunken ist
n	Ganze Zahl

n_c	Natürliche Zahl, welche die maximale Verzögerungszeit einer Verkehrsklasse bzw. Verzögerungspriorität c als Vielfaches einer zu definierenden Zeitspanne angibt
$\boldsymbol{P^T}$	Vektor der Schattenpreise für die Nutzung einer Menge von Ressourcen
P_A	Für ein Verbindungsaggregat vereinbarte Spitzenrate
P_{Loss}	Verlustwahrscheinlichkeit
$P_M(t)$	Bei aktivem Puffermanagement die von der Zeit t abhängige Wahrscheinlichkeit, daß ein Paket markiert oder verworfen wird
$P_M(\overline{A})$	Bei aktivem Puffermanagement die von der mittleren Belegung $\overline{A}$ des Wartespeichers abhängige Wahrscheinlichkeit, daß ein Paket markiert oder verworfen wird
$P_M'(\overline{A})$	In einem Zwischenschritt zur Berechnung von $P_M(\overline{A})$ bei RED [71] benötigte Variable
$P_j(y)$	Schattenpreis für die Nutzung einer Ressource j, d. h. Komponente des Vektors $\boldsymbol{P^T}$
$P_j(y)$	Schattenpreis für die Nutzung einer Ressource j bei der Last y
p	Für eine Verbindung vereinbarte Spitzenrate
p_i	Für eine Verbindung i vereinbarte Spitzenrate
p_{0i}	Modifizierte Spitzenrate einer Verbindung i [128]
Q	Füllstand einer Warteschlange
$\overline{Q}$	(Gemessener) mittlerer Füllstand einer Warteschlange
$Q(t)$	Zufallsprozeß des Füllstandes einer Warteschlange
$Q_N(t)$	Zufallsprozeß des Füllstandes einer Warteschlange in dem mit dem Faktor N skalierten Bediensystem [30]
R	Reservierte Rate
R_A	Für ein Aggregat vereinbarte mittlere Rate
R_i	Für Verbindung i reservierte Rate

r	Für eine Verbindung vereinbarte mittlere Rate (*Sustainable Rate* [11])
r_i	Für eine Verbindung *i* vereinbarte mittlere Rate
r_{0i}	Modifizierte mittlere Rate einer Verbindung *i* [128]
$r(t)$	Momentanrate eines Verkehrsstroms
$r_{in}(t)$	Momentanrate eines Verkehrsstroms am Eingang eines Netzelements
$r_{out}(t)$	Momentanrate eines Verkehrsstroms am Ausgang eines Netzelements
S	Wartespeicher in einem Bediensystem
$S_1, S_2, ...$	Wartespeicher im Bediensystem am Eingang derÜbertragungsabschnitte 1, 2, ...
S_A	Für ein Aggregat reservierter Wartespeicher
$S_{A,1}, S_{A,2}, ...$	Für die Teilaggregate 1, 2, ... reservierter Wartespeicher
S_i^{GS}	*Slack Term* einer Verbindung *i* [176]
S_{min}	Mittlere Belegung des Wartespeichers eines Routers, ab der bei RED [71] das zufällige Verwerfen von Paketen einsetzt
S_{max}	Mittlere Belegung des Wartespeichers, bis zu der bei RED von S_{min} an die Wahrscheinlichkeit des Markierens oder Verwerfens von Paketen linear ansteigt
SND.NXT	Bei TCP die Sendefolgenummer, die als nächste übertragen werden soll
SND.UNA	Das erste unquittierte Byte des Datenstroms bei TCP
s	Komplexe Variable in Integraltransformationen und reelle Variable der momentenerzeugenden Funktion
$s(x)$	Reelle Variable der momentenerzeugenden Funktion bei der *Probability Shift* Methode
$s_a(t), s_v(t)$	Skalierungsfunktion, mit deren Hilfe ein gegenüber dem Theorem von Gärtner und Ellis verallgemeinertes Large Deviation Prinzip etabliert werden kann [54, 55, 56]
$s*$	Extremstelle der aus dem Kontext hevorgehenden Funktion von *s*

s_{0i}	Von einer Verbindung i nach Modifikation der *Token Bucket* Parameter beanspruchter Wartespeicher [128]
$ssthresh(t)$	Slow Start Threshold von TCP
T	Endezeitpunkt einer *Busy Period*
T_B	Beobachtungszeitraum
$T_P, T_{P,1}, T_{P,2}$	Periodendauern periodischer deterministischer Quellen
T_r	Die Umlaufzeit (*Round Trip Time*) einer TCP-Verbindung, d. h. die Zeit, die vom Senden eines Segmentes bis zum Empfang der Quittierung vergeht
T_{0r}	Feste minimale Umlaufzeit (*Round Trip Time*)
$T_{0r,1}, T_{0r,2}, \ldots$	Beiträge der Übertragungsabschnitte 1, 2, ... zur festen minimalen Umlaufzeit T_{0r}
$T_r(t)$	Von der Zeit abhängige Umlaufzeit (*Round Trip Time*)
T_φ	Phasenlage einer periodischen deterministischen Quelle bezogen auf den Zeitpunkt 0
T_{EIN}	Zeitspanne, während der eine Ein-Aus-Quelle einen Datenstrom konstanter Rate erzeugt
T_{AUS}	Zeitspanne, während der eine Ein-Aus-Quelle pausiert
t	Variable der Zeit
t^*	Extremstelle der aus dem Kontext hervorgehenden Funktion von t
$t_1, t_2, \ldots$	Punkte auf der Zeitachse
$t_{1,in}, t_{j,in}, \ldots$	Empfangszeitpunkte von Paket 1, j, ...
$t_{1,out}, t_{j,out}, \ldots$	Sendezeitpunkte von Paket 1, j, ...
$U(\boldsymbol{x})$	Gesamtzufriedenheit in einem Netz, in welchem dem Vektor der Verbindungen ein Vektor $\boldsymbol{x}$ von Raten zugeordnet ist
$U_r(x_r)$	Zufriedenheit eines Benutzers, dessen Verbindung über die Route r läuft und der dabei die Rate x_r zur Verfügung steht

u_i Oberer Randzustand der *Working Zone i* im Markoff-Modell des dynamischen Bandbreitemanagements [155]

$V(\boldsymbol{x}(t),t)$ Ljapunov-Funktion eines durch die Gleichung $\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}(t)=f(\boldsymbol{x}(t),t)$ beschriebenen Systems

$W(t)$ Zufallsprozeß der Restarbeit in einem Bediensystem

$W_N(t)$ Zufallsprozeß der Restarbeit in dem mit dem Faktor *N* skalierten Bediensystem [30]

$W_{i,j}^{S}(t_1,t)$ Die von einer Bedieneinheit *S* für in der *j*-ten *Busy Period* angekommenen Anforderungen der Verbindung *i* im Intervall $(t_1,t]$ geleistete Arbeit

$W_{i,j}^{S_j}(t_1,t)$ Die von einer Bedieneinheit S_j für in der *j*-ten *Busy Period* angekommenen Anforderungen der Verbindung *i* im Intervall $(t_1,t]$ geleistete Arbeit

w_i Durch ein virtuelles deterministisches Bediensystem modifizierte Aktivität einer Verbindung *i* [59]

$w_r(t)$ Permanentes Anheben der Rate einer Verbindung *r*

$X, X_1, \ldots$ Zufallsvariablen

X_i Zufallvariable für die Anzahl aktiver Verbindungen während des Aufenthaltes in *Working Zone i* im Markoff-Modell des dynamischen Bandbreitemanagements [155]

$\boldsymbol{x}$ Vektor von Ressourcen, der auf eine aus dem jeweiligen Zusammenhang hervorgehende Weise einem Vektor von Verkehrsströmen zugeordnet wird

$\boldsymbol{x}(t)$ Lage eines Systems abhängig von der Zeit

$\boldsymbol{x}_G$ Ruhelage eines Systems

x_i Ressource, die einer Verbindung *i* auf eine aus dem jeweiligen Zusammenhang hervorgehende Weise zugeordnet ist

Y Verkehrswert, d. h. die mittlere Anzahl belegter Bedieneinheiten in einem Bediensystem

$Z(t)$ Normalverteilte Zufallsvariable

$\boldsymbol{z}$	Vektor von Schlupfvariablen in Optimierungsaufgaben
α	Anrufrate freier Quellen
α_P	Formparameter der Pareto-Verteilung
β	Exponent, mit dem die Varianz des empirischen Mittelwertes eines Zufallsprozesses bzw. der Autokorrelationsfunktion charakterisiert werden kann
Δ	Rotationsintervall bei RPQ [133, 134]
ΔC_A	Effektive Bandbreite einer Reservierungsanforderungen
$\overline{\Delta C_A}$	(Gemessene) mittlere effektive Bandbreite einer Reservierungsanforderung
Δt	Länge eines Zeitintervalls
δ_C	Konstante des *Exponentially Weighted Moving Average* zur Berechnung der mittleren effektiven Bandbreite von Reservierungsanforderungen für das dynamische Bandbreitemanagement
δ_D	Konstante des *Exponentially Weighted Moving Average* zur Berechnung der mittleren Abweichung der effektiven Bandbreite von Reservierungsanforderungen vom geschätzten Mittelwert
δ_E	Beim Algorithmus für dynamisches Bandbreitemanagement werden extreme und seltene Abweichungen *Err* der effektiven Bandbreite ΔC_A einer Reservierungsanforderung vom geschätzten Mittelwert $\overline{\Delta C_A}$ werden mit Hilfe des Faktors δ_E auf $\delta_E \overline{\Delta C_A}$ beschränkt und auf diese Weise herausgefiltert
δ_G	Faktor, mit dem die geschätzte mittlere Abweichung $\overline{D_A}$ der effektiven Bandbreite in die Berechnung der Schrittweite, der Granularität G_A des heuristischen dynamischen Bandbreitemanagement aus Kapitel 4 eingeht
δ_Q	Konstante des *Exponentially Weighted Moving Average* zur Berechnung der mittleren Belegung des Wartespeichers aus einer Folge von Meßwerten
ϵ	Kleine reele Zahl größer als null
ϵ_B	Obere Schranke für Rufblockierwahrscheinlichkeit

ϵ_C	Bruchteil ϵ_C, auf den sich das Gewicht eines Meßwertes im *Exponentially Weighted Moving Average* des heuristischen dynamischen Bandbreitemanagements nach $K_{Ü} N_{\delta}$ weiteren Meßwerten reduziert hat
ϵ_L	Obere Schranke für Paketverlustwahrscheinlichkeit
ϵ_O	Obere Schranke für eine Überschreitungswahrscheinlichkeit
$\phi(x)$	Konvexe Funktion
$\gamma_X(t)$	Autokorrelationsfunktion eines stationären Zufallsprozesses $X(t)$
λ	Mittlere Rate eines Ankunftsprozesses mit negativ exponentiell verteilten Ankunftsabständen
μ	Enderate aktiver Quellen
Σ	Platzhalter für Bedienstrategie
Θ_i	*Latency* einer *Latency Rate* Bedieneinheit [186] bezüglich Verbindung i
$\Theta_i^{S_j}$	*Latency* einer als S_j bezeichneten *Latency Rate* Bedieneinheit [186] bezüglich Verbindung i
σ^2	Parameter eines *Fractional Brownian Motion* Prozesses
τ	Variable der Zeit eines Integranden
ξ, υ	Bei Adaptivem RED [76] benötigte Konstanten zur Berechnung der Wahrscheinlichkeit, daß ein Paket markiert oder verworfen wird

1 Einführung

Paketvermittelnde Netze sind bezüglich der Bereitstellung von Bandbreite flexibler und effizienter als leitungsvermittelnde Netze. Statt eines Kanals fester Bandbreite werden im Idealfall die Ressourcen des Netzes nur dann belegt, wenn ein Paket übertragen werden muß. Aus diesem Grunde ist bereits Ende der 60er Jahre mit der Entwicklung und dem Aufbau eines verbindungslosen paketvermittelnden Datennetzes begonnen worden. Aus diesem ARPANET hat sich das heute weltumspannende Internet entwickelt, das bedeutendste der auf den TCP/IP-Protokollen basierenden Netze, das zunächst ausschließlich zum Austausch von Daten und Nachrichten genutzt worden ist.

Parallel ist die Entwicklung von verbindungsorientierten paketvermittelnden Netzen vorangetrieben worden. Aus diesen Bestrebungen sind X.25, *Frame Relay*, ATM und zuletzt MPLS entstanden. Alle diese Protokolle stellen bei der Vermittlung von Paketen eine Zuordnung von Paketen zu zuvor eingerichteten virtuellen Verbindungen her. Der Verbindungspfad wird bereits beim Verbindungsaufbau festgelegt und durch Tabellen innerhalb der Netzknoten fixiert. Je Übertragungsabschnitt wird dabei jeder Verbindung eine eindeutige Verbindungskennung zugeordnet und vor der Übertragung eines Paketes in dessen Kopf eingetragen, damit die Knoten den Verbindungsbezug herstellen können und jedes Paket in der für die Verbindung vorgesehenen Art und Weise verarbeiten und weiterleiten können. So werden einerseits die bei IP notwendige vollständige und für jedes Paket einzeln durchgeführte Wegewahl vermieden und andererseits die mit dem Verbindungsbezug einhergehenden Möglichkeiten des Verkehrsmanagements einbezogen, so daß Anwendungen mit ganz unterschiedlichen Dienstgüteanforderungen unterstützt werden können.

In den letzen Jahren hat sich aber gezeigt, daß der Datenverkehr, d. h. der Verkehr von in bezug auf Verzögerungen und Lastschwankungen im Netz unempfindlichen Anwendungen, weit stärker zunimmt als Telefonie, Videokonferenzen und andere Anwendungen, die aufgrund ihrer meist sehr strikten Dienstgüteanforderungen von verbindungsorientierter Verkehrssteuerung profitieren.

Für Anwendungen, die Datenverkehr erzeugen, sind verbindungslose paketvermittelnde Netze offensichtlich die bessere Lösung. In diesen Netzen entfällt der für die Verbindungssteuerung notwendige Aufwand, der zu einem beträchtlichen Teil auf die Maßnahmen der Verkehrssteuerung zur Sicherung einer für diese Anwendungen gar nicht benötigten Dienstgüte zurückzuführen ist. Deshalb liegt es näher, ausgehend von einem verbindungslosen Netz, das für den größeren Teil des Verkehrs günstige Eigenschaften aufweist, nach einer Lösung zu suchen, die auch Anwendungen mit strikten Dienstgüteanforderungen einschließt, als umgekehrt in verbindungsorientierten paketvermittelnden Netzen, die für den kleineren Teil der Anwendungen Vorteile bieten, Nachbesserungen zum effizienteren Transport dieser Klasse von Anwendungen vorzunehmen.

Zusätzliche technische und wirtschaftliche Vorteile können dann realisiert werden, wenn diese paketvermittelnden Netze mit Hilfe einer allgegenwärtigen Protokollarchitektur gesteuert werden. IP basierte Architekturkonzepte dringen wohl deshalb zunehmend auch in Domänen ein, die bislang mit IP konkurrierenden Architekturen vorbehalten waren. Die bekanntesten Beispiele sind *Voice over IP*, das sicherlich die klassische leitungsvermittelnde Telefonie ablösen und um Multimedia-Anwendungen bereichern wird, und mehr noch die auf die zweite Generation der Mobilfunknetze folgenden nächsten Generationen. Es beginnt sich die Konvergenz all dieser Netze auf der Basis von IP abzuzeichnen, die angesichts der Schwächen der IP-Architektur diesbezüglich verstärkte Anstrengungen zur Realisierung einer durchgängigen und leistungsfähigen Verkehrssteuerungsarchitektur zwingend erforderlich macht.

1.1 Verkehrssteuerung in diensteintegrierenden paketvermittelnden Netzen

Für verbindungsorientierte Verkehrssteuerungsfunktionen müssen alle Netzknoten Verbindungszustände einrichten und verwalten können. Ohne das Zusammenfassen von Verkehrsströmen zu Aggregaten werden in Weitverkehrsnetzen die Knoten vermutlich die dafür notwendigen Ressourcen nicht oder jedenfalls nicht wirtschaftlich bereitstellen können.

In der *Differentiated Services* Architektur wird aus diesem Grunde sogar völlig auf Verbindungsbezug verzichtet. Gleichzeitig wird die für verbindungsorientiertes Verkehrsmanagement typische enge Kopplung der im Verkehrsvertrag festgelegten deterministischen oder statistischen Dienstgüteparametern und der Verkehrssteuerungsfunktionen aufgehoben. Statt dessen kann das Netz zusichern, bestimmte Verkehrsströme innerhalb der im Verkehrsvertrag spezifizierten Schranken einer

bestimmten Verkehrsklasse zuzuordnen und sie am Eingang des Netzes entsprechend zu markieren, um sie innerhalb des Netzes mit Priorität übertragen oder ihnen auf andere Weise eine Sonderbehandlung zukommen lassen zu können. Wenn die Ressourcen des Netzes überdimensioniert oder aufgrund von Verkehrsmessungen der Nachfrage entsprechend angepaßt werden, kann zwar nicht für einzelne Verkehrsströme, möglicherweise aber im statistischen Mittel über eine längere Zeit und größere Anzahl von Verkehrsströmen die gewünschte Dienstgüte erreicht werden. Der völlige Verzicht auf Verbindungsbezug verbaut jedoch eine Reihe von Möglichkeiten eines sich unmittelbar an den Anforderungen der Anwendungen orientierenden Verkehrsmanagements.

In der vorliegenden Arbeit werden daher verbindungsorientierte und verbindungslose Mechanismen untersucht und zu einem Verkehrsmanagementkonzept für paketvermittelnde IP-Netze kombiniert, deren Knoten sowohl virtuelle Verbindungen und Verbindungsaggregate als auch die verbindungslose Vermittlung von Datagrammen unterstützen. Im Mittelpunkt des ersten Teils der Arbeit (Kapitel 3 und 4) steht dabei die Realisierung und umfassende theoretische und experimentelle Untersuchung des Verkehrsmanagements eines durchgängig verbindungsorientierten Aggregationsknotens zur Unterstützung von Anwendungen mit strikten Dienstgüteanforderungen. Im zweiten Teil der Arbeit (Kapitel 5 und 6) werden verbindungslose Mechanismen des Verkehrsmanagements am Beispiel elastischer Datenströme untersucht, deren Rate mit Hilfe von TCP an die Auslastung des Netzes angepaßt wird. Nur Algorithmen, die innerhalb des Netzes ohne Verbindungsbezug arbeiten, werden in die Betrachtung einbezogen. Im Vordergrund steht die Weiterentwicklung von systematischen Methoden zur simulativen Leistungsuntersuchung, um den Einfluß von Faktoren wie dem Verkehrsangebot, der Charakteristik der Quellen, die Reaktion der Nutzer und unterschiedlicher Algorithmen auch in großen Systemen besser bewerten zu können.

Durch die Aggregation virtueller Verbindungen für Anwendungen mit strikten Dienstgüteanforderungen und den völligen Verzicht auf Zustandsinformation für elastische Datenströme erfüllt das Verkehrsmanagement die Anforderung der Skalierbarkeit. Eine differenzierte Behandlung von Verkehrsströmen mit Hilfe von *Differentiated Services* wird an dieser Stelle ausdrücklich nicht ausgeschlossen, ist aber nicht Gegenstand der im Rahmen der vorliegenden Arbeit durchgeführten Untersuchungen.

1.2 Übersicht über die Arbeit

Die Entscheidung für bzw. der Verzicht auf Verbindungsorientierung haben einen unmittelbaren und sehr starken Einfluß auf den Umfang und die Qualität der einsetzbaren Funktionen des

Verkehrsmanagements und demzufolge auch auf die zwischen Nutzer und Netz vereinbarte Dienstgüte. Diese Zusammenhänge werden in Kapitel 2 am Beispiel der wichtigsten Verkehrsmanagementarchitekturen für IP-Netze, RSVP/*Integrated Services*, *Differentiated Services* und an der ursprünglich auf Endsysteme beschränkten, nunmehr aber zunehmend die Netzknoten einbeziehenden Überlastregelung elastischer Datenströme diskutiert. Dabei werden Kriterien erarbeitet, die für die Entscheidung für oder gegen Verbindungsorientierung maßgeblich sind.

In Kapitel 3 werden für das Zusammenfassen von Verbindungen zu Aggregaten benötigte Methoden unter dem Gesichtspunkt einer praktischen Implementierung bewertet und ergänzt. Hierzu zählen vor allem Verfahren zur Berechnung des Ressourcenbedarfs und dynamisches Bandbreitemanagement. Mehr als die zwar sehr einfachen, dafür aber die Netzressourcen wenig effizient ausnutzenden, auf dem deterministischen *Network Calculus* basierenden Verfahren zur Berechnung des Ressourcenbedarfs sind die Verfahren von praktischer Bedeutung, die den Ressourcenbedarf unter Ausnutzung statistischen Multiplexens ermitteln. Die Betrachtung konzentriert sich hier auf Verfahren, die sich auf die *Large Deviation* Theorie stützen.

Die Ergebnisse dieses Kapitels sind in die prototypische Realisierung der Verkehrsmanagementfunktionen eines Aggregationsknotens eingeflossen, dessen Architektur in Kapitel 4 beschrieben wird. Er dient als Plattform für die Leistungsuntersuchung. Zunächst wird demonstriert, wie mit Hilfe von Signalisierung der Aufbau der Aggregate und die Bedieneinheiten so gesteuert werden können, daß die Verbindungen am Eingang des Aggregates entsprechend ihrer Anforderungen eine differenzierte Behandlung erfahren. Der Ressourcenbedarf des Aggregates kann mit den in Kapitel 3 untersuchten deterministischen oder statistischen Methoden berechnet werden. Der Vergleich dieser Methoden beschränkt sich nicht auf den errechneten Bedarf an Bandbreite und Wartespeicher, sondern bezieht auch die für die Berechnung notwendige Bearbeitungszeit ein.

Kapitel 5 ist den Verfahren zur Regelung elastischen Verkehrs bei Überlast gewidmet. Da der Großteil elastischer Anwendungen zur Übertragung der Daten und zur Anpassung der Datenrate TCP einsetzt, stehen TCP sowie die auf TCP zugeschnittenen verbindungslos arbeitenden Mechanismen der Netzknoten im Mittelpunkt. Darüber hinaus wird eine Reihe von Leistungsmaßen zur systematischen Untersuchung von TCP vorgestellt und ihre Integration in die im Rahmen der vorliegenden Arbeit entwickelten Simulationsumgebung erläutert.

Die in Kapitel 6 zusammengefaßten Ergebnisse einer Simulationsstudie zeigen, wie sich der Einfluß von Faktoren wie der Aktivität von Quellen und Länge von Verbindungen, möglichen Interaktionen des Nutzers sowie optionale Ergänzungen der Überlastregelung mit Hilfe von TCP und verschiedener Mechanismen innerhalb der Netzknoten in diesen Leistungsmaßen widerspiegelt.

2 Verkehrssteuerung und Überlastregelung in diensteintegrierenden IP-Netzen

In idealen paketvermittelnden Netzen werden die Ressourcen des Netzes nur dann belegt, wenn ein Paket übertragen werden muß. Wenn aber die Ressourcen nicht mehr vorab fest zugewiesen werden und Nutzer unkoordiniert und unkontrolliert auf sie zugreifen, sind Überlastsituationen unvermeidlich. Die Datenströme können sich also in unerwünschter Weise beeinflussen. Es muß daher ein Kompromiß zwischen der Forderung einer optimalen Ausnutzung der Netzressourcen und den Dienstgüteanforderungen der Nutzer gefunden werden. In der Praxis bedeutet dies, daß der Zustrom von Paketen gesteuert oder geregelt werden muß, um Überlast zu vermeiden oder zu begrenzen. Die Gesamtheit der Funktionen, Mechanismen und Protokolle, die diese Aufgabe erfüllen, wird als Verkehrsmanagement bezeichnet [121].

2.1 Verkehrsmanagementfunktionen in paketvermittelnden Netzen

Anwendungen oder Nutzer, die auch in Überlastsituationen keine Einbußen bezüglich der Dienstgüte hinnehmen können oder wollen, vereinbaren mit dem Netz einen Verkehrsvertrag. Dieser umfaßt neben dem Verkehrsdeskriptor, der einem Datenstrom oder einer Gruppe von Datenströmen obere Schranken auferlegt, auch Festlegungen zur Dienstgüte. Das Netz verpflichtet sich, die verabredete Dienstgüte einzuhalten, solange der Datenstrom innerhalb der vereinbarten Schranken bleibt, und teilt mit Hilfe der Verkehrsmanagementfunktionen die Ressourcen des Netzes entsprechend zu. Insofern ist der Verkehrsvertrag das Bindeglied zwischen den Anforderungen des Nutzers und dem Verkehrsmanagement des Netzes.

Wenn zur Vereinbarung der Dienstgüte statistische Kenngrößen verwendet werden, ist statistisches Multiplexen möglich, allerdings in geringerem Maße, als dies bei unkontrolliertem Zufluß der Datenströme der Fall wäre. Wenn wie bei ATM oder der RSVP/*Integrated Services* Architektur das Verkehrsmanagement mit dem Konzept virtueller Verbindungen gekoppelt wird, erhält man dafür aber auch für die Einzelströme sehr gut vorhersagbare Ergebnisse [167].

Dazu müssen alle Knoten im Netz oder eine sie steuernde zentrale Einheit über eine Verbindungsannahmesteuerung (*Connection Admission Control*, CAC) verfügen. Diese entscheidet über die Annahme oder Ablehnung eines Verbindungsaufbau- oder -veränderungswunsches, nachdem sie unter Berücksichtigung des Verkehrsdeskriptors und der Dienstgüteanforderungen den aus der Annahme einer neuen Verbindung oder der Modifikation der Parameter einer bestehenden Verbindungen zusätzlich erwachsenden Bedarf an Ressourcen (meist werden lediglich Wartespeicher und die Übertragungskapazität eines Übertragungsabschnittes betrachtet) berechnet hat.

Die Quellflußsteuerung gewährleistet, daß die Verkehrsdeskriptoren das tatsächliche Verkehrsaufkommen widerspiegeln, indem sie jeden einzelnen Datenstrom, für den ein Verkehrsvertrag vereinbart worden ist, überwacht und gegebenenfalls nicht zum Verkehrsdeskriptor konforme Pakete verwirft oder deren Priorität verändert. Sie wird in der Regel am Eingang eines Netzes oder an Übergängen zwischen Netzen implementiert und bei ATM dementsprechend als *Usage Parameter Control* (UPC) [11, 103] bzw. *Network Parameter Control* (NPC) [103] bezeichnet. Gebräuchlicher, nicht zuletzt auch in den Spezifikationen von RSVP/*Integrated Services* oder *Differentiated Services*, ist jedoch die Bezeichnung *Policing.*

In ATM-, *Integrated Services* und *Differentiated Services* Netzen basieren die Quellflußsteuerung und die Verkehrsdeskriptoren gleichermaßen auf einem Algorithmus, der je nach Kontext unter der Bezeichnung *Leaky Bucket*, *Generic Cell Rate Algorithm* (GCRA) [11] oder *Token Bucket* [177, 176, 198, 24, 106, 91] eingeführt wird. Pro Verbindung sind mehrere *Leaky Buckets* möglich, die dann jeweils die Einhaltung eines einzelnen Parameters, wie z. B. einer Spitzenrate (*Peak Cell Rate*, PCR), einer langzeitigen Rate (*Sustainable Cell Rate*, SCR) [11, 103, 120] oder entsprechende auf die Einheit *Byte* normierte Größen [177] im Rahmen einer ebenfalls vorab vereinbarten Toleranz überwachen.

Ein *Leaky Bucket* ist ein durch zwei Parameter r und b gekennzeichneter Zähler. Bei der Ankunft eines Paketes der Länge L wird vom aktuellen Zählerstand zunächst das Produkt aus der seit der letzten Ankunft eines Paketes verstrichenen Zeit Δt und der für den *Leaky Bucket* zu überwachenden Rate r, also $r \Delta t$, subtrahiert. Dies ist die Datenmenge, die aufgrund des Verkehrsvertrages seit dem letzten Paket hätte ankommen dürfen. Ist das Resultat dieser Subtraktion allerdings kleiner

null, wird der Zählerstand auf null korrigiert, damit eine inaktive Quelle nicht beliebig viel Sendekredit akkumulieren kann. Wenn die Summe des so aktualisierten Zählerstandes und der Länge *L* des Paketes unterhalb der Schranke *b* bleibt, gilt das Paket als konform zum *Leaky Bucket* mit den Parametern *r* und *b* und wird folgerichtig akzeptiert. Die Addition wird durchgeführt. Sollte das Gegenteil zutreffen und die Schranke *b* überschritten werden, ist das Paket nicht konform. Die Quellflußsteuerung wird das Paket dann normalerweise verwerfen oder ihm eine höhere Verlustpriorität zuordnen.

Normalerweise ist es schwierig, den von einer Anwendung generierten Datenstrom so präzise vorherzusagen, daß einerseits Verluste durch die Quellflußsteuerung unwahrscheinlich sind und andererseits die Ressourcen des Netzes effizient genutzt werden. Deshalb ist vor allem in den Endgeräten oder an Netzübergängen Verkehrsformung (*Traffic Shaping*) wirkungsvoll. Dabei werden Pakete jeweils so lange zwischengespeichert, bis sie zum vereinbarten Verkehrsdeskriptor konform sind.

Verluste aufgrund von Quellflußsteuerung und Verzögerungen aufgrund von Verkehrsformung sollten auf jeden Fall der Anwendung signalisiert werden. Adaptiven Anwendungen wird so die Möglichkeit gegeben, die Senderate zu reduzieren, während nicht-adaptive, stromförmige Anwendungen die Reservierung mit dem Netz neu aushandeln könnten.

Alle Knoten, in denen Verbindungsannahmesteuerung, Quellflußsteuerung und Verkehrsformung realisiert werden sollen, müssen Verbindungszustände einrichten und verwalten können. Wird wie bei *Differentiated Services* gänzlich (abgesehen von den Routern am Rande des Netzes) auf Verbindungsbezug verzichtet, so wird die Vereinbarung der Dienstgüte zwangsläufig „unverbindlicher“. Sie könnte z. B. so aussehen, daß das Netz zusichert, einen Datenstrom einer Anwendung oder eines Nutzers innerhalb der im Verkehrsvertrag spezifizierten Schranken mit einer bestimmten Priorität zu übertragen und die Ressourcen des Netzes so zu (über)dimensionieren und gegebenenfalls aufgrund von Verkehrsmessungen anzupassen, daß im statistischem Mittel über eine längere Zeit oder eine größere Anzahl an Verkehrsströmen die gewünschte Dienstgüte erreicht wird.

Außer Prioritäten können Warteschlangendisziplinen, in denen Pakete in der Reihenfolge der ihnen zugewiesenen Zeitstempel bedient werden, die gegenseitige Beeinflussung weiter reduzieren [84, 159, 169, 200]. Diese Mechanismen sind besonders bei Verbindungsbezug sehr wirkungsvoll, denn dann kann sich die Berechnung der Zeitstempel nach den für die Verbindung im Verkehrsvertrag vereinbarten Parametern richten. Am Beispiel der gegen Ende dieses Kapitels noch etwas ausführlicher besprochenen Warteschlangendisziplin *Weighted Earliest Due Date* (WEDD) [25] wird jedoch deutlich, daß auch in Knoten ohne Verbindungsbezug diese oft auch als *Fair Queueing* bezeich-

neten Warteschlangendisziplinen Vorteile gegenüber Verzögerungsprioritäten aufweisen. Über die Zuordnung von Zeitstempeln können die Verkehrsströme weitaus besser gesteuert werden.

Grundsätzlich muß auch die Zuteilung von Wartespeicher in einer auf die Bedienstrategie abgestimmten Weise gesteuert werden. Für diese Aufgabe ist der Begriff Puffermanagement gebräuchlich. Eine formale Definition wird in [121] gegeben. Die in Netzknoten mit Verbindungsbezug verfügbaren Informationen sind auch dafür hilfreich. Bonaventure hat dazu in [27] ein Konzept für die Implementierung von Puffermanagement für die Dienstklasse *Guaranteed Frame Rate* (GFR) in ATM-Vermittlungsknoten vorgestellt, mit dessen Hilfe sich einerseits die durchschnittliche Belegung und andererseits der Anteil einer einzelnen Verbindung sehr präzise steuern läßt, vgl. dazu auch die Ergebnisse in [162]. GFR ist nicht zuletzt auch für Datenströme geeignet, die unter Verwendung von TCP auf Überlast im Netz reagieren.

Verkehrsverträge, wie sie auch bei der Dienstklasse GFR vorgesehen sind, sind dagegen charakteristisch für präventives [167] Verkehrsmanagement. Sie werden vor der Nutzung des Netzes ausgehandelt und sehen die Reservierung von Ressourcen oder die Zuweisung von Prioritäten für ein mit Hilfe der Verkehrsdeskriptoren recht präzise beschriebenes, aber vor allem begrenztes Verkehrsaufkommen vor. Statt eines Verkehrsvertrages verlangen reaktive Verfahren nicht selten ein mehr oder weniger explizit spezifiziertes Verhalten der Quelle, spätestens wenn Überlast auftritt. Bei *Available Bit Rate* (ABR), einer der Dienstklassen von ATM, laufen in den Datenstrom eingestreute *Resource Management* Zellen um, die zumindest in der Variante *Explicit Rate* jeder Verbindung eine der augenblicklichen Verkehrssituation angemessene Rate mitteilen. Die Quelle muß dann entsprechend ihre Senderate anpassen [11]. In verbindungslosen IP-Netzen läßt sich eine vergleichbare Präzision nicht realisieren. Dort kann praktisch nur aufgrund von Paketverlusten oder einer expliziten Anzeige in Paketköpfen auf Überlast [166] geschlossen werden. Erst in den Endsystemen werden diese Signale aus dem Netz auf Verbindungen abgebildet und eine detailliert zu spezifizierende Reaktion der Quelle eingeleitet. Der bekannteste Vertreter dieser Form reaktiven Verkehrsmanagements ist TCP [165, 31, 105, 143, 2, 73]. Erfolgreich kann sie nur sein, wenn sich alle Quellen entsprechend der Spezifikationen an der Überlastregelung beteiligen. Der Verzicht auf Verbindungsbezug erschwert die Überwachung der Einhaltung der Spezifikationen. Zusätzliche Probleme werden daher spätestens dann entstehen, wenn ein größerer Anteil des Internet-Verkehrs als heute TCP nicht verwendet, beispielsweise weil die Wiederholung von fehlerhaft übertragenen oder gänzlich verloren gegangenen Paketen wegen der damit verbundenen Verzögerungen für eine zunehmende Anzahl von Anwendungen nutzlos ist. Zwar könnte man von diesen Anwendungen verlangen, nicht mehr Verkehr zu erzeugen, als dies TCP-Verbindungen unter den gleichen Umständen erlauben würden, d. h. sich „freundlich“ zu TCP zu verhalten (*TCP friendly* [52, 199]).

Das würde aber voraussetzen, daß die Inhalte so kodiert werden können, daß sich praktisch beliebige Raten realisieren lassen und dem Benutzer zuzumuten ist, entsprechende Einbußen bezüglich der Qualität seiner Verbindung hinzunehmen. Das hieße, am *Best Effort* Paradigma festzuhalten.

Verkehrslenkung (*Routing*) kann sowohl in präventivem als auch in reaktivem Verkehrsmanagement eine bedeutende Rolle spielen. Wenn Routing-Protokolle ausreichende Informationen zur Auslastung aller Übertragungsabschnitte verteilen und die Auswahl unter mehreren Routen erlauben, können Netze mit virtuellen Verbindungen gleichmäßiger ausgelastet werden, als dies ohne Unterstützung der Routing-Protokolle möglich ist. In verbindungslos betriebenen Netzen ist eine solche Unterstützung eine sehr viel anspruchsvollere, wenn nicht sogar unlösbare Aufgabe. Denn es sollte vermieden werden, aufeinanderfolgende Datagramme einer Transportverbindung über unterschiedliche Pfade zu routen, da sonst die beim Empfänger eintreffenden Pakete vor der Übergabe an die Anwendung erst noch sortiert werden müssen. Dies ist nur dann möglich, wenn Routen unter Verwendung von virtuellen Verbindungen stabilisiert werden. Auf der Grundlage von *Multiprotocol Label Switching* (MPLS) [171, 39] könnte ein entsprechendes Konzept realisiert werden. Im Unterschied zu ATM und RSVP/*Integrated Services* werden diese für Routen eingerichteten virtuellen Verbindungen nicht von einer oder mehreren Anwendungen, sondern von Routern am Anfang und/oder am Ende eines Pfades gesteuert.

In der Literatur wird außer den bereits angesprochenen Verfahren der Einsatz zahlloser weiterer präventiver und reaktiver Verkehrssteuerungsfunktionen vorgeschlagen, die in aller Regel besondere Ausprägungen und Kombinationen der hier beschriebenen elementaren Funktionen sind.

2.2 Vor- und Nachteile virtueller Verbindungen

Paketvermittelnde Netze routen Pakete entweder einzeln und unabhängig voneinander als Datagramme, oder sie stellen eine Assoziation von Paketen zu zuvor eingerichteten virtuellen Verbindungen her [189]. Häufig wird diese Assoziation mit Hilfe von beim Verbindungsaufbau abschnittsweise zugewiesenen, relativ kurzen Verbindungskennungen im Paketkopf kenntlich gemacht. Die bekanntesten Beispiele sind der *Data Link Connection Identifier* (DLCI) bei *Frame Relay* [126], die Kombination aus *Virtual Channel Identifier* (VCI) und *Virtual Path Identifier* (VPI) bei ATM [121] und das *Label* bei MPLS [171]. Bereits beim Verbindungsaufbau wird der Verbindungspfad festgelegt und durch Tabellen innerhalb der Vermittlungsknoten fixiert, die pro Verbindung Ein- und Ausgangsport einander zuordnen. Obwohl RSVP/*Integrated Services* ohne Verbindungskennungen auskommt und statt dessen Verbindungen anhand der Absender- und Ziela-

dresse in Kombination mit den Portnummern identifizieren können, kommt dieses Konzept virtuellen Verbindungen sehr nahe. Denn sowohl in der Steuerungs- als auch in der Nutzerebene werden Verbindungszustände eingerichtet. Manche Implementierungen routen zwar Pakete zunächst wie Datagramme auf den Ausgangsport und weisen sie erst danach entsprechend der Verbindungstabelle in dessen Warteschlangendisziplin ein, aber auf diese Weise ist nicht sichergestellt, daß die Pakete jederzeit den Pfad nehmen, auf dem auch entsprechend den Anforderungen der Verbindungen Ressourcen reserviert sind.

Zum verbindungslosen Weiterleiten von Datagrammen wird dagegen deren Zieladresse in jedem Router neu ausgewertet und zur Auswahl des nächsten Knotens (*Next Hop)* nach einem passenden Eintrag in der Routing-Tabelle gesucht (*Hop-by-Hop Datagram Routing*). Da der kürzeste Pfad stets auf der Basis der neuesten verfügbaren Informationen über die Topologie des Netzes errechnet und in die Routing-Tabelle eingetragen wird und überdies unter Umständen der Verkehr auf gleichwertige Pfade verteilt werden kann [152], ist keineswegs sicher, daß zwei aufeinanderfolgende, aber unabhängig voneinander geroutete Pakete einer Kommunikationsbeziehung den gleichen Weg nehmen. Andererseits erfordert das Routen von Paketen als voneinander unabhängige Datagramme weit weniger Protokollunterstützung, als für die Steuerung von virtuellen Verbindungen notwendig ist. Beim Ausfall von Knoten oder Übertragungsabschnitten gehen lediglich einige Pakete verloren. Verbindungszustände müssen nicht wiederhergestellt werden.

Selbstverständlich sind Anpassungen an aufgrund von Änderungen der Netztopologie wechselnde Pfade auch bei Verbindungsbezug möglich und oft auch notwendig. Zuvor sollten allerdings auf dem neuen Pfad zunächst die erforderlichen Ressourcen angefordert und bereitgestellt werden. Der dafür zu erbringende Aufwand ist beträchtlich, so daß sicherlich nicht jede Änderung des kürzesten Pfades zu einem Ziel nachvollzogen werden kann, sondern vor allem solche, die aufgrund von Knotenausfällen notwendig werden. Da *Link State* Routing Protokolle [189] die Netzknoten mit vollständigen Informationen über die Topologie des Netzes versorgen, sind Erweiterungen von Routing-Tabellen zur Stabilisierung von Routen durchaus denkbar.

Der Aufwand für die Steuerung von Verbindungen ist deshalb insbesondere dann gerechtfertigt, wenn die Kommunikationsbeziehung zwischen den Endpunkten der Verbindung vergleichsweise lange aufrechterhalten wird, wenn sich die erforderliche Dienstgüte und das Datenaufkommen mit Hilfe eines Verkehrsdeskriptors und QoS-Parametern darstellen lassen oder ihr Datenaufkommen so hoch ist, daß die Vorteile von verbindungsbezogenem Verkehrsmanagement wirklich zum Tragen kommen. Auf einen Großteil der Kommunikationsbeziehungen trifft keine dieser drei Bedingungen zu. Besonders die Nutzung des Internets ist durch vergleichsweise kurze Transaktionen geprägt.

RSVP/*Integrated Services* Router zeigen, daß sich das Konzept der virtuellen Verbindung und das Routen von Datagrammen nicht ausschließen. So liegt es nahe, für Anwendungen mit präzisen Dienstgüteanforderungen zwar virtuelle Verbindungen einzurichten, aber Methoden zur Reduzierung des Aufwandes für die Verbindungssteuerung zu untersuchen. Der Verzicht auf virtuelle Verbindungen zum Transport der verbleibenden Datenströme reduziert den Aufwand hier zwar auf ein Minimum, eine Einschätzung der Wirksamkeit der unterschiedlichen Algorithmen zur Steuerung des Datenflusses muß aber unter einheitlichen und weit realistischeren Bedingungen verifiziert werden, als dies bisher der Fall ist.

2.3 *Integrated Services* und *Resource Reservation Protocol*

Traditionell bietet das Internet einen zwar flexiblen aber einfachen verbindungslosen *Best-Effort* Netzdienst an. Im Falle von Überlastsituationen verwerfen Router Pakete, ohne dabei die ihnen unbekannten Anforderungen der Anwendungen zu berücksichtigen. Aus diesem Grunde hat die IETF weitergehende Architekturen erarbeitet, die abhängig von den Anforderungen der Anwendungen Dienstgüte garantieren (*Integrated Services*) oder zumindest eine differenzierte Behandlung zusichern (*Differentiated Services*).

In der *Integrated Services* Architektur handeln Anwendungen für einen sogenannten *Flow*, einen Datenstrom, einen Verkehrsvertrag mit dem Netz aus, in dem der Datenstrom in Form eines Verkehrsdeskriptors beschrieben und die vom Netz zugesicherte Dienstgüte fixiert wird. Ein Signalisierungsprotokoll, das *Resource Reservation Protocol* (RSVP), unterstützt die Aushandlung dieses Verkehrsvertrages. Wie andere Architekturen zur Realisierung diensteintegrierender Netze, beispielsweise ATM, stellen *Integrated Services* einen Verbindungsbezug im Netz her, um unter Berücksichtigung der Anforderungen dieser Verbindungen Verkehrssteuerungsfunktionen in den Knoten des Netzes so zu konfigurieren, daß Verkehrsströme mit besonderen Anforderungen auch in Überlastsituationen geschützt sind und die für sie vereinbarte Dienstgüte eingehalten wird. Da die *Integrated Services* auf bestehende Routing-Mechanismen aufsetzen, schließen sie im Gegensatz zu ATM die Bereitstellung verbindungsloser Netzdienste nicht aus.

Die Spezifikationen von *Integrated Services* [33, 197, 176, 198, 177] und RSVP [32, 34] sind die Grundlage der folgenden Darstellung der wesentlichen Merkmale dieser Architektur. Einen guten Überblick geben auch [201] und [196].

2.3.1 Modell eines *Integrated Services* unterstützenden Routers

Die Steuerungsebene eines *Integrated Services* unterstützenden Routers wertet Reservierungsanforderungen aus und bildet sie auf dessen Verkehrssteuerungsarchitektur ab. In der hier betrachteten Kombination von RSVP und *Integrated Services* übernimmt ein RSVP-Prozeß diese zentrale Steuerungsaufgabe. Zuvor ist jedoch zu prüfen, ob das zwischen dem Netzteilnehmer und dem Netzbetreiber vereinbarte Teilnehmerprofil die mit der Reservierungsnachricht signalisierte Anforderung abdeckt. Diese Funktion wird in [32] als *Policy Control* bezeichnet. Erst anschließend wird die Verbindungsannahmesteuerung aufgerufen. Wie eingangs erwähnt, entscheidet diese Funktion in der Regel auf der Basis eines Bedienmodells, das sowohl das Verhalten der Verkehrsquellen in den Sendern als auch die Verkehrssteuerungsfunktionen der Nutzerebene des Knotens einbezieht, ob noch genügend Ressourcen für die neue Reservierungsanforderung zur Verfügung stehen. Nur wenn dies der Fall ist, kann ein Pfad durch den Router eingerichtet werden, der den Dienstgüteanforderungen gerecht wird.

Bei Verwendung von RSVP werden am Eingang dieses Pfades in Übereinstimmung mit dem in der Reservierungsnachricht enthaltenen *Filterspec* Paketfilter eingefügt, die entweder alle (*Wildcard Filter*) oder die von den explizit aufgeführten Sendern (*Fixed Filter*, *Shared Explicit*) stammenden und die *Session* adressierenden Datenpakete der Quellflußsteuerung unterwerfen. Nachdem die Paketfilter die Pakete mit Hilfe eines entsprechend dem Verkehrsdeskriptor (*Tspec*) eingestellten *Leaky Bucket* überprüft haben, übergeben sie die Pakete schließlich der für sie vorgesehenen Warteschlange. Diese hochauflösende Form der Klassifizierung, in der selbst im einfachsten Fall (*Wildcard Filter*) Pakete durch die Auswertung von IP-Adressen und Portnummern im Paketkopf klassifiziert werden, wird *Microflow* Klassifizierung genannt. Sie gilt als sehr aufwendig und gehört neben der zu erwartenden großen Anzahl von Verbindungszuständen (*Path/Resv State*) zu den am meisten kritisierten Komponenten des RSVP/*Integrated Services* Verkehrssteuerungskonzeptes, obwohl in ATM-Vermittlungsknoten Mechanismen vergleichbarer Komplexität zum Einsatz kommen und überdies *Integrated Services* andere Formen der Klassifizierung nicht ausschließen.

Im Gegensatz zu *Differentiated Services* unterscheiden die ursprünglichen Spezifikationen von RSVP und *Integrated Services* [33, 197, 176, 198, 177, 32, 34] nicht zwischen Routern am Netzrand und Routern im Kernnetz. Allenfalls die Quellflußsteuerung könnte in Routern des Kernnetzes entfallen. Wie in allen auf Verkehrsdeskriptoren beruhenden Konzepten zur Realisierung eines diensteintegrierenden paketvermittelnden Netzes sind Knoten in einem *Integrated Services* Netz in der Lage, durch Verzögerung der Bedienung oder Verwerfen von Paketen einen zum vereinbarten Verkehrsdeskriptor konformen Ausgangsstrom zu erzeugen. Verkehrsformung, Puffermanagement

und die Warteschlangendisziplin sind in Abb. 2.1 vereinfachend in dem als Bedieneinheit (*Packet Scheduler*) ausgewiesenen Block zusammengefaßt worden.

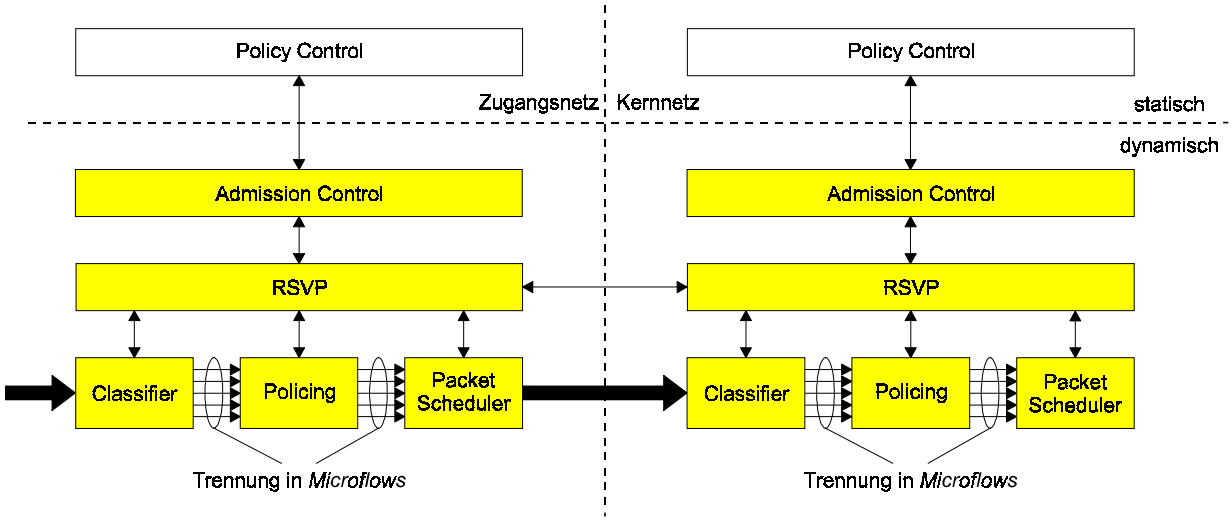

Abb. 2.1: *Modell eines Integrated Services unterstützenden Routers. Die Router im Zugangsnetz und Kernnetz sind identisch. Beide halten Zustände für Microflows.*

2.3.2 *Integrated Services* Dienstklassen

Die *Integrated Services* Architektur schreibt die Verwendung eines Signalisierungsprotokolls zwingend vor. Denn während die *Differentiated Services* Architektur voneinander entkoppelte *Per-Hop Behaviour* realisiert, verknüpft die *Integrated Services* Architektur die Verkehrssteuerungsfunktionen der Einzelknoten so, daß ein den unterschiedlichen Dienstgüteanforderungen gerecht werdender Netzdienst angeboten wird, mit im Verkehrsvertrag vereinbarten, garantierten und damit auch nachprüfbaren Eigenschaften [33].

Die Dienstklasse *Guaranteed Quality of Service* [176] realisiert auf der Grundlage von Ergebnissen des deterministischen *Network Calculus*, insbesondere Gleichung (3.13), die im nachfolgenden Kapitel eingehender besprochen werden, einen Netzdienst, der durch völlige Verlustfreiheit und eine garantierte Obergrenze für die Verzögerungszeit Ende-zu-Ende gekennzeichnet ist. Diese Eigenschaften könnten allenfalls durch den Ausfall von Netzelementen oder die Änderung von Routen kurzzeitig beeinträchtigt werden.

Guaranteed Quality of Service macht die Verwendung eines Signalisierungsprotokolls erforderlich, das nicht nur einen Verkehrsdeskriptor (*Tspec*) transportieren kann, sondern auch von den Netzknoten exportierte Informationen (*Adspec*), mit denen diese implementierungsabhängige Obergrenzen der Abweichungen der maximalen Verzögerungszeit von der aufgrund einer bestimmten

reservierten Rate zu erwartenden mitteilen. Bei RSVP sind dazu in den Pfadnachrichten die Objekte *Tspec* und *Adspec* vorgesehen. Wenn im Zusammenhang mit RSVP von Objekten die Rede ist, so sind damit den Informationselementen in ATM-Signalisierungsprotokollen [154] vergleichbare Nachrichtenteile gemeint. Insofern setzt die Dienstklasse auch voraus, daß die Verkehrssteuerungsmechanismen der Knoten so beschaffen sind, daß sie solche Obergrenzen garantieren können, also einem Verkehrsstrom eine bestimmte Rate mit einer gewissen Toleranz dediziert bereitstellen können. Nur dann können Empfänger in Kenntnis von Gleichung (3.13) mit diesen Informationen die zu reservierende Rate so bestimmen, daß keines der Pakete eines zum deterministischen Verkehrsdeskriptor konformen Verkehrsstromes über die gewünschte Obergrenze hinaus verzögert wird. Mit dem sogenannten *Slack*-Term signalisiert der Empfänger, in welchem Maße er für die Aussicht auf eine höhere Wahrscheinlichkeit der erfolgreichen Verbindungsannahme zu Abstrichen bezüglich der maximalen Verzögerungszeit bereit ist. Die zu reservierende Rate und der *Slack*-Term bilden gemeinsam den *Rspec*.

Falls RSVP zur Signalisierung der Reservierungsanforderungen eingesetzt wird, werden *Tspec*, *Rspec* und die vom Empfänger angeforderte Dienstklasse in einem gemeinsamen Objekt der Reservierungsnachricht, dem *Flowspec*, zum Sender übertragen [197]. Der vom Empfänger in die Reservierungsnachricht eingefügte *Tspec* muß nicht unbedingt mit dem in der Pfadnachricht eines Senders enthaltenen *Tspec* übereinstimmen. Im Falle von Abweichungen muß jedoch einer der Netzknoten im Pfad oder der Sender selbst durch Verkehrsformung einen zum *Tspec* des Empfängers konformen Paketstrom erzeugen.

Deutlich unverbindlicher ist die Dienstgüte, die einer Verbindung der Dienstklasse *Controlled-Load Service* [198] zugesichert wird. Verbindungen dieser Dienstklasse sollen unabhängig von der tatsächlichen Lastsituation im Netz immer etwa die Dienstgüte erfahren, wie sie auch bei geringer Last auftreten würde. Moderate Verluste und gelegentlich über die im günstigsten Fall hinausgehenden Verzögerungszeiten werden ausdrücklich nicht ausgeschlossen. Das Netz garantiert lediglich, bei der Dimensionierung der für die Verbindungen zuzuweisenden Ressourcen den Verkehrsdeskriptor angemessen zu berücksichtigen. Ganz anders als die Dienstklasse *Guaranteed Quality of Service*, die mit dem deterministischen *Network Calculus* einen sehr engen Rahmen für die Wahl eines Verfahrens zur Verbindungsannahmesteuerung bzw. zur Berechnung der Ressourcenbedarfs einer Verbindung setzt, haben Netzelemente zahlreiche Optionen, statistisches Multiplexen in die Berechnung des Ressourcenbedarfs einzubeziehen und so die Ressourcen besser auszunutzen, als dies bei *Guaranteed Quality of Service* möglich ist. Jeder einzelne Knoten im Pfad hat allerdings dafür Sorge zu tragen, daß im Sinne der Dienstgütespezifikation ausreichend Ressourcen bereitstehen. In dieser Hinsicht unterscheiden sich *Integrated Services* und *Differen-*

tiated Services grundlegend.

Controlled Load Service stellt weit geringere Anforderungen an das zum Transport der Reservierungsanforderungen eingesetzte Signalisierungsprotokoll als *Guaranteed Quality of Service*. So sind z. B. die von den Netzknoten exportierten Daten ohne Belang, die über implementierungsabhängige Abweichungen der maximalen Verzögerungszeit von der aufgrund einer bestimmten reservierten Rate zu erwartenden angeben. Der *Adspec* verliert damit einen wesentlichen Teil seiner Funktion. Eine explizite Ratenanforderung in Form des *Rspec* wird ebenfalls nicht benötigt.

2.3.3 Resource Reservation Protocol

RSVP ist ein im Einklang mit den Anforderungen der Dienstklassen *Guaranteed Quality of Service* und *Controlled Load Service* entwickeltes Signalisierungsprotokoll, mit dem Endsysteme Netzressourcen für einen unidirektionalen Datenstrom anfordern und so unmittelbar auf die Dienstgüte einwirken können. Bei RSVP erzeugen die Empfänger die Reservierungsanforderungen. Auf diese Weise kann Dienstgüte nicht nur für *Unicast*, also Kommunikationsbeziehungen Punkt-zu-Punkt, sondern auch für *Multicast*, Punkt-zu-Mehrpunkt, realisiert werden. Dies gilt selbst dann, wenn unterschiedliche Dienstgüteanforderungen für einzelne Zweige des mit Hilfe von *Group Membership* [50, 68] und *Multicast Routing* [145] Protokollen (unabhängig von RSVP) aufgebauten Baumes zu erfüllen sind und das dynamische Zuschalten oder Entfernen von Empfängern zu *Multicast* Kommunikationsbeziehungen unterstützt werden soll. Die Empfänger berücksichtigen die dazu zuvor von den Sendern in den oben erwähnten Objekten *Tspec* und *Adspec* der Pfadnachrichten übermittelten Informationen über die Verkehrscharakteristik des Datenstroms bzw. die Beschaffenheit des Pfades zwischen Sender und Empfänger.

Die Pfadnachrichten erfüllen über den Transport dieser Informationen hinaus einen weiteren Zweck. Da in verbindungslosen IP-basierten Netzen keineswegs davon auszugehen ist, daß Pakete vom Sender zum Empfänger dieselbe Route nehmen wie Pakete in umgekehrter Richtung, ordnen Router bei der Verarbeitung von Pfadnachrichten dem Eingangsport, an dem sie die Pfadnachricht erhalten haben, einen Pfadzustand zu, in dem unter anderem auch die IP-Adresse des Knotens gespeichert wird, der zuletzt die Pfadnachricht verarbeitet und sich in das dafür vorgesehene Objekt eingetragen hat. Die Reservierungsnachrichten werden dann unter Verwendung dieser Adressen von Router zu Router weitergereicht, so daß unter der Voraussetzung einer stabilen Route zwischen Sender und Empfänger die Ressourcen genau entlang des Pfades reserviert werden können, den anschließend die Datenpakete nehmen werden. Der Inhalt der Reservierungsnachricht wird in einem Reservierungszustand gespeichert und logisch mit dem Ausgangsport (bezogen auf die Reservierungsrich-

tung von Sender zu Empfänger) verknüpft, auf dem der Router die Reservierungsnachricht erhalten hat. Außerdem werden in der Nutzerebene für die Verbindung ein Pfad eingerichtet und entsprechend der Reservierungsanforderung Ressourcen reserviert.

Anders als Reservierungsnachrichten werden Pfadnachrichten von den Sendern an die Zieladresse adressiert, denn der Sender kennt den Pfad zum Ziel nicht. Damit die Router im Pfad dennoch den Inhalt von Pfadnachrichten ähnlich wie den von Reservierungsnachrichten auswerten und einen Pfadzustand einrichten, muß daher die *IP Router Alert Option* [112] eingefügt werden.

Es ist durchaus denkbar, daß nicht nur ein, sondern mehrere Sender an einer RSVP *Session* teilnehmen. Alle an einer solchen Mehrpunkt-zu-Mehrpunkt-Kommunikationsbeziehung teilnehmenden Sender tragen jeweils dasselbe *Session* Objekt in die Pfadnachrichten ein, adressieren also dieselbe (*Multicast*-)IP-Adresse und Portnummer. Auf diese Weise werden dem *Multicast* Baum weitere Zweige hinzugefügt. Ob sich die Sender in den gemeinsam genutzten Zweigen des Baumes die reservierten Ressourcen teilen oder ihnen die Ressourcen stets dediziert zugeordnet sind, können die Empfänger durch die Angabe eines *Style* Objektes bestimmen. Eine *Fixed-Filter* Reservierung bezieht sich immer auf einen einzigen Sender oder – präziser formuliert – auf das von diesem in seiner Pfadnachricht angegebene *Sender Template*, das neben dessen IP-Adresse auch die IP-Portnummer angibt, also den *Socket*, den die Anwendung zum Senden der Pakete verwendet. In der Reservierungsnachricht wird dazu jedem *Flowspec* genau ein *Filterspec* zugeordnet. Diese strikte Trennung der Datenströme der Sender wird in der Nutzerebene in der oben beschriebenen Weise durch Paketfilter nachvollzogen und spiegelt sich in der Berechnung der benötigten Ressourcen zur Verbindungsannahmesteuerung wider. Dagegen teilen sich Sender eine Reservierung, wenn einem *Flowspec* mehrere *Filterspec* Objekte zugeordnet werden. Diese Form der Reservierung ist in Abb. 2.2 für zwei Sender S1 und S2 angedeutet. In [32] wird von einer *Shared-Explicit* Reservierung gesprochen. Die *Wildcard-Filter* Reservierung verzichtet auf *Filterspec* Objekte. Dann können alle Sender innerhalb einer RSVP *Session* auf die reservierten Ressourcen zugreifen.

In jedem Falle werden nach Möglichkeit die Reservierungsanforderungen in Verzweigungspunkten des Baumes, in denen die Reservierungsnachrichten mehrerer Empfänger zusammentreffen, zusammengefaßt (*Merging*). Die resultierende Reservierungsanforderung muß natürlich alle Einzelreservierungen der Zweige enthalten. Unter welchen Bedingungen das der Fall ist, hängt nicht zuletzt auch von der Dienstklasse ab [32, 197]. Die Liste der *Filterspec* Objekte geht aber unabhängig von dienstklassenspezifischen Regeln aus der Vereinigung der Listen der Einzelzweige hervor.

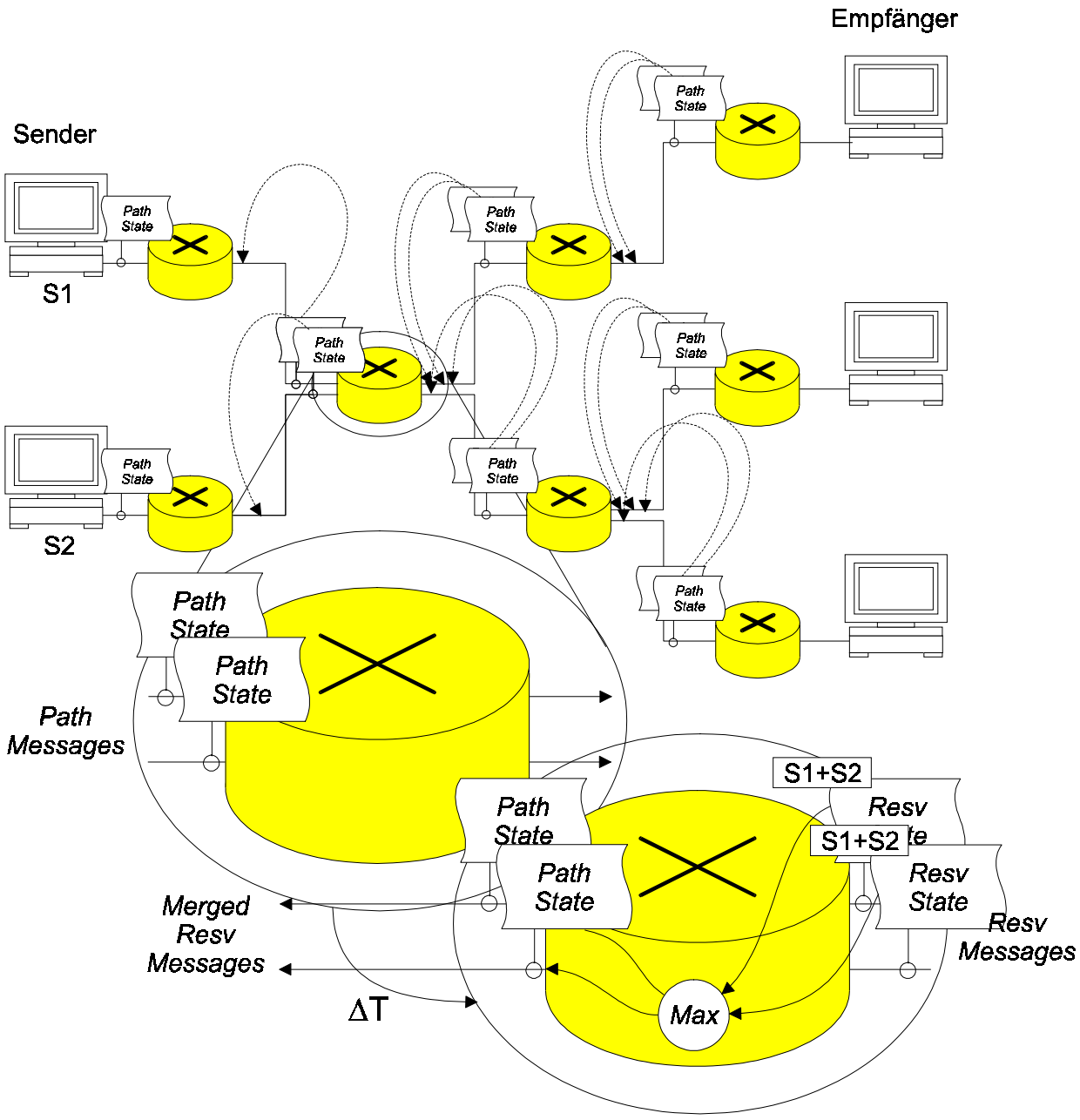

Abb. 2.2: *Das Einrichten einer Reservierung in einer Multicast-Verbindung (RSVP Session). Reservierungen sind grundsätzlich unidirektional. Von den Sendern ausgehende Pfadnachrichten (Path Messages) richten Pfadzustände (Path States) ein, die logisch den Eingängen (aus der Perspektive dieser Sender) der Router zugeordnet werden können. Auf der Basis der in den Pfadnachrichten transportierten Informationen fordern die Empfänger mit Hilfe von Reservierungsnachrichten (Resv Messages) Ressourcen an, gegebenenfalls unter expliziter Angabe der Sender, die diese Ressourcen nutzen dürfen (S1+S2). Die Reservierungsnachrichten folgen dem durch die Pfadzustände vorgegebenen Weg. Unterschiedliche Anforderungen werden an Verzweigungspunkten zusammengeführt (Merging). In jedem Router werden entsprechende Reservierungszustände (Resv State) eingerichtet. Logisch sind sie den Ausgängen der Router zugeordnet.*

Die Einführung von verbindungsbezogenen Pfadzuständen, Reservierungszuständen und die entsprechende Konfiguration der Nutzerebene ist eine Abkehr vom gelegentlich zum Dogma erhobenen Prinzip der Verbindungslosigkeit im Innern von IP-Netzen. Davon betroffen sind aber nur

Kommunikationsbeziehungen, die eine Verbindung mit Dienstgüteattributen anfordern. Der Austausch von Datenpaketen ist auch ohne weiteres ohne oder vor dem Einrichten dieser Zustände und einer erfolgreichen Reservierung von Ressourcen möglich. Die Zustände bleiben auch nur dann erhalten, wenn sie regelmäßig durch weitere Pfad- oder Reservierungsnachrichten aufgefrischt werden. In [32] werden sie deshalb auch als *Soft States* bezeichnet. Dieser Begriff ist sicherlich auch deshalb gerechtfertigt, weil sie Sender und Empfänger durch neue Pfad- bzw. Reservierungsnachrichten zu beliebigen Zeitpunkten unterbrechungslos ihren neuen Anforderungen anpassen können. Anwendungen wird so die Möglichkeit gegeben, die vom Netz realisierte Dienstgüte fortwährend zu bewerten und gegebenenfalls notwendige Anpassungen der Netzressourcen vorzunehmen.

Der Nachrichtenaustausch bis zum Zustandekommen einer Reservierung Ende-zu-Ende folgt also einem in der Literatur [175] als *One Pass With Advertising* (OPWA) bezeichneten Reservierungsmodell. Router im Pfad vom Sender zum Empfänger fügen dem Objekt *Adspec* der Pfadnachrichten jeweils lokale Informationen hinzu, so daß beim Empfänger relativ umfassende Informationen über die prinzipiell auf diesem Pfad mögliche Dienstgüte vorliegen.

Der überwiegende Teil dieser Informationen wird zu den *General Characterization* Parametern [177] gerechnet. Dazu gehören die Anzahl von *Integrated Services* fähigen Routern im Pfad, gegebenenfalls eine Warnung (*Break Bit*), falls nicht alle Router im Pfad *Integrated Services* unterstützen, die maximal für eine Verbindung realisierbare Reservierung, die Paketgröße, ab der einer der Router Datenpakete fragmentieren müßte, und schließlich der Verkehrsdeskriptor *Tspec* (er wird in den Routern nicht verändert und wird deshalb als eigenständiges Objekt und nicht als Teil des *Adspec* behandelt). Neben diesen dienstklassenunabhängigen Parametern des Pfades werden außerdem *Per-Service Characterization* Parameter ermittelt [197]. Im *Guaranteed Service Fragment* des Objektes *Adspec* sind dies vor allem die für die Reservierung unentbehrlichen Informationen über implementierungsabhängige Obergrenzen der Abweichungen der maximalen Verzögerungszeit von der aufgrund einer bestimmten reservierten Rate zu erwartenden (die Summen $\sum C_j^{GS}$ und $\sum D_j^{GS}$ in Gleichung (3.13) des nachfolgenden Kapitels) und darüber hinaus ein dienstklassenspezifisches *Break Bit*. Das *Controlled Load Service Fragment* enthält dagegen lediglich ein solches dienstklassenspezifisches *Break Bit*. Ein Empfänger kann die jeweilige Dienstklasse nur dann anfordern, wenn der Sender die entsprechenden Fragmente in die Pfadnachrichten einträgt.

Aus dem Inhalt des Objektes *Adspec* kann der Empfänger trotzdem nicht zuverlässig schließen, daß seine Reservierungsanforderung angenommen wird. Denn die Router reservieren bei der Verarbeitung der Pfadnachrichten noch keine Ressourcen, nicht einmal vorläufig. Dies ist ein wichtiges Kennzeichen von OPWA.

2.3.4 Konkurrierende Reservierungsprotokolle

Neben Ansätzen für die Erweiterung von RSVP zur Unterstützung von Aggregation [22, 88, 17] haben einige Autoren auch neue Reservierungsprotokolle entworfen, um unter Umgehung einiger Nachteile von RSVP zu effizienteren Lösungen zu gelangen. Folgt man der Argumentation in [157], die durch Messungen untermauert wird, dann ist RSVP hauptsächlich deshalb so komplex, weil es Anwendungen mit unterschiedlichen Dienstgüteanforderungen in ein und derselben Sitzung unterstützt. Würde man von vornherein eine solche Heterogenität ausschließen, dann könnte man unter Umständen auch auf das *One Pass with Advertising* (OPWA) genannte Reservierungmodell von RSVP verzichten. Zwar sammelt das in den Pfadnachrichten mitgeführte Objekt *Adspec* Informationen über die prinzipiell entlang des Pfades realisierbare Dienstgüte [197], insbesondere über die abhängig von der späteren Reservierung realisierbaren oberen Grenzen für die Verzögerung [176], zeigt dem Empfänger jedoch keineswegs an, ob eine bestimmte Reservierung zur Zeit angenommen werden kann oder nicht.

Daher kehren die beiden neuen Ansätze, YESSIR [157] und *Boomerang* [64], die Reservierungsrichtung um. Da in diesem Falle die Reservierungsrichtung und die Datenflußrichtung übereinstimmen, kommen sie ohne Pfadzustand (*Path State*) in den Routern aus. Dies vereinfacht die Protokollprozeduren spürbar. Statt Reservierungsanforderungen werden lediglich Benachrichtigungen in Rückrichtung gesendet, unter Verwendung des Protokolls *Real-Time Control Protocol* (RTCP) [174] bei YESSIR bzw. bei *Boomerang Internet Control Message Protoco*l (ICMP) [164], in das *Boomerang* eingebettet ist. Die Bestätigungen müssen nicht zuletzt deshalb nicht denselben Weg wie die Reservierungsanforderungen nehmen, weil von RSVP das *Soft State* Prinzip übernommen wird: Reservierungen, die nicht rechtzeitig aufgefrischt werden, gehen verloren, auch und insbesondere dann, wenn eine Reservierung nur auf einem Teil des Pfades zustande gekommen ist. Im Gegensatz zu RSVP und YESSIR spezifiziert bei *Boomerang* der Initiator die Reservierungsanforderungen für beide Richtungen. Die Reservierungsnachricht wird vom Empfänger verarbeitet, kehrt danach auf demselben oder einem anderen Weg zum Sender zurück und baut unterwegs die Reservierung in Gegenrichtung auf. Die Effizienz von *Boomerang* und YESSIR kann ebenso wie die von RSVP mit Hilfe von verbindungsorientierter Aggregation weiter gesteigert werden.

Das *Scalable Reservation Protocol* (SRP) [69] versucht die Vorzüge eines auf Reservierung beruhenden Dienstgütekonzepts zu erhalten und trotzdem eine so skalierbare Lösung wie die *Differentiated Services* Architektur zu realisieren. Reservierungen werden dazu nicht mehr auf einen *Flow*, mit welcher Granularität er auch immer definiert sein mag, sondern auf sogenannte *Behaviour Aggregates* bezogen. Wie im Zusammenhang der *Differentiated Services* Architektur näher erläutert

werden wird, sind dies Aggregate, die in dieser Zusammensetzung in der Regel nur innerhalb eines Knotens bestehen und verbindungslos zu unterschiedlichen nächsten Knoten geroutet werden, denen aber aufgrund eines besonderen Feldes im Paketkopfes eine aus einer Menge genau spezifizierter Sonderbehandlungen zuteil wird.

Auch ohne Verbindungsbezug im Netz kommt bei SRP eine Reservierung zustande, wenn Sender ihre Datenpakete als *Request* zu markieren beginnen. Die Knoten messen die Gesamtrate dieser so markierten Pakete ohne Beachtung ihrer Herkunft und routen sie unverändert weiter, sofern die durch die Rate implizit angeforderte zusätzliche Bandbreite verfügbar ist. Sonst löschen sie die Markierung. Da ein Empfänger problemlos den Verbindungsbezug wiederherstellen kann, ist er imstande, aus der Rate der empfangenen *Request* Pakete auf die zusätzlich vom Netz bereitgestellte Bandbreite zu schließen und diese Information dem Sender zu übermitteln. Der darf von nun an seine Pakete bis zu der vom Empfänger gemessenen Rate als *Reserved* markieren. Da die Router und der Empfänger die Rate der Pakete zur Abschätzung des momentan akzeptierten Bandbreitebedarfs messen, die *Reserved* anders als die *Request* Markierung jedoch nicht gelöscht werden, behauptet ein Sender so „seine" Reservierung. Sendet er mehr als seine Reservierung abdeckt, dann muß er die überschüssigen Pakete als *Request* markieren.

Im Unterschied zu den *Integrated Services* Reservierungsprotokollen im engeren Sinne kommt SRP ohne Verkehrsdeskriptoren, Dienstgüteparameter und vom Datenstrom separierte Signalisierung aus. Trotzdem werden faktisch Ressourcen reserviert. Auf der anderen Seite gibt SRP mit dem Verbindungsbezug auch einen Großteil der Verbindungssteuerung auf. Insbesondere die Quellflußsteuerung wird extrem erschwert. Da zudem die Reservierung an den Datenstrom gekoppelt ist, kann es zu Verzögerungen und Instabilitäten beim Aufbau und Aufrechterhalten von Reservierungen kommen [78]. Deshalb sollte SRP vor allem für Anwendungen eingesetzt werden, die sich bis zu einem gewissen Grad an wechselnde Bedingungen im Netz anpassen, und weniger für Anwendungen mit sehr strikten Dienstgüteanforderungen [90].

Die in [187] vorgeschlagene Architektur *Scalable Core* vermeidet zumindest im Kernnetz Verbindungsbezug gänzlich, führt aber so viel Verbindungsinformation in jedem Paket mit, wie sie für eine der Dienstgüte angemessene Behandlung im Netz erforderlich ist. Reservierungsanforderungen werden im Aggregationsbereich Verbindungen oder Verbindungsaggregaten nicht zugeordnet. Dies hat gegenüber verbindungsbezogener Aggregation den Vorteil, daß keine konsistenten Verbindungszustände auf den Routern aufrechterhalten werden müssen. Andererseits wird die Verbindungsannahmesteuerung zu einer komplizierten Schätzaufgabe.

2.4 *Differentiated Services* (DS)

Mehr noch als Ansätze zu protokolltechnischen Verbesserungen von RSVP und Erweiterungen zur verbindungsorientierten Aggregation von Datenströmen haben in den letzten Jahren die Spezifikationen zu *Differentiated Services* (DS) Anklang gefunden. In dieser Architektur wird auf das Einrichten von Verbindungen, das Zuordnen von Paketen zu Verbindungen auf der Basis von Absenderadresse, Zieladresse und Portnummern (*Microflow* Klassifizierung) und die damit verbundenen zum Teil aufwendigen Protokollabläufe zur Steuerung dieser Verbindungen verzichtet. Nur einmal, nämlich in Routern an den Eingängen (*Ingress Router*) zu DS realisierenden Netzen, werden Pakete wie oben beschrieben klassifiziert. Sie werden dabei jedoch nicht Verbindungen, sondern einem *Behaviour Aggregate* (BA) zugeordnet und im dafür vorgesehenen DS-Feld des IP-Paketkopfes [153] entsprechend markiert. Indem sie in ihrem *BA Classifier* lediglich das DS-Feld auswerten, können nachfolgende Router (*Interior Router*) jedem Paket die für das *Behaviour Aggregate* vorgesehene Sonderbehandlung (*Per-Hop Behaviour*) zuteil werden lassen. Diese Zusammenhänge sind in Abb. 2.3 dargestellt. Die Sonderbehandlung ist die einzige Gemeinsamkeit von Paketen in einem *Behaviour Aggregate*, nicht etwa ein gemeinsames Ziel oder wenigstens ein Stück gemeinsamen Weges wie im Falle von Verbindungsaggregaten. Mit der Zuordnung von Paketen zu einem *Behaviour Aggregate* wird folglich ein möglicherweise bis zum *Ingress Router* noch bestehender Verbindungsbezug aufgelöst, etwa wenn wie in Abb. 2.3 RSVP/*Integrated Services* bis zum *Ingress Router* verwendet wird. Ganz ohne Verbindungsbezug kann der *Ingress Router* nicht feststellen, wie viele Ressourcen auf Übertragungsabschnitten irgendwo im Netz zur Verfügung stehen. Verläßliche Voraussagen der Dienstgüte sind dementsprechend unmöglich. Datenströme erfahren aber eine ihren unterschiedlichen Anforderungen gemäße differenzierte Behandlung.

2.4.1 Grundelemente der *Differentiated Services*

Zur Nutzung der Möglichkeiten der differenzierten Behandlung von Verkehr muß zwischen dem Netzbetreiber und Kunden ein Verkehrsvertrag vereinbart werden. Die *Service Level Specification* (SLS) legt den Netzdienst fest, den das Netz einem Datenstrom (*Microflow*), einer Gruppe von Datenströmen oder einem *Behaviour Aggregate* offeriert. Diese Vereinbarung schließt die *Traffic Conditioning Specification* (TCS) ein. Sie ist das Gegenstück des unter anderem von ATM und *Integrated Services* vertrauten Verkehrsdeskriptors. Mit ihrer Hilfe werden die Komponenten des *Traffic Conditioner* konfiguriert, um den Zufluß an Paketen in das vereinbarte *Behaviour Aggregate*

zu steuern und die bevorzugte Behandlung beispielsweise auf eine vorab vereinbarte Rate oder ein vorab vereinbartes Datenaufkommen zu begrenzen.

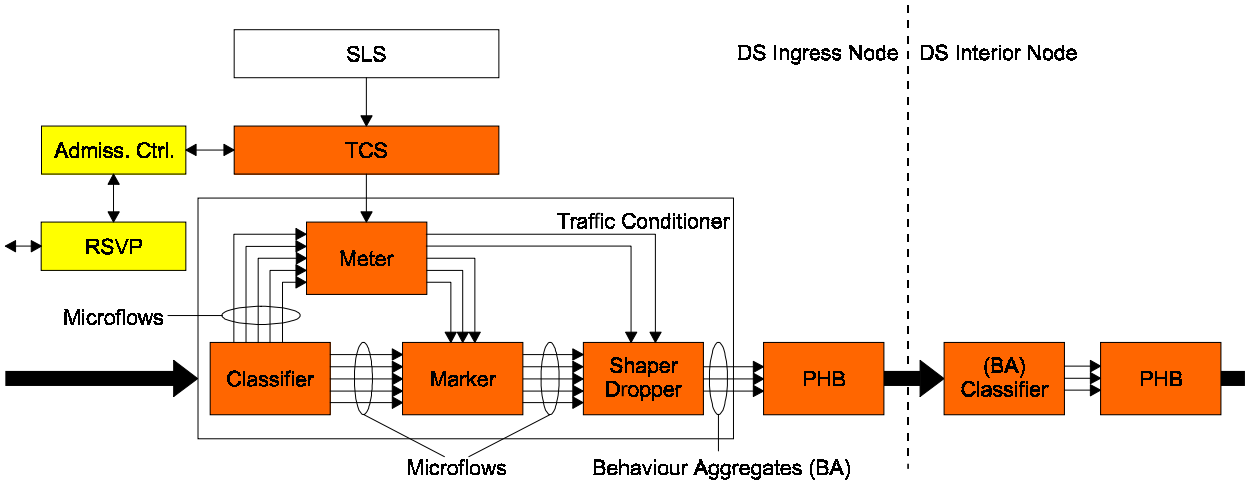

Abb. 2.3: *Differentiated Services (DS) Router. Ein Traffic Conditioner ist nur am Eingang zu DS-Domänen notwendig. Für jeden Microflow sind dessen Parameter zu einer Traffic Conditioning Specification (TCS) zusammengefaßt [87], die entweder dynamisch (z. B. mit RSVP) oder statisch eingerichtet ist. Innerhalb einer DS-Domäne werden statt Microflows nur noch Behaviour Aggregates unterschieden. DS Ingress und DS Interior Node unterscheiden sich also fundamental.*

Wie Abb. 2.3 zeigt, wird die Struktur des *Traffic Conditioner* in [24] noch weiter verfeinert. Zunächst wird der Datenstrom, die Gruppe von Datenströmen oder das *Behaviour Aggregate* im *Classifier* identifiziert. Man sieht unmittelbar, daß dieser *Classifier* ähnlich wie ein *Classifier* in *Integrated Services* realisierenden Routern Verkehrsströme mit variabler Granularität auflösen können muß. Er wird in [24] deshalb auch *Multifield Classifier* genannt. Im *Meter* wird geprüft, ob die tatsächliche Verkehrscharakteristik des Stromes der vereinbarten entspricht. Wenn das nicht der Fall ist, kann er unter Umständen in einen zur vorab vereinbarten SLS konformen Strom umgeformt werden (Verkehrsformung, *Shaping*). Gelingt dies nicht, besteht noch die Möglichkeit, einen Teil der Pakete (*out-of-profile packets*) einem weniger attraktiven *Behaviour Aggregat*e als dem eigentlich vorgesehenen zuzuordnen oder einen Teil der Pakete direkt zu verwerfen.

2.4.2 *Per-Hop Behaviour*

Eine besondere Rolle in der Entstehungsgeschichte der DS-Architektur hat das schließlich unter der Bezeichnung *Expedited Forwarding* standardisierte *Per-Hop Behaviour* (PHB) gespielt [106]. In einem *Expedited Forwarding* Netzknoten soll die maximale Paketankunftsrate eines Aggregates kleiner als die minimale Paketbedienrate sein. Abgesehen von kurzfristiger Belegung durch Paketebeneneffekte [120] bleibt die Warteschlange unter dieser Bedingung weitgehend unbelegt, so daß

Verzögerungen in nur sehr begrenztem Umfang auftreten. Man spricht in diesem Zusammenhang auch von einem *Premium Service*. Da die DS-Architektur die Verknüpfung von PHB in den Netzknoten zu einem Ende-zu-Ende Netzdienst jedoch nicht als Bestandteil der Standardisierung unterstützt, ist die Bezeichnung Dienst eigentlich nicht ganz gerechtfertigt. Möglicherweise soll sie auch lediglich andeuten, daß Pakete dieser Klasse beispielsweise durch Verzögerungsprioritäten gegenüber *Best-Effort* Verkehr bevorzugt bedient werden sollen. Mehr läßt sich mit DS ohnehin nicht realisieren. Um nämlich in allen Netzknoten die maximale Paketankunftsrate eines Aggregates auf die minimale Bedienrate zu beschränken, wäre Verbindungsbezug und Verbindungsannahmesteuerung notwendig, es sei denn man würde bei (ohne Verbindungsbezug) unvorhersehbarer Konzentration von Verkehr der *Expedited Forwarding* Klasse in einem Netzknoten notfalls massive Verluste durch *Policing* hinnehmen.

Verlust- und/oder Verzögerungsprioritäten oder eine auf Klassen bezogene Zuteilung von Wartespeicher und Bandbreite allein reichen nicht aus, um bei variabler Belastung des Netzes dennoch bezogen auf einen bestimmten Nutzer, einen bestimmten Pfad oder gar einen bestimmten Datenstrom eine mit statistischen Methoden quantifizierbare Dienstgüte zu etablieren. Die Dienstqualität kann mit ihnen aber durchaus beeinflußt werden. Netzbetreiber und Kunde können beispielsweise eine Vereinbarung (SLS und TCS) dergestalt treffen, daß der Netzbetreiber zusichert, den Verkehr seines Kunden bis zu einer im Kundenprofil festgeschriebenen Rate mit erhöhter Priorität zu transportieren. Von der höheren Priorität verspricht sich der Kunde, daß seine Erwartungen an die Dienstgüte mit hoher Wahrscheinlichkeit erfüllt werden. Dazu kann in Netzknoten die *Assured Forwarding* PHB Gruppe [91] realisiert sein.

Der relativ strikten Terminologie in [87] folgend, wird *Assured Forwarding* als PHB Gruppe eingeführt. Eine PHB Gruppe besteht der Definition nach aus einer Menge von PHB, die sinnvoll nur gemeinsam spezifiziert und implementiert werden können, z. B. weil sie zu einem genau definierten System aus Warteschlangen angeordnet werden sollen.

In der *Assured Forwarding* PHB Gruppe sind vier Klassen und innerhalb dieser vier Klassen noch einmal jeweils drei Verlustprioritäten vorgesehen. Jeder der Klassen werden Wartespeicher und Bedienkapazität zugeordnet. Diese Ressourcen sollten nach Möglichkeit so dimensioniert sein, daß die einzelnen Klassen jeweils nicht schon mit Paketen der niedrigsten Verlustpriorität (geringste Verlustwahrscheinlichkeit) ausgelastet sind und statt dessen die Ressourcen nur gemeinsam mit Paketen der höheren Verlustprioritäten ausschöpfen. Darüber hinaus wird vorgeschrieben, in länger andauernden Überlastsituationen durch Puffermanagement eine für alle Verkehrsströmen gleiche Verlustquote sicherzustellen. Diese Verluste sollen auch nicht wie in gewöhnlichen Warteschlangen erst bei Pufferüberlauf, sondern schon früher einsetzen, wenn sich abzeichnet, daß die augenblick-

liche Lastsituation mit hoher Wahrscheinlichkeit zu Pufferüberlauf führen wird. In [91] wird wie in der Literatur üblich in diesem Zusammenhang von aktivem Puffermanagement (*Active Queue Management*, AQM) gesprochen.

Die Autoren der *Assured Forwarding* PHB Gruppe gehen davon aus, daß der Nutzer sich für eine der vier Klassen entscheidet und die Verlustprioritäten aus den Werten der Variablen eines dem Verkehrsvertrag entsprechend konfigurierten *Leaky Bucket* am Netzeingang bestimmt werden. Die niedrigste Verlustpriorität könnte dann beispielsweise dann zugewiesen werden, wenn der *Bucket* null ist, die mittlere wenn er größer als null ist, aber noch unterhalb der zulässigen Obergrenze.

Bis heute sind nur das *Expedited Forwarding* PHB und die *Assured Forwarding* PHB Gruppe von der DS-Arbeitsgruppe der IETF als Standards vorgeschlagen worden. Natürlich sind noch zahlreiche weitere Methoden zur differenzierten Behandlung von Verkehr mit Verlust- und Verzögerungsprioritäten möglich.

Mit der Bedienstrategie *Weighted Earliest Due Date* (WEDD) [25] kann eine sehr präzise einstellbare relative Differenzierung zwischen Verkehrsklassen erreicht werden. Im Normalbetrieb ist WEDD identisch zur Bedienstrategie EDD (*Earliest Due Date*)[1]. WEDD berechnet also für Pakete bei ihrer Ankunft einen Übertragungstermin als Summe aus der Ankunftszeit und der für die Verbindung oder Verkehrsklasse festgelegten maximalen Verzögerungszeit und ordnet ihnen diesen in Form eines Zeitstempels zu. Pakete werden dann in der Reihenfolge ihrer Zeitstempel bedient. Wenn sich eine Überlastsituation abzeichnet und nicht alle Pakete rechtzeitig bedient werden können, wird statt des Zeitstempels den Paketen ein sogenannter *Congestion Tag* zugeordnet als Quotient aus der für die Klasse relativ zu anderen Klassen vereinbarten und der gemessenen momentanen Wahrscheinlichkeit, daß die vereinbarten Bedientermine nicht eingehalten werden. Solange die Überlast anhält, werden die Pakete nicht mehr in der Reihenfolge ihres Zeitstempels, sondern in der Reihenfolge ihres *Congestion Tag* bedient. WEDD ist insofern eine Erweiterung von EDD, als nicht mehr nur die Paketverzögerungsobergrenzen, sondern auch die Wahrscheinlichkeiten, daß diese nicht eingehalten werden können, relativ zu anderen Klassen eingestellt werden können. Damit ist ein auf relativer Differenzierung basierender Netzdienst möglich, der dem Nutzer die Auswahl einer Verkehrsklasse erleichtern könnte.

SIMA (*Simple Integrated Media Access*) [111] geht einen etwas anderen Weg. Zwar ist auch hier eine Trennung in zwei Verkehrsklassen vorgesehen, um verzögerungskritischen Verkehr vom restlichen Verkehr zu schützen, der Nutzer kann allerdings durch die Angabe einer nominalen Bitrate die Behandlung seines Verkehrs innerhalb einer Klasse in einer Verkehrsdeskriptoren ähnelnden Weise

1 Neben EDD ist für diese Bedienstrategie auch die Bezeichnung EDF (*Earliest Deadline First*) gebräuchlich.

beeinflussen. Aus dem Verhältnis von der gemessenen zur nominalen Bitrate berechnen Router am Rande eines Kernnetzes dynamisch eine Verlustpriorität. Je kleiner dieses Verhältnis ist, desto niedriger ist die Wahrscheinlichkeit für Paketverluste.

ABE (*Alternative Best Effort*) [99] bietet einer Anwendung nicht solch weitreichende Eingriffsmöglichkeiten. Indem sie ihre Pakete „einfärbt“, kann sie aber Netzelementen immerhin signalisieren, ob sie in Überlastsituationen eher an geringen Verzögerungszeiten – dafür steht die Farbe Grün – oder an hohem Durchsatz – symbolisiert durch die Farbe Blau – interessiert ist. Der Dienst wird mit Hilfe der neuen Puffermanagement- und Bedienstrategie DSD (*Duplicate Scheduling with Deadlines*) so ausgestaltet, daß blaue Pakete nicht unter grünen leiden. Dies ist natürlich nur dann möglich, wenn auf der anderen Seite die Entscheidung für Grün und dementsprechend geringe Verzögerungszeiten mit einer höheren Verlustwahrscheinlichkeit einhergeht. Welche der Farben vorteilhafter ist, hängt davon ab, ob die Anwendung aus niedrigeren Verzögerungszeiten oder höherem Durchsatz größeren Nutzen zieht. Abhängig von der Anwendung, der aktuellen Verzögerung oder dem aktuellen Durchsatz kann diese Entscheidung ganz unterschiedlich ausfallen. Insofern kommt ABE sogar ohne Quellflußsteuerung und damit wirklich gänzlich ohne Verbindungsbezug aus. DSD arbeitet ähnlich wie EDD mit Zeitstempeln, diese werden jedoch abhängig von der Farbe der Pakete auf unterschiedliche Art und Weise berechnet. Für grüne ist die vorab festzulegende maximale Verzögerungszeit maßgeblich. Für blaue Pakete berechnet DSD dagegen den Zeitpunkt, zu dem in einem FIFO-Bediensystem ihre Bedienung abgeschlossen wäre. Grüne Pakete werden nur dann in den Wartespeicher aufgenommen, wenn nicht mehr Pakete mit kleineren Zeitstempeln auf ihre Bedienung warten, als bis zum Ablauf der maximalen Verzögerungszeit bedient werden können. Die Bedienreihenfolge folgt nicht in jedem Fall der Reihenfolge der Zeitstempel. Statt dessen werden grüne und blaue Pakete getrennten logischen Warteschlangen zugeordnet. Wenn unabhängig von der Auswahl des nächsten Paketes sowohl das erste grüne und das erste blaue Paket noch rechtzeitig bedient werden können, wird mit der Wahrscheinlichkeit g das grüne zuerst genommen. Mit Hilfe von g läßt sich das Verhältnis der Bedienraten von grünen und blauen Paketen auf einen Sollwert regeln.

2.5 Überlastregelung elastischer Verkehrsströme bei verbindungsloser Paketvermittlung

Der Verzicht auf Verbindungsbezug in den Knoten im Innern des Netzes schränkt die Möglichkeiten, mit Verkehrsmanagement auf die Dienstgüte Einfluß einzunehmen, doch sehr deutlich ein.

Nur in den Endsystemen sind noch verbindungsorientierte Mechanismen möglich. In den Netzknoten kommen dagegen nur noch Mechanismen in Betracht, die auf Paketströme einwirken können, ohne daß sie die Pakete mit Verbindungen assoziieren können. Hier sind insbesondere die im Zusammenhang mit DS erwähnten Verfahren aktiven Puffermanagements, wie z. B. *Random Early Detection* (RED) [71, 76, 66] oder *Random Early Marking* (REM) [96, 15], zu nennen, die Pakete kontrolliert zur Anzeige von Überlast markieren oder verwerfen, bevor der Wartespeicher überläuft. Diese Mechanismen funktionieren aber auch nur unter der Voraussetzung, daß die Endsysteme kooperativ am Verkehrsmanagement mitwirken. Sie sind nämlich die einzige Instanz, welche die Verbindung zwischen den Rückmeldungen des Netzes und den Verbindungen wieder herstellen können.

Da die meisten elastischen Anwendungen in IP-Netzen TCP verwenden, bestimmt dessen Überlastregelung die Reaktion auf die Überlast. Hier sind natürlich eine Reihe verschiedener Algorithmen denkbar. TCP *Reno* und davon abgeleitete Varianten reagieren auf das Ausbleiben von Quittierungen [2, 73, 143] bzw. auf Überlastanzeigen in Quittierungen [166] mit einer Halbierung des Sendefensters und als Folge davon der Senderate. Dagegen wird beim Eintreffen neuer Quittierungen das Sendefenster erhöht, und zwar abhängig vom Zustand (*Slow Start* oder *Congestion Avoidance*), in der sich die Überlastregelung befindet, exponentiell oder linear (unter der Voraussetzung einer konstanten Umlaufzeit vom Absenden des Paketes bis zu dessen Quittierung).

TCP *Reno* und seine Varianten *New Reno* und SACK reagieren also erst, wenn in Folge von Überlast Pakete verloren oder markiert werden. Aktives Puffermanagement, beispielsweise RED und REM, können allerdings eine vorzeitige Reaktion von Sendern auslösen.

TCP *Vegas* [35], eine weitere Variante von TCP, setzt andere Algorithmen zur Wiederholung von Paketen und zur Anpassung des Sendefensters im Zustand *Slow Start* ein. Um auch ohne aktives Puffermanagement schon vor dem Auftreten von Verlusten die Senderate an die Lastsituation im Netz anpassen zu können, vergleichen die Sender vor allem im Zustand *Congestion Avoidance* einmal pro Umlaufzeit den erzielten Durchsatz mit dem Durchsatz, mit dem aufgrund der Fenstergröße und der kleinsten gemessenen Umlaufzeit (*Basic RTT*) ohne Überlast im Netz gerechnet werden kann. Wenn die Differenz klein ist, vergrößern die Sender ihre Fenster linear, wenn sie dagegen groß ist, reduzieren sie ihr Sendefenster linear. Neben der Koexistenz mit TCP *Reno* ist vor allem die zuverlässige Messung der *Basic RTT* problematisch [149]. TCP *Vegas* hat wohl deshalb nie praktische Bedeutung erlangt.

Ohne Maßnahmen zur Überlastabwehr besteht die Gefahr, daß überlastete Netze und Netzelemente einen Zustand erreichen, in dem bei weiter wachsendem Verkehrsangebot der Nutzdurchsatz nicht

nur nicht mehr zunimmt, sondern sogar abnimmt. Die vorliegende Arbeit beschäftigt sich mit der grundlegenden Frage, ob die und welche der im Zusammenhang mit TCP diskutierten Mechanismen diesen Überlastsituationen am effektivsten entgegenwirken. Natürlich muß aber auch der Einsatz der Mechanismen, die in den Endsystemen implementiert werden müssen, erzwungen werden können. Die in [74] vorgeschlagenen Verfahren setzen voraus, daß Router Pakete aufgrund der Felder des IP-Paketes Verbindungen zuordnen (können) und mit Hilfe unterschiedlicher makroskopischer, d. h. nicht den Zustandsautomaten von TCP nachbildenden Modelle, Verbindungen isolieren und bestrafen, die ihre Datenrate bei Überlast nicht wie vereinbart reduzieren. Unter der Annahme, daß sich nur wenige Verbindungen nicht konform zu den vereinbarten Mechanismen der Überlastregelung verhalten, kann – das zeigt *Stochastic Fair Blue* [67] – die Menge der notwendigen Zustandsinformation reduziert werden.

2.6 Kombinierte Ansätze

Zweifellos ist eine Differenzierung von Verkehr sinnvoll, wenn auf diese Weise elastischer Datenverkehr, also Verkehr, dessen Rate beispielsweise durch die Überlastregelung von TCP an die Auslastung des Netzes angepaßt werden kann, von stromförmigem Verkehr bei Telefongesprächen oder Videokonferenzen, für den dies aufgrund seiner zumeist rigiden Anforderungen bezüglich der Ende-zu-Ende-Verzögerung nur bedingt (etwa durch eine Veränderung der Kodierung, also einer Reduzierung des Verkehrsangebotes) gilt, getrennt wird. Selbst in diesem Fall bleibt jedoch die Dienstgüte unspezifiziert. Aus diesem Grunde wird eine weitergehende Differenzierung ohne verbindliche Zusagen, die DS ohne Verbindungsbezug (zumindest in der Steuerungsebene) nun einmal nicht einhalten kann, kaum überzeugen können.

Der völlige Verzicht auf Verbindungsbezug verbaut eine Reihe von Möglichkeiten zur Verkehrslenkung. Auf der anderen Seite ist die Kritik an der ohne Aggregation großen Anzahl von Verbindungen und dem damit verbundenen hohen Steuerungsaufwand, der aus Verbindungsbezug resultierenden komplexeren Protokollarchitektur im allgemeinen und von RSVP im besonderen durchaus gerechtfertigt. Eine völlige Abkehr von verbindungsorientierten Konzepten, ohne zuvor das Optimierungspotential von RSVP/*Integrated Services* durch protokolltechnische Verbesserungen [157, 64] und mehr noch durch Aufgreifen der Vorschläge zur Aggregation von Verbindungen [22, 88, 17] auszuschöpfen, erscheint dennoch vorschnell.

Die Unterstützung von Prioritäten durch Router allein reicht nicht aus, um auch einen entsprechenden Netzdienst Ende zu Ende anbieten zu können. Gerade die *Differentiated Services* Archi-

tektur zeigt dies nachdrücklich. Nicht anders als bei *Integrated Services* auch, wird die Nutzung des Netzes auf der Grundlage von Verkehrsverträgen (SLS) gesteuert. Im Rahmen dieser Verkehrsverträge müssen auch Verkehrsdeskriptoren festgelegt werden. Wenn ein Netzbetreiber Netzanwendungen und deren Nutzern die Option anbieten möchte, die Dienstgüte ihren individuellen Ansprüchen gemäß zu gestalten und dynamisch anzupassen, dann muß an der Schnittstelle zwischen Nutzer und Netz der Austausch von Signalisierungsnachrichten möglich sein. Die Schnittstelle zwischen der Anwendung oder deren Nutzer vereinfacht sich durch *Differentiated Services* dann aber nicht signifikant. Ob zwischen Netzelementen jenseits dieser Schnittstelle dann noch signalisiert wird oder nicht, ist für den Nutzer von untergeordneter Bedeutung. Entscheidend für ihn ist die Dienstgüte, die das Netz für seine Anwendung Ende-zu-Ende realisiert. Die ganze Vielfalt der Mechanismen des Verkehrsmanagements, das die Dienstgüte durch das Netz absichern soll, steht jedoch nur bei Verwendung von Signalisierung und Verbindungsbezug offen.

So überrascht es eigentlich nicht, daß zur Ergänzung der *Differentiated Services* Architektur der Einsatz sogenannter *Bandwidth Broker* erwogen wird. Das sind Rechner, die jeweils für einen bestimmten Bereich des Netzes verantwortlich sind, aktuelle Informationen über die dort verfügbaren Ressourcen einholen und zentral zuteilen. Abhängig davon, inwieweit die *Bandwidth Broker* zu einer bereichsübergreifenden Infrastruktur zur Bereitstellung von Ressourcen integriert und entweder mittelbar oder unmittelbar mit Anwendungen und Netzelementen kooperieren, variiert die Verbindungsorientierung, also auch die Präzision, mit der Ressourcen bereitgestellt werden [190]. Am Ende könnten daher in der um *Bandwidth Broker* ergänzten *Differentiated Services* Architektur ganz ähnliche Mechanismen, Protokolle und Algorithmen zur Bereitstellung von Ressourcen zum Einsatz kommen wie in einer um durchgängig verbindungsorientierte Aggregation erweiterten *Integrated Services* Architektur: Signalisierung zur Steuerung von Verbindungsaggregaten, dienstklassenabhängige Berechnung des Ressourcenbedarfs und passende Bedienstrategien sowie dynamisches Bandbreitemanagement. Die Implementierung zentraler *Bandwidth Broker* hat also vor allem protokolltechnische und weniger verkehrstheoretische Implikationen. Mögliche Vorteile dieser zentralisierten Ansätze konnten nach Ansicht des Autors bislang nicht überzeugend dargestellt werden. Es besteht daher überhaupt kein Anlaß, von dem bewährten Konzept einer dezentralen, verteilten Verbindungsannahmesteuerung abzurücken.

Die in der Literatur vorgeschlagenen Erweiterungen von RSVP [22, 88, 17] ebenso wie die ATM-Dienstklasse *Differentiated* UBR [13] sehen zwar alle einen Verbindungsbezug in der Steuerungsebene vor, ziehen aber zur Reduzierung der Komplexität die Auswertung des DS-Feldes der Herstellung des Verbindungsbezugs durch Auswertung praktisch des gesamten Paketkopfes vor. Der Verbindungsbezug bleibt also auf die Steuerungsebene begrenzt. Das Für und Wider dieser Kombi-

nationen aus Konzepten der *Integrated Services* und der *Differentiated Services* Architektur wird in Kapitel 4 diskutiert. Gegenüber einer Architektur mit durchgängiger verbindungsorientierter Aggregation ändert sich vor allem die Klassifizierung von Paketen.

In der vorliegenden Arbeit werden geeignete Mechanismen und Algorithmen des Verkehrsmanagements eines durchgängig verbindungsorientierten Aggregationsknotens ausgewählt, unter dem Gesichtspunkt einer vollständigen prototypischen Realisierung verfeinert und durchgängig analytisch, simulativ und experimentell evaluiert. Verbindungslose Mechanismen des Verkehrsmanagements werden am Beispiel elastischer Datenströme untersucht, deren Rate mit Hilfe von TCP an die Auslastung des Netzes angepaßt wird. Nur Algorithmen, die völlig ohne Verbindungsbezug arbeiten, werden in die Betrachtung einbezogen. Denn wie oben erläutert, erfüllen diese Datenströme die Voraussetzungen, die Verbindungsbezug rechtfertigen würden, nicht. Aufgrund der komplexen Einwirkungen und Zusammenhänge der zahlreichen relevanten Parameter, die nur zum Teil und noch dazu unter sehr idealisierenden Annahmen analytisch erfaßt werden können [156], und der großen Anzahl an Algorithmen des Verkehrsmanagements wird für diese Datenströme auf eine analytische Leistungsuntersuchung verzichtet. Vielmehr liegt der Schwerpunkt auf der Weiterentwicklung von systematischen Methoden zur simulativen Leistungsuntersuchung, um den Einfluß von Faktoren wie dem Verkehrsangebot, der Charakteristik der Quellen, der Reaktion der Nutzer und unterschiedlicher Algorithmen auch in großen Systemen besser isolieren sowie qualitativ und quantitativ bewerten zu können. Diese Untersuchungen verfolgen das Ziel, Schlußfolgerungen für die Gestaltung des Verkehrsmanagements für elastische Datenströme ziehen zu können.

Auf diese Weise entsteht das Bild eines paketvermittelnden Netzes, dessen Knoten sowohl virtuelle Verbindungen und Verbindungsaggregate unter Einhaltung einer vorab vereinbarten Dienstgüte als auch Datagramme mit Unterstützung der jeweils geeigneten Verkehrssteuerungsmechanismen vermitteln. Wie eingangs erwähnt, erfüllt das Verkehrsmanagement dieses Netzes aufgrund der Aggregation virtueller Verbindungen und den völligen Verzicht auf Zustandsinformation für elastische Datenströme die Anforderung der Skalierbarkeit. Eine differenzierte Behandlung von Datagrammen mit Hilfe von *Differentiated Services* ist zwar möglich, ist aber nicht Gegenstand der im Rahmen der vorliegenden Arbeit durchgeführten Untersuchungen.

3 Berechnung des Ressourcenbedarfs von Verbindungsaggregaten

Die Aggregation von Verkehrsströmen kann den für die Verbindungssteuerung notwendigen Aufwand reduzieren, zum einen weil die Netzknoten im Aggregationsbereich weniger Verbindungszustände verwalten müssen, zum anderen weil zur Steuerung dieser Zustände durch den Bündelungsgewinn schon bei vertretbarer Überreservierung von Bandbreite weniger Signalisierungsnachrichten ausgetauscht werden müssen.

Um zu gewährleisten, daß die Aggregation die Dienstgüte der Verbindungen im Aggregat nicht oder nur minimal beeinträchtigt, muß die Zuordnung von Verbindungen zu einem Verbindungsaggregat, die Abbildung von Paketen auf eine Verbindung (Klassifizierung), die Berechnung des Ressourcenbedarfs des Aggregates, die Bedieneinheit am Eingang des Aggregates und dynamisches Bandbreitemanagement zu einem konsistenten Verkehrssteuerungskonzept kombiniert werden, das noch dazu der unter Umständen sehr stark variierenden Größe und Zusammensetzung von Aggregaten gewachsen sein muß. Dabei sind eine Reihe von Abhängigkeiten zu beachten. Beispielsweise beeinflußt die Spezifikation des Netzdienstes die Wahl geeigneter Bedieneinheiten, beide gemeinsam die Algorithmen zur Berechnung des Ressourcenbedarfs und diese wiederum die Größe (ausgedrückt in effektiver Bandbreite) der Verbindungen im Aggregat, also letztlich auch das dynamische Bandbreitemanagement.

Als Ressourcen des Aggregationsknotens werden nicht nur Wartespeicher und Übertragungskapazität betrachtet, sondern auch die Rate, mit der ein Aggregationsknoten und nachfolgende Knoten im Aggregationsbereich Signalisierungsnachrichten zur Anpassung der Rate des Aggregates verarbeiten können.

Entsprechend gliedert sich die Berechnung des Ressourcenbedarfs in zwei Phasen. In der ersten Phase wird die notwendige Menge an Bandbreite und Wartespeicher berechnet, die tatsächlich benötigten Ressourcen der Nutzerebene. Hier ist das Konzept der effektiven Bandbreite von großer

Bedeutung. Die effektive Bandbreite kann als ein Maß für den Einzelbeitrag einer Quelle zum Gesamtressourcenbedarf verstanden werden. Sie darf aber nicht als absolute Größe mißverstanden werden, denn sie hängt vom zur Berechnung des Ressourcenbedarfs beim (statistischen oder deterministischen) Multiplexen verwendeten verkehrstheoretischen Modell und von Wechselwirkungen (wenn diese im Modell berücksichtigt werden) zwischen den Verkehrsströmen ab.

In der zweiten Phase wird unter Abwägung der Kosten für die Reservierung zusätzlicher Bandbreite auf der einen und den Austausch von Signalisierungsnachrichten auf der anderen Seite die letztlich für das Aggregat zu reservierende Bandbreite bestimmt.

Im folgenden werden sowohl deterministische als auch statistisches Multiplexen ausnutzende Verfahren zur Berechnung des Ressourcenbedarfs von Aggregaten behandelt. Darüber hinaus werden die Möglichkeiten und Grenzen dynamischen Bandbreitemanagements aufgezeigt.

Die Bewertung von Verfahren zur Berechnung des Ressourcenbedarfs von Verbindungsaggregaten, die Untersuchung ihres Verhaltens und des Potentials dynamischen Bandbreitemanagements kann auf zahlreiche theoretische Vorarbeiten zurückgreifen. Vor dem Hintergrund der im nächsten Kapitel vorgestellten praktischen Realisierung eines Aggregationsknotens bedürfen diese Vorarbeiten jedoch einiger Ergänzungen, aber vor allem einer Untersuchung und Neubewertung unter Berücksichtigung nicht nur theoretischer Kriterien.

Unter diesem Gesichtspunkt ist auch die wichtige Rolle zu verstehen, welche in diesem Abschnitt ebenso wie bei ATM, IP *Integrated Services* und *Differentiated Services* Verkehrsdeskriptoren spielen, die den von einer Quelle ausgehenden Verkehr in Form deterministischer oberer Schranken spezifizieren. Unter Inkaufnahme einer mehr oder weniger niedrigen Paketverlustwahrscheinlichkeit kann grundsätzlich jeder realistische Datenstrom durch Verkehrsformung in einen zu solchen deterministischen Schranken konformen Verkehr umgewandelt werden. Man muß jedoch beachten, daß für nur als langzeitkorrelierte Ankunftsprozesse modellierbare Datenströme, die in paketvermittelnden Netzen auftreten können, dazu aber unverhältnismäßig viele Ressourcen notwendig sind, vgl. dazu die Diskussion im Zusammenhang mit den Gleichungen (3.65) und (3.66). Als Alternative zu deterministischen Verkehrsdeskriptoren kommen die in Netzen nur schwer implementierbaren Modelle langzeitkorrelierter Prozesse allerdings kaum in Frage.

Meßbasierte Verfahren zur Berechnung des tatsächlichen Ressourcenbedarfs von Verbindungsaggregaten in Knoten des Netzes bleiben praktisch unberücksichtigt, da der Autor Verfahren favorisiert, die es Applikationen erlauben, ihren Verkehr durch die Übergabe von Parametern zu beschreiben, die von ihnen wahrgenommene Dienstgüte zu analysieren und gegebenenfalls auf diese

unmittelbar durch die explizite Anforderung weiterer oder Freigabe im Übermaß reservierter Ressourcen einzuwirken.

3.1 Deterministische Verfahren

Paketvermittelnde diensteintegrierende Netze können die von leitungsvermittelnden Netzen gewohnte Dienstgüte nur mit Hilfe einer Verkehrssteuerungsarchitektur erbringen, die Mechanismen wie Verbindungsannahmesteuerung, Quellenflußsteuerung und Verlust- oder Verzögerungsprioritäten zu einem funktionierendem Ganzen kombiniert. Grundlage ist der Verkehrsvertrag, der eine Dienstklasse, einen Verkehrsdeskriptor, Dienstgüteparameter, eventuell Toleranzen für diese u. ä. festlegt [103]. Die Quellflußsteuerung (*Usage/Network Parameter Control*) stellt sicher, daß keine Verbindung mehr Verkehr erzeugt, als im Verkehrsdeskriptor zugesichert worden ist. Mit Verkehrsdeskriptoren, die als deterministische obere Schranken für den von einer Quelle in einem Intervall erzeugten Verkehr zu interpretieren sind, können am Netzrand am schnellsten Verstöße gegen den Verkehrsvertrag erkannt und geeignete Gegenmaßnahmen ergriffen werden. Andere Verbindungen sind so vor Beeinträchtigungen geschützt.

Deterministische Verfahren sehen davon ab, solche deterministischen oberen Schranken unterworfenen Prozesse dennoch weiterhin als Zufallsprozesse zu modellieren. Statt dessen arbeiten sie mit deterministischen Ankunftsprozessen.

Diese Betrachtungsweise ist im Zuge der Entwicklung solcher Netze auch theoretisch unter der von Cruz [49] geprägten Bezeichnung *Network Calculus* immer weiter verfeinert worden, so daß heute nicht mehr nur einfache Netzelemente wie Multiplexer, Wartespeicher usw., sondern auch komplexe Bedienstrategien und die Verknüpfung von Netzknoten zu einem Ende-zu-Ende-Dienst analysiert werden können.

3.1.1 *Network Calculus*

Bereits Cruz [49] leitet für die wichtigen Netzelemente wie verzögerungsfreie Übertragungsabschnitte, Empfangsspeicher, Demultiplexer, Wartespeicher, Multiplexer mit diversen Bedienstrategien und *Shaper* den Zusammenhang zwischen deterministischen Schranken des Datenstroms am Eingang und Ausgang her. Unter Ausnutzung dieser Ergebnisse lassen sich diese Netzelemente zu beliebig komplexen Netzmodellen zusammenfassen.

Ist die in einem beliebigen Intervall der Länge *t* ankommende Datenmenge nie größer als $A_{in}(t)$ und beträgt die Verzögerung von Dateneinheiten maximal $D_{max} < \infty$, so ist eine obere Schranke für den in einem wiederum beliebigen Intervall der Länge *t* das Netzelement verlassende Datenmenge gegeben durch

$$A_{out}(t) = A_{in}(t + D_{max}) \tag{3.1}$$

Am Ausgang von Netzelementen, in denen große und variable Verzögerungen auftreten, ist demzufolge mit büschelförmigerem Verkehr zu rechnen. Cruz [49] zeigt jedoch, daß diese Schranke verbessert werden kann, wenn man die innere Struktur des Netzelementes kennt und berücksichtigt. Beispielsweise gilt für einen Übertragungsabschnitt konstanter Verzögerung

$$A_{out}(t) = A_{in}(t) \tag{3.2}$$

Bei einem Multiplexer mit zwei Eingängen, in dem Pakete in der Reihenfolge ihres Eintreffens (FIFO) bedient werden, ist die Verzögerung, die Pakete vom Eingang 1 erfahren, natürlich vom Verkehr am Eingang 2 abhängig und umgekehrt. Wenn die Verkehrsströme als Flüssigkeitsströme modelliert werden, ist die maximale Verzögerung für beide Eingänge gleich und kann aus dem in einem beliebigen Zeitintervall *t* maximal möglichen Vorlauf des Gesamtankunftsstroms gegenüber der in diesem Intervall bedienten Datenmenge berechnet werden. Wenn statt Flüssigkeiten bis zu *L* Dateneinheiten große Pakete mit der Rate *C* bedient werden, ist die maximale Verzögerung, die an Eingang 1 eintreffende Pakete erfahren,

$$D_{1,max} = \frac{1}{C} \max_{t \geq 0} \{ A_{1,in}(t) + A_{2,in}(t + \frac{L}{C_2}) - Ct \} \tag{3.3}$$

wenn die Datenströme an den Eingängen 1 und 2 den deterministischen Schranken $A_{1,in}(t)$ bzw. $A_{2,in}(t)$ gehorchen und C_2 die Übertragungsrate an Eingang 2 ist. Mit der Zeitverschiebung LC_2^{-1} wird ein möglicherweise an Eingang 2 vor dem Beginn des betrachteten Intervalls *t* zwar noch (fast) vollständig eingetroffenes Paket berücksichtigt, dessen Bedienung jedoch erst bei Empfang der letzten Dateneinheit begonnen werden kann.

Als obere Schranke für die das Netzelement in einem Intervall der Länge *t* verlassende Datenmenge des an Eingang 1 eintreffenden Stromes wird in [49]

$$\begin{aligned} &A_{1,out}(\Delta t) \\ &= \min \{ C\Delta t, \max_{\substack{t_1 \geq 0 \\ D \geq 0}} \{ \min \{ A_{1,in}(\Delta t + D), A_{1,in}(\Delta t + t_1 + D) + A_{2,in}(t_1 + \tfrac{L}{C_2}) - C(t_1 + D) \} \} \} \end{aligned} \tag{3.4}$$

hergeleitet. Für die Schranke $C\Delta t$ in Formel (3.4) ist die endliche Übertragungsrate C am Ausgang des Multiplexers verantwortlich. Die zweite Schranke $A_{1,in}(\Delta t+D)$ folgt aus Formel (3.1), der allgemeinen oberen Schranke. Die sich an Cruz' Beweis [49] anlehnende Darstellung in Abb. 3.1 veranschaulicht die verbleibende Schranke

$$A_{1,in}(\Delta t+t_1+D)+A_{2,in}(t_1+\tfrac{L}{C_2})-C(t_1+D)$$

Dazu wird ein beliebiger Ausschnitt $[t_2, t_2+\Delta t]$ einer zum Zeitpunkt 0 beginnenden Belegtperiode der Bedieneinheit betrachtet. Zum Zeitpunkt $(t_1+D)\in[t_2, t_2+\Delta t]$ werde zum ersten Mal in diesem Intervall eine Dateneinheit von Eingang 1 bedient. Ist D die Zeit, die diese Dateneinheit auf die Bedienung warten mußte, müssen bis t_1+D alle Dateneinheiten, die vor t_1 angekommen sind, bedient sein (FIFO). Umgekehrt können Dateneinheiten, die nach t_1 eintreffen, zu $A_{1,out}(\Delta t)$ beitragen. Obwohl wir uns in einer Belegtperiode befinden, d. h. Dateneinheiten die Bedieneinheit ohne Unterbrechung mit der Rate C verlassen, werden unter Umständen nur

$$C(t_1+D)-A_{2,in}(t_1+\tfrac{L}{C_2})$$

Dateneinheiten von Eingang 1 vor t_2 bedient. Dies ist der Anteil dieses Verkehrs an der insgesamt bedienten Datenmenge $C(t_1+D)$ nach Abzug von bis $t_1<t_2$ bedienten Dateneinheiten $A_{2,in}(t_1+\tfrac{L}{C_2})$ von Eingang 2. Im (für die Rate des Ausgangsstroms der von Eingang 1 stammenden Dateneinheiten) schlimmsten Fall verhindern von nun an keine Datenpakete von Eingang 2 die Bedienung. Damit ist

$$A_{1,in}(t_2+\Delta t)-(C(t_1+D)-A_{2,in}(t_1+\tfrac{L}{C_2}))$$

eine Obergrenze für $A_{1,out}(\Delta t)$ und damit natürlich auch die oben angegebene Schranke.

Im in Abb. 3.1 dargestellten Fall erreichen im Intervall $[t_1, t_2+\Delta t]$ beispielsweise überhaupt keine Dateneinheiten Eingang 2, $\int r_2(t)\,dt$ bleibt konstant, während an Eingang 1 die obere Schranke $A_{1,in}(t_2+\Delta t-t_1)$ voll ausgeschöpft wird und $\int r_1(t)\,dt$ entsprechend maximal ansteigt.

Ein durch das Tupel (b_δ, r) gekennzeichneter *Shaper* wird definiert als ein Netzelement mit einem Eingang und einem Ausgang, der ungeachtet seiner am Ein- und Ausgang gleichen Übertragungsrate C nur dann Pakete überträgt – und zwar in der Reihenfolge ihres Eintreffens (FIFO) – falls zum Sendezeitpunkt $t_{j,out}$ die Bedingung

$$\max_{t_1\le t_{j,out}}\{\int_{t_1}^{t_{j,out}} r_{out}(\tau)\,\mathrm{d}\,\tau - r\cdot(t_{j,out}-t_1)\}\le b_\delta \qquad (3.5)$$

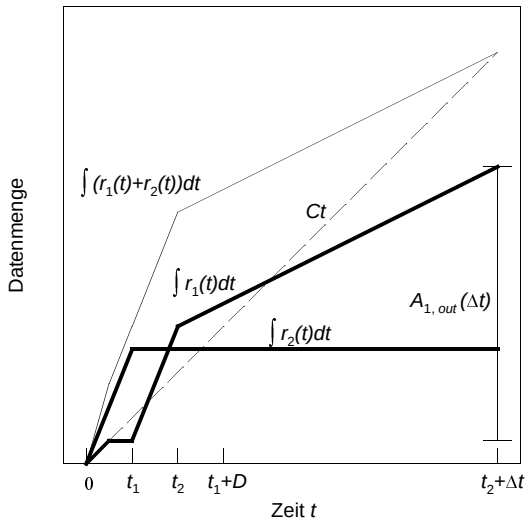

Abb. 3.1: *Darstellung der der Herleitung von Formel (3.4) für einen FIFO-Multiplexer durch Cruz [49] zugrundeliegenden Konstruktion mit zwei Eingangsströmen*

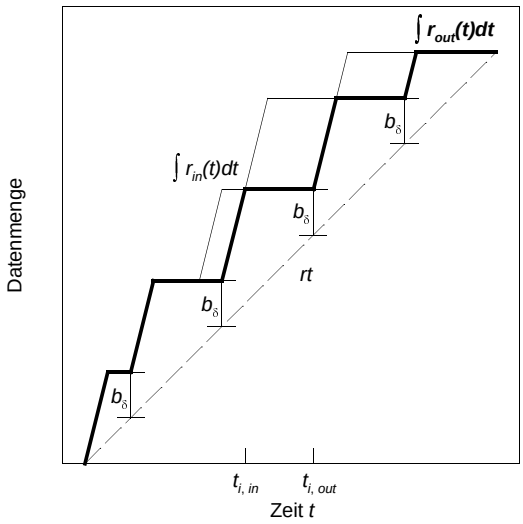

Abb. 3.2: *Funktionsweise eines (b_δ, r) Shapers. Pakete werden so lange verzögert, bis die in einem Intervall gesendete Datenmenge um nicht mehr als b_δ über rt liegt.*

erfüllt ist. Gesendet wird ein Paket j, sobald zu einem Zeitpunkt $t_{j,out}$ kein t_1 mehr existiert, so daß die im Intervall $[t_1, t_{j,out})$ übertragene Datenmenge

$$\int_{t_1}^{t_{j,out}} r_{out}(\tau)\,\mathrm{d}\,\tau$$

um mehr als b_δ von der aufgrund der vereinbarten mittlere Rate $r \le C$ zu erwartenden Datenmenge $r\cdot(t_{j,out}-t_1)$ abweicht. Selbst unter Berücksichtigung der maximalen Paketlänge L ist die übertragene Datenmenge im Intervall $[t_1, t]$ durch die folgende Beziehung beschränkt:

$$\max_{t_1 \le t}\{\int_{t_1}^{t} r_{out}(\tau)\,\mathrm{d}\,\tau - r\cdot(t-t_1)\} \le b_\delta + (1-\frac{r}{C})L \qquad (3.6)$$

Ein Shaper reduziert also die Büschelförmigkeit eines Datenstroms. Wie Abb. 3.2 zu entnehmen ist, muß er dazu eintreffende Pakete um genau

$$t_{j,out} - t_{j,in} = \max\{0, \frac{1}{r}(\max_{t_1 \le t_{j,in}}\{\int_{t_1}^{t_{j,in}} r_{in}(\tau)\,\mathrm{d}\,\tau - r\cdot(t_{j,in}-t_1)\} - b_\delta)\} \qquad (3.7)$$

verzögern.

3.1.2 Berechnung von maximalen Ende-zu-Ende Verzögerungszeiten

Zur Realisierung eines Netzdienstes mit garantierten Dienstgüteeigenschaften müssen die Verkehrssteuerungsfunktionen der Einzelknoten zusammenwirken und die Verkehrsströme so durch die Einzelknoten leiten und voneinander isolieren, daß für die Nutzer der Eindruck einer isolierten Ende-zu-Ende-Netzverbindung entsteht, welche die gewünschten Dienstgüteattribute unter allen Umständen beibehält. Den Protokollen und Algorithmen zur Verkehrslenkung fällt dabei die Aufgabe zu, Pfade zu finden, die Ressourcen auf geeignete Weise und in ausreichendem Umfang bereitstellen können. Im Einklang mit der in Kapitel 2 formulierten Strategie einer dezentralen Verbindungssteuerung sollten Netzknoten entlang dieses Pfades dann in die Lage sein, lokal und weitgehend unabhängig voneinander die notwendigen Konfigurationsschritte einzuleiten und ausreichend Ressourcen zu reservieren. Stiliadis und Varma stellen in [186] ein als *Latency-Rate* Bedieneinheit bezeichnetes Modell vor, mit dessen Hilfe eine obere Grenze für die Verzögerungszeiten durch eine Reihe von Bediensystemen angegeben werden kann, wenn – genauso wie bei Cruz [49] – der Verkehr am Eingang deterministischen oberen Schranken gehorcht. Dem Modell liegt das Konzept von invarianten *Busy Periods* zugrunde, die als die Zeitspannen definiert werden, während der die durchschnittliche Ankunftsrate an einem Bediensystem größer als die für die Verbindung reservierte Rate ist. Diese *Busy Periods* sind nicht deckungsgleich mit den Zeitspannen, während der Dateneinheiten der Verbindung im Bediensystem sind, da die dafür entscheidende tatsächliche Bedienrate in allen realisierbaren Bedieneinheiten nur im Mittel der reservierten Rate entspricht, kurzfristig aber abhängig von der Bedienstrategie mehr oder weniger stark davon abweichen kann. Eine Bedieneinheit wird nun genau dann als *Latency-Rate* Bedieneinheit bezeichnet, wenn die mittlere Rate, mit der sie eine Verbindung bedienen kann, in jedem Intervall, das nicht vor einer Zeit Θ nach dem Beginn einer *Busy Period* beginnt, nicht kleiner als die reservierte Rate R ist.

Definition [186]

Seien t_1 der Startzeitpunkt der j-ten *Busy Period* einer Verbindung i in der Bedieneinheit S und t_2 der Zeitpunkt, zu dem die letzte Anforderung, die während der j-ten *Busy Period* angekommen ist, die Bedieneinheit verläßt. Die Bedieneinheit S ist eine *Latency-Rate* Bedieneinheit genau dann, wenn eine nicht negative Konstante C_i^S existiert, so daß für jeden Zeitpunkt t im Intervall $(t_1, t_2]$ für die Arbeit $W_{i,j}^S(t_1, t)$, welche die Bedieneinheit S für in der j-ten *Busy Period* angekommenen Anforderungen der Verbindung i im Intervall $(t_1, t]$ geleistet hat, gilt:

$$W_{i,j}^S(t_1, t) \geq \max\{0, R_i(t - t_1 - C_i^S)\} \tag{3.8}$$

Die kleinste nicht negative Konstante C_i^S, welche der Ungleichung (3.8) genügt, wird als *Latency* Θ_i der Bedieneinheit *S* bezüglich Verbindung *i* bezeichnet.

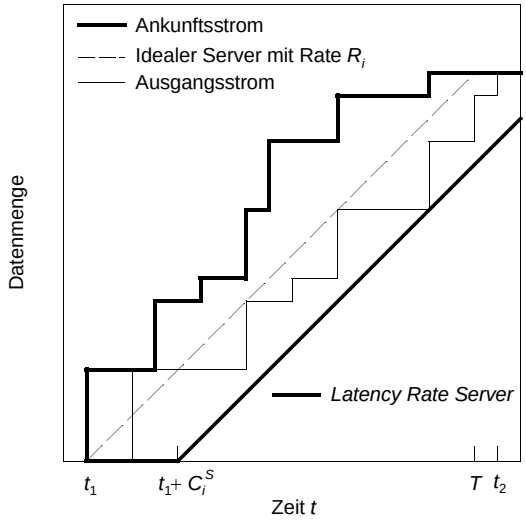

***Abb. 3.3**: Beispiel für das Verhalten einer Latency Rate Bedieneinheit in einer Busy Period* $(t_1, t]$

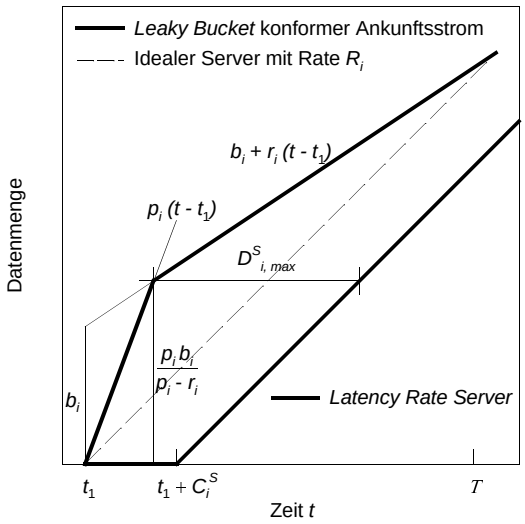

***Abb. 3.4**: Maximale Verzögerungszeit von Dateneinheiten einer Leaky Bucket konformen Quelle durch einen Latency Rate Bedieneinheit*

Wie in Abb. 3.3 an dem treppenförmigen Verlauf der bis zum jeweiligen Zeitpunkt *t* angekommenen bzw. bedienten Datenmenge zu erkennen ist, werden sowohl der Ankunftsstrom als auch der Ausgangsstrom als Punktprozesse modelliert. Demzufolge löst die Ankunft der ersten Dateneinheit an der ersten Bedieneinheit in einer Reihe auch eine *Busy Period* im nachfolgenden Netzknoten aus, sobald die Dateneinheit vom jeweiligen Vorgängerknoten bedient worden ist. Weitere Pakete können diese *Busy Period* im zweiten Knoten verlängern oder neue auslösen. Die erste *Busy Period* im zweiten Knoten kann bereits Pakete der zweiten *Busy Period* im ersten Knoten enthalten. Dieser Unterschied zwischen der ersten und nachfolgenden *Busy Periods* am ersten Knoten kann jedoch mittels vollständiger Induktion aus der Betrachtung der ersten *Busy Period* behandelt werden. Damit sind zunächst drei Fälle zu unterscheiden. Der erste Fall ist trivial und erstreckt sich auf die Zeitspanne von der Ankunft des ersten Pakets am ersten Knoten bis zum Beginn der Bedienung im zweiten Knoten, also auf die Zeitspanne vor dem Beginn der Bedienung, also der ersten *Busy Period* im zweiten Knoten. Der zweite Fall behandelt alle durch die *Busy Period* am ersten Knoten ausgelösten *Busy Periods* im zweiten Knoten bis zum Ende der Bedienung und der dritte Fall die Zeitspannen zwischen *Busy Periods* im zweiten Knoten. Die Autoren von [186] weisen nach, daß in all diesen drei Fällen und auch für nachfolgende *Busy Periods* am ersten Knoten die minimale

Arbeit $W_{i,j}^{S_k}(t_1,t)$, welche eine Reihe von *k Latency Rate* Bedieneinheiten für eine Verbindung *i* der *j*-ten *Busy Period* im Intervall $(t_1,t]$ leistet, gegeben ist durch

$$W_{i,j}^{S_k}(t_1,t) \geq \max\{0, R_i(t-t_1-\sum_{j=1}^{k}\Theta_i^{S_j})\} \tag{3.9}$$

Dabei bezeichnet $\Theta_i^{S_j}$ die *Latency* der *j*-ten Bedieneinheit bezüglich Verbindung *i*. Beginnend mit der Betrachtung von zwei hintereinandergeschalteten Netzknoten kann eine Reihe von *Latency Rate* Bedieneinheiten also zu einer einzigen *Latency Rate* Bedieneinheit mit *Latency*

$$\sum_{j=1}^{k}\Theta_i^{S_j}$$

abstrahiert werden, so daß sich die weitere Betrachtung zunächst auf einen Einzelknoten beschränken kann.

Am Ende einer *Busy Period*, in Abb. 3.3 und 3.4 als *T* bezeichnet, ist definitionsgemäß die Datenmenge $R_i(T-t_1)$ angekommen, die dann spätestens zur Zeit $T+\Theta_i$ bedient worden ist. Die maximal mögliche Abweichung der tatsächlich seit dem Startzeitpunkt t_1 einer *Busy Period* bis zu einem beliebigen Zeitpunkt *t*, $t \leq T$ ankommenden Datenmenge von der mindestens bedienten Datenmenge $W_{i,j}^{S}(t_1,t)$ gemäß Ungleichung (3.8) bestimmt demzufolge die maximal auftretende Verzögerung.

Kann für die im Intervall $(t_1,t]$ ankommende Datenmenge der Verbindung *i* eine deterministische obere Grenze $A_i(t-t_1)$ angegeben werden, beispielsweise

$$A_i(t-t_1) \leq \min\{b_i + r_i(t-t_1), p_i(t-t_1)\} \tag{3.10}$$

wie in Abb. 3.4 für eine Quelle mit mittlerer Rate $r_i \leq R_i$, Büscheltoleranz b_i und endlicher Spitzenrate $p_i \geq R_i$, kann mit Hilfe der in der Abbildung dargestellten Konstruktion als obere Grenze für die Verzögerung

$$D_{i,max}^{S} = \frac{b_i}{R_i} \cdot \frac{p_i - R_i}{p_i - r_i} + \Theta_i \tag{3.11}$$

und damit entsprechend für eine Kette von *k Latency Rate* Bediensystemen

$$D_{i,max} = \frac{b_i}{R_i} \cdot \frac{p_i - R_i}{p_i - r_i} + \sum_{j=1}^{k}\Theta_i^{S_j} \tag{3.12}$$

angegeben werden.

Im folgenden werden solche deterministische obere Schranken wie (3.10) für die in einem Intervall einer bestimmten Länge ankommende Datenmenge einer Verbindung als (deterministische) Verkehrsankunftsfunktionen bezeichnet.

3.1.3 Verteilte Verbindungsannahmesteuerung

Unter Verwendung dieser Ergebnisse koordiniert die Verbindungssteuerung von zur Spezifikation von *Guaranteed Service* [176] konformen Netzknoten die Verkehrssteuerungsfunktionen der einzelnen Knoten auf dem Pfad vom Sender zum Empfänger so, daß ein verlustfreier Netzdienst mit garantierten oberen Grenzen für die Paketverzögerung Ende-zu-Ende angeboten werden kann. Dazu zerlegen Netzknoten den Beitrag ihrer Bedieneinheit zur *Latency* des Pfades in eine von der reservierten Rate R_i abhängige Komponente $C_j^{GS} R^{-1}$ und eine konstante Komponente D_j^{GS}. Bei der Verarbeitung von RSVP-Pfadnachrichten exportieren sie diese Information durch Addition zu den entsprechenden Feldern des Objektes *Adspec*.

Die Kenntnis der Formel

$$D_{i,max} = \frac{b_i - L_i}{R_i} \cdot \frac{p_i - R_i}{p_i - r_i} + \frac{L_i}{R_i} + \sum_{j=1}^{k} \frac{C_j^{GS}}{R_i} + D_j^{GS} \tag{3.13}$$

zur Berechnung der maximalen Paketverzögerung Ende-zu-Ende, welche die maximale Paketlänge L_i der Verbindung i berücksichtigt und für welche die deterministische obere Grenze

$$A_i(t - t_1) \leq \min\{b_i + r_i(t - t_1), L_i + p_i(t - t_1)\} \tag{3.14}$$

der im Intervall $(t_1, t]$ ankommenden Datenmenge gilt, ermöglicht es Empfängern, abhängig vom Verkehrsdeskriptor in der Reservierungsanforderung (*Rspec*) die Rate R_i anzugeben, aus der die gewünschte maximale Verzögerung resultiert. Gleichung (3.13) geht unmittelbar aus Gleichung (3.12) hervor, wenn die Obergrenze für die im Intervall $(t_1, t]$ ankommende Datenmenge („Leaky Bucket konformer Ankunftsstrom“ in Abb. 3.4) bei t_1 mit einem Sprung auf L_i startet, dafür aber b_i durch $b_i - L_i$ ersetzt wird.

Reserviert ein Empfänger über die tatsächlich benötigte Rate hinaus Bandbreite, kann er der Reservierungsanforderung in der RSVP-Reservierungsnachricht den sogenannten *Slack Term* S_i^{GS} hinzufügen, eine Toleranz für die maximale Verzögerung, die diejenigen Knoten im Pfad verbrauchen können, die Bandbreite nicht in dem eigentlich erforderlichen Maße bereitstellen können und daher die Verbindung ansonsten ablehnen müßten.

Wie die Gegenüberstellung von Formel (3.1) mit Formel (3.2) zeigt, verändert der variable Anteil der Verzögerung in jedem Knoten die Verkehrscharakteristik der Datenströme. Die Abweichung von der ursprünglichen deterministischen Obergrenze $A_i(t_1,t)$ für die im Intervall $(t_1,t]$ ankommende Datenmenge und dementsprechend der beanspruchte Wartespeicher ist um so größer, je größer die *Latency* der einzelnen Bedieneinheiten ist und je mehr Knoten passiert worden sind.

Aus diesem Grunde kombinieren die Autoren von [80] verbindungsbezogene *Shaper* und die Bedieneinheit zu einer *Rate Controlled Service* (RCS) *Discipline*. Als Bedienstrategie kommt *Earliest Deadline First* (EDF) zum Einsatz. Diese Bedienstrategie berechnet für ankommende Pakete einer Verbindung *i* einen Übertragungstermin als Summe aus der Ankunftszeit und der für die Verbindung festgelegten maximalen Verzögerungszeit $D^S_{i,max}$. Wenn ein Paket die Bedieneinheit verläßt, beginnt als nächstes das Paket mit dem frühesten Übertragungstermin die Bedienung. EDF ist in bezug auf die Anzahl der Verbindungen, die bei gegebener Bedienkapazität zugelassen werden können, die optimale Bedienstrategie. Liebeherr et al. [133] drücken diesen Sachverhalt folgendermaßen aus:

Definition [133]

Gegeben sei eine Bedieneinheit *S* mit arbeitserhaltender Bedienstrategie Σ und eine Menge *M* von Verbindungen, deren Eigenschaften vollständig beschrieben sind durch die Tupel $(A_i(t), D^S_{i,max})$ aus rechtsseitig stetigen, subadditiven, deterministischen Verkehrsankunftsfunktionen $A_i(t)$ und maximalen Verzögerungszeiten $D^S_{i,max}$, die Dateneinheiten von Verbindung *i* durch die Bedieneinheit *S* erfahren dürfen. Es gelte ferner die Stabilitätsbedingung

$$\lim_{t\to\infty} \frac{1}{C_A t} \sum_{i\in M} A_i(t) < 1 \qquad (3.15)$$

Dann heißt die Menge *M* von Verbindungen Σ*-schedulable*, wenn für alle $t>0$ keine Dateneinheit mehr als die ihrer Verbindung zugeordnete maximale erlaubte Zeit $D^S_{i,max}$ verzögert wird. *M* heißt *schedulable*, wenn eine Bedienstrategie Σ existiert, so daß *M* Σ*-schedulable* ist.

Eine Funktion $A_i(t)$ ist genau dann subadditiv, wenn für alle $t_1,\ t_2 \geq 0$ gilt:

$$A_i(t_1+t_2) \leq A_i(t_1)+A_i(t_2) \qquad (3.16)$$

Satz [133]

Wenn eine Menge *M* von Verbindungen *schedulable* ist, dann ist sie EDF-*schedulable*.

Da sich die Bedienreihenfolge bei EDF nicht direkt an der für die Verbindung reservierten Rate R_i orientiert, sondern an der maximal erlaubten Verzögerung $D^S_{i,max}$, tritt die gesamte Verzögerung in Form von *Latency* auf, die sich gemäß Formel (3.12) in einem längeren Pfad zu beträchtlichen Beträgen summieren kann. Der *Shaper* wird in [80] allerdings so konfiguriert, daß der Datenstrom der Verbindung konform zu einem Verkehrsdeskriptor ist, dessen Spitzenrate $\max\{p_i, R_i\}$ ist und dessen weitere Parameter der am Netzeingang vereinbarten Büscheltoleranz b_i bzw. der Rate r_i entsprechen. Er richtet sich also direkt nach der für die Verbindung vereinbarten Raten. Auf die Bedieneinheit, die nach der Strategie EDF bedient, entfällt nach dieser Verkehrsformung nur noch der (unvermeidliche und in der Regel kleine) Teil der Verzögerung aufgrund der Segmentierung des Datenstroms in Pakete. Nur dieser Teil wird in Form der Parameter C^{GS}_j und D^{GS}_j exportiert.

3.1.4 Berechnung des Ressourcenbedarfs eines Aggregates mit der deterministischen Methode

Für die Berechnung der Bedienrate C_A des Bediensystems, als die auch ein Aggregat modelliert werden kann, sind sogenannte *Schedulability Conditions* heranzuziehen. Das sind notwendige und hinreichende Bedingungen in Form von Ungleichungen, die durch die Wahl einer ausreichend großen Bedienrate C_A zu erfüllen sind, wenn sich kein Paket länger als die für seine festgelegte Zeit $D^S_{i,max}$ bis zu seiner vollständigen Übertragung im Bediensystem aufhalten soll. Per Konvention sind in diesen Ungleichungen die Verbindungen nach den für sie festgelegten Verzögerungszeiten sortiert, d. h. es gilt $D^S_{i,max} \le D^S_{j,max}$, wenn $i < j$ ist.

In [133] wird gezeigt, daß eine Menge von M Verbindungen genau dann *schedulable* ist, wenn für alle $t \ge D^S_{1,max}$ die Bandbreite eines Übertragungsabschnittes bzw. die für ein Aggregat zu reservierende Rate C_A die Ungleichung

$$C_A t \ge \sum_{i \in M} A_i(t - D^S_{i,max}) + \max_{i: D^S_{i,max} > t} \{L_i\} \qquad (3.17)$$

erfüllt. Zu jedem Zeitpunkt t muß demnach der im schlimmsten Fall in einem Intervall bis $t - D^S_{i,max}$ ankommende Verkehr der entsprechenden Verzögerungspriorität bedient sein. Die Verkehrsankunftsfunktion, also die deterministische obere Schranke des in einem beliebigen Intervall der Länge t ankommenden relevanten Gesamtverkehrs, erhält man durch Addition der Verkehrsankunftsfunktionen der Einzelbeiträge, $\sum_{i \in M} A_i(t - D^S_{i,max})$. Möglicherweise trägt auch noch ein Paket zum relevanten Gesamtverkehr bei, das aufgrund seines Termins noch nicht bedient werden müßte,

aber dessen Bedienung schon vor der Ankunft der Pakete mit früheren Terminen begonnen hat und jetzt nicht mehr unterbrochen werden kann.

Auf Ungleichung (3.17) aufbauend, arbeiten Firoiu et al. [70] eine effiziente Lösung für die Verbindungsannahmesteuerung formal aus. Im folgenden werden jedoch die entsprechenden, aber weniger formalen Überlegungen in [1] für die Bedienstrategien RPQ (*Rotating Priority Queues*) [133] und RPQ+ [134] weitergeführt.

Bei Bedienstrategien wie EDF, die den Paketen zum Ermitteln der Bedienreihenfolge Zeitstempel zuordnen, muß die Menge der Verbindungen sortiert werden, sobald ein Paket das Bediensystem verläßt und das nächste zur Bedienung ausgewählt werden soll. Angesichts stetig wachsender Übertragungskapazitäten, angesichts des Informationsverlustes, der mit der Beschreibung von Verkehr in Form von Verkehrsdeskriptoren einhergeht, angesichts der Segmentierung von Verkehrsströmen in Pakete variabler Größe und nicht zuletzt angesichts der mit der deterministischen Methode verbundenen konservativen Zuteilung von Bandbreite und Wartespeicher ist es zumindest fraglich, ob diese Komplexität notwendig und gerechtfertigt ist.

Tatsächlich nähern die Bedienstrategien RPQ [133] und RPQ+ [134] das Verhalten von EDF bei weit geringerem Aufwand durchaus zufriedenstellend an. Eine Implementierung von RPQ besteht aus N_c+1 nach Priorität geordneten FIFO-Warteschlangen, im folgenden mit $0, 1, ..., N_c$ indiziert. Das Warteschlangensystem rotiert in konstanten Abständen Δ um eine Position weiter, so daß die Warteschlange mit der bisher höchsten Priorität an die letzte Stelle tritt, alle anderen dafür aber eine Position nach vorne rücken. Wenn nun einer Verbindung *i* die Priorität c, $c=1, ..., N_c$ zugeordnet ist, wenn Pakete dieser Verbindung immer in die Warteschlange eingefügt werden, die zum Ankunftszeitpunkt mit n_c indiziert ist, und wenn die Bedieneinheit die Warteschlangen in der durch die Priorität vorgegebenen Reihenfolge bedient, lassen sich bis zu N_c diskrete Prioritäten mit maximalen Verzögerungszeiten $n_c\Delta$ realisieren. Natürlich müssen wie bei EDF *Schedulability Conditions* erfüllt sein, wenn alle Pakete verlustfrei innerhalb der spezifizierten maximalen Verzögerungszeiten bedient werden sollen. In [133] wird die Beziehung

$$C_A t \geq \sum_{i \in M_1} A_i(t - D^S_{1,max}) + \sum_{c=2}^{N_c} \sum_{i \in M_c} A_i(t + \Delta - D^S_{c,max}) + \max_{i: D^S_{i,max} > t+\Delta} \{L_i\} \qquad (3.18)$$

als notwendige und hinreichende Bedingung für die RPQ-*Schedulability* einer Menge *M* von Verbindungen hergeleitet. Der besseren Übersichtlichkeit wegen bezeichnen die neuen Symbole M_1 bzw. M_c die Mengen der Verbindungen, die zur Prioritätsklasse 1 bzw. *c* gehören. Wie oben ange-

deutet, sind nur diskrete $D^S_{i,max}=D^S_{c,max}=n_c\Delta$ erlaubt. Ungleichung (3.18) ist für alle $t\geq D^S_{1,max}$ zu verifizieren. Für $\Delta\to 0$ geht (3.18) in (3.17) über, d. h. RPQ approximiert EDF mit abnehmendem Δ immer besser.

Des besseren Verständnisses der nachfolgenden Abschnitte wegen betrachte man zur Begründung von (3.18) ein Paket einer Verbindung der Prioritätsklasse j, das zum Zeitpunkt t am Bediensystem eintrifft, und zwar zwischen den Rotationszeitpunkten $t-t_1$ und $t-t_1+\Delta$. Das Paket wird rechtzeitig bedient, wenn zum Zeitpunkt $t+n_j\Delta$ die Restarbeit für Pakete mit Priorität kleiner oder gleich j kleiner oder gleich null ist. Diese Restarbeit sei zum Zeitpunkt 0 zum letzten Mal vor der Ankunft des betrachteten Paketes 0. Zum Zeitpunkt $t+n_j\Delta$ setzt sie sich dann zusammen aus den Paketen von Verbindungen der Prioritätsklassen $c=1, ..., j-1$, die nach dem Zeitpunkt 0, aber vor dem betrachteten Paket oder zumindest früher als $t-t_1+(n_j-n_c)\Delta$ ankommen, also je nach Priorität im nächsten, übernächsten usw. Rotationsintervall danach, aus dem vor dem Paket eintreffenden Verkehr gleicher Priorität j und schließlich aus dem Verkehr der niederen Prioritätsklassen $c=j+1, ..., N_c$, der schon früher ankommt, nämlich wiederum abhängig von der Prioritätsklasse bis $t-t_1+\Delta+(n_j-n_c)\Delta$. Diese Beiträge sind deterministisch begrenzt durch

$$\sum_{c=1}^{j-1}\sum_{i\in M_c} A_i(t-t_1+(n_j-n_c)\Delta)+\sum_{i\in M_j} A_i(t)+\sum_{c=j+1}^{N_c}\sum_{i\in M_c} A_i(t-t_1+\Delta+(n_j-n_c)\Delta) \qquad (3.19)$$

Möglicherweise wird zur Zeit 0, wenn zum ersten Mal wieder ein Paket der Prioritätsklassen $c=1, ..., j$ das Bediensystem erreicht, noch ein Paket niederer Priorität bedient, das eigentlich nach $t+D^S_{j,max}=t+n_j\Delta$ bedient werden könnte. Seine Restbedienzeit schlägt mit maximal

$$\max_{i:D^S_{i,max}>t+D^S_{j,max}+\Delta}\left\{\frac{L_i}{C_A}\right\} \qquad (3.20)$$

zu Buche. Da sich zur Zeit 0 definitionsgemäß zum letzten Mal keine Pakete der Prioritäten $c=1, ..., j$ im Bediensystem aufgehalten haben, können bis zum Bedientermin $t+n_j\Delta$ des betrachteten Paketes $C_A(t+n_j\Delta)$ Dateneinheiten der Prioritäten $c=1, ..., j$ bedient werden. Subtrahiert man $C_A(t+n_j\Delta)$ von der Summe der Terme (3.19) und (3.20), so erhält man die gesuchte Restarbeit zum Zeitpunkt $t+n_j\Delta$. Im schlimmsten Fall trifft das betrachtete Paket unmittelbar nach einer Rotation ein. Dann ist $t_1=0$. Für diesen Fall ergibt sich nach Substitution von $t+n_j\Delta$ durch t die *Schedulability Condition* (3.18) für $j=1$.

Bei RPQ müssen sich neue Pakete grundsätzlich hinter den Paketen der Verbindungen mit niedrigerer Priorität einreihen, die in vergangenen Rotationsintervallen in die Warteschlange einsortiert worden sind, als dieser noch die entsprechende niedrigere Priorität zugeordnet war. Dies benachteiligt Pakete, die kurz nach einer Rotation ankommen und deswegen bis zu Δ früher bedient werden müssen als Pakete niederer Prioritäten, die kurz vor einer Rotation angekommen sind und deswegen im Grunde weniger kritisch sind.

Um dies zu vermeiden, wird in [134] jeder der N_c Warteschlangen eine zweite, mit c^+ indizierte zur Seite gestellt, in die im Gegensatz zu den mit c indizierten Warteschlangen auf direktem Wege keine Pakete eingefügt werden. Dieses RPQ+ genannte Bediensystem hängt bei einer Rotation die Warteschlangen c^+ an c an und weist der so neu entstehenden Warteschlange danach den Index $(c-1)^+$ zu. Da die Bedieneinheit die Warteschlangen in der Reihenfolge 0^+, 1, 1^+, ..., c, c^+, ..., N_c bedient, werden innerhalb einer Warteschlange die Pakete immer strikt in der Reihenfolge ihrer Verzögerungspriorität bedient, im Unterschied zu RPQ.

Überlegungen analog zu den zur Begründung von Ungleichung (3.18) angeführten zeigen [134], daß eine Menge M von Verbindungen dann und nur dann RPQ+ *schedulable* ist, wenn für alle Prioritäten j und alle $t \geq 0$ ein t_1 im Intervall $[0, D^S_{j,max} - \min\{L_i C_A^{-1}\}]$ existiert, so daß gilt

$$\begin{aligned} C_A(t+t_1) \geq & \sum_{c=1}^{j-1} \sum_{i \in M_c} A_i(\min\{t+t_1, t+D^S_{j,max} - D^S_{c,max} + \Delta\}) \\ & + \sum_{c=j}^{N_c} \sum_{i \in M_c} A_i(t+D^S_{j,max} - D^S_{c,max}) - \min\{L_i\} + \max_{i: D^S_{i,max} > t + D^S_{j,max}} \{L_i\} \end{aligned} \tag{3.21}$$

Eine ausreichende Bedingung ist mit $t_1 = D^S_{j,max} - \min\{L_i C_A^{-1}\}$ nach Substitution von $t + D^S_{j,max}$ durch t, $t \geq D^S_{j,max}$, auch gegeben durch

$$C_A t \geq \sum_{c=1}^{j-1} \sum_{i \in M_c} A_i(t - D^S_{c,max} + \Delta) + \sum_{c=j}^{N_c} \sum_{i \in M_c} A_i(t - D^S_{c,max}) + \max_{i: D^S_{i,max} > t} \{L_i\} \tag{3.22}$$

3.1.5 Praktische Realisierung

Die *Schedulability Conditions* (3.17) für EDF, (3.18) für RPQ und (3.22) für RPQ+ müssen für alle $t \geq D^S_{1,max}$ bzw. $t \geq D^S_{j,max}$ im Falle von RPQ+ verifiziert werden. Mit abschnittsweise linearen Verkehrsankunftsfunktionen wie etwa (3.14) ist so ein Ansatzpunkt für ein praktisch realisierbares Verfahrens zur Berechnung der Bandbreite C_A eines Verbindungsaggregates gegeben. Die rechten Seiten der Ungleichungen (3.17), (3.18) und (3.22), allesamt Summen solcher abschnittsweise

linearen Funktionen, sind dann ebenfalls stückweise lineare Funktionen, und es genügt, die *Schedulability Conditions* zu einer endlichen Anzahl diskreter, im folgenden auch als kritisch bezeichneter Zeitpunkte, zu betrachten. Kritisch sind alle Zeitpunkte, in denen sich die Steigung der rechten Ungleichungsseite ändert. Denn ist für zwei solche Zeitpunkte die *Schedulability Condition* erfüllt, dann ist sie auch für alle Zeitpunkte dazwischen erfüllt, weil die Gerade $C_A t$ die Strecke zwischen den diesen Zeitpunkten zugeordneten Funktionswerte der rechten Ungleichungsseite höchstens einmal schneiden kann.

Es ist leicht einzusehen, daß sich die Steigung der Summe auf der rechten Seite der Ungleichungen der *Schedulability Conditions* dann ändert, wenn sich die Steigung eines der Einzelbeiträge ändert. Die Berechnung des Ressourcenbedarfs eines Aggregates reduziert sich also darauf, für die deterministische Schranke einer jeden Verbindung in jeder Ungleichung abhängig von der zeitlichen Verschiebung der Variable der Zeit in $A_i(t-...)$ die kritischen Zeiten zu bestimmen, danach für jede kritische Zeit das kleinste C_A zu berechnen, so daß die Ungleichung zur kritischen Zeit erfüllt ist, und am Ende als Bandbreite des Aggregats das größte C_A zu übernehmen.

Eine deterministische Schranke der Form (3.14) führt zu maximal zwei kritischen Zeiten je Verbindung. Werden nur diskrete Verzögerungsprioritäten angeboten, sei es weil die Bedienstrategie das so erzwingt wie bei RPQ oder RPQ+ oder weil der Netzdienst entsprechend spezifiziert ist, und werden Verbindungen außerdem zu Verbindungstypen gruppiert, sinkt der Beitrag einer einzelnen Verbindung zur Komplexität des Berechnungsverfahrens natürlich erheblich. Denn dann sind die kritischen Zeiten aller Verbindungen identisch, deren Typ und Verzögerungspriorität übereinstimmen. Die Berechnung der Bandbreite C_A eines Aggregats ist dabei bei RPQ+ etwas komplexer als bei RPQ und EDF, weil die *Schedulability Condition* (3.22) entsprechend der Anzahl der Verzögerungsprioritäten in N_c Ungleichungen aufzuspalten ist und Verbindungen bzw. Verbindungstypen dadurch mit bis zu $2N_c$ kritischen Zeiten in die Berechnung eingehen können, obwohl die Ungleichungen jeweils nur für $t \geq D^S_{j,max}$ zu verifizieren sind. Dies zeigt die höhere Anzahl der zur Verdeutlichung verbundenen Berechnungspunkte für die Verifikation der *Schedulability Condition* (3.22) von RPQ+ in Abb. 3.7 und 3.8.

Trägt man für jede Bedienstrategie Σ für alle Kombinationen von zu *n* Typen gehörenden Verbindungen, die Σ*-schedulable* sind, die Summe der mittleren Raten von Verbindungen in eine dem jeweiligen Typ zugeordnete Koordinatenrichtung eines kartesischen Koordinatensystems auf, so erhält man einen Bereich im $\mathbb{R}^n$. Den Rand dieses Bereichs bezeichnet man auch als Verbindungsannahmegrenzkurve.

Tabelle 3.1: *Die für die Abb. 3.5-3.8 verwendeten Quellen. Die Typen 1-3 haben einen Büschelfaktor* $pr^{-1}=2$ *bei relativ kurzen Büscheldauern* B_T*, die Typen 4-6 einen Büschelfaktor* $pr^{-1}=10$ *bei längeren Büscheldauern* B_T*.*

					Anzahl			
Typ	$\frac{D^S_{i,max}}{ms}$	$\frac{r}{C}$	$\frac{p}{C}$	$\frac{B_T}{ms}$	Abb. 3.5	Abb. 3.6	Abb. 3.7	Abb. 3.8
1	20	0,005	0,010	10	Alle Kombi-nationen	0	50	0
2	60	0,005	0,010	30			38	
3	100	0,005	0,010	50			80	
4	20	0,005	0,050	90	0	Alle Kombi-nationen	0	58
5	60	0,005	0,050	100				26
6	100	0,005	0,050	110				14

Tatsächlich ist das Volumen des von der Verbindungsannahmegrenzkurve eingeschlossenen zulässigen Bereichs bei RPQ+ größer als bei RPQ, wie beispielhaft Abb. 3.5 und 3.6 zeigen. Besonders für die büschelförmigeren Quelltypen 4 bis 6 aus Tab. 3.1 nähert RPQ+ das optimale EDF bereits bei einem relativ großen Rotationsintervall Δ gut an. In der Darstellung ist das Volumen des zulässigen Bereichs bezogen auf das Volumen, das von der Verbindungannahmegrenzkurve bei Reservierung nur der mittleren Rate einer jeden Verbindung eingeschlossen wäre.

Wenn sich die Beiträge der Verbindungen ausgewogen auf die verschiedenen kritischen Zeiten verteilen, ist auch beim deterministischen Multiplexen ein Multiplexgewinn, eine höhere Bandbreiteausbeute als bei Trennung der Verbindungen, möglich. Diese Bedingung können natürlich nur Kombinationen von Verbindungen verschiedenen Typs erfüllen, ganz im Gegensatz zur Berechnung des Bandbreitebedarfs unter Berücksichtigung statistischen Multiplexens. Die Abb. 3.7 und 3.8 bestätigen dies für ausgewählte Punkte auf der Verbindungsannahmegrenzkurve der Verbindungen aus Tab. 3.1. Während die Verbindungen in Abb. 3.7 vor allem Beiträge bei großen kritischen Zeiten leisten, ist in Abb. 3.8 die Last doch recht gleichmäßig über die Zeit verteilt und der Multiplexgewinn gegenüber nach Typen getrennten Aggregaten dementsprechend deutlich höher.

Le Boudec [127] ordnet Verbindungen eine deterministische effektive Bandbreite

$$e_{D,i} = \sup_{t \geq 0} \frac{A_i(t)}{t + D^S_{i,max}} \tag{3.23}$$

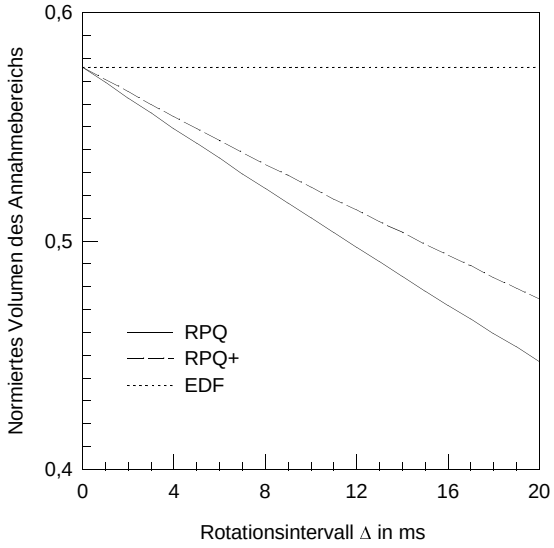

Abb. 3.5: *Volumen des zulässigen Bereiches für Verbindungen der Typen 1-3 in Tab. 3.1 als Funktion des Rotationsintervalls Δ von RPQ/RPQ+*

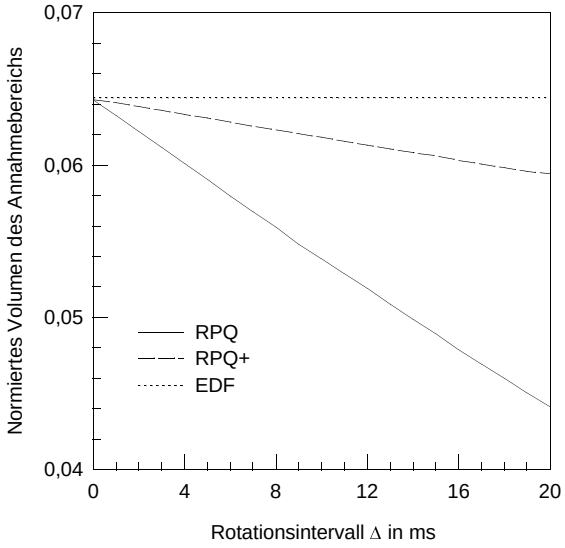

Abb. 3.6: *Volumen des zulässigen Bereiches für Verbindungen der Typen 4-6 in Tab. 3.1 als Funktion des Rotationsintervalls Δ von RPQ/RPQ+*

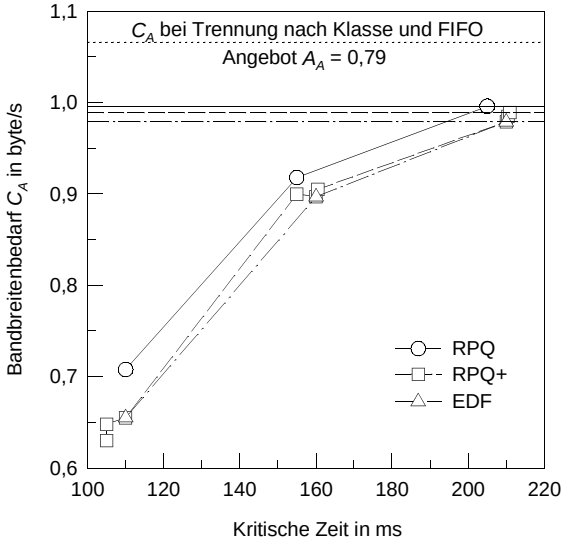

Abb. 3.7: *Verifikation der Schedulability Conditions der Kombination von Verbindungen der Typen 1-3 für RPQ, RPQ+, EDF. Jeder Punkt repräsentiert eine Ungleichung. Das größte C_A bestimmt den Bandbreitebedarf.*

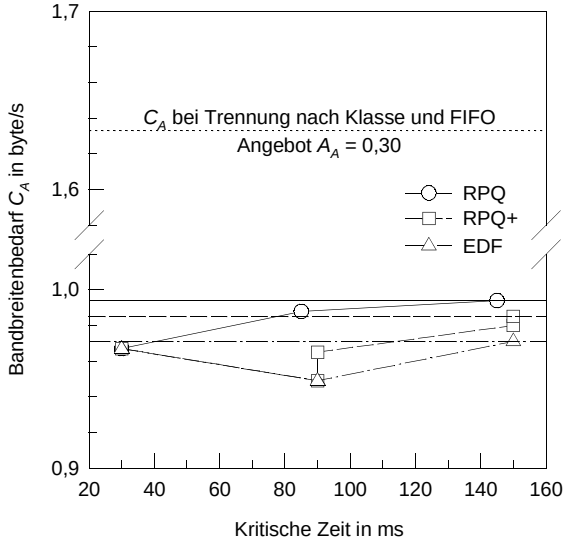

Abb. 3.8: *Im Gegensatz zu den Typen 1-3 in Abb. 3.7 verteilt sich der Bandbreitebedarf bei den Typen 4-6 gleichmäßiger auf die Ungleichungen. Entsprechend größer ist der Gewinn gegenüber getrennten Aggregaten.*

zu. Diese Definition entspricht für $\max\{L_i\}=0$ und für nur eine Verbindung genau der Suche nach der kritischen Zeit, aus der der maximale Bandbreitebedarf C_A der *Schedulability Condition* (3.17) für EDF resultiert. Über diesen Zusammenhang lassen sich Le Boudecs Schlußfolgerungen für eine auf dieser deterministischen Bandbreite beruhende, also ideale deterministische Verbindungsannahmesteuerung unmittelbar auf EDF und näherungsweise auf RPQ sowie RPQ+ übertragen. Im

Hinblick auf die nachfolgende Diskussion von Verfahren zur Berechnung des Ressourcenbedarfs unter Berücksichtigung statistischen Multiplexens ist vor allem die aus den deterministischen Verfahren resultierende Form des von der Verbindungsannahmegrenzkurve eingeschlossenen Bereichs interessant. Wie die Betrachtungen zu einem möglichen deterministischen Multiplexgewinn oben schon nahelegen, führt eine auf der deterministischen effektiven Bandbreite (3.23) basierende Verbindungsannahme dazu, daß die Verbindungsannahmegrenzkurve einen konvexen zulässigen Bereich einschließt, zumindest dann wenn man den diskreten Charakter der Annahmeentscheidung für eine natürliche Anzahl von Verbindungen außer Acht läßt.

Die Verbindungen in Tab. 3.1 und dementsprechend die Abb. 3.5-3.8 vernachlässigen den Einfluß der Segmentierung des Datenstroms in Pakete, der in der Realität in die Berechnung des Ressourcenbedarfs einzubeziehen ist. Zur Berücksichtigung von Paketlängen $L_i > 0$ reicht die Verkehrsankunftsfunktion (3.14). Man sollte darin auf jeden Fall die Spitzenrate p_i auf Paketebene spezifizieren, also nach Möglichkeit nicht die Kapazität des Übertragungsabschnittes als Spitzenrate deklarieren. Im Falle von RPQ oder RPQ+ kann $L_i > 0$ auch implizit in die Verkehrsankunftsfunktion der Form (3.10) einfließen. Solange innerhalb eines Rotationsintervalls Δ garantiert nicht mehr Daten ankommen, als aufgrund der Verkehrsankunftsfunktion zu erwarten sind, können die zugesagten maximalen Verzögerungszeiten und die Verlustfreiheit eingehalten werden. p_i muß also gegebenenfalls intern so geändert werden, daß $p_i \Delta L_i^{-1} \in \mathbb{N}$ ist. Dann ist $p_i \Delta$ tatsächlich eine obere Grenze für die in Intervallen der Länge Δ ankommende Datenmenge. Hier wird dann auch deutlich, daß es nicht sinnvoll ist, Verzögerungsprioritäten mit beliebig feiner Granularität anzubieten.

Verzichtet man darauf, EDF, RPQ und RPQ+ mit einem verbindungsbezogenen *Shaper* zu kombinieren, so vergrößert sich die vom Bediensystem verursachte Verzögerung und muß als *Latency* zur Ende-zu-Ende-Verzögerungszeit addiert werden. Will man aufgrund der Kostenverteilung unter den Knoten entlang des Pfades vor allem hinsichtlich der Bandbreiteausbeute einigermaßen effiziente Aggregate erhalten, liegt es nahe, vor dem Aggregat einen beträchtlichen Teil der Ende-zu-Ende zulässigen *Latency* zu verbrauchen, also relativ große maximale Knotenverzögerungszeiten $D^S_{i,max}$ zuzulassen.

Leider sieht zumindest die *Guaranteed Service* Spezifikation außer der konstanten Komponente D_j^{GS} als von der reservierten Rate R_i abhängige Komponente der *Latency* nur den Term $C_j^{GS} R_i^{-1}$vor. Das sind also die einzigen beiden Parameter, die ein Knoten j zur Spezifikation seiner *Latency* exportieren kann. Übernimmt man die vom Empfänger reservierte Rate R_i auch intern im Aggregationsknoten, die im Idealfall der deterministischen effektiven Bandbreite (3.23) entspricht, so kann man

aus (3.23) die gesuchte *Latency* $D^S_{i,max}$ errechnen, die gleich dem vom Verkehrsdeskriptor mit den Parametern p_i, r_i und b_i abhängigen Teil in Formel (3.13) ist, wenn man auch dort zur Vereinfachung $L=0$ setzt. Wenn das Aggregat einen Großteil der Strecke zwischen Sender und Empfänger ausmacht und im Aggregat selbst die Verzögerung gering ist, mag diese Größenordnung gerechtfertigt sein. Ansonsten sollte man von der direkten Übernahme von R_i für die interne Zuordnung einer Verzögerungspriorität absehen. Immerhin kann der angesprochene, vom Verkehrsdeskriptor abhängige Teil von (3.13) auf die Form $C^{GS}_j R_i^{-1}+D^{GS}_j$ gebracht werden. Wenn allerdings der Empfänger den Verkehrsdeskriptor in der *Resv*-Nachricht ändern sollte, müßte der Aggregationsknoten eine Korrekturrechnung vornehmen.

3.1.6 Die deterministische Methode vor dem Hintergrund statistischer Methoden

Es ist offensichtlich, daß die Reservierung von Ressourcen für einen Netzdienst, der für Pakete einer Verbindung im Rahmen der vereinbarten Verkehrsparameter Verlustfreiheit und deterministische obere Grenzen für Paketverzögerung Ende-zu-Ende garantiert, konservativ sein muß und daher für sich allein betrachtet die vorhandenen Ressourcen ineffizient auslastet. Da jedoch andererseits das Volumen von Verbindungen, die derart hohe Anforderungen an den Netzdienst stellen, klein ist, die Verbindungsentgelte dagegen recht hoch sein werden und außerdem die ungenutzten Kapazitäten zum Teil *Best Effort* Verkehr zugute kommen, wird der Gesichtspunkt einer optimalen Auslastung auch innerhalb der hervorgehobenen Dienstklasse möglicherweise gegenüber anderen, höher zu gewichtenden technischen und wirtschaftlichen Erwägungen zurückstehen.

Eine Verkehrsklasse, die obere Grenzen für die Paketverzögerung nur statistisch garantiert, so daß mit einer Wahrscheinlichkeit größer null die vereinbarte maximale Verzögerungszeit überschritten wird, könnte die reservierte Bandbreite effizienter nutzen. Ein Lösungsansatz wird für EDF in [182] aufgezeigt. Dazu wird zur Modellierung der Bedieneinheit ein Ersatzbediensystem zum Zeitpunkt $t=0$ betrachtet, das im Intervall $[-D^S_{c,max},0)$ alle Pakete der Verzögerungspriorität *c* verwirft. Diese Pakete müßten bei EDF nämlich erst nach $t=0$ bedient werden. Aus der Belegung des Ersatzbediensystems zur Zeit $t=0$ kann die gesuchte Überschreitungswahrscheinlichkeit berechnet werden, allerdings nicht aufgeschlüsselt nach Verzögerungsprioritäten. Unter der Annahme *Leaky Bucket* konformer Quellen, die mit einer zufällig verteilten und von den anderen Quellen unabhängigen Phasenlage periodisch senden und pausieren, lösen die Autoren dieses System näherungsweise unter Anwendung des Konzept der virtuellen Wartezeit von Beneš. Da Quellen der Verzögerungspriorität

c nach $t=-D^S_{c,max}$ nicht mehr im Ersatzbediensystem berücksichtigt werden dürfen, erweist sich die Berechnung des Gesamtankunftsprozesses aus den Einzelbeiträgen der Quellen als äußerst schwierig. Die Komplexität des Verfahrens wächst exponentiell mit der Anzahl der Quelltypen und stellt die praktische Bedeutung des Verfahrens in Frage.

3.2 Statistisches Multiplexen

Die Verkehrssteuerung eines paketvermittelnden Netzes kann statistisches Multiplexen zur besseren Auslastung der Ressourcen zulassen, also sozusagen auf die statistischen zeitlichen Schwankungen der Quellen spekulieren und weit weniger Ressourcen reservieren als es der Summe der Spitzenraten der Ströme entspricht, sofern ein gewisses Maß an Paketverlusten im Rahmen des Verkehrsvertrages vereinbart worden ist. Allerdings macht statistisches Multiplexen die Berechnung des Ressourcenbedarfs eines Aggregates aus Verkehrsströmen mit im allgemeinen sehr variablen statistischen Eigenschaften zu einer äußerst komplexen Aufgabe. Der Einzelbeitrag einer Quelle zum Ressourcenbedarf ist nämlich nicht nur von deren individuellen Eigenschaften bestimmt, sondern hängt auch ab von der aktuellen Zusammensetzung des Aggregates und vom Verhältnis, mit dem Wartespeicher und Bandbreite bereitgestellt werden. In der wissenschaftliche Literatur, namentlich von Kelly [114] und dort referenzierten Autoren, ist mit dem Konzept einer effektiven Bandbreite ein Maß für den Einzelbeitrag einer Quelle zum Gesamtressourcenbedarf entwickelt worden, das von diesen schwierigen Zusammenhängen abstrahiert. Ziel solcher Betrachtungen ist es, Verkehrsströme in paketvermittelnden Netzen ähnlich wie Verbindungen in leitungsvermittelten Netzen zu modellieren und damit zahlreiche Aufgaben, beispielsweise dynamisches Bandbreitemanagement, Routing, Entgelterhebung und Netzplanung, mit vertrauten Methoden lösen zu können.

Wirklich nützlich ist diese Abstraktion aber nur, solange sich die aktuellen Randbedingungen, unter denen die effektive Bandbreite, der Einzelbetrag einer Verbindung, berechnet worden ist, nicht allzu sehr ändern. Wenn sich dagegen die effektive Bandbreite aufgrund der Dynamik eines Aggregates häufig ändert, verliert dieses abstrakte Konzept an Bedeutung. Dann muß der Gesamtbedarf an Ressourcen immer wieder mit meist relativ komplexen Modellen neu berechnet werden.

Auf der *Large Deviation* Theorie beruhende Verfahren stellen all diese komplexen Aspekte der Berechnung des Ressourcenbedarfs dar. Sie führen auf einfache Formeln für die effektive Bandbreite, zeigen deren Abhängigkeit von der Zusammensetzung des Aggregates, vermitteln Einsichten zur Effizienz von Aggregation und zeigen den Einfluß beliebiger Strategien der Bereitstellung von Bandbreite und Wartespeicher. Darüber hinaus erfaßt die *Large Deviation* Theorie deterministi-

sches Multiplexen sowie statistisches Multiplexen mit und ohne Wartespeicher als Sonderfälle eines allgemeinen theoretischen Konzeptes und stellt insofern die für das Verständnis der Zusammenhänge so wichtigen Querbezüge her.

3.2.1 Grundlagen der *Large Deviation* Theorie

Die *Large Deviation* Theorie ist eine nützliche Methode zur Schätzung der Wahrscheinlichkeiten seltener Ereignisse. In der Verkehrstheorie werden ihre Ergebnisse vor allem zur Abschätzung von Überschreitungswahrscheinlichkeiten von Summen von Zufallsvariablen eingesetzt. In der Tat kann die *Large Deviation* Theorie ein wichtiges Hilfsmittel bei der Berechnung des Ressourcenbedarfs von Quellen unter Ausnutzung statistischen Multiplexens sein. Dabei ist die Kapazität einer Bedieneinheit (und eventuell der Wartespeicher) so zu dimensionieren, daß bei Überlagerung einer größeren Anzahl von als Zufallsprozesse beschreibbaren Quellen eine vorgegebene maximale Verlustwahrscheinlichkeit (die durch die Überschreitungswahrscheinlichkeit abgeschätzt werden kann) nicht überschritten wird. Die Randbedingungen, welche die Anwendung der *Large Deviation* Theorie nahelegen, nämlich eine größere Anzahl von zu addierenden Zufallsvariablen, die Notwendigkeit, kleine Überschreitungswahrscheinlichkeiten zu berechnen und nicht zuletzt die Forderung, daß die Dimensionierungsaufgabe schnell, d. h. mit beherrschbarem numerischen Aufwand, gelöst werden sollte, sind also gegeben.

3.2.1.1 Elementare Ungleichungen

Einem der zentralen Ergebnisse der *Large Deviation* Theorie, der Tschernoff-Schranke, liegen einige elementare und sehr einfach zu zeigende Ungleichungen zur Abschätzung von Überschreitungswahrscheinlichkeiten zugrunde. Da sie dem Gesamtverständnis der nachfolgenden Abschnitte sehr dienlich sind, werden sie an dieser Stelle in gegenüber [23] gekürzter Form zusammengestellt.

Für eine nicht negative Zufallsvariable X mit beliebiger Wahrscheinlichkeitsverteilungsfunktion $F_X(x)$ und positivem c_0 gelten die Beziehungen

$$\mathrm{E}\{X\} = \int_0^\infty \xi \,\mathrm{d}F_X(\xi) \geq \int_{c_0}^\infty \xi \,\mathrm{d}F_X(\xi) \geq c_0 \int_{c_0}^\infty \mathrm{d}F_X(\xi) = c_0 \,\mathrm{P}\{X \geq c_0\} \tag{3.24}$$

Ersetzt man in (3.24) X und c_0 durch $|X|^k$ und ${c_0}^k$, so erhält man die Ungleichung von Markoff

$$\mathrm{P}\{|X| \geq c_0\} \leq \frac{1}{{c_0}^k} \mathrm{E}\{|X|^k\} \tag{3.25}$$

aus der für $k=2$ und $X := X - \mathrm{E}\{X\}$ die wohlbekannte Ungleichung von Tschebyscheff

$$\mathrm{P}\{|X-\mathrm{E}\{X\}|\geq c_0\}\leq\frac{1}{c_0^{\,2}}\mathrm{Var}\{|X|\} \tag{3.26}$$

folgt. In [23] werden weitere Ungleichungen für Erwartungswerte von Funktionen von Zufallsvariablen hergeleitet. Dazu zählt die Ungleichung von Jensen

$$\phi(\mathrm{E}\{X\})\leq\mathrm{E}\{\phi(X)\} \tag{3.27}$$

für konvexe Funktionen $\phi(x)$, mit deren Hilfe auch die Ungleichung von Hölder

$$\mathrm{E}\{|X_1 X_2|\}\leq\mathrm{E}^{1/c_1}\{|X_1|^{c_1}\}\cdot\mathrm{E}^{1/c_2}\{|X_2|^{c_2}\} \tag{3.28}$$

gezeigt werden kann für zwei Zufallsvariablen X_1 und X_2 mit zwei reellen Zahlen $c_1>1, c_2>1$, für die $c_1^{-1}+c_2^{-1}=1$ gilt. Als Spezialfall folgt aus (3.28) noch die Ungleichung von Ljapunoff

$$\mathrm{E}^{1/c_3}\{|X|^{c_3}\}\leq\mathrm{E}^{1/c_4}\{|X|^{c_4}\} \tag{3.29}$$

mit $0<c_3\leq c_4$.

3.2.1.2 Momentenerzeugende Funktion

Von herausragender Bedeutung für die Theorie ist die momentenerzeugende Funktion

$$M_X(s)=\mathrm{E}\{\mathrm{e}^{sX}\} \tag{3.30}$$

$s\in\mathbb{R}$, die eine Reihe von interessanten Eigenschaften besitzt. Insbesondere lassen sich durch Ableitung von (3.30) alle gewöhnlichen Momente der Zufallsvariable X bestimmen:

$$\frac{\mathrm{d}^k}{\mathrm{d}\,s^k}M_X(s)\,|_{s=0}\;=\mathrm{E}\{X^k\} \tag{3.31}$$

Für die momentenerzeugende Funktion einer Summe von unabhängigen Zufallsvariablen $\sum X_i$ gilt

$$M_{\sum X_i}(s)=\prod M_i(s) \tag{3.32}$$

Ferner kann mit der Ungleichung von Hölder gezeigt werden [23], daß nicht nur $M_X(s)$, sondern auch $\ln M_X(s)$ konvexe Funktionen sind.

3.2.1.3 Tschernoff-Schranke und *Large Deviation Rate Function*

Unter Verwendung der Ungleichung von Markoff (3.25) für $k=1$ kann mit Hilfe der momentenerzeugenden Funktion die Wahrscheinlichkeit, daß eine Zufallsvariable X einen Wert c_0 ($>\mathrm{E}\{X\}$) überschreitet, durch die Ungleichung

$$\mathrm{P}\{X \geq c_0\} = \mathrm{P}\{\mathrm{e}^{sX} \geq \mathrm{e}^{sc_0}\} \leq \mathrm{e}^{-sc_0} M_X(s) \tag{3.33}$$

abgeschätzt werden. Diese Ungleichung gilt für alle *s*, also insbesondere für

$$\begin{aligned} \mathrm{P}\{X \geq c_0\} &\leq \inf_s \{\mathrm{e}^{-sc_0} M_X(s)\} \\ &= \exp(\inf_s \{\ln \mathrm{E}\{\mathrm{e}^{sX}\} - sc_0\}) = \exp(-\sup_s \{sc_0 - \ln \mathrm{E}\{\mathrm{e}^{sX}\}\}) = \mathrm{e}^{-I(c_0)} \end{aligned} \tag{3.34}$$

Dies ist die eingangs erwähnte Tschernoff-Schranke. Die auch *Large Deviation Rate Function* genannte Funktion

$$I(c_0) = \sup_s \{sc_0 - \ln \mathrm{E}\{\mathrm{e}^{sX}\}\} \tag{3.35}$$

ist konvex und nimmt ihr Minimum 0 an der Stelle $c_0 = \mathrm{E}\{X\}$ an [37]. Gewöhnlich wird wie in [132] statt einer Zufallsvariable X der empirische Mittelwert einer Serie von immer gleichen und voneinander unabhängig ausgeführten Zufallsexperimenten X betrachtet und die Überschreitungswahrscheinlichkeit des empirischen Mittelwertes $\sum_{i=1}^{n} X_i / n$ untersucht. Substituiert man dazu in (3.34) $X := \sum_{i=1}^{n} X_i$ und $c_0 := nc_0$, so ergibt sich nach einigen Umformungen mit (3.32)

$$\ln \mathrm{P}\{\frac{1}{n}\sum_{i=1}^{n} X_i \geq c_0\} \leq -n \sup_s \{sc_0 - \ln \mathrm{E}\{\mathrm{e}^{sX_1}\}\} = -n I_1(c_0) \tag{3.36}$$

Zwar ist bei kleinem *n* die Steigung der Überschreitungswahrscheinlichkeit in Abhängigkeit von *n* im logarithmischen Maßstab noch deutlich kleiner als $-I_1(c_0)$ (von bei diskreten Zufallsvariablen auftretenden Sprünge nach oben und unten sowieso abgesehen), mit wachsendem *n* nähert sie sich dieser Abschätzung jedoch immer mehr an und stimmt für $n \to \infty$ schließlich überein (Theorem von Cramer [37], Theorem von Tschernoff [23]). Mit wachsendem *n* nimmt dann die Überschreitungswahrscheinlichkeit exponentiell exakt mit der durch die *Large Deviation Rate Function* berechneten Rate weiter ab.

3.2.1.4 Theorem von Gärtner und Ellis

Rückt man von der Annahme voneinander unabhängiger und identischer Zufallsvariablen ab, ist die Zerlegung (3.32) nicht mehr möglich und man erhält statt (3.36)

$$\ln \mathrm{P}\{\frac{1}{n}\sum_{i=1}^{n} X_i \geq c_0\} \leq -n \sup_s \{sc_0 - \ln \mathrm{E}\{\mathrm{e}^{s\frac{1}{n}\sum_{i=1}^{n} X_i}\}\} = -n I(c_0) \tag{3.37}$$

Auf der Basis dieser Ungleichung etablieren Gärtner und Ellis folgenden, gegenüber dem Theorem von Cramer verallgemeinerten Satz:

Satz (Theorem von Gärtner und Ellis) [37]:

Es existiere $\lim_{n\to\infty} \ln \mathrm{E}\{\mathrm{e}^{sn^{-1}\sum_{i=1}^{n} X_i}\}$ für alle $s\in\mathbb{R}$ (muß nicht endlich sein). Dann ist $\{s:\lim_{n\to\infty} \ln \mathrm{E}\{\mathrm{e}^{sn^{-1}\sum_{i=1}^{n} X_i}\}<\infty\}$ eine konvexe Menge und $\lim_{n\to\infty} \ln \mathrm{E}\{\mathrm{e}^{sn^{-1}\sum_{i=1}^{n} X_i}\}$ eine konvexe Funktion von s. Ferner sei $[c_0,c_1]\cap\{x: I_G(x)<\infty\}\neq\emptyset$. Dann gilt

$$\limsup_{n\to\infty} \frac{1}{n} \ln P\{\frac{1}{n}\sum_{i=1}^{n} X_i \in [c_0,c_1]\} \;\leq\; -\inf_{x\in[c_0,c_1]} I_G(x) \tag{3.38}$$

mit der *Large Deviation Rate Function*

$$I_G(x)=\sup_s\{sc_0-\lim_{n\to\infty}\frac{1}{n}\ln \mathrm{E}\{\mathrm{e}^{s\sum_{i=1}^{n} X_i}\}\} \tag{3.39}$$

Wenn außerdem die Funktion $\lim_{n\to\infty} \ln \mathrm{E}\{\mathrm{e}^{sn^{-1}\sum_{i=1}^{n} X_i}\}$ in $D_{I_G}=\{s:\lim_{n\to\infty} \ln \mathrm{E}\{\mathrm{e}^{sn^{-1}\sum_{i=1}^{n} X_i}\}<\infty\}$ differenzierbar ist und $(c_0,c_1)\subset\{\frac{\mathrm{d}}{\mathrm{d}s}\lim_{n\to\infty} \ln \mathrm{E}\{\mathrm{e}^{sn^{-1}\sum_{i=1}^{n} X_i}\}: s\in D_{I_G}\}$, gilt

$$\liminf_{n\to\infty} \frac{1}{n} \ln \mathrm{P}\{\frac{1}{n}\sum_{i=1}^{n} X_i \in (c_0,c_1)\} \;\geq\; -\inf_{x\in(c_0,c_1)} I_G(x) \tag{3.40}$$

Betrachtet man statt (c_0,c_1) wie bisher auch $(c_0,+\infty)$, liegt c_0 im Innern von D_{I_G} und ist die konvexe Funktion $I_G(x)$ streng monoton steigend, so daß $\inf_{x\in(c_0,\infty)} I_G(x)=I_G(c_0)$, folgt aus (3.38) und (3.40) [170]

$$\lim_{n\to\infty}\frac{1}{n}\ln \mathrm{P}\{\frac{1}{n}\sum_{i=1}^{n} X_i > c_0\}=-I_G(c_0) \tag{3.41}$$

Das Theorem von Cramer kann in einer zu (3.38) und (3.40) analogen Form formuliert werden. Da jedoch meist Überschreitungswahrscheinlichkeiten über eine Schranke $c_0>\mathrm{E}\{X_1\}$ hinaus interessieren, das konvexe $I_1(x)$ von seinem Minimum an der Stelle $x=\mathrm{E}\{X_1\}$ aus aber streng monoton ansteigt [37], gilt (3.41).

3.2.1.5 Verbesserung der Approximation mit der *Probability Shift* Methode

In der Literatur wird häufig zur Berechnung von Überschreitungswahrscheinlichkeiten für die Summe von unabhängigen Zufallsvariablen die *Probability Shift* Methode verwendet, sei es zur Herleitung [170] der gegenüber der Tschernoff-Schranke (in weiten Bereichen) verbesserten Rao-Baradur-Abschätzung [37] oder zur Approximation der Verteilungsdichtefunktion einer Summe von

Zufallsvariablen wie etwa in [182, 193]. Diese Methode verdankt ihren Namen der Transformation der Wahrscheinlichkeitsverteilungsfunktion F_X der Zufallsvariable X,

$$\mathrm{d}\,F_{X,s}(x)=\frac{\mathrm{e}^{sx}}{\mathrm{E}\{\mathrm{e}^{sX}\}}\mathrm{d}\,F_X(x) \tag{3.42}$$

die für alle $s\in\{s:\ln \mathrm{E}\{\mathrm{e}^{sX}\}<\infty\}$ definiert ist. Durch die Transformation verschieben sich abhängig von s der Mittelwert und die Varianz, und es gelten die Zusammenhänge

$$\mathrm{E}_s\{X\}=\int_{-\infty}^{+\infty}\xi\,\mathrm{d}\,F_{x,s}(\xi)=\frac{\mathrm{E}\{X\,\mathrm{e}^{sX}\}}{\mathrm{E}\{\mathrm{e}^{sX}\}}=\frac{\mathrm{d}}{\mathrm{d}\,s}\ln \mathrm{E}\{\mathrm{e}^{sX}\} \tag{3.43}$$

$$\mathrm{Var}_s\{X\}=\int_{-\infty}^{+\infty}(\xi-\mathrm{E}_s\{X\})^2\mathrm{d}\,F_{x,s}(\xi)=\frac{\mathrm{E}\{X^2\mathrm{e}^{sX}\}}{\mathrm{E}\{\mathrm{e}^{sX}\}}-\frac{\mathrm{E}^2\{X\,\mathrm{e}^{sX}\}}{\mathrm{E}^2\{\mathrm{e}^{sX}\}}=\frac{\mathrm{d}^2}{\mathrm{d}\,s^2}\ln \mathrm{E}\{\mathrm{e}^{sX}\} \tag{3.44}$$

Wenn die Zufallsvariable X also nicht konstant ist, d. h. $\mathrm{Var}_s\{X\}>0$, dann ist $\mathrm{E}_s\{X\}$, die Stammfunktion, streng monoton wachsend im Definitionsbereich von *s*. In diesem Fall kann eine Abbildung $s(x)$ definiert werden, die jedem Wert *x*, für den die Wahrscheinlichkeiten $\mathrm{P}\{X>x\}$ und $\mathrm{P}\{X<x\}$ positiv sind, ein *s* eineindeutig so zuordnet, daß der Erwartungswert der Zufallsvariablen X nach der Transformation nach *x* verschoben worden ist: $\mathrm{E}_s\{X\}=x$. Wenn *s* also so gewählt wird, daß

$$x=\frac{\mathrm{d}}{\mathrm{d}\,s}\ln \mathrm{E}\{\mathrm{e}^{sX}\}=\mathrm{E}_s\{X\} \tag{3.45}$$

dann wird auch gleichzeitig der Exponent in der Tschernoff-Schranke (Gleichung (3.34) mit *x* statt c_0) optimiert. Da

$$s(x)=\frac{\mathrm{d}}{\mathrm{d}\,x}(s(x)\,x-\ln \mathrm{E}\{\mathrm{e}^{s(x)X}\}) \tag{3.46}$$

auch und insbesondere dann gilt, wenn *s* als von *x* abhängige Funktion $s(x)$ betrachtet wird (wie sich durch Ableitung von (3.46) mit Produkt- und Kettenregel leicht zeigen läßt), kann das gesuchte $s(x)$ leicht aus Gleichung (3.46) gewonnen werden. Statt (3.46) kann man auch

$$s(x)=\frac{\mathrm{d}}{\mathrm{d}\,x}\inf_s\{\ln \mathrm{E}\{\mathrm{e}^{sX}\}-sx\} \tag{3.47}$$

schreiben. In dem Ansatz

$$\mathrm{P}\{X\geq x\}=\int_x^\infty \mathrm{d}\,F_X(\xi)=\int_0^\infty \exp(\ln \mathrm{E}\{\mathrm{e}^{s(x)X}\}-s(x)\xi)\,1\{\xi\geq x\}\,\mathrm{d}\,F_{X,s(x)}(\xi) \tag{3.48}$$

zur Berechnung der Überschreitungswahrscheinlichkeit kann man nun in einfacher Weise die bekannte Tschernoff-Schranke (3.34) isolieren und erhält

$$\mathrm{P}\{X \geq x\} = \exp(\inf_{s} \{\ln \mathrm{E}\{\mathrm{e}^{sX}\} - sx\}) \, \mathrm{E}_{s(x)}\{\mathrm{e}^{-s(x)(X-x)} 1\{X \geq x\}\} \tag{3.49}$$

Die Zufallsvariable X ist nach der Transformation (3.42) um den neuen, gewünschten Erwartungswert $\mathrm{E}_{s(x)}\{X\} = x$ verteilt. Deshalb kann der neben der Tschernoff-Schranke in (3.49) verbliebene Erwartungswert

$$\mathrm{E}_{s(x)}\{\mathrm{e}^{-s(x)(X-x)} 1\{X \geq x\}\} = \int_{x}^{\infty} \mathrm{e}^{-s(x)(\xi - x)} \, \mathrm{d}F_{X,s(x)}(\xi) = \int_{0}^{\infty} \mathrm{e}^{-s(x)\xi} \, \mathrm{d}F_{X,s(x)}(\xi + x) \tag{3.50}$$

näherungsweise mit der Wahrscheinlichkeitsdichte der Normalverteilung berechnet werden, wenn X entweder die Summe einer sehr großen Anzahl von beliebigen unabhängigen Zufallsvariablen oder die Summe von Zufallsvariablen mit Verteilungsfunktionen ist, die in der Nähe ihres Mittelwertes ohnehin gut durch die Normalverteilung angenähert werden können (beispielsweise weil sie selbst aus der Summation einer größeren Zahl von Zufallsvariablen hervorgegangen sind). Also gilt:

$$\int_{0}^{\infty} \mathrm{e}^{-s(x)\xi} \, \mathrm{d}F_{X,s(x)}(\xi + x) \approx \frac{1}{\sqrt{2\pi \, \mathrm{Var}_{s(x)}\{X\}}} \int_{0}^{\infty} \exp(-s(x)\xi - \frac{1}{2} \frac{\xi^2}{\mathrm{Var}_{s(x)}\{X\}}) \, \mathrm{d}\xi \tag{3.51}$$

Für das Integral auf der rechten Seite von Gleichung (3.51) ist eine Näherung bekannt [170], die auf die eingangs erwähnte Rao-Baradur-Abschätzung

$$\mathrm{P}\{X \geq x\} \approx \frac{1}{\sqrt{2\pi \, \mathrm{Var}_{s(x)}\{X\}}} \; \frac{1}{s(x)} \; \exp(\inf_{s} \{\ln \mathrm{E}\{\mathrm{e}^{sX}\} - sx\}) \tag{3.52}$$

führt.

3.2.2 *Large Deviation Theorie* und *Network Calculus*

Chang beschäftigt sich in [40] mit der Frage, wie das Konzept des *Network Calculus* von Cruz [49] für deterministischen Grenzen unterworfene Ankunftsprozesse auf stochastische Ankunftsprozesse übertragen werden kann. Er entwickelt dazu eine zu der deterministischen Beschreibung von Ankunftsprozessen im *Network Calculus* analoge Beschreibung für stochastische Ankunftsprozesse. (In [40] werden sowohl die deterministische als auch die stochastische Beschreibung als *Envelope Process* bezeichnet.) Damit kann er die Überlagerung von Ankunftsprozessen beschreiben, den Zusammenhang zwischen den Prozessen am Eingang und Ausgang von Bediensystemen herleiten und schließlich die Stabilität komplexer Netzmodelle mit Routing untersuchen.

Nach Chang ist eine minimale deterministische Verkehrsankunftsfunktion $A^*(t)$ (*Minimum Envelope Process*), also eine Funktion, die für alle t kleiner als alle anderen möglichen Verkehrsankunftsfunktionen $A(t)$ ist, gegeben durch die Funktion

$$A^*(t) = \sup_{t_0 \geq 0} \{ A(t_0, t_0+t) \} \tag{3.53}$$

$A(t_1, t_2) = \sum_{t=t_1}^{t_2-1} a(t)$, die Zufallsvariable der im Intervall $[t_1, t_2)$ ankommende Datenmenge, setzt sich zusammen aus einer Folge von zu diskreten Zeitpunkten gemessenen zufälligen Zuwächsen $\{a(t), t = 0,1,2,...\}$. Da $A^*(t)$ wie alle Verkehrsankunftsfunktionen monoton wachsend und aufgrund von (3.53) subadditiv ist, existiert der von Chang als *Minimum Envelope Rate* bezeichnete Grenzwert $a^* = \lim_{t\to\infty} A^*(t) / t$, und es gilt

$$a^* = \lim_{t\to\infty} \frac{A^*(t)}{t} = \inf_{t>0} \frac{A^*(t)}{t} \tag{3.54}$$

Chang weist weiter nach, daß eine deterministische Obergrenze für die Verzögerung durch das Bediensystem mit konstanter Bedienrate C existiert, wenn $a^* < C$, und umgekehrt, wenn $a^* > C$, die Verzögerung unbegrenzt ist.

Für stochastische Prozesse definiert Chang einen *Minimum Envelope Process* bezüglich s mit

$$A^*(s,t) = \sup_{t_0 \geq 0} \{ \frac{1}{s} \ln \mathrm{E} \{ \mathrm{e}^{sA(t_0, t_0+t)} \} \} \tag{3.55}$$

$0 < s < \infty$, der für deterministische $A(t_1, t_2)$ mit (3.53) übereinstimmt. Daraus gewinnt er unter den drei Voraussetzungen, daß die Folge $\{a(t), t \geq 0\}$ stationär und ergodisch ist, der Grenzwert $a^*(s) = \lim_{t\to\infty} A^*(s,t) / t$ existiert und $sa^*(s)$ konvex und differenzierbar ist für alle $0 < s < \infty$, die *Minimum Envelope Rate* $a^*(s)$ mit einer (3.54) vergleichbaren Aussagekraft. Sowohl $a^*(s)$ als auch $sa^*(s)$ sind streng monoton wachsend. All diese Eigenschaften genügen auch, um die Verbindung zur *Large Deviation* Theorie herzustellen: Für $n := t$ und $X_n := a(t)$ gilt das Theorem von Gärtner und Ellis (3.41) mit der *Large Deviation Rate Function* $I_G(c_0) = \inf_{s>c_0} \{ sc_0 - sa^*(s) \}$. Wenn $\{a(t), t = 0,1,2,...\}$ außerdem eine Folge von voneinander unabhängigen Zufallsvariablen ist, ist $A^*(s,t)$ sogar subadditiv und erfüllt eine zu (3.54) vollkommen analoge Beziehung.

Es läßt sich sehr leicht zeigen, daß für eine Überlagerung von N voneinander unabhängigen Ankunftsprozessen die Ungleichungen $A^*(s,t) \leq \sum_{i=1}^{N} A_i^*(s,t)$ bzw. $a^*(s) \leq \sum_{i=1}^{N} a_i^*(s)$ gelten und sogar $a^*(s) = \sum_{i=1}^{N} a_i^*(s)$, wenn die $a_i^*(s)$ die o. g. Voraussetzungen erfüllen.

Ausgehend von einem auf Lindley zurückgehenden klassischen Ansatz zur Bestimmung der Verteilung des Füllstandes der Warteschlange $Q(t)$ in einem G/G/1-Bediensystem [118]

$$Q(t)=\max\{0, a(t-1)-C, a(t-1)+a(t-2)-2C, \ldots, a(0)-tC\} \tag{3.56}$$

(wenn zum Zeitpunkt $t=0$ mit einem leeren System gestartet worden ist) kann Chang den Zusammenhang zwischen dem Ankunftsprozeß und dem Füllstand des Wartespeichers abschätzen. Er erhält mit (3.55) die Ungleichung

$$\mathrm{E}\{\mathrm{e}^{sQ(t)}\} \le \sum_{t_0=0}^{t} \mathrm{E}\{\mathrm{e}^{s(A(t-t_0,t)-t_0C)}\} \le \sum_{t_0=0}^{t} \mathrm{e}^{s(A^*(s,t_0)-t_0C)} \tag{3.57}$$

Da sich die Überschreitungswahrscheinlichkeit $\mathrm{P}\{Q(t)>c_0\}$ unter Einbeziehung von (3.57) durch die Tschernoff-Schranke, vgl. (3.33) abschätzen läßt, kann nach dem Grenzübergang $t\to\infty$ und mit (3.54) als Stabilitätskriterium, also als Grenze, bis zu der der die Überschreitungswahrscheinlichkeit mit wachsendem c_0 exponentiell abnimmt, $a^*(s)<C$ etabliert werden.

Solange die Bedienrate *C* größer als die mittlere Ankunftsrate und die Überschreitungswahrscheinlichkeit $\mathrm{P}\{Q(t)>c_0\}$ keinen Grenzen unterworfen ist, kann Chang den Arbeitspunkt *s* seines Systems so wählen, daß sein Stabilitätskriterium $a^*(s)<C$ erfüllt ist.

3.2.3 Approximation der Verteilungsfunktion des Pufferfüllstandes mit der *Large Deviation Theorie*

In der Literatur, z. B. in [54], findet man den Zusammenhang (3.56) zwischen der Restarbeit $W(t)$ im Bediensystem und dem Füllstand der Warteschlange $Q(t)$ häufig in der Form

$$Q(t)=\sup_{t\ge 0}\{A(-t,0)-Ct\}=\sup_{t\ge 0}\{W(t)\} \tag{3.58}$$

Für diskrete Zeitpunkte *t* ergibt sich Changs *Minimum Envelope Process* $A^*(s,t)$ (3.55) ähnlich wie in (3.57) dann aus der Überlegung

$$\mathrm{P}\{Q(t)>c_0\}=\mathrm{P}\{\sup_{t\ge 0}\{W(t)\}>c_0\}=\mathrm{P}\{\bigcup_{t}\{W(t)>c_0\}\}\le\sum_{t}\mathrm{P}\{W(t)>c_0\} \tag{3.59}$$

Da die Wahrscheinlichkeit, daß das unter allen Zeitpunkten *t* größte $W(t)$ einen Wert c_0 überschreitet, größer ist als die Wahrscheinlichkeit, daß dies zu einem bestimmten, dem wahrscheinlichsten Zeitpunkt *t* passiert, kann für die Überschreitungswahrscheinlichkeit auch die untere Grenze

$$\mathrm{P}\{\sup_{t\ge 0}\{W(t)\}>c_0\}\ge\sup_{t\ge 0}\mathrm{P}\{W(t)>c_0\} \tag{3.60}$$

angegeben werden. Wenn $\mathrm{P}\{W(t)>c_0\}$ hinreichend schnell, z. B. exponentiell, mit wachsendem c_0 abfällt, dominiert der größte Summand in $\sum_t \mathrm{P}\{W(t)>c_0\}$, und es gilt näherungsweise [54]

$$\mathrm{P}\{Q(t)>c_0\}=\mathrm{P}\{\sup_{t\geq 0}\{W(t)\}>c_0\}\simeq\sup_{t\geq 0}\mathrm{P}\{W(t)>c_0\} \tag{3.61}$$

3.2.3.1 *Large Buffer Asymptotic* Modell

Dies kann mit der Tschernoff-Schranke (3.34) kombiniert werden zu

$$\mathrm{P}\{Q(t)>c_0\}\simeq\sup_{t\geq 0}\{\inf_{s\geq 0}\{\mathrm{e}^{-sc_0}\mathrm{E}\{\mathrm{e}^{sW(t)}\}\}\}=\sup_{t\geq 0}\{\inf_{s\geq 0}\{\mathrm{e}^{-sc_0}\mathrm{E}\{\mathrm{e}^{s(A(-t,0)-tC)}\}\}\} \tag{3.62}$$

Mit Changs Abschätzung (3.57) wird daraus die obere Schranke

$$\mathrm{P}\{Q(t)>c_0\}\leq\sup_{t\geq 0}\{\inf_{s\geq 0}\{\mathrm{e}^{-sc_0}\mathrm{e}^{s(A^*(s,t)-tC)}\}\} \tag{3.63}$$

oder – wenn $c_0\to\infty$ und $\lim_{c_0\to\infty} 1/c_0(A^*(s,t)-tC)=0$, also nur der Wartespeicher, nicht aber die Anzahl der Quellen und die Bedienrate erhöht werden (*Large Buffer Asymptotic* [46]) –

$$\lim_{c_0\to\infty}\frac{1}{c_0}\ln\mathrm{P}\{Q(t)>c_0\}\leq -s^*=-\sup\{s:a^*(s)<C\}=-\sup\{s:\lim_{t\to\infty}\frac{1}{t}\ln\mathrm{E}\{\mathrm{e}^{sW(t)}\}<0\} \tag{3.64}$$

Das größte s, das noch die Stabilitätsbedingung $a^*(s)<C$ erfüllt, liefert die mit dieser Methode bestmögliche obere Grenze. Eine untere Grenze für $\mathrm{P}\{W(t)>c_0\}$ kann durch Anwendung des Theorems von Gärtner und Ellis (3.41) auf $A(-t,0)$ hergeleitet werden. Unter den oben erwähnten drei Voraussetzungen läßt sich mit diesen beiden Grenzen das *Large Deviation* Prinzip

$$\lim_{c_0\to\infty}\frac{1}{c_0}\ln\mathrm{P}\{Q(t)>c_0\}=-s^* \tag{3.65}$$

für den Zusammenhang zwischen dem Füllstand des Wartespeichers $Q(t)$ des Bediensystems und der Restarbeit $W(t)$ beweisen [40]. Alternativ zu (3.64) kann s^* auch mit

$$s^*=\sup\{s:\lim_{t\to\infty}\frac{1}{t}\ln\mathrm{E}\{\mathrm{e}^{sW(t)}\}<0\}=\inf_{t\geq 0}t\sup_s\{\frac{1}{t}s-\lim_{t\to\infty}\frac{1}{t}\ln\mathrm{E}\{e^{sW(t)}\}\} \tag{3.66}$$

bestimmt werden [30].

Das durch die Gleichungen (3.65) und (3.66) beschriebene *Large Deviation* Prinzip kann auf beliebige stabile Bediensysteme ($a^*(s)<C$) und Ankunftsprozesse mit stationären Zuwächsen angewendet werden, vorausgesetzt diese sind nicht langzeitkorreliert [54]. Dagegen strebt die Autokorrelationsfunktion

$$\gamma_a(t) = \frac{1}{\mathrm{Var}\{a(t)\}} \mathrm{E}\{(a(t_0)-\mathrm{E}\{a(t)\})(a(t_0+t)-\mathrm{E}\{a(t)\})\} \tag{3.67}$$

langzeitkorrelierter Ankunftsprozesse mit stationären Zuwächsen $a(t)$ nur sehr langsam gegen null. Es gilt mit $0<\beta<1$

$$\lim_{t\to\infty} \gamma_a(t) \sim t^{-\beta} \tag{3.68}$$

Zwischen der Autokorrelationsfunktion und der in der Taylorreihenentwicklung von $sA^*(s,t)$ um $s=0$

$$sA^*(s,t)=\mathrm{E}\{\sum_{\tau=t_0}^{t_0+t-1} a(\tau)\}s+\frac{1}{2}\mathrm{Var}\{\sum_{\tau=t_0}^{t_0+t-1} a(\tau)\}s^2+... \tag{3.69}$$

auftretenden Varianz $\mathrm{Var}\{\sum_{\tau=t_0}^{t_0+t-1} a(\tau)\}$ der Summe einer Folge von Zuwächsen des Ankunftsprozesses gilt für stationäre Prozesse aber der mit einigen vergleichsweise einfachen Umformungen herzuleitende Zusammenhang [19]

$$\mathrm{Var}\{\sum_{\tau=t_0}^{t_0+t-1} a(\tau)\}=\mathrm{Var}\{a(t)\}(1+2\sum_{\tau=1}^{t-1}(1-\frac{\tau}{t})\gamma_a(\tau))t \tag{3.70}$$

so daß

$$\lim_{t\to\infty} \mathrm{Var}\{\sum_{\tau=t_0}^{t_0+t-1} a(\tau)\} \sim t^{2-\beta} \tag{3.71}$$

und der Grenzwert $a^*(s)=\lim_{t\to\infty} A^*(s,t)/t$ nicht existiert. Somit kann das durch die Gleichungen (3.65) und (3.66) beschriebene *Large Deviation* Prinzip nicht aufrechterhalten werden.

Unter einer Reihe von Bedingungen und unter Zuhilfenahme von Skalierungsfunktionen haben Duffield und O´Connell ausgehend vom Theorem von Gärtner und Ellis [37] ein verallgemeinertes *Large Deviation* Prinzip für die Restarbeit $W(t)$ und den Füllstand des Wartespeichers $Q(t)$ gefunden [54]. Da dieses Ergebnis zwar für die Weiterentwicklung der *Large Deviation* Theorie von Bedeutung ist, jedoch nicht so sehr für die Praxis, soll an dieser Stelle auf eine zusammenfassende Darstellung verzichtet werden.

3.2.3.2 *Many Sources Asymptotic* Modell

In [46] und [30] wird nämlich statt des Gleichung (3.65) zugrundeliegenden Modells eines Bediensystems, in dem jeder Verbindung praktisch unendlich viel Wartespeicher bereitgestellt werden kann (*Large Buffer Asymptotic)*, ein Modell behandelt, in dem Wartespeicher *NS*, Bedienrate *NC*

und Anzahl der Quellen N gleichzeitig mit einem gemeinsamen Faktor skaliert werden (*Many Sources Asymptotic* [46], *Large Multiplexer* [30]). Dieses Modell ist sicherlich auch für dynamische Aggregate das geeignetere. Schließlich müssen ja die dem Aggregat zugeordnete Rate und der Wartespeicher mit der Anzahl der Verbindungen in irgendeiner Form gekoppelt werden, d. h. die insgesamt zur Verfügung stehenden Ressourcen entsprechend der Größe der Aggregate verteilt werden.

Wenn $W_N(t)$ die Restarbeit in dem mit dem Faktor N skalierten Bediensystem bezeichnet, läßt sich das Theorem von Gärtner und Ellis für dieses Modell in der Form

$$\limsup_{N\to\infty} \frac{1}{N} \ln \mathrm{P}\{\frac{W_N(t)}{N} \in [c_0, c_1]\} \;\le\; -\inf_{x\in[c_0,c_1]} I_G(x) \tag{3.72}$$

$$\liminf_{N\to\infty} \frac{1}{N} \ln \mathrm{P}\{\frac{W_N(t)}{N} \in (c_0, c_1)\} \;\ge\; -\inf_{x\in(c_0,c_1)} I_G(x) \tag{3.73}$$

schreiben. Die *Large Deviation Rate Function* in diesen Ungleichungen ist

$$I_G(x) = \sup_s \{ s\,x - \lim_{N\to\infty} \frac{1}{N} \ln \mathrm{E}\{\mathrm{e}^{sW_N(t)}\}\} \tag{3.74}$$

Diese Beziehungen gelten unter der Bedingung, daß die Grenzwerte $\lim_{N\to\infty}(N\,t)^{-1}\ln \mathrm{E}\{\mathrm{e}^{sW_N(t)}\}$ und $\lim_{t\to\infty}\lim_{N\to\infty}(N\,t)^{-1}\ln \mathrm{E}\{\mathrm{e}^{sW_N(t)}\}$ existieren, im Innern des Gebietes, in denen sie endlich sind, $D_{I_G} = \{ s : \lim_{t\to\infty}\lim_{N\to\infty}(N\,t)^{-1}\ln \mathrm{E}\{\mathrm{e}^{sW_N(t)}\} < \infty\}$, differenzierbar in s sind und außerdem der Betrag der Ableitung für jede Folge s_i, die für $i\to\infty$ gegen einen Punkt auf dem Rande des Gebietes konvergiert, gegen unendlich strebt. Ferner muß es ein s geben, so daß die Stabilitätsbedingung $\lim_{N\to\infty}(N\,t)^{-1}\ln \mathrm{E}\{\mathrm{e}^{sW_N(t)}\} < 0$ für alle ausreichend großen t erfüllt ist.

Von Interesse ist natürlich vor allem die Abschätzung von Überschreitungswahrscheinlichkeiten, also Intervallen der Form $(c_0, +\infty)$. Wenn c_0 größer ist als das $c_0{}^*$, das $I_G(c_0{}^*) = 0$ löst, wird aufgrund der allgemeinen Eigenschaften und der Stabilitätsbedingung $\inf_{x\in(c_0,\infty)} I_G(x) = I_G(c_0)$ am linken Rande des betrachteten Intervalls angenommen.

Letztlich kann das durch (3.72)-(3.74) gegebene *Large Deviation* Prinzip für die Restarbeit $W_N(t)$ wieder auf der Basis der oberen Schranke (3.59) zu einer Abschätzung des Füllstandes der Warteschlange $Q_N(t)$ verknüpft werden [30]:

$$\limsup_{N\to\infty} \frac{1}{N} \ln \mathrm{P}\{Q_N(t) > NS\} \;\le\; -\inf_{t>0} I_G(S) \tag{3.75}$$

$$\liminf_{N\to\infty} \frac{1}{N} \ln \mathrm{P}\{Q_N(t) > NS\} \;\geq\; -\inf_{t>0} I_G(S^+) \tag{3.76}$$

mit der *Large Deviation Rate Function* $I_G(S)$ aus (3.74). (3.75) ist übrigens auch für $S\to\infty$ eine gegenüber dem *Large Buffer Asymptotic* (3.64) verbesserte obere Schranke, wenn die Zuwächse des Ankunftsprozesses stationär und assoziiert sind, also ihre Autokorrelationsfunktion positiv ist. In Bediensystemen ohne Puffer, $S=0$, wird das Infimum von $\inf_{t>0} I_G(0)$ bei $t=0$ (bzw. in zeitdiskreten Modellen bei $t=1$) angenommen [30], wenn die Zuwächse stationär sind. Die Autokorrelation spielt in diesem speziellen Falle keine Rolle.

Die Herleitung der oberen Schranke für das *Many Sources Asymptotic* beruht genauso wie die oben kurz erwähnte Erweiterung des *Large Buffer Asymptotic* für langzeitkorrelierte Ankunftsprozesse nicht mehr auf der kritischen Abschätzung (3.61). Statt dessen wird die Summe $\sum_t \mathrm{P}\{W(t) > c_0\}$ aus (3.59) in eine Teilsumme mit den Summanden bis zu $t\leq t_0$ und in eine Teilsumme mit den Summanden für $t>t_0$ zerlegt. Im Falle des *Many Sources Asymptotic* gewährleisten die Bedingungen, unter denen (3.72)-(3.74) und folglich (3.75) gelten, daß zum einen der Beitrag der zweiten Teilsumme nach dem Grenzübergang $N\to\infty$ gegen null strebt (insbesondere die Bedingung, daß $\lim_{N\to\infty} N\, t^{-1} \ln \mathrm{E}\{\mathrm{e}^{sW_N(t)}\} < 0$ für alle ausreichend großen t, beispielsweise für $t>t_0$) und zum anderen die erste Teilsumme nach weiteren elementaren Abschätzungen (Tschernoff-Schranke (3.34)) und Umformungen auf die *Large Deviation Rate Function* $I_G(S)$ aus (3.74) führt [46, 30].

Obgleich langzeitkorrelierte Ankunftsprozesse diese Bedingungen nicht erfüllen, halten die Beziehungen (3.75) und (3.76) bei ebenfalls unveränderter *Large Deviation Rate Function* $I_G(S)$ aus (3.74) der notwendigen Verallgemeinerung stand [55, 56].

Eine solche Verallgemeinerung auf langzeitkorrelierte Ankunftsprozesse ist möglich unter der auch von langzeitkorrelierten Ankunftsprozessen erfüllbaren Bedingung, daß die Grenzwerte $\lim_{N\to\infty} (N\, s_v(t))^{-1} \ln \mathrm{E}\{\mathrm{e}^{sW_N(t) s_v(t)/s_a(t)}\}$ und $\lim_{t\to\infty} \lim_{N\to\infty} (N\, s_v(t))^{-1} \ln \mathrm{E}\{\mathrm{e}^{sW_N(t) s_v(t)/s_a(t)}\}$ existieren, im Innern des Gebietes, in denen sie endlich sind, $D_{I_G} = \{s : \lim_{t\to\infty} \lim_{N\to\infty} (N\, s_v(t))^{-1} \ln \mathrm{E}\{e^{sW_N(t) s_v(t)/s_a(t)}\} < \infty\}$, differenzierbar in s sind, und außerdem der Betrag der Ableitung für jede Folge s_i, die für $i\to\infty$ gegen einen Punkt auf dem Rande des Gebietes konvergiert, gegen unendlich strebt. Die Funktionen $s_a(t)$ und $s_v(t)$ sind sogenannte Skalierungsfunktionen, also wachsende Funktionen, welche die positiven Halbachsen aufeinander abbilden. Sie müssen abhängig vom Modell des Ankunftsprozesses so gewählt werden, daß alle notwendigen Bedingungen für die Anwendung des *Many Sources Asymptotic* erfüllt sind

[54, 55, 56]. Die ursprünglich [30] für das *Many Sources Asymptotic* angegebenen Voraussetzungen beziehen sich auf den Spezialfall $s_a(t)=s_v(t)=t$. Zu den bereits genannten Bedingungen kommen zwei weitere hinzu, mit deren Hilfe die Summe $\sum_t \mathrm{P}\{W(t)>c_0\}$ in (3.59) erneut so in Teilsummen zerlegt und abgeschätzt werden kann, daß die Beziehungen (3.74)-(3.76) aufrecht erhalten werden können. Zum einen muß wiederum ein s existieren, so daß für alle hinreichend großen t der Grenzwert $\lim_{N\to\infty}(N\,s_v(t))^{-1}\ln \mathrm{E}\{e^{sW_N(t)s_v(t)/s_a(t)}\}<0$ ist. Zum anderen muß für alle $\epsilon<0$ der Grenzwert $\lim_{t_0\to\infty}\limsup_{N\to\infty} N^{-1}\ln\sum_{t\geq t_0} e^{\epsilon N s_v(t)}=-\infty$ sein.

Das Ergebnis, daß (3.72)-(3.76) auch für langzeitkorrelierte Prozesse gültig bleiben [55, 56], erlaubt es, die in [135] unter einschränkenden Voraussetzungen bewiesene und in [47] verwendete Rao-Baradur-Verfeinerung weiter zur Verbesserung der Abschätzung der Überschreitungswahrscheinlichkeit einzusetzen, und zwar in der Form

$$\mathrm{P}\{Q_N(t)>NS\}\approx \frac{1}{\sqrt{2\pi\,\mathrm{Var}_{s^*}\{A_N(-t^*,0)\}}}\;\frac{1}{s^*}\;\mathrm{e}^{-N\inf_t\sup_s\{s(Ct+S)-\frac{1}{N}\ln \mathrm{E}\{\mathrm{e}^{sA_N(-t,0)}\}\}} \tag{3.77}$$

wobei die Parameter s^* und t^* aus der Operation $\inf_t \sup_s\{s(Ct+S)-N^{-1}\ln \mathrm{E}\{e^{sA_N(-t,0)}\}\}$ hervorgehen.

Eine noch präzisere Vorhersage der Dienstgüte erhält man, wenn man es wie in [135] nicht bei der Abschätzung der Überschreitungswahrscheinlichkeit beläßt, sondern ausgehend vom Quotienten aus mittlerer Verlustrate und Angebot

$$P_{Loss}=\frac{\mathrm{E}\{\max\{0,Q_N(t-1)+a_N(t)-N(C\cdot 1+S)\}\}}{\mathrm{E}\{a_N(t)\}} \tag{3.78}$$

die Verlustwahrscheinlichkeit mit Hilfe des *Many Sources Asymptotic* in einem zeitdiskreten System abschätzt. Man sollte dabei jedoch nicht außer acht lassen, daß eine wirklich präzise und dennoch konservative Abschätzung mit dem *Large Deviation Asymptotic* nur bei einer hinreichend großen Anzahl von Quellen möglich ist.

3.2.4 Verfahren zur Berechnung der effektiven Bandbreite auf der Basis der *Large Deviation* Theorie

Aufgrund der Eigenschaften des in Formel (3.55) definierten *Minimum Envelope Process* $A^*(s,t)$ und seiner Rolle bei der Abschätzung der Überschreitungswahrscheinlichkeit $\mathrm{P}\{W(t)>b\}$ beispielsweise in (3.63) leitet Kelly [114] mit dem Ausdruck

$$a(s,t)=\frac{1}{st}\ln \mathrm{E}\left\{e^{sA(0,t)}\right\} \tag{3.79}$$

$(0<s,t<\infty)$ eine Definition für die effektive Bandbreite ab. Aus den bereits von Chang beobachteten Eigenschaften kann gefolgert werden, daß $a(s,t)$ immer zwischen der mittleren und maximalen in einem Intervall der Länge t gemessenen Rate des Ankunftsprozesses liegt. Werden voneinander unabhängige Ankunftsprozesse überlagert, so kann die effektive Bandbreite des resultierenden Gesamtankunftsprozesse als Summe der effektiven Bandbreiten der einzelnen Prozesse berechnet werden. Deshalb ist mit Formel (3.79) tatsächlich die effektive Bandbreite einer Einzelverbindung definiert. Abhängig von dem der Berechnung der effektiven Bandbreite zugrundeliegenden stochastischen Modell, also abhängig von der Anzahl und den stochastischen Eigenschaften der Ankunftsprozesse und abhängig von der Zuteilung von Wartespeicher und Bedienkapazität können die Parameter s und t ganz unterschiedliche Werte annehmen. Mitunter kann die Abhängigkeit von t ganz verschwinden, z. B. im Falle von von Prozessen mit unabhängigen und stationären Zuwächsen $a(\tau)$. Dann gilt nämlich mit $A(0,t)=\sum_{\tau=0}^{t} a(\tau)$

$$a(s,t)=\frac{1}{st}\ln \mathrm{E}\left\{e^{s\sum_{\tau=0}^{t} a(\tau)}\right\}=\frac{1}{st}\ln \prod_{\tau=0}^{t} \mathrm{E}\left\{e^{sa(\tau)}\right\}=\frac{1}{st}\ln \mathrm{E}^{t}\left\{e^{sa(t)}\right\}=\frac{1}{s}\ln \mathrm{E}\left\{e^{sa(t)}\right\} \tag{3.80}$$

mit dem zeitunabhängigen Erwartungswert $\mathrm{E}\{s\,a(t)\}$.

Aus Sicht des Autors ist Kellys Definition nicht ganz befriedigend, da sie weder die wechselseitige Abhängigkeit von s und t, noch deren Abhängigkeit von der Dimensionierung von Wartespeicher und Bedienkapazität darstellt.

Auf der anderen Seite begegnet uns die effektive Bandbreite nach (3.79) in Form des Logarithmus der momentenerzeugenden Funktion, so daß ihre qualitativen Eigenschaften für die Verbindungsannahmesteuerung bzw. für die Berechnung des Ressourcenbedarfs eines Aggregates bedeutsam sind. Darüber hinaus entspricht bei der Überlagerung von Verkehr in einem Bediensystem ohne Wartespeicher ($t=1$) die Summe der effektiven Bandbreite der Verkehrsquellen nach (3.79) der Rate NC, die auf der Basis des *Many Sources Asymptotic* (3.72)-(3.76) zur Begrenzung der Überschreitungswahrscheinlichkeit auf einen vorgegebenen Schwellwert notwendig ist. Außerdem wird in den meisten Fällen eine einzelne Verbindung die aktuellen Parameter s und t eines größeren Aggregats nicht allzu sehr beeinflussen, wie in [47] an ausgewählten Beispielen aufgezeigt wird. Aus diesem Grunde läßt die effektive Bandbreite nach (3.79) zumindest eine gute Abschätzung der Ressourcen zu, die nach dem Hinzufügen oder Entfernen einer Verbindung zur Sicherstellung der Dienstgüte der Verbindungen im Aggregat mehr oder weniger benötigt werden, was in der Praxis von großer Bedeutung sein kann.

3.2.4.1 Verfahren von Elwalid et al.

Wie eingangs ausgeführt, beschränkt sich diese Arbeit auf die Bewertung von auf der *Large Deviation* Theorie beruhenden Verfahren zur Berechnung des Ressourcenbedarfs der Überlagerung von stochastischen Ankunftsprozessen der Verbindungen eines Aggregates. Wegen ihrer insbesondere gegenüber dem *Many Sources Asymptotic* geringeren Komplexität sind mit Blick auf den realisierten Prototypen eines Aggregationsknotens zunächst die Verfahren interessant, welche die Tschernoff-Schranke (3.34) allein oder in Verbindung mit der Rao-Baradur-Abschätzung (3.52) zur Berechnung der benötigten Bedienkapazität einsetzen. An sich ist diese auf [97] zurückgehende Vorgehensweise zwar nur auf Bediensystemmodelle ohne Wartespeicher anwendbar. Elwalid et al. [59] haben jedoch speziell für *Leaky Bucket* konforme Ein-Aus-Quellen einen heuristischen Ansatz vorgestellt, mit dem ein Bediensystem, in dem Wartespeicher proportional zur Bandbreite zugeteilt wird, in ein System ohne Puffer übergeführt werden kann.

Ausgangspunkt dieses Verfahrens ist die Hypothese, daß sich Verkehrsströme, die zu einem *Leaky Bucket* konform sind und sich in zufällig verteilter Phasenlage zueinander befinden, so verhalten, als ob sie zunächst einzelne, vollkommen voneinander entkoppelte Bediensysteme durchlaufen und anschließend konform zu dadurch modifizierten *Leaky Bucket* Parametern auf ein Bediensystem ohne Wartespeicher treffen würden. Die einzelnen, virtuellen Bediensysteme verfügen, so die Hypothese, über Wartespeicher und Bedienkapazität entsprechend dem festen Verhältnis S_A/C_A von Wartespeicher und Bedienkapazität des Aggregates in dem Umfange, wie für die verlustfreie Bedienung der *Leaky Bucket* konformen Verbindung notwendig ist. Dazu muß jede virtuellen Bedieneinheit i den Verkehrsstrom der ihr zugeordneten Verbindung mit einer von der Spitzenrate p_i, mittleren Rate r_i und Büscheltoleranz b_i des *Leaky Bucket* Verkehrsdeskriptor abhängigen Rate

$$e_i = \frac{p_i}{1 + S_A C_A^{-1} b_i^{-1} (p_i - r_i)} \tag{3.81}$$

bedienen, sofern $r_i \leq S_A^{-1} C_A b_i$ ist, bzw. $e_i = r_i$, falls diese Bedingung nicht erfüllt ist [59].

Diese deterministische Betrachtung voneinander unabhängiger virtueller Bediensysteme ist nun aber in mehrfacher Hinsicht sehr konservativ, vgl. dazu auch [85]. Sie schöpft nicht einmal den im Zusammenhang der Verbindungsannahmesteuerung mit der deterministischen Methode angesprochenen deterministischen Multiplexgewinn aus, der bei Betrachtung des Bediensystems als Ganzes möglich wäre. Darüber hinaus schätzen die Autoren von [59] die Belegung des Wartespeichers in den virtuellen Bediensystemen so ab, als ob die Belegung des Wartespeichers während der Aktivitätsphasen einer Ein-Aus-Quelle durchgehend auf dem maximal möglichen Niveau verharren

würde, um das Problem des statistischen Multiplexens von Wartespeicher und Bedienkapazität auf ein Ein-Ressourcenproblem in einem Bediensystemmodell ohne Wartespeicher zurückführen zu können. Diese Vereinfachungen spiegeln die komplexen Zusammenhänge des statistischen Multiplexens der Ressourcen Wartespeicher und Bedienkapazität natürlich nicht so präzise wider wie das *Many Sources Asymptotic* (3.72)-(3.76). Entsprechend wird die Belegung des Wartespeichers zum Teil sehr deutlich überschätzt.

Dafür ist dieses konservative Verfahren nicht sehr aufwendig und nicht zuletzt auch deswegen sehr attraktiv, da es sich sehr leicht auf Verbindungsaggregate anwenden läßt [140]. Die Tschernoff-Schranke (3.34) liefert die für die Bestimmung der effektiven Bandbreite notwendige Ungleichung

$$\ln \mathrm{P}\{\sum_i X_i \geq C_A\} \leq -\sup_{s\geq 0}\{s C_A - \sum_{j=1}^{N_c} N_j \ln M_j(s)\} \leq \ln \epsilon_O \tag{3.82}$$

in der $M_j(s)$ die momentenerzeugende Funktion der als Zufallsgröße modellierten (und durch die virtuellen Bediensysteme modifizierte) Momentanrate einer Verbindung des Typs j ist, d. h.

$$M_j(s) = 1 - w_j + w_j \mathrm{e}^{s e_i} \tag{3.83}$$

wenn $w_j = r_j / e_j$ die Aktivität der Verbindungen des Typs j am Ausgang der virtuellen Bediensysteme ist. N_j ist die Anzahl der Verbindungen dieses Typs im Aggregat und ϵ_O das in Form einer Überschreitungswahrscheinlichkeit angegebene mindestens zu erfüllende Dienstgütekriterium. C_A, die Bandbreite des Aggregats, kann und soll abhängig von der Zusammensetzung des Aggregates nun so gewählt werden, daß die Maximumstelle der Funktion $s C_A - \sum_{j=1}^{N_c} N_j \ln \mathrm{M}_j(s)$ und deren Schnittpunkt mit der Schranke $\ln \epsilon_O$ identisch sind. Die dazu notwendige Bedingung

$$C_A = C_A(s) = \sum_{j=1}^{N_c} N_j \frac{\frac{\mathrm{d}}{\mathrm{d}s} M_j(s)}{M_j(s)} \tag{3.84}$$

kann in (3.82) eingesetzt werden. Mit der Verfeinerung von Rao-Baradur (3.52) erhält man schließlich die Gleichung

$$\begin{aligned} & s \sum_{j=1}^{N_c} N_j \frac{\frac{\mathrm{d}}{\mathrm{d}s} M_j(s)}{M_j(s)} - \sum_{j=1}^{N_c} N_j \ln M_j(s) \\ & + \frac{1}{2} \ln 2\pi s^2 \sum_{j=1}^{N_c} N_j \left(\frac{\frac{\mathrm{d}^2}{\mathrm{d}s^2} M_j(s)}{M_j(s)} - \left(\frac{\frac{\mathrm{d}}{\mathrm{d}s} M_j(s)}{M_j(s)} \right)^2 \right) = \ln \frac{1}{\epsilon_O} \end{aligned} \tag{3.85}$$

wenn man die Varianz $\mathrm{Var}\{\sum_{j=1}^{N_c} N_j X_j\}$ als zweite Ableitung der Funktion $\ln M_j(s)$ nach s berechnet. Die Lösung s^* dieser Gleichung kann als Arbeitspunkt des statistischen Multiplexens interpretiert werden, der immer wieder neu berechnet werden muß, wenn eine neue Verbindung hinzugefügt oder entfernt wird. Mit wachsender Zahl von Verbindungen werden s^* und damit verbunden der Bandbreitebedarf pro Quelle, die effektive Bandbreite, kleiner. Dies zeigen Abb. 3.9 und Abb. 3.10 am Beispiel der Verbindungstypen 1 und 6 aus Tabelle 4.5 in Kapitel 4.

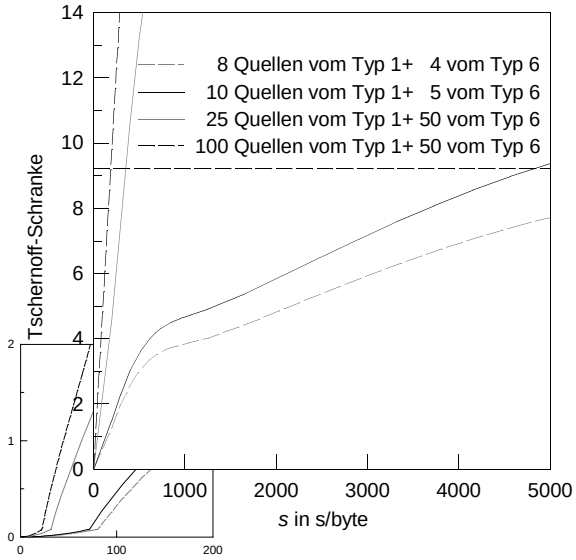

Abb. 3.9: *Lösung von Gleichung (3.85). Die untere Kurve hat keinen Schnittpunkt mit der Schranke* $\ln \epsilon_O^{-1}$*: Das Aggregat erreicht nicht die kritische Größe. Die Vergrößerung zeigt den Wegfall des Rao-Baradur-Terms.*

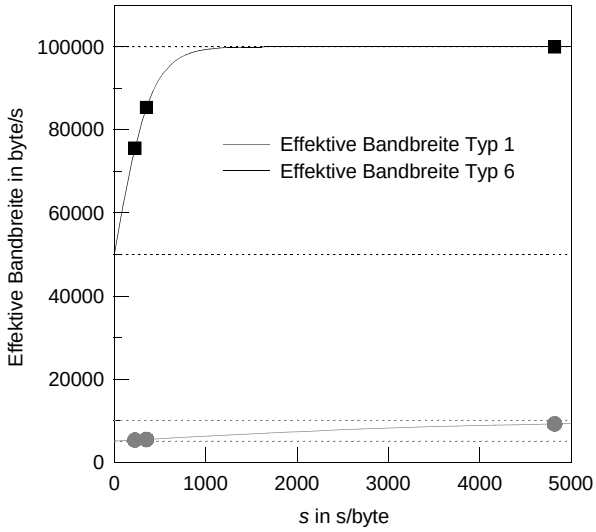

Abb. 3.10: *Die effektive Bandbreite in Abhängigkeit von s. Alle Werte liegen in* $[r_i, e_i]$*. Die Lösungen s*, die Schnittpunkte der Tschernoff-Schranke mit* $\ln \epsilon_O^{-1}$ *in Abb. 3.9, sind hervorgehoben.*

3.2.4.2 Verfahren von Elwalid et al. in der Praxis

Wenn nur wenige Verbindungen das Aggregat bilden, ist (3.85) unter Umständen nicht lösbar, denn für $s \to \infty$ strebt diese Gleichung gegen

$$\sum_{j=1}^{N_c} N_j \ln \frac{1}{w_j} = \ln \frac{1}{\epsilon_O} \tag{3.86}$$

Aufgrund der Eigenschaften der linken Seite in Gleichung (3.85), die im Zusammenhang mit der Bewertung der Aggregation noch näher diskutiert werden, ist (3.85) nur dann lösbar, wenn $\sum_{j=1}^{N_c} N_j \ln w_j^{-1} > \ln \epsilon_O^{-1}$. Statistisches Multiplexen mit einem Arbeitspunkt $s^* < \infty$ setzt also erst bei einer bestimmten kritischen Anzahl von Verbindungen ein. Solange diese nicht erreicht ist, muß

eine aus einer deterministischen Betrachtung resultierende Rate C_A für das Aggregat gewählt werden, also z. B. $\sum_{j=1}^{N_c} N_j e_j$.

In der Praxis verläuft der Übergang vom deterministischen zum statistischen Multiplexen nicht ganz problemlos. Denn im Gegensatz zu Aggregaten, deren Arbeitspunkt $s*$ statistisches Multiplexen zuläßt, ist bei Aggregaten, bei denen eine neu hinzukommende Verbindung überhaupt erst statistisches Multiplexen ermöglicht, noch nicht der gute Startwert zur Lösung von Gleichung (3.85) mit dem Newton-Algorithmus bekannt, der das $s*$ vor der Veränderung des Aggregates nun einmal ist [140]. Es stellt sich heraus, vgl. dazu auch die Untersuchungen im nächsten Kapitel, daß nicht nur viele Iterationen, sondern auch zusätzliche Maßnahmen zur Sicherstellung der Konvergenz des Lösungsverfahrens notwendig sind, um die Berechnung des Ressourcenbedarfs robust gestalten zu können. Ist der Übergang zum statistischen Multiplexen aber erst einmal erfolgt, genügt wie im pufferlosen Modell von Hui [97] meist eine Iteration, um ausgehend vom Arbeitspunkt $s*$ vor der Veränderung des Aggregates die neue Rate mit hinreichender Genauigkeit bestimmen zu können.

Da es sich bei der Tschernoff-Schranke (3.82) um eine konservative obere Schranke für die Überschreitungswahrscheinlichkeit handelt, sollte der zur Verbesserung dieser Abschätzung in Gleichung (3.85) ergänzte, auf Rao-Baradur zurückgehende Term eigentlich nur dann berücksichtigt werden, wenn er einen positiven Beitrag leistet. Den daraus resultierenden Kurvenverlauf zeigt die Ausschnittsvergrößerung von Abb. 3.9 für kleine s. Mit dieser Vorgehensweise geht jedoch der Verlust der Eigenschaft der Differenzierbarkeit der linken Seite von (3.85) einher, was unter Umständen zu Problemen bei der Lösung dieser Gleichung mit dem Newton-Algorithmus führen könnte. Bisher konnte der Autor solche Probleme in der Praxis jedoch nicht beobachten.

3.2.4.3 Berechnung auf der Grundlage des *Many Sources Asymptotic*

Zur Berechnung des Ressourcenbedarfs eines Aggregates mit dem *Many Sources Asymptotic* muß primär die *Large Deviation Rate Function* aus (3.72)-(3.77) bestimmt werden, also das Optimierungsproblem

$$N\, I_G(C_A, S_A) = \inf_{t \geq 0} \sup_{s \geq 0} \left\{ s(C_A t + S_A) - \ln \mathrm{E}\left\{ \mathrm{e}^{s A_N(-t,0)} \right\} \right\} \tag{3.87}$$

gelöst werden. In (3.87) ersetzten $C_A = NC$ und $S_A = NS$ die mit dem Faktor N zu skalierenden Ressourcen aus (3.74) bzw. (3.77). Zur Betonung der Abhängigkeit der *Large Deviation Rate Function* $I_G(S)$ aus (3.74) von den Ressourcen C_A und S_A wird die Schreibweise $I_G(C_A, S_A)$ gewählt. Dies ist auch als Hinweis darauf zu verstehen, daß im Gegensatz zur Verbindungsannahmesteuerung für einen Übertragungsabschnitt fester Kapazität die Rate C_A und der Wartespeicher S_A am

Eingang des Aggregates nach Möglichkeit auch noch so zu wählen sind, daß die Ressourcen gerade so die mit (3.77) abgeschätzte Überschreitungswahrscheinlichkeit auf das erforderliche Maß begrenzen. Dazu müssen nicht nur wie bei dem auf der Verbindungsannahmesteuerung von Elwalid et al. [59] beruhenden Verfahren optimale Lösungen für C_A und *s* bestimmt werden, sondern auch noch S_A und *t*. Nur wenn die Reservierung von Bandbreite C_A und Wartespeicher S_A gekoppelt werden, etwa über einen konstanten Faktor oder eine Kostenfunktion, ist das System überhaupt lösbar.

Leider ist Gleichung (3.87) auch dann nur in Ausnahmen geschlossen lösbar. Zu diesen Ausnahmen gehören als *Fractional Brownian Motion* modellierte Ankunftsprozesse

$$A(0,t)=m\,t+Z(t) \tag{3.88}$$

In (3.88) bezeichnet *m* die mittlere Rate und $Z(t)$ einen normalverteilten Zufallsprozeß, für den $\mathrm{E}\{Z(t)\}=0$ und $\mathrm{Var}\{Z(t)\}=\sigma^2 t^{2H}$ mit dem Hurst-Parameter $H\in(0,1)$ und der Konstante σ^2 ist. Der Logarithmus der momentenerzeugenden Funktion eines derart modellierten Prozesses läßt sich nämlich als eine nach dem zweiten Summanden abbrechende und leicht bezüglich *t* und *s* differenzierbare Taylorreihe darstellen [114, 57],

$$\ln \mathrm{E}\{\mathrm{e}^{s\,A(0,t)}\}=m\,t\,s+\frac{1}{2}\sigma^2 t^{2H}\,s^2 \tag{3.89}$$

weil für normalverteilte Zufallsvariablen *X*, deren Erwartungswert $\mathrm{E}\{X\}=0$ beträgt, auch $\mathrm{E}\{X^n\}=0$ für ungerade *n* folgt [160].

Im ungünstigsten Fall ist aber nicht einmal ein Modell des Ankunftsprozesses bekannt, so daß erst einmal die Verteilungsfunktion von $A_N(-t,0)$ gemessen werden muß. Immerhin kann selbst dann für jedes feste *t* das optimale *s* (näherungsweise) numerisch berechnet werden. Dazu nutzen die Autoren von [47] die allgemeine Eigenschaft der Konvexität der momentenerzeugenden Funktion aus. Dagegen ist die Abhängigkeit der momentenerzeugenden Funktion von *t* so komplex, daß die Suche nach *t**, der Infimumstelle von (3.87), auf ein mehr oder weniger genaues Absuchen des gesamten erlaubten Bereiches hinausläuft. Für periodische Quellen kann der abzusuchende Bereich allerdings eingeschränkt werden. Siris beweist in [180], daß für periodische Quellen konstanter Rate das *t*, das (3.87) minimiert und die *Large Deviation Rate Function* des *Many Buffer Asymptotic* bestimmt, *t**, im Intervall $[0,T_P)$ zu finden ist. Es müssen also nur Werte für *t* bis zur Periodendauer T_P überprüft werden. Auch ein Aggregat aus periodischen Ein-Aus-Quellen erreicht das Infimum in (3.87) in $[0,T_P)$, weil (analog zum Beweis in [180]) für alle $t_2=t_1-T_P\geq 0$ gilt:

$$
\begin{aligned}
& s(C_A t_2 + S_A) - \ln \mathrm{E}\left\{ e^{s A_N(-t_2, 0)} \right\} \\
&= s(C_A t_1 + S_A) - s C_A T_P - \ln \mathrm{E}\left\{ e^{s A_N(-t_1, 0)} \right\} + \ln \mathrm{E}\left\{ e^{s A_N(-t_1, -t_2)} \right\} \\
&= s(C_A t_1 + S_A) - s C_A T_P - \ln \mathrm{E}\left\{ e^{s A_N(-t_1, 0)} \right\} + \ln \mathrm{E}\left\{ e^{s A_N(-T_P, 0)} \right\} \\
&= s(C_A t_1 + S_A) - s C_A T_P - \ln \mathrm{E}\left\{ e^{s A_N(-t_1, 0)} \right\} + s T_P m_A \\
&\le s(C_A t_1 + S_A) - \ln \mathrm{E}\left\{ e^{s A_N(-t_1, 0)} \right\}
\end{aligned}
\tag{3.90}
$$

Die Ungleichung $C_A T_P \geq T_P m_A$, die in (3.90) verwendet wird, folgt aus der Stabilitätsbedingung, daß die mittlere Rate m_A des Aggregates kleiner als die reservierte Rate C_A sein muß. T_P ist nun aber die Periodendauer des Aggregates, das kleinste gemeinsame Vielfache der Periodendauern der einzelnen Verbindungen. Um aus (3.90) überhaupt Vorteile für die Suche nach t^* ziehen zu können, müssen die Periodendauern folglich diskretisiert werden.

Danach kann mit Hilfe des im Anhang angegebenen mathematischen Ausdrucks für die momentenerzeugende Funktion solcher Quellen die *Large Deviation Rate Function* (3.87) näherungsweise berechnet und die Bandbreite C_A so bestimmt werden, daß die obere Schranke ϵ_O für die Überschreitungswahrscheinlichkeit eingehalten wird. Um die Effizienz der heuristischen Methode von Elwalid et al. [59] mit den diversen auf dem *Many Sources Asymptotic* basierenden Verfahren zu vergleichen, zeigt Abb. 3.11 den Verlauf der mittleren effektiven Bandbreite pro Quelle abhängig von der Größe des Aggregates am Beispiel der Quellen aus Tab. 3.2. Die Verfahren sind natürlich um so präziser, je genauer sie die tatsächliche Überschreitungswahrscheinlichkeit erfassen. Abb. 3.12 zeigt die Überschreitungswahrscheinlichkeit in einem Aggregat konstanter Größe, wenn die reservierte Rate C_A und proportional dazu der Wartespeicher S_A variiert werden. Die für Abb. 3.11, 3.13 und 3.14 durchgeführten Berechnungen des *Many Sources Asymptotic* nähern sich von der Ausgangsrate, der aufgrund der Methode von Elwalid et al. [59] zu erwartenden Rate, in sich halbierenden Schritten nach und nach der Rate, die aufgrund der jeweiligen Formel für die Überschreitungswahrscheinlichkeit benötigt wird. Der zulässige Bereich für t wird für jedes C_A durch äquidistante Stützstellen im Abstand 0,02 untersucht. Die Suche wird abgebrochen, wenn der Logarithmus der prognostizierten Überschreitungswahrscheinlichkeit nur noch um 20% oder weniger über der Schranke $\ln \epsilon_O$ liegt, oder spätestens nach 10 Schritten. Die bis zu diesem Zeitpunkt ermittelten kleinste zulässige Rate C_A wird als vermeintlich (im Sinne des Verfahrens) konservative Lösung akzeptiert. Am Ende wird diese vorläufige Lösung C_A in der Umgebung von t^* mit einer um den Faktor 10 erhöhten Zeitauflösung verifiziert. Man erkennt unmittelbar den immensen Aufwand für die Berechnungen, wenn man bedenkt, daß für jeden Wert, den man als Lösung für C_A in Betracht zieht, die *Large Deviation Rate Function* für alle Werte von t mit Hilfe des Newton-Verfahrens berechnet werden muß.

Tabelle 3.2: *Verkehrsparameter der periodischen Ein-Aus-Quellen, mit denen die Untersuchungen der Verfahren zur Berechnung des Ressourcenbedarfs von Aggregaten durchgeführt werden. Die Parameter sind auf eine Rate von* $20 \cdot 10^6\, byte \cdot s^{-1}$ *normiert. N ist der im Zusammenhang mit dem Many Sources Asymptotic eingeführte Skalierungsfaktor. Im folgenden wird jeweils* $NC_A \cdot 50\, ms$ *Wartespeicher reserviert und wie auch bisher schon als obere Schranke für die Überschreitungswahrscheinlichkeit* $\epsilon_O = 10^{-4}$ gesetzt.

Typ	1	2
Anzahl	N	N
p	0,006250	0,002500
r	0,000625	0,001250
b	0,0005625	0,0001250
T_P	1,0	0,2

Abb. 3.11 und 3.12 zeigen, daß das *Many Sources Asymptotic* das Verhalten mittlerer und großer Aggregate sehr präzise beschreibt. Ergänzt um die Verfeinerungen von Rao-Baradur, Gleichung (3.77), oder um die Herleitung der Verlust- statt Überschreitungswahrscheinlichkeit, wie mit Gleichung (3.78) angedeutet, kann es das Verhalten des Aggregates noch präziser vorhersagen, allerdings bleibt die Vorhersage des Ressourcenbedarfs dann nicht mehr unbedingt konservativ. In allen drei Fällen bereitet die Berechnung des Ressourcenbedarfs des kleinsten Aggregates bei $N = 1$ Probleme. Es ist zweckmäßig, hier analog zum Verfahren von Elwalid et al. [59] die deterministische effektive Bandbreite als beste Lösung zu akzeptieren.

Dagegen erfaßt das auf der Verbindungsannahmesteuerung von Elwalid et al. [59] beruhende Verfahren zur Berechnung des Ressourcenbedarfs von Aggregaten aufgrund seines heuristischen Ansatzes die beim statistischen Multiplexen auftretenden Effekte nur zum Teil. Dies bestätigt entsprechende Aussagen in [85]. Andererseits relativieren sich die Vorteile des *Many Sources Asymptotic* bei einer ganzheitlichen Betrachtung von Aggregation unter besonderer Berücksichtigung praktischer Gesichtspunkte, dem Anliegen der vorliegenden Arbeit: Die effiziente Ausnutzung der Ressourcen der Datenebene, Wartespeicher und Übertragungskapazität, wird durch einen hohen Berechnungsaufwand erkauft.

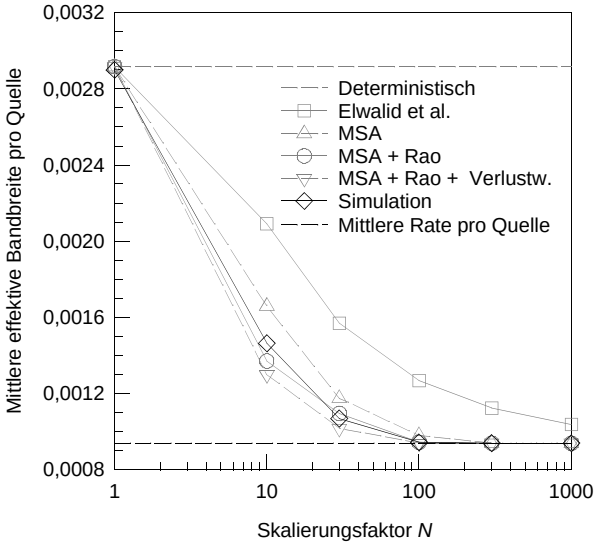

Abb. 3.11: *Mittlere effektive Bandbreite pro Quelle bei einer wachsenden Anzahl von Verbindungen der Typen aus Tab. 3.2. Die Erweiterungen des Many Sources Asymptotic (MSA) erweisen sich als nicht konservativ.*

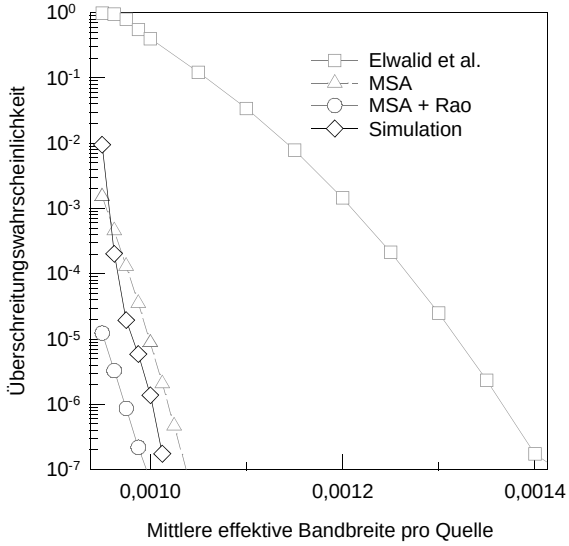

Abb. 3.12: *Abhängigkeit der Wahrscheinlichkeit, daß die Belegung des Wartespeichers $N\,C_A\,50$ ms überschreitet, mit wachsender Rate C_A für jeweils $N = 100$ Verbindungen der beiden Typen aus Tab. 3.2.*

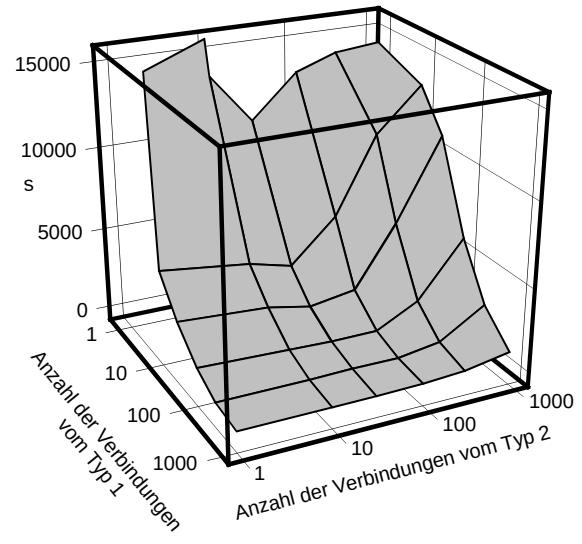

Abb. 3.13: *Arbeitspunkt s^* in (3.77) bei variabler Zusammensetzung des Aggregates mit den normierten Quellen aus Tab. 3.2.*

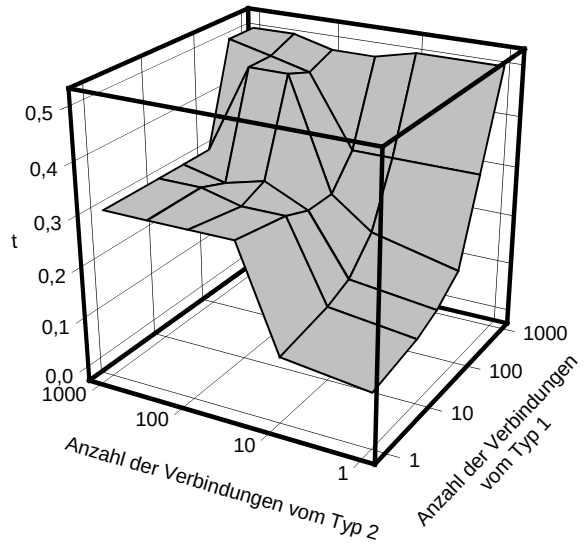

Abb. 3.14: *Arbeitspunkt t^* in (3.77) bei variabler Zusammensetzung des Aggregates mit den normierten Quellen aus Tab. 3.2.*

Obwohl für die Untersuchungen – nicht zuletzt zum Vergleich mit dem auf dem Verfahren von Elwalid et al. [59] – aus vergleichsweise einfachen periodischen Ein-Aus-Quellen bestehende Aggregate betrachtet werden, geht aus den Abb. 3.13 und 3.14 die Komplexität des Zusammenhangs von s^* und t^* hervor. Andererseits sind keine eklatanten Unstetigkeiten zu beobachten, so

daß ein heuristischer Ansatz zu einer praktikablen Lösung für die Berechnung des Ressourcenbedarfs mit dem *Many Sources Asymptotic* führen könnte.[2]

3.2.4.4 Vorschlag für den praktischen Einsatz des *Many Sources Asymptotic* mit periodischen Ein-Aus-Quellen

Einige wenige Verbindungen, die in ein nicht zu kleines Aggregat aufgenommen oder aus diesem entfernt werden, verändern normalerweise auch den Arbeitspunkt $(s*, t*)$ wenig. Diese Beobachtung motiviert folgendes heuristische Verfahren zur Berechnung der für das Aggregat benötigten Ressourcen.

Solange ein Aggregat klein ist, berechnet man seinen Ressourcenbedarf mit dem sehr robusten und schnellen Verfahren von Elwalid et al. Sobald die kritische Anzahl der Verbindungen so deutlich überschritten ist, daß zur Berechnung des neuen Ressourcenbedarfs eine oder zwei Iterationen des Newton-Algorithmus genügen, setzt ein Verfahren ein, das den für die genauere Berechnung mit dem *Many Sources Asymptotic* nötigen Rechenaufwand begrenzt, indem es die Berechnung auf einen größeren Zeitraum verteilt. Mit jeder Änderung des Aggregates wird auf diese Weise die Berechnung präzisiert.

Dazu wird zunächst die mit dem Verfahren von Elwalid et al. errechnete Rate als konservative Lösung C_A in die *Large Deviation Rate Function* des *Many Sources Asymptotic* (3.87) eingesetzt und eine erste Lösung für $s*$ und $t*$ gesucht. In dieser Phase muß auf jeden Fall noch der ganze erlaubte Bereich von t bis zur Periodendauer T_P mit Hilfe von Stützstellen untersucht werden. Die Berechnungszeit dafür würde zu inakzeptablen Antwortzeiten auf die Anforderungen zur Änderung des Aggregates führen. Da sich $s*$ und $t*$, so die Annahme, bei dem jetzt ja schon nicht mehr kleinen Aggregat von Anforderung zu Anforderung nicht sehr stark verändert, wird die Suche nach $t*$ bei der ersten Anforderung begonnen, nach wenigen Stützstellen jedoch unterbrochen, um schließlich mit jeder neuen Anforderung ein Stück weit fortgesetzt. Bis zum Ende dieser Prozedur wird dem Aggregat noch die mit der Methode von Elwalid et al. bestimmte Rate zugewiesen. Nach dem Ende dieser Initialisierungsphase sollte es gemäß der oben erwähnten Annahme ausreichen, eine neue Lösung $t*$ nur noch in der näheren Umgebung des alten $t*$ zu suchen.

Wenn bei der letzten Berechnung eine konservative Lösung für C_A gefunden worden ist (und S_A mit dieser über die maximal zulässige Verzögerungszeit verbunden ist), ist beim Entfernen einer

2 Die oben angesprochenen Schwierigkeiten für $N = 1$ treten natürlich auch in Abb. 3.13 und 3.14 auf. Die dargestellte Lösung konnte nicht in der oben beschriebenen Weise verifiziert werden und ist nur aus technischen Gründen in den Graphen aufgenommen worden.

Verbindung i nach Abzug der vereinbarten Rate r_i und beim Hinzufügen nach Addition der bei Verwendung des Verfahrens von Elwalid et al. [59] fälligen Differenz jeweils eine konservative Lösung gegeben, auf die man zurückfallen kann, falls sich eine spekulative Schätzung als zu optimistisch erweisen sollte. Spekulativ und optimistisch wäre es, genau umgekehrt r_i zu addieren, wenn eine Verbindung zum Aggregat hinzukommt, bzw. die beim Verfahren von Elwalid et al. auftretende Differenz zu subtrahieren, wenn eine Verbindung wegfällt. Diese neue Rate wird gleichsam ausprobiert. Leinmüller untersucht in [129] die Effizienz dieses kombinierten Verfahrens.

Eine Begrenzung der Anzahl der Stützstellen, mit denen man die Abhängigkeit der momentenerzeugenden Funktion von t zu erfassen versucht, reduziert zwar auf der einen Seite den Aufwand für die Berechnung des Ressourcenbedarfs, birgt auf der anderen Seite aber das Risiko, daß die Überschreitungswahrscheinlichkeit unterschätzt wird und die berechneten Ressourcen nicht genügen, um die vereinbarte Dienstgüte der Verbindungen im Aggregat einzuhalten. Wenn sich beispielsweise das für die Berechnung der in Abb. 3.13 und 3.14 dargestellten Ergebnisse herangezogene Aggregat aus nur wenigen Verbindungen zusammensetzt, der statistische Multiplexgewinn mithin nur gering ist und die Berechnung des Ressourcenbedarfs zu hohen Werten von s^* führt, ist der Logarithmus der momentenerzeugenden Funktion sehr sensitiv gegenüber Änderungen von t. Insofern ist es nicht unkritisch, bei Aggregaten dynamischer Zusammensetzung den erlaubten Bereich für t wie in [47] mit immer gleichen äquidistanten Stützstellen abzusuchen.

3.2.5 Einordnung der auf der *Large Deviation* Theorie beruhenden Verfahren

Tab. 3.3 stellt die wichtigsten Eigenschaften der in der vorliegenden Arbeit besprochenen, auf der *Large Deviation* Theorie beruhenden Verfahren zusammen.

Außer Hui [97] haben weitere Autoren Verfahren zur Verbindungsannahmesteuerung in Bediensystemen ohne Wartespeicher vorgestellt. Kröner untersucht einen Großteil dieser Verfahren in [120]. Huis auf der *Large Deviation* Theorie beruhendes Verfahren schneidet dabei sehr gut ab.

In der Literatur wird eine Reihe von Modellen behandelt, die genau so wie das *Large Buffer Asymptotic* auf einen exponentiellen Zusammenhang zwischen der Wahrscheinlichkeit, daß die Belegung des Puffers eine vorgegebene Schwelle überschreitet, und dieser Schwelle führen. Kelly [116] beispielsweise findet einen solchen Zusammenhang für ein Modell, in dem der Verkehr der Quellen in Form von in negativ exponentiell verteilten Abständen ankommenden Büscheln beliebig verteilter Länge beschrieben wird, auf der Basis einer klassischen verkehrstheoretischen Methode zur Berechnung der Wahrscheinlichkeit für Ruin bei *Compound Poisson* Prozessen [65]. In [26]

wird auf einen ursprünglich auf Guérin zurückgehenden praktischen Ansatz hingewiesen, der sich eine analytische Lösung für die Überschreitungswahrscheinlichkeit in einem Bediensystem zu Nutze macht, in dem Quellen mit negativ-exponentiell verteilten Ein- und Aus-Phasen Daten in Form eines Flüssigkeitsstroms erzeugen oder pausieren [6].

Tabelle 3.3: *Zusammenfassung einiger wichtiger Eigenschaften der in der vorliegenden Arbeit besprochenen, auf der Large Deviation Theorie beruhenden Verfahren zur Berechnung des Ressourcenbedarfs von Aggregaten*

	Tschernoff-Schranke (Hui)	Elwalid et al.	Large Buffer Asymptotic	Many Sources Asymptotic
Modell	Pufferloses Bediensystem	Bediensystem mit Wartespeicher	Bediensystem mit unendlich großem Wartespeicher	Wartespeicher beliebig, Bediensystem skaliert mit Quellenzahl N
Formeln	(3.37), (3.41), (3.52)	(3.82), (3.83), (3.85)	(3.65), (3.66)	(3.72)-(3.76)
Quellen	beliebige	periodische Ein-Aus-Quellen	beliebige, nicht langzeitkorrelierte	beliebige
Genauigkeit	Asymptotisch korrekt unter den Voraussetzungen des Gärtner-Ellis-Theorems	unzulängliches Modell	unzulängliches Modell	offenbar gute Abbildung des Systemverhaltens
Obere Schranke	ja	ja	meistens ja	nachgewiesen nur für $N \to \infty$
Rechenaufwand	minimal	minimal	nicht betrachtet	abhängig vom Quellenmodell, u. U. sehr hoch
Bemerkungen		geringfügige Verbesserungen möglich, z. B. [85]	wird im Anschluß noch ausführlicher diskutiert	Ansätze wie der aus Kap. 3.2.4.4 reduzieren ggf. Rechenaufwand

Das durch die Gleichungen (3.65) und (3.66) gegebene *Large Buffer Asymptotic* ist in der Literatur intensiv untersucht worden, zunächst vornehmlich mit dem Ziel, die in vielen Fällen zu konservative Näherung zu verbessern. Eine sehr einleuchtende Lösung wird in [119] referenziert. Dort wird für Quellen, die als durch Markoff-Prozesse modulierte Flüssigkeitsströme modelliert werden, das *Large Buffer Asymptotic* mit Huis [97] Lösung des Systems ohne Wartespeicher multipliziert. In einem Graphen, in dem die Überschreitungswahrscheinlichkeit als Funktion der Schwelle aufgetragen wird, senkt dies die Kurve um eine Konstante ab. Offensichtlich werden die beim statistischen Multiplexen auftretenden Effekte dadurch besser berücksichtigt. Natürlich ist diese Lösung

wie das *Large Buffer Asymptotic* selbst auch nicht auf langzeitkorrelierte Prozesse anwendbar. Bodamer und Charzinski [26] wenden diese Idee jedoch auch auf die (übrigens mit dem *Many Sources Asymptotic* für diese Quellen übereinstimmende [114]) Formel von Norros für als *Fractional Brownian Motion* modellierte Ankunftsprozesse an.

Für spezielle Quellenmodelle, beispielsweise Markoffsche Ankunftsprozesse (MAP) [30] oder Gaußsche Prozesse [43], kann das durch die Gleichungen (3.65) und (3.66) beschriebene *Large Buffer Asymptotic* in eine andere Richtung verbessert werden. Dem Exponenten $-c_0 s^*$ in (3.66) wird dann ein zweiter zur Seite gestellt, der bei positiver Autokorrelationsfunktion (3.67) mit wachsender Anzahl der Quellen kleiner wird. Im Falle der Gaußschen Ankunftsprozesse erhält man schließlich mit einem anderen analytischen Verfahren (*Extreme Value Theory for Gaussian Processes*) sogar die gleiche Näherung wie im *Many Sources Asymptotic* [43]. Dieses unter der Bezeichnung *Maximum Variance Asymptotic* bekannte Ergebnis stellt dabei die beim *Large Buffer Asymptotic* nicht unkritische Näherung (3.61) des Zusammenhangs zwischen der Restarbeit $W(t)$ im Bediensystem und dem Füllstand der Warteschlange $Q(t)$ auf eine neue Basis [119].

Das *Large Buffer Asymptotic* in der Form (3.65) und (3.66) kann dagegen nicht alle entscheidenden Effekte in Bediensystemen mit Wartespeicher beschreiben. Die oben erwähnte Arbeit von Duffield und O´Connell [54] etabliert eine (3.65) und (3.66) formal ähnelnde Asymptotik, die deutlich macht, daß im Falle langzeitkorrelierter Prozesse die Wahrscheinlichkeit, daß die Belegung des Puffers eine vorgegebene Schwelle überschreitet, nicht mehr exponentiell mit wachsendem Speicher abnimmt.

Das *Many Sources Asymptotic* (3.72)-(3.76) bleibt davon unberührt. Da es sich nicht zuletzt deshalb auf beliebige Quelltypen anwenden läßt, mit wichtigen in der Literatur behandelte andere Näherungslösungen für spezielle Quellen korrespondiert [43, 114] und dabei den Einfluß von Bandbreite und Wartespeicher auf den Ressourcenbedarf von Quellen zeigt, ist es nach Auffassung des Autors zur Zeit das mächtigste Konzept zur Berechnung der effektiven Bandbreite von Verbindungen.

In [28] und [29] ist eine neue Methode vorgestellt worden, mit deren Hilfe ausgehend vom deterministischen *Network Calculus* für Aggregate aus Quellströmen, die deterministischen oberen Schranken unterworfen sind, Verkehrsankunftsfunktionen bestimmt werden können, die mit hoher Wahrscheinlichkeit obere Schranken für den Gesamtstrom sind. Diese Verkehrsankunftsfunktionen lassen ganz im Gegensatz zu ihren deterministischen Pendants die Berücksichtigung statistischen Multiplexens zu und werden daher auch *Effective Envelopes* genannt. Unter einem *Local Effective Envelope* der im Intervall $[t, t+\Delta t)$ ankommende Datenmenge $A(t, t+\Delta t)$ eines Zufallsan-

kunftsprozesses wird beispielsweise eine Funktion $E_L(\Delta t, \epsilon)$ verstanden, die für alle t und Δt die Ungleichung

$$P\{A(t, t+\Delta t) \leq E_L(\Delta, \epsilon)\} \geq 1-\epsilon \tag{3.91}$$

erfüllt. Wie in [29] gezeigt wird, lassen sich mit einem solchen deterministischen *Local Effective Envelope* $E_L(\Delta t, \epsilon)$ Überschreitungswahrscheinlichkeiten der Form $P\{A(t, t+\Delta t) \geq c_0\}$ abschätzen. Nach der Umwandlung von deterministischen in stochastische *Schedulability Conditions* (zur Ausnutzung statistischen Multiplexens) sind nun aber genau solche Überschreitungswahrscheinlichkeiten zu bestimmen. Die eben noch in statistische Bedingungen umgewandelten *Schedulability* Conditions lassen sich dann doch wieder auf deterministische zurückführen, wenn die statistische Verkehrsankunftsfunktionen durch deren deterministischen *Local Effective Envelope* $E_L(\Delta t, \epsilon)$ ersetzt werden. Diese müssen natürlich zuvor mit Hilfe stochastischer Methoden, etwa dem zentralen Grenzwertsatz, der Tschernoff-Schranke, möglicherweise auch mit anderen in der vorliegenden Arbeit besprochenen Verfahren bestimmt werden. Als Hauptvorteil dieses Verfahrens ist sicherlich seine Anwendbarkeit auf beliebige Bedienstrategien zu nennen. Offensichtlich lassen sich aber mit präziseren verkehrstheoretischen Modellen bessere Ergebnisse erzielen, wenn auch mit deutlich größerem Aufwand, wie das Beispiel EDF zeigt [182].

3.3 Auswirkungen auf die Aggregation von Verkehrsströmen

In Multiplexern ohne Wartespeicher tragen manche Quellen mehr, andere weniger zum statistischen Multiplexgewinn bei [140]. In der Literatur zum *Many Sources Asymptotic* wird sogar davon gesprochen, daß die Aggregation von Ankunftsprozessen, die gut zueinander passen, zu einer gegenüber Teilaggregaten besonders effizienten Auslastung der Ressourcen führt [57]. Im Umkehrschluß kann dementsprechend die Aggregation von Ankunftsprozessen, die nicht gut zueinander passen, weit weniger effizient sein.

Dennoch, so stellt sich heraus, werden jeweils die Ressourcen bei identischen Anforderungen an die Dienstgüte durch ein Gesamtaggregat effizienter genutzt als durch alle möglichen Teilaggregate. Diese Aussage gilt natürlich nur insoweit, als die Verkehrsmodelle (Flüssigkeitsströme) und Asymptotiken (*Large Deviation* Prinzip, *Many Sources Asymptotic*) eigentlich das Verhalten von Aggregaten einer größeren Anzahl von Quellen genau approximieren. Diese Annahme liegt allerdings

auch schon den Verfahren zur Berechnung des Ressourcenbedarfs zugrunde. Aggregation macht ohnehin nur unter der Randbedingung Sinn, daß Verbindungen in hinreichend großer Zahl zusammengefaßt werden können.

Einige Modelle beruhen auf deterministischen Verkehrsankunftsfunktionen und korrespondieren deshalb nur dann mit der Realität, wenn Verkehrsformung im Aggregationsknoten dafür sorgt, daß die Verkehrsströme konform zu der im Verkehrsdeskriptor spezifizierten Verkehrsankunftsfunktion sind. Inwieweit der Verzicht auf eine solche Verkehrsformung den Ressourcenbedarf verändert, bedarf einer weitergehenden Studie, die sehr komplexe Randbedingungen einbeziehen sollte.

3.3.1 Aggregation bei deterministischem Multiplexen

Man betrachte in einem Aggregationsknoten zunächst ein Aggregat, an dessen Eingang EDF als Bedienstrategie eingesetzt wird. Für dieses Aggregat muß nach Prüfung der kritischen Zeiten der *Schedulability Condition* die kleinste Rate C_A reserviert werden, welche die Ungleichung für alle diese Zeiten erfüllt. Wenn man außerdem vom Idealfall $\max\{L_i\}=0$ ausgeht, gilt also

$$C_A = \sup_{t\geq 0}\{\frac{1}{t}\sum_{i\in M} A_i(t-D^S_{i,max})\} \tag{3.92}$$

Wird die Menge M von Verbindungen beispielsweise in zwei beliebige Teilmengen M_1 und M_2 aufgeteilt, $M_1 \cup M_2 = M$, so gilt wegen der Dreiecksungleichung

$$C_A = \sup_{t\geq 0}\{\frac{1}{t}\sum_{i\in M} A_i(t-D^S_{i,max})\} \leq \sup_{t\geq 0}\{\frac{1}{t}\sum_{i\in M_1} A_i(t-D^S_{i,max})\} + \sup_{t\geq 0}\{\frac{1}{t}\sum_{i\in M_2} A_i(t-D^S_{i,max})\} \tag{3.93}$$

Unter den o. g. Voraussetzungen ist das Multiplexen der Verbindungen in ein Aggregat also niemals schlechter als die Trennung in Teilaggregate.

Alle realisierbaren Bedieneinheiten weichen natürlich vom idealen EDF mehr oder weniger ab. Bei Berücksichtigung der Paketlänge kann prinzipiell sogar bei EDF in allerdings für die Praxis nicht relevanten Ausnahmefällen die Trennung in Teilaggregate vorteilhaft sein.

Bei RPQ und RPQ+ kommt das Rotationsintervall $\Delta > 0$ als ein die Aggregation behindernder Faktor hinzu. Zunächst können nunmehr nur diskrete Verzögerungsprioritäten angeboten werden, die unter theoretischen Gesichtspunkten natürlich nicht exakt mit den tatsächlichen Anforderungen der Verbindungen übereinstimmen, so daß die jeweils höhere Prioritätsstufe gewählt und deshalb mehr Bandbreite als eigentlich nötig reserviert werden muß. In den *Schedulability Conditions* (3.18) für RPQ und (3.22) für RPQ+ entfallen durch Δ erhöhte Beiträge, wenn in einem Teilaggregat nur

eine Prioritätsstufe bedient oder eine Prioritätsstufe durch Trennung zur kleinsten im abgetrennten Teilaggregat wird. Bei hinreichend kleinem Rotationsintervall Δ ist ist der durch eine Trennung zu erzielende Gewinn jedoch praktisch vernachlässigbar.

Für FIFO-Bedieneinheiten am Eingang des Aggregates trifft dies keineswegs zu [191]. Hier muß für das alle Verbindungen umfassende Aggregat

$$C_A = \sup_{t \geq 0} \{ \frac{1}{t} \sum_{i \in M} A_i(t - \min D^S_{i,max}) \} \tag{3.94}$$

für nach Verzögerungsprioritäten getrennte Teilaggregate dagegen

$$C_A = \sum_{c=1}^{N_c} \sup_{t \geq 0} \{ \frac{1}{t} \sum_{i \in M_c} A_i(t - D^S_{i,max}) \} \tag{3.95}$$

Bandbreite reserviert werden, was beispielsweise bei Verbindungen mit identischen Verkehrsparametern günstiger ist.

Wenn ein Aggregat unterschiedliche Verzögerungsprioritäten umfaßt, muß in nachfolgenden Knoten die Verzögerung so gering sein, daß für keine Verbindung die vereinbarte Gesamtverzögerung Ende-zu-Ende überschritten wird. Sofern im Aggregationsknoten ein Großteil der insgesamt erlaubten Verzögerungszeit zur Reduzierung des Bandbreitebedarfs des Aggregates eingesetzt werden kann, haben alle Verbindungen im Aggregat ungefähr gleich hohe Anforderungen an die Verzögerung im Aggregationsbereich. Am einfachsten kann dies im Kernnetz mit Aggregaten konstanter Bandbreite gewährleistet werden. Ob unter Berücksichtigung aller Bewertungsfaktoren nach Verzögerungsprioritäten getrennte Aggregate wirklich zu einer besseren Lösung führen, darf zumindest bezweifelt werden. Zur abschließenden Bewertung einer solchen Strategie müßten Statistiken über die Anzahl, die geforderte Dienstgüte, die Ziele und die Pfade der Verkehrsströme vorliegen und für eine gegebene Netztopologie ausgewertet werden. Überdies müßten die aus der Trennung eines Aggregates resultierenden Kosten für zusätzliche Verbindungszustände, Verbindungsannahmesteuerung, Steuerung der Wartespeicher sowie Bedieneinheiten bekannt sein.

Le Boudec [127] schlägt dennoch Aggregate variabler Rate mit Verkehrsankunftsfunktionen der Form (3.10) vor. Setzt man darin $p_i = C_A$, so wird die maximale, für alle Verbindungen des Aggregats als konstant angenommene Verzögerung D^S_{max} bei Verkehrsformung des Aggregats auf einen Verkehrsdeskriptor der Form (3.10) dann eingehalten, wenn die verbleibenden Parameter r_i und b_i der Ungleichung

$$A_{in}(t) \leq b_i + (t + D^S_{max}) r_i \tag{3.96}$$

genügen. Eine lineare Kostenfunktion der Form

$$c_1 r_i + c_2 b_i \tag{3.97}$$

wird für den Punkt (r_i, b_i) auf dem Rande des konvexen Gebietes (3.96) minimal, der auf einer zum Ortsvektor (c_1, c_2) orthogonalen Tangent(ialeben)e liegt. Dies ist eine Gerade (Fläche), auf der abgesehen von der gesuchten Lösung nur Punkte (r_i, b_i) (außerhalb des erlaubten Bereiches) liegen, für die die Kostenfunktion konstant bleibt. Der Normalenvektor der gesuchten Tangentialebene des durch die Gleichung

$$f(b_i, r_i) = b_i + (t + D^S_{max})\, r_i - A_{in}(t) = 0 \tag{3.98}$$

definierten Randes ist gegeben durch

$$\textbf{grad}\ f(b_i, r_i) = \left(\frac{\partial}{\partial b_i}, \frac{\partial}{\partial r_i}\right) f(b_i, r_i) \tag{3.99}$$

Dieser Normalenvektor auf den Rand des Gebietes im gesuchten optimalen Punkt muß parallel zu (c_1, c_2) liegen. Diese Bedingung genügt, um die in [127] angegebenen Bestimmungsgleichungen für die gesuchten optimalen Parameter herzuleiten.

3.3.2 Aggregation von stochastischen Ankunftsprozessen in pufferlosen Bediensystemen

Aufgrund der allgemeinen Eigenschaften der momentenerzeugenden Funktion in bezug auf *s*, der einzigen Variablen, können aus Gleichung (3.85) sehr einfach Schlußfolgerungen zur Aggregation von Verkehrsströmen in pufferlosen Bediensystemen gezogen werden. Bereits bei den Betrachtungen zu den Abb. 3.9 und 3.10 konnte beobachtet werden, daß mit wachsender Zahl von Verbindungen $s*$ und damit verbunden der Bandbreitebedarf pro Quelle, die effektive Bandbreite für dieses Modell, kleiner wird. Zum Nachweis der Allgemeingültigkeit dieser Beobachtung kann man auf die Ungleichung

$$\left(\frac{\mathrm{d}}{\mathrm{d}\,s} M_j(s)\right)^2 \le \left(\frac{\mathrm{d}^2}{\mathrm{d}\,s^2} M_j(s)\right) M_j(s) \tag{3.100}$$

zurückgreifen [23]. Man sieht mit ihrer Hilfe sofort, daß $\frac{\mathrm{d}}{\mathrm{d}s} M_j(s) M_j(s)^{-1}$, die effektive Bandbreite der Verbindungen jedes beliebigen Typs *j*, vgl. Formel (3.84), streng monoton wachsend ist, also mit fallendem *s* abnimmt. Wenn nun zu einem bestehenden Aggregat eine Verbindung des Typs *j* hinzugefügt wird, müssen in Gleichung (3.85) die Terme $s \frac{\mathrm{d}}{\mathrm{d}s} M_j(s) M_j(s)^{-1} - \ln M_j(s)$ auf der

linken Seite und im Argument des Logarithmus $2\pi s^2(M_j(s)^{-1}\frac{\mathrm{d}^2}{\mathrm{d}s^2}M_j(s)-(M_j(s)^{-1}\frac{\mathrm{d}}{\mathrm{d}s}M_j(s))^2)$ addiert werden. Der erste Term ist für $s=0$ gleich null und für $s>0$ streng monoton wachsend, der zweite Term nicht negativ, was wiederum mit der Ungleichung (3.100) bestätigt werden kann. Aus diesem Grunde vergrößert jede Verbindung, die in das Aggregat aufgenommen werden soll, die linke Seite (das sind die in Abb. 3.9 dargestellten Kurven) von Gleichung (3.85) für alle *s*, so daß die Lösung von (3.85), s^*, kleiner wird, und aufgrund ihrer strengen Monotonie, auch die effektive Bandbreite. Diese Aussage gilt natürlich auch dann noch, wenn die Rao-Baradur-Verfeinerung negativ wird und deswegen entfällt. Durch neue Verbindungen verbessert sich also grundsätzlich der statistische Multiplexgewinn des Aggregates, obwohl die einzelnen Verbindungstypen durchaus unterschiedlich beitragen und die Verbindungsannahmegrenzkurve bei vorgegebener Bandbreite deshalb im Gegensatz zum deterministischen Multiplexen einen konkaven zulässigen Bereich einschließt.

Daraus folgt, daß das Multiplexen von Verbindungen in ein Aggregat niemals schlechter als die Trennung in Teilaggregate ist. Man betrachte dazu zwei Aggregate mit den Arbeitspunkten s_1^* bzw. s_2^*, $s_1^*<s_2^*$. Entfernt man aus dem zweiten Aggregat eine Verbindung und fügt sie dem ersten Aggregat hinzu, so reduziert sich s_1^* weiter. Setzt man diese Prozedur fort, bis das zweite Aggregat leer ist, erhält man als Lösung von Gleichung (3.85) $s^*<s_1^*<s_2^*$. Also ist die effektive Bandbreite aller Verbindungen kleiner als vor dem Zusammenfassen der beiden Aggregate zu einem.

Im Gegensatz zu den deterministischen Verfahren zur Berechnung des Ressourcenbedarfs wird statistisches Multiplexen mit wachsender Anzahl von Verbindungen immer effizienter, bis sich die effektive Bandbreite einer Verbindung ihrer mittleren Rate nähert. In Verbindung mit einer Verkehrsklasse ohne präzise Erwartungen an die maximale Verzögerung wie etwa *Controlled Load* mag die Abbildung in ein Aggregat mit variabler Rate Vorteile haben. Folgt man dem Vorschlag in [128], muß dazu analog zu dem oben zur Berechnung des Ressourcenbedarfs von *Controlled Load* Aggregaten eingesetzten Verfahren [59] jede Quelle wieder so umgeformt werden, als ob sie statt eines gemeinsam genutzten Wartespeichers ein getrenntes, verlustfreies Bediensystem durchlaufen würde. Während in [59] dieses imaginäre Bediensystem eine Quelle so auf eine von den *Leaky Bucket* Parametern abhängige Spitzenrate e_i umformt, daß sie von der dem Aggregat zugeordneten konstanten Rate C_A und dem Wartespeicher S_A einen jeweils gleichen Anteil belegt, wird in [128] angenommen, daß die Quelle unter Ausnutzung eines jeweils gleichen Anteils nicht nur an der Spitzenrate P_A und am Wartespeicher S_A des Aggregates, sondern auch an dessen Büscheltoleranz B_A und mittleren Rate R_A analog umgeformt wird. Es wird angenommen, daß sie sich anschließend wie

eine Quelle verhält, die bis zur Ausschöpfung einer entsprechend umgeformten Büscheltoleranz b_{0i} mit einer von der tatsächlichen Spitzenrate p_i auf p_{0i} reduzierten Spitzenrate sendet, danach solange mit einer dafür erhöhten mittleren Rate r_{0i} weitersendet, bis eine dem tatsächlichen Büschel entsprechende Datenmenge gesendet worden ist, um schließlich zu pausieren, bis der *Leaky Bucket* sich wieder vollständig erholt hat. Durch den Zusammenhang

$$\frac{p_{0i}}{P_A}=\frac{r_{0i}}{R_A}=\frac{s_{0i}}{S_A}=\frac{b_{0i}}{B_A} \tag{3.101}$$

wird das Problem der Berechnung der für das Aggregat benötigten Ressourcen bzw. des zu vereinbarenden Verkehrsdeskriptors wiederum auf den Fall eines Systems mit nur einer Ressource reduziert. Damit ergibt sich eine einfache Analogie zu dem einfacheren Problem in [59]. In [128] wird zur Lösung nun eine vergleichsweise komplexe momentenerzeugende Funktion $M_j(s)$ hergeleitet. Die allgemeinen Eigenschaften momentenerzeugender Funktionen gelten selbstverständlich unverändert weiter und – als Folge davon – auch die Betrachtungen zur Effizienz der Aggregation im Falle heterogener Quellen. Ob die heuristischen Annahmen wirklich den realen Gegebenheiten im Bediensystem entsprechen, bleibt ebenso offen wie in [59]. In [128] werden die Vorteile von Aggregaten variabler Rate überdies mit relativ büschelförmigen Quellen bei sehr kleinen Verlustobergrenzen nachgewiesen, d. h. für Quellen, die ein großes Potential hinsichtlich des erzielbaren statistischen Multiplexgewinns haben, dieses aber vor allem bei Aggregaten, für die eine konstante Rate reserviert wird, unter diesen Bedingungen erst bei einer relativ großen Quellenzahl ausschöpfen können. Kleine Aggregate sollten aber eher die Ausnahme als die Regel sein. Im übrigen könnten Aggregate mit konstanter vereinbarter Rate in höheren Ebenen ohne weitere Prüfung wiederum aggregiert werden, ohne daß dies unter Umständen proprietäre Netzdienstangebote im Zugangsbereich beeinträchtigt. Eine abschließende Bewertung der Vor- und Nachteile von Aggregaten variabler Rate ist also nur nach Integration in ein mehrstufiges Aggregationskonzept möglich, in der dann auch die erhöhte Komplexität, die durch Pufferung auf jeder Aggregationsebene anwachsende *Latency* und nicht zuletzt die Kosten für zusätzliche Aggregate einfließen sollten.

3.3.3 Aggregation von stochastischen Ankunftsprozessen auf der Basis des *Many Sources Asymptotic*

Die Ausnutzung allgemeiner Eigenschaften der momentenerzeugenden Funktion läßt vergleichsweise präzise Aussagen zum Verhalten von s^*, dem Arbeitspunkt des statistischen Multiplexens,

im pufferlosen Fall zu, die sogar den Korrekturterm von Rao-Baradur einbeziehen: Jede neue Verbindung verringert den Wert von s^*. Dies impliziert, daß Teilaggregate immer vorteilhaft zu einem einzigen Aggregat zusammengefaßt werden können.

Aus der mathematischen Literatur [100] sind Eigenschaften der auch als Young-Fenchel-Transformierte der Funktion f bezeichneten Funktion

$$f^*(x) = \sup_y \{\langle x|y\rangle - f(y)\} \tag{3.102}$$

bekannt, mit deren Hilfe über den pufferlosen Fall hinaus Aussagen zur Aggregation gemacht werden können, obwohl – von Spezialfällen abgesehen [100, 57] – weder klare Aussagen zur (zu den) Stelle(n) (s^*, t^*) möglich sind, an der (denen) die *Large Deviation Rate Function* (3.87) einen Sattelpunkt aufweist, noch der Term von Rao-Baradur in die Überlegungen mit einbezogen werden kann. Das lineare Funktional $\langle x|y\rangle$ in (3.102) kann durch eine der linearen Funktionen ersetzt werden, die uns in den Verfahren zur Verbindungsannahmesteuerung auf der Basis der *Large Deviation* Theorie begegnen. Unter anderem gilt

$$(f_1 + f_2 + ... + f_n)^* \leq f_1^* \oplus f_2^* \oplus ... \oplus f_n^* \tag{3.103}$$

In dieser Ungleichung ist mit $f_1 \oplus f_2$ die Faltungsoperation

$$(f_1 \oplus f_2)(x) = \inf\{f_1(x_1) + f_2(x_2) \mid x_1 + x_2 = x\} \tag{3.104}$$

gemeint. Zum Beweis dieser und folgender Eigenschaften der Young-Fenchel-Transformierten sei auf die bereits angeführte Literatur verwiesen [100].

Zur Übertragung dieser Beziehung beispielsweise auf die Ungleichungen (3.82) oder (3.72) mit (3.87), auf deren Grundlage der Ressourcenbedarf eines Aggregates berechnet werden kann, müssen lediglich die Funktionen f_1, ..., f_n durch $\ln M_1$, ..., $\ln M_n$ sowie x_1, ..., x_n durch die Ressourcen $C_{A,1}t + S_{A,1}$, ..., $C_{A,n}t + S_{A,n}$ möglicher Teilaggregate und schließlich y durch s ersetzt werden. Man sieht dann, daß die *Large Deviation Rate Function* eines Gesamtaggregates der linken Seite der Ungleichung (3.103) entspricht, wenn für dieses Aggregat die Ressourcen $x = \sum x_i$ reserviert würden, und die rechte Seite der Summe der *Large Deviation Rate Functions* der Teilaggregate. Ungleichung (3.103) schließt also noch nicht aus, daß Teilaggregate effizienter sein können.

Sind allerdings die Funktionen f_1, ..., f_n so wie im Falle von $\ln M_1$, ..., $\ln M_n$ konvex und haben ihre effektiven Definitionsbereiche, d. h. die Teile ihres Definitionsbereiches, in denen sie endlich sind, einen gemeinsamen Punkt, an denen sie (mit höchstens einer Ausnahme) stetig sind, gilt die Gleichung [100]

$$(f_1+f_2+...+f_n)^*=f_1^*\oplus f_2^*\oplus...\oplus f_n^* \tag{3.105}$$

Es existieren sogar Ressourcenanteile x_1, ..., x_n mit $x=\sum x_i$, so daß

$$(f_1+f_2+...+f_n)^*(x)=f_1^*(x_1)+f_2^*(x_2)+...+f_n^*(x_n) \tag{3.106}$$

Die *Large Deviation Rate Function* des Gesamtaggregates kann bei einer bestimmten Verteilung der Ressourcen auf die zum Vergleich herangezogenen Teilaggregate folglich die Summe der *Large Deviation Rate Functions* der Teilaggregate erreichen. Die Überschreitungswahrscheinlichkeit für den Füllstand des Wartespeichers wäre in diesem Fall also bedeutend geringer als bei den Teilaggregaten.

Gleichung (3.106), verknüpft mit einer vergleichsweise einfachen Überlegung, führt nun auf die gewünschten Aussagen zur Aggregation [57]. Spaltet man nämlich das Gesamtaggregat in Teilaggregate auf, denen jeweils die Kapazität $C_{A,i}t+S_{A,i}$ zugeordnet ist, so wird mindestens ein Aggregat über nur gleich viele oder weniger Ressourcen verfügen können als x_i aus Gleichung (3.106): $C_{A,i}t+S_{A,i}\le x_i$. Andernfalls wäre $\sum x_i<\sum C_{A,i}t+S_{A,i}$.

An dieser Stelle kommen die bei der Einführung der Tschernoff-Schranke erwähnten besonderen Eigenschaften der Young-Fenchel-Transformierten des Logarithmus der momentenerzeugenden Funktion $I(C_At+S_A)$ des Ankunftsprozesses des Gesamtaggregates bzw. $I_i(C_{A,i}t+S_{A,i})$ der Teilaggregate, vgl. (3.35), zum Tragen. Jenseits der Stelle $C_At+S_A=\mathrm{E}\{A_N(-t,0)\}$ bzw. der entsprechenden Stellen der Teilaggregate, wo die Transformierte ihr Minimum 0 erreicht, ist diese konvex, also streng monoton wachsend und damit auch positiv. Mit dem Teilaggregat *i*, für das $C_{A,i}t+S_{A,i}\le x_i$ gilt, kann deshalb folgende Ungleichungskette verifiziert werden:

$$\begin{aligned} &I(x=C_At^*+S_A)\ge I_i(x_i)\ge I_i(C_{A,i}t^*+S_{A,i})\ge I_i(C_{A,i}t_i^*+S_{A,i})\\ &\ge \min_j I_j(x=C_{A,j}t_j^*+S_{A,j}) \end{aligned} \tag{3.107}$$

t^* bezeichnet in dieser Kette das *t*, für das die *Large Deviation Rate Function* $I(C_At+S_A)$ des Gesamtaggregates ihr Minimum erreicht, t_i^* die entsprechende, in der Regel natürlich nicht mit t^* übereinstimmende Stelle für $I_i(C_{A,i}t+S_{A,i})$. In (3.107) folgt die erste Relation aus Gleichung (3.106), weil die speziellen Summanden $f_i=I_i$ alle positiv sind. Da x_i so ausgewählt worden ist, daß $x_i\ge C_{A,i}t+S_{A,i}$ und I_i streng monoton wachsend ist, ist die zweite Relation korrekt. Die *Large Deviation Rate Function* des Teilaggregates *i* erhält man erst, nachdem man I_i bezüglich *t* minimiert hat. Die entsprechende Stelle sei t_i^*. Obwohl der letzte Schritt im Grunde die Abschätzung verschlechtert, kann aus der Ungleichungskette (3.107) die Schlußfolgerung gezogen werden,

daß die Überschreitungswahrscheinlichkeit des Aggregates nicht schlechter werden kann als die des schlechtesten Teilaggregates, wenn die insgesamt zur Verfügung stehenden Ressourcen in beiden Fällen gleich sind. Teilaggregate, deren Dienstgüteanforderungen übereinstimmen, benötigen daher bei vollständiger Aggregation weniger Ressourcen.

Wenn die Teilaggregate sehr gut zueinander passen, kann die aus der vollständigen Aggregation resultierende *Large Deviation Rate Function* sogar die Summe der *Large Deviation Rate Functions* der Teilaggregate übertreffen:

$$I(C_A t^* + S_A) \geq \sum_j I_j(C_{A,j} t_j^* + S_{A,j}) \tag{3.108}$$

Dies trifft zumindest dann zu, wenn alle Teilaggregate (und entsprechend auch das Gesamtaggregat) für $t = t^*$, der Infimumstelle des Gesamtaggregates, ihr Supremum bezüglich s einheitlich an der Stelle s^* annehmen [57].

Um die Verbindung zu den früheren Überlegungen zum pufferlosen Fall herstellen zu können, benötigt man eine weitere, Gleichung (3.106) ergänzende Eigenschaft der Young-Fenchel-Transformierten. Wenn die Funktionen f_i differenzierbar in s sind und außerdem der Betrag der Ableitung für jede Folge s_i, die für $i \to \infty$ gegen einen Punkt auf dem Rande des Gebietes konvergiert, in denen sie endlich sind, gegen unendlich strebt, können die x_i aus Gleichung (3.106) berechnet werden aus $x_i = \frac{\mathrm{d}}{\mathrm{d}s} f_i(s)\big|_{s=s^*}$, wobei s^* die eindeutige Lösung der Gleichung $x = \frac{\mathrm{d}}{\mathrm{d}s} f(s)$ des Gesamtaggregates bezeichnet [57].

Angewandt auf das Problem der Aggregation im pufferlosen Fall ($t = 1$ in Formel (3.87)) sollte zunächst der Arbeitspunkt s^* des Gesamtaggregates bestimmt werden, so wie im Zusammenhang mit Abb. 3.9 diskutiert. Bei dieser Vorgehensweise steht danach auch dann die insgesamt zur Verfügung stehende Rate C_A fest, wenn die obere Schranke für die Überschreitungswahrscheinlichkeit und eben nicht die Rate C_A vorgegeben ist. Würde nun die insgesamt zur Verfügung stehende Rate C_A auf die Teilaggregate nach der Vorschrift $C_{A,i} = x_i = \frac{\mathrm{d}}{\mathrm{d}s} f_i(s)\big|_{s=s^*}$ verteilt, entstünden Aggregate mit ganz unterschiedlichen Überschreitungswahrscheinlichkeiten. Wegen Gleichung (3.106) steht aber fest, daß keine der Kurven $s\frac{\mathrm{d}}{\mathrm{d}s} f_i(s) - f_i(s)$ (entspricht der Abb. 3.9 zugrundeliegenden Funktion (3.85) für $f_i(s) = \ln M_i(s)$ ohne den Korrekturterm von Rao-Baradur) an der Stelle $s = s^*$ den Wert $s\frac{\mathrm{d}}{\mathrm{d}s} f(s^*) - f(s^*)$ erreichen würde, der zur Begrenzung der Überschreitungswahrscheinlichkeit auf die vorab festgesetzte Grenze notwendig ist. Da – wie im Zusammenhang mit den früheren Überlegungen zur Aggregation im pufferlosen Fall festgestellt – die Funktionen $s\frac{\mathrm{d}}{\mathrm{d}s} f_i(s) - f_i(s)$

streng monoton wachsend sind, müßten dazu die Arbeitspunkte s_i^* der Teilaggregate größer als s^* sein, also mehr Bandbreite reserviert werden. Diese Schlußfolgerung stimmt mit den früheren Überlegungen überein.

Die Berücksichtigung von Wartespeicher bei der Untersuchung der Effizienz der Aggregation ist nicht zuletzt deswegen etwas schwieriger, weil die Abhängigkeit der momentenerzeugenden Funktion von t weitaus komplizierter als die von s ist. Wird einem Aggregat eine Verbindung hinzugefügt oder entnommen, ändern sich meist sowohl s^* als auch t^* in einer allgemein nicht näher quantifizierbaren Weise. Um so bemerkenswerter ist es, daß die Eigenschaften der Young-Fenchel-Transformierten des Logarithmus von momentenerzeugenden Funktionen bei Vernachlässigung der Verfeinerung von Rao-Baradur dennoch die Vorteile der vollständigen Aggregation von Verbindungen im Vergleich zu Teilaggregaten aufzeigen.

Die von diesen Aussagen nicht abgedeckte Verfeinerung von Rao-Baradur ist für $N \to \infty$, also große Aggregate, zunehmend von untergeordneter Bedeutung. Weitergehende Aussagen für kleine und mittlere Aggregate sind schwierig. Immerhin enthält (3.77) mit dem Term $\mathrm{Var}_{s^*}\{A_N(-t^*,0)\}s^{*2}$ unter der Wurzel das zweite Glied der Taylorreihenentwicklung des Logarithmus der momentenerzeugenden Funktion, also einen signifikanten Teil der *Large Deviation Rate Function* des *Many Sources Asymptotic*. Bei als *Fractional Brownian Motion* modellierten Ankunftsprozessen mit identischem Hurst-Parameter H läßt sich der Term $\mathrm{Var}_{s^*}\{A_N(-t^*,0)\}s^{*2}$ sogar leicht auf die Form $2N^2 I(C_A t^* + S_A)$ bringen. Dazu muß der analytisch leicht zu gewinnende Zusammenhang zwischen t^* auf der einen und C_A, S_A sowie den Parametern der Ankunftsprozesse auf der anderen Seite lediglich in die (nicht mehr von s abhängige) zweite Ableitung des Logarithmus der momentenerzeugenden Funktion eingesetzt und auch s^{*2} entsprechend ersetzt werden. Mit den allgemeinen Eigenschaften der Funktion $I(.)$ kann gezeigt werden, daß auch hier wiederum das Aggregat besser abschneidet als das schlechteste der Teilaggregate.

Verbunden mit dem Hinweis auf weiterführende Literatur [114] sei abschließend noch darauf hingewiesen, daß Verbindungsannahmeentscheidungen mit dem *Many Sources Asymptotic* nicht immer auf konkave von den Verbindungsannahmegrenzkurven eingeschlossene Bereiche führt.

3.3.4 Aggregationsstrategie

Keines der hier im Detail besprochenen Verfahren zur Berechnung des Ressourcenbedarfs von Aggregaten, d. h. weder das deterministische, noch das auf den pufferlosen Fall angewandte *Large Deviation* Prinzip oder das *Many Sources Asymptotic*, liefert einen theoretisch motivierten Ansatz-

punkt für eine Strategie, welche Verbindungen mit identischen Anforderungen an die Paketverlustwahrscheinlichkeit und maximalen Verzögerungszeiten trennt. Das deterministische Verfahren läßt sich sogar auf hinsichtlich der maximalen Verzögerungszeiten nicht identische Verbindungen anwenden. Daher kann man sich bei der Wahl einer geeigneten Aggregationsstrategie von praktischen Erfordernissen leiten lassen.

Fraglos liegt die Zuordnung von Verkehrsströmen zu unterschiedlichen Aggregaten nahe, wenn auf diese Art und Weise Netzdienste mit ganz unterschiedlichen Dienstgüteparadigmen angeboten werden. Die Optionen reichen von völliger Verlustfreiheit, deterministischen Obergrenzen für die Ende-zu-Ende-Verzögerungszeiten, über Zusagen statistischer Natur, wie etwa die Zusicherung beschränkter Überschreitungswahrscheinlichkeiten, zu qualitativen Aussagen, wie etwa das Versprechen, grundsätzlich geringe Verluste und niedrige Verzögerungszeiten sicherstellen zu können.

In einigen Fällen können technische Faktoren die Aggregationsentscheidung beeinflussen. Das Problem der Berechnung des Ressourcenbedarfs eines Aggregates aus als *Fractional Brownian Motion* modellierten Ankunftsprozessen mit identischem Hurst-Parameter H kann beispielsweise mit dem *Many Sources Asymptotic* geschlossen gelöst werden, während periodische Ein-Aus-Quellen und viele andere denkbare Ankunftsprozesse eine kostspielige Iteration durch eine große Anzahl von Werten von t erfordern. Auch hier macht eine Trennung in Teilaggregate Sinn, um den geringeren Berechnungsaufwand für den geschlossen lösbaren Teil des Aggregates stärker zur Geltung zu bringen.

Auch innerhalb einer Menge von periodischen Ein-Aus-Quellen ist eine Trennung denkbar. Die Berechnung der benötigten Ressourcen mit dem *Many Sources Asymptotic* muß sich ja auf eine nicht zu große Anzahl ausgewählter diskreter Werte von t beschränken. Da der zu untersuchende Wertebereich durch das kleinste gemeinsame Vielfache der Periodenlänge der Quellen bestimmt ist, könnte bei Trennung der Verbindungen der Ressourcenbedarf eines Teilaggregates aus Quellen kleiner Periodenlänge ebenfalls entweder genauer oder schneller ermittelt werden als der eines Gesamtaggregates.

3.4 Dynamisches Bandbreitemanagement

Das Zusammenfassen von Verbindungen zu Aggregaten allein reduziert nur den Aufwand für das Speichern von und den Zugriff auf Zustandsinformationen. Solange jeder Verbindungsauf- und

-abbau oder die Änderung einzelner Verbindungen auch unmittelbar entsprechende Steuerungsaktivitäten auf der Ebene des Aggregates auslösen, wird sich der Aufwand für die Verbindungsteuerung nicht merklich reduzieren. Dieses Problem kann nur durch dynamisches Bandbreitemanagement gelöst werden.

Unter dynamischem Bandbreitemanagement wird in dieser Arbeit eine in der Steuerungsebene eines Aggregationsknotens angesiedelte verbindungsorientierte Verkehrssteuerungsfunktion verstanden, die für ein Aggregat eine dem aktuellen Bedarf der Verbindungen dynamisch angepaßte Bandbreite reserviert und den Aufwand für die Verbindungssteuerung im Aggregationsbereich durch Überreservierung reduziert. Dem übergeordneten Ziel folgend, die Verkehrssteuerung dezentral zu organisieren, werden hier nur Verfahren in Betracht gezogen, die nicht mehr Informationen benötigen, als während der Signalisierung zur Modifikation der Reservierung unter realistischen Randbedingungen ermittelt werden können.

Unter anderen Vorzeichen sind für Knoten in ATM-Netzen verbindungsorientierte Mechanismen untersucht worden, welche die Rate von in ATM als virtuelle Pfade bezeichneten Verbindungen für Aggregate dynamisch an den aktuellen Bedarf anpassen. Im Unterschied zu *Integrated Services* Netzen ist bei ATM der gesamte Verkehr Verbindungen zugeordnet. Solange für jede Verbindung eine Rate reserviert ist, kann das dynamische Bandbreitemanagement die gesamte verfügbare Bandbreite auf virtuelle Pfade so verteilen, daß der Durchsatz, d. h. die Summe der effektiven Bandbreite aller Verbindungen integriert über die Zeit, maximiert wird [178]. Ohne zusätzliche Zielfunktionen und/oder Randbedingungen würden in wirklich diensteintegrierenden Netzen die Verkehrsklassen mit Dienstgütegarantien den verbindungslosen oder zumindest reservierungslosen *Best-Effort*-Verkehr komplett verdrängen.

In [150, 193, 194, 123, 124] werden Verfahren untersucht, die in regelmäßigen Zeitabständen, den Nachverhandlungszeitpunkten, die Anzahl der Verbindungen in der virtuellen Pfadverbindung feststellen und für das Intervall bis zur nächsten Nachverhandlung eine neue Bandbreite so bestimmen, daß im zeitlichen Mittel eine vorgegebene Blockierungswahrscheinlichkeit ϵ_B nicht überschritten wird.

In [124] wird für diese Aufgabe die Anwendbarkeit einer einfachen Reservierungsvorschrift der Form

$$C_A(N)=\lceil N + f(\epsilon_B, \Delta t, N)\sqrt{N}\rceil \tag{3.109}$$

für eine virtuelle Pfadverbindung untersucht, die aus N identischen aktiven Verbindungen mit konstanter Bandbreite 1 besteht. Der Vorfaktor $f(\epsilon_B, \Delta t, N)$ soll einerseits den Einfluß der Länge

des Intervalls Δt zwischen periodisch ausgeführten Nachverhandlungen erfassen und andererseits durch Anpassung an die Anzahl N aktiver Verbindungen einen größeren Teil des statistischen Multiplexgewinns realisieren, der bei Berücksichtigung ganz oder abschnittsweise paralleler anderer virtueller Pfadverbindungen möglich ist [194]. Wie zuvor auch sind die Ankunftsabstände und Haltedauern der Verbindungen negativ exponentiell verteilt. Formel (3.109) liegt die Annahme zugrunde [150, 194], daß im blockierungsfreien Bediensystem M/M/∞ die Anzahl der Verbindungen im Bediensystem poissonverteilt mit Mittelwert A (Angebot) und Standardabweichung $\sqrt{A}$ sind und so die Zahl der aktiven Verbindungen N zum Nachverhandlungszeitpunkt meistens eine gute Schätzung für das Verkehrsangebot A ist.

Zwei Studien [123, 124] versuchen simulativ die Abhängigkeiten des Vorfaktors $f(\epsilon_B, \Delta t, N)$ genauer zu quantifizieren, um so Ansatzpunkte für einen heuristischen Algorithmus zu finden. Die Ergebnisse in [123] zeigen unter anderem, daß die Abhängigkeit des Faktors $f(\epsilon_B, \Delta t, N)$ von N nicht sehr stark ausgeprägt ist, wenn die Übertragungsabschnitte so dimensioniert sind, daß auch ohne statistischen Multiplexgewinn die angestrebte Rufblockierungswahrscheinlichkeit erreicht wird. Das hat den Vorteil, daß bei vorgegebenen Kosten für die Signalisierung auch ein optimales Nachverhandlungsintervall simulativ bestimmt werden kann. Weiter an der Idee eines adaptiven Vorfaktors festhaltend, ersetzt die Nachfolgearbeit [124] $f(\epsilon_B, \Delta t, N)$ durch einen Vorfaktor, der statt unmittelbar von N nunmehr abhängig von der Bandbreite ist, die dem virtuellen Pfad zusätzlich zur momentanen Bandbreite zugeteilt werden kann, wenn in jedem Knoten entlang des Pfades die noch freie Bandbreite gleichmäßig auf die einzelnen virtuellen Pfadverbindungen verteilt wird. Die Auswertung von Simulationsergebnissen führt zu einem durch eine Gerade annäherbaren Zusammenhang zwischen f und der auf die Anzahl der Verbindungen bezogenen freien Bandbreite. Unter Verwendung dieses Zusammenhangs könnte ein Aggregationsknoten den optimalen, also die Rufblockierungswahrscheinlichkeit minimierenden Vorfaktor f in Abhängigkeit von der aktuellen Auslastung entlang des Pfades bestimmen. Diese Konzept wirft jedoch einige Fragen auf. Zum einen geht aus den zitierten Arbeiten [123] und [124] nicht hervor, wie die Bestimmung der momentanen Lastsituation zeitlich mit der zur eigentlichen Nachverhandlung erforderlichen Signalisierungsprozedur gekoppelt ist. Zum anderen ist die für das im Grunde heuristische Verfahren zugrunde gelegte Annahme homogener Aggregate zu sehr einschränkend.

Berücksichtigt man die Interaktionen zwischen virtuellen Pfadverbindungen nicht, kann das System als Markoff-Zustandsprozeß modelliert und exakt mit Hilfe der Kolmogoroff-Vorwärtsgleichung für die zeitabhängigen Zustandswahrscheinlichkeiten gelöst werden [122]. Die Berechnung der erforderlichen Rate für die virtuelle Pfadverbindung ist auf diese Weise jedoch zu aufwendig, nicht

zuletzt weil das System von Differentialgleichungen eines Bediensystems des Typs M/M/N für verschiedene Bedienkapazitäten *N*, also verschiedene Blockierungszustände gelöst werden muß. *N* entspricht der momentanen Bedienkapazität des Aggregates.

Virtamo und Aalto [193] schlagen aus diesem Grunde ein Verfahren vor, das ausgehend von einem transienten blockierungsfreien Bediensystem M/M/∞ die zeitabhängigen Zustandswahrscheinlichkeiten berechnet, die endliche, regelmäßig anzupassende Bedienkapazität *N* des Aggregates zunächst außer acht lassend. Die zeitabhängigen Zustandswahrscheinlichkeiten gewinnen sie wiederum unter der Voraussetzung identischer Einzelverbindungen durch diskrete Faltung der Wahrscheinlichkeitsverteilungen der Anzahl der Quellen, die zum Zeitpunkt der Nachverhandlung schon aktiv waren und immer noch aktiv sind, und des Beitrages der Quellen, die seit der Nachverhandlung hinzugekommen sind und auch noch aktiv sind. Die resultierende, zeitabhängige Zustandsverteilung im System M/M/∞ muß eigentlich durch diskrete Faltung der Wahrscheinlichkeitsverteilungen der Anzahl dieser Quellen bzw. bei Verwendung der erzeugenden Funktion [122] durch die Rücktransformation ermittelt werden. Zur Vereinfachung wird aber statt dessen die oben angesprochene *Probability Shift* Methode verwendet. Zur Berechnung der im zeitlichen Mittel auftretenden Blockierungswahrscheinlichkeit im nun begrenzten System M/M/N wird abhängig von der zu prüfenden Rate *N* das System M/M/∞ in ein System M/M/N übergeführt, indem am Zustand *N* ein Quellenterm hinzugefügt wird. Die gesuchte Blockierungswahrscheinlichkeit erhält man dann durch Lösen einer Integralgleichung mit den zuvor näherungsweise unabhängig von *N* errechneten zeitabhängigen Zustandsvariablen des Systems M/M/∞.

Ein im Rahmen des ACTS REFORM Projektes für das Management von virtuellen Pfaden in ATM-Netzen entwickeltes Konzept [192] rückt weitgehend von der Rufblockierung als primär die Dimensionierung von virtuellen Pfaden bestimmende Größe ab. Nach Auffassung des Autors ist dies ein berechtigter Ansatz. Denn in einem Netz, das Verkehr sowohl verbindungsorientiert als auch verbindungslos vermittelt, ohne die Ressourcen statisch zu partitionieren, wird Rufblockierung vermutlich von so geringer Bedeutung sein, das sie als Dimensionierungskriterium für Verbindungsaggregate ausscheidet.

Das Konzept sieht vor, innerhalb einer *Admissable Zone*, vgl. Abb. 3.15, die längerfristig durch die Rufblockierung bestimmt sein mag, eine Nachverhandlung nicht mehr in konstanten Abständen, sondern immer dann auszulösen, wenn der Ressourcenbedarf der Verbindungen des virtuellen Pfades eine sogenannte *Working Zone* verläßt. Eine oberhalb der *Working Zone*, des eigentlich erlaubten Arbeitsbereichs, angeordnete *Buffer Zone* kann verhindern, daß die eventuell zeitaufwendige Nachverhandlung des virtuellen Pfades den Verbindungsaufbau der neu hinzukommenden Verbindung unnötig verzögert. Sofern diese Reserve für die neue Verbindung ausreicht, kann die

Bandbreite des virtuellen Pfades dann asynchron nachverhandelt werden. Die zentrale Frage, wie die neue Bandbreite und *Working Zone*, also die obere und untere Nachverhandlungsschwelle, zu bestimmen sind, bleibt offen.

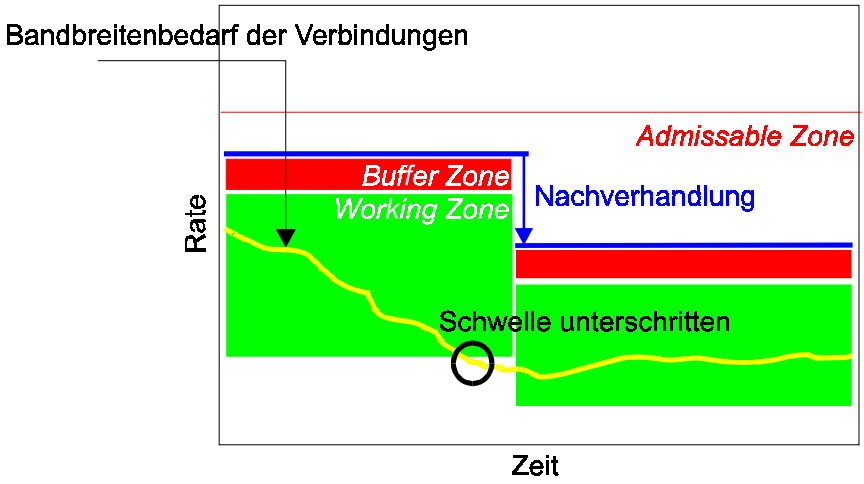

Abb. 3.15: *Das in [192] vorgeschlagene dynamische Bandbreitemanagement*

Abgesehen von der *Buffer Zone*, die an sich auch nur dafür sorgt, daß asynchron nachverhandelt werden kann, wird aber in [155] dieses Verfahren analytisch untersucht. Identisch negativ exponentiell verteilte Ankunfts- und Haltedauern sowie einheitliche Bandbreite vorausgesetzt, kann der Zustand des Bediensystems innerhalb einer *Working Zone*, die Anzahl der Verbindungen im Aggregat, als Markoff-Prozeß modelliert werden. Dieser Prozeß startet unmittelbar nach der Nachverhandlung in dem Zustand, der die Nachverhandlung ausgelöst hat (dem in Abb. 3.16 grau hinterlegten sogenannten *Allocation State*) und endet durch Übergang in einen neuen Markoff-Prozeß, wenn die untere oder obere Nachverhandlungsschwelle durchbrochen wird. Da der in Abb. 3.16 angedeutete Gesamtprozeß irreduzibel ist, wird jeder der Teilprozesse unendlich oft durchlaufen. Es sind nur Wechsel aus einer der beiden benachbarten *Working Zones* in den *Allocation State* möglich. Die Autoren von [155] nutzen diesen Umstand aus, indem sie die Zustandsübergänge zwischen den *Working Zones*, $l_i \rightarrow k_{i-1}$ (nicht dargestellt) und $k_{i+1} \rightarrow u_i$ ersetzen durch neue Übergänge $l_i \rightarrow k_i$ bzw. $u_i \rightarrow k_i$ und ignorieren die Zeitspanne, die nach dem Verlassen der *Working Zone* über einen der beiden Randzustände bis zum Übergang in den *Allocation State* vergeht. Danach sind die Prozesse innerhalb der *Working Zones* voneinander entkoppelt und stationär und können durch voneinander unabhängige Zufallsvariablen X_i beschrieben werden.

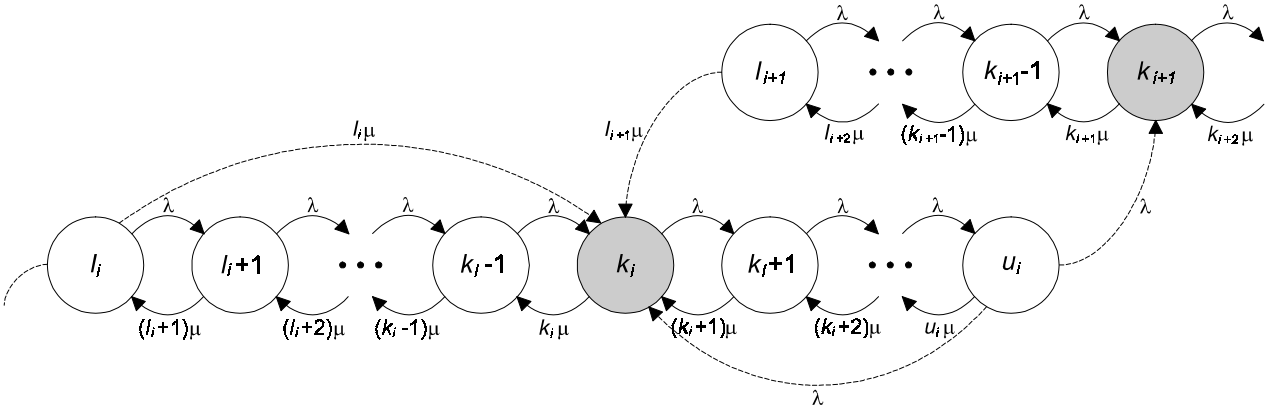

Abb. 3.16: *Markoff-Modell für den Zustandsprozeß innerhalb einer Working Zone [155], wenn das Aggregat aus identischen Quellen besteht.*

Die stationären Zustandswahrscheinlichkeiten können auf diese Weise unabhängig für jede *Working Zone* bestimmt werden. Die mittlere Nachverhandlungsrate aus der *Working Zone i* hinaus ist dann

$$\mathrm{P}\{X_i = u_i\}\cdot\lambda + \mathrm{P}\{X_i = l_i\}\cdot l_i\cdot\mu$$

und die im Mittel während des Aufenthaltes ungenützte Bandbreite

$$\sum_{j=1}^{u_i} \mathrm{P}\{X_i = j\}\cdot(u_i - j)$$

Multipliziert man beide Größen beispielsweise mit konstanten Kostenfaktoren, erhält man die im Mittel während eines Aufenthaltes in *Working Zone i* anfallenden Kosten. Diese können durch eine geeignete Wahl von l_i und u_i minimiert werden. Zur Berechnung der Gesamtkosten werden die Zustände innerhalb einer *Working Zone i* zu einem Überzustand gruppiert. Man erhält so einen Geburts-Sterbe-Prozeß, also einen Markoff-Prozeß, in dem nur Übergänge zwischen benachbarten Zuständen erlaubt sind. Dessen Zustandswahrscheinlichkeiten, die Aufenthaltswahrscheinlichkeiten der *Working Zones,* können sehr leicht berechnet werden. Der Erwartungswert der Gesamtkosten pro Zeiteinheit ist dann die Summe der mit ihrer Aufenthaltswahrscheinlichkeit gewichteten Kosten der einzelnen *Working Zones*.

Leider führt die lokale Optimierung nicht auf ein globales Optimum, weil die Optimierung einer einzelnen *Working Zone* die Größe und Lage aller anderen *Working Zones* mitbestimmt. Um dennoch auch mit vertretbarem Rechenaufwand eine gute Lösung zu erhalten, wird in der folgenden Untersuchung zunächst um den Systemzustand, der dem Verkehrswert *Y*, der mittleren Anzahl von aktiven Verbindungen, am nächsten liegt, eine erste, optimale *Working Zone* angeordnet. Das System befindet sich im stationären Fall meistens in einem Zustand in der Nähe, so daß die lokale Optimierung auch eine gute Gesamtlösung verspricht. Diese erste *Working Zone* bestimmt dann

schon die obere Nachverhandlungschwelle u_{i-1} und die untere Nachverhandlungsschwelle l_{i+1} der benachbarten *Working Zones*, so daß dort jeweils nur noch die fehlende zweite Schwelle frei bestimmt werden kann. Diese Vorgehensweise wird fortgesetzt, bis der ganze Zustandsraum von *Working Zones* abgedeckt ist. Auf diese Weise entstehen *Working Zones* variabler Größe. Zum Vergleich wird auch der Fall von *Working Zones* konstanter Größe betrachtet.

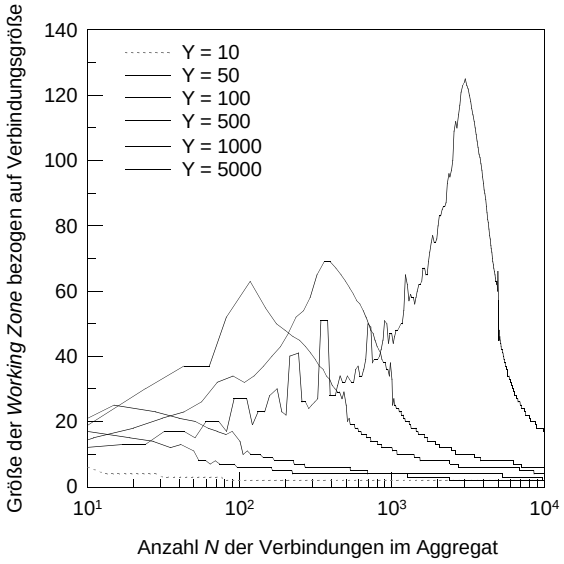

Abb. 3.17: *Größe der der mit dem Markoff-Modell aus [155] berechneten Working Zones (variabler Größe) in Abhängigkeit vom Verkehrswert Y (= Angebot A)*

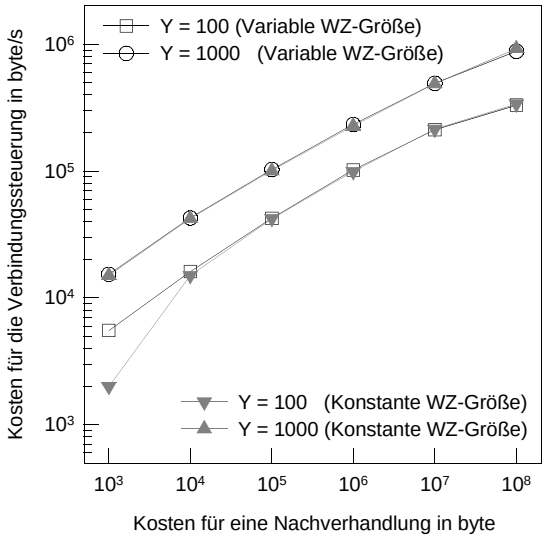

Abb. 3.18: *Vergleich der mit dem Markoff-Modell aus [155] berechneten Gesamtkosten für die Verbindungssteuerung mit Working Zones konstanter bzw. variabler Größe*

Abb. 3.17 bestätigt die Vermutung, daß die Größe der *Working Zones* doch recht stark vom Verkehrswert des Systems abhängt, ganz wie dies auch die in der Literatur [124] verwendete Formel (3.109) auszudrücken versucht. Bei starken Angebotsschwankungen (beispielsweise im Tagesverlauf) sollten deshalb Algorithmen für dynamisches Bandbreitemanagement eine dynamische Anpassung der Größe der *Working Zones* anstreben. Außerdem fällt auf, daß bei statistischen zeitlichen Schwankungen des Aggregates um den Verkehrswert *Working Zones* variabler Größe durchlaufen werden, wenn man das Markoff-Modell aus [155] in der oben beschriebenen zur Minimierung der Kosten für die Verbindungssteuerung einsetzt.

Dagegen widerlegt Abb. 3.18 die in [155] vertretene These, daß sich *Working Zones* variabler Größe bei statistischen Schwankungen des Aggregats um einen konstanten Verkehrswert *Y* positiv auf die Effizienz des dynamischen Bandbreitemanagements auswirken. Dazu sind größere statistische Schwankungen um den Mittelwert *Y* im zugrundeliegenden Markoff-Modell auch einfach zu unwahrscheinlich. Dieses Ergebnis deutet darauf hin, daß bei konstantem Verkehrswert *Y Working Zones* variabler Größe weniger wichtig sind als das Verfahren mit den um die *Allocation States*

angeordneten *Working Zones* an sich (vgl. Abb. 3.15), das neue *Working Zones* um den jeweils aktuellen Zustand des Aggregates herum anordnet und auf diese Weise stets mit Hysterese zwischen *Working Zones* wechselt.

Ein Aggregat mit unterschiedlichen Typen von Verbindungen könnte im Prinzip durch einen mehrdimensionalen Markoff-Prozeß modelliert werden. Anders als in [155] dargestellt, ist eine sinnvolle *Working Zone* in diesem System eindeutig durch ihre untere und obere Nachverhandlungsschwelle gegeben als ein von zwei (Hyper-)Ebenen eingeschlossener Bereich im Zustandsraum. Zwar treten auch hier nur Wechsel zwischen benachbarten *Working Zones* auf, Ausgang- und Zielzustand des Übergangs sind jetzt aber nicht mehr eindeutig bestimmt. Aus diesem Grunde kann eine einzelne *Working Zone* nicht mehr isoliert betrachtet werden.

Wie alle dem Autor bekannten analytischen Verfahren ist also auch dieses auf aus Quellen mit identischer und konstanter Reservierung zusammengesetzte Aggregate beschränkt. In [123] wird diese Vereinfachung sogar mit der sicherlich nicht zutreffenden These gerechtfertigt, daß in zukünftigen diensteintegrierenden Netzen der Verkehr in homogene Teilaggregate zerlegt werden wird. Zudem sind die Ankunftsabstände und Haltedauern der Verbindungen identisch negativ exponentiell verteilt. Obwohl die Analysen so nicht unmittelbar zu einer praktischen Lösung führen, bestätigt vor allem [155] die Bedeutung von dynamischem Bandbreitemanagement für eine skalierbare Verkehrssteuerungsarchitektur für *Integrated Services* Netze und eignet sich sowohl als guter Ausgangspunkt als auch als Referenz zur Leistungsbewertung von heuristischen Verfahren.

4 Verbindungsaggregation am Beispiel von RSVP über ATM

Die in Kapitel 3 besprochenen und theoretisch untersuchten Mechanismen zur Aggregation von Verkehrsströmen sind im Rahmen der vorliegenden Arbeit in einem Linux-basierten Router prototypisch implementiert worden. Dieser Router bildet *Integrated Services* und RSVP auf einen als ATM-Netz realisierten Aggregationsbereich ab und ist daher hinsichtlich der Protokolle und der zweistufigen Netztopologie als ein spezielles Beispiel eines Aggregationsknotens anzusehen. Ausschlaggebend für die Wahl von IP *Integrated Services* und RSVP über ATM war neben aktuellen Fragestellungen zum Zeitpunkt des Beginns der Implementierungsarbeiten [137] vor allem der Umstand, daß es eine solche Lösung erlaubt, die vorrangig interessanten Probleme der Verkehrssteuerung bei nur geringfügigen Eingriffen in die aus wissenschaftlicher Sicht weniger interessanten Protokollabläufe zu untersuchen.

Die wesentlichen Bestandteile der Verkehrssteuerungsarchitektur eines Aggregationsknotens, das Klassifizieren von Paketen, die Warteschlangen und Bedieneinheiten am Eingang von Aggregaten, das Schalten von Aggregatsverbindungen, die Berechnung und Zuweisung der vom Aggregat benötigten Ressourcen und dynamisches Bandbreitemanagement zur Reduzierung des Steuerungsaufwandes, sind ohnehin von den Protokollabläufen weitgehend unabhängig und können daher unverändert in verallgemeinerten, mehrstufige Aggregation einbeziehenden Konzepte aufgehen.

4.1 Integration von IP und ATM

Vor allem bis Mitte der neunziger Jahre galt ATM (*Asynchronous Transfer Mode*), dessen Grundlagen unter anderem in [179] und [89] beschrieben werden, als vielversprechende Lösung zum Aufbau eines diensteintegrierenden paketvermittelnden Breitbandnetzes. ATM entwickelt im Grunde vom Schmalband-ISDN (*Integrated Services Digital Network*) her bekannte, dort aber für

ein leitungsvermittelndes Netz entworfene Dienstkonzepte und Protokolle weiter. Virtuelle Kanäle, Verkehrsdeskriptoren, Quellflußsteuerung und nicht zuletzt Verbindungsannahmesteuerung auf der Basis von effektiver Bandbreite abstrahieren dabei von der Paketebene mit dem Ziel, die Vorteile von Verbindungsbezug und Signalisierung bei der Steuerung von Diensten in leitungsvermittelnden Netzen mit der flexibleren Zuordnung von Übertragungskapazität paketvermittelnder Netze zu kombinieren. Ein umfassendes Verkehrsmanagementkonzept [103, 11] definiert für die Dienstklassen CBR (*Constant Bit Rate*), rt/nrt-VBR (*Real-Time/Non-Real-Time Variable Bit Rate*), UBR (*Unspecified Bit Rate*), ABR (*Available Bit Rate*) und GFR (*Guaranteed Frame Rate*) [11], wie qualitativ doch sehr unterschiedliche Dienstgüteanforderungen konsistent auf Funktionen der Netzelemente abgebildet werden können. Während aber die Standardisierung von ATM um die Bereitstellung gelegentlich mit dem Attribut *Killer* versehener Anwendungen wie *Video on Demand*, Multimedia-Konferenzen und *Virtual Reality* kreiste, welche die neue Technologie zwingend erforderlich machen würden, oder mit einer zwar effizienten, aber sehr komplexen Ratenregelung für Datenanwendungen (ABR) beschäftigt war, hat das *World Wide Web* die an sich viel ältere IP-Technologie gleichsam als Synonym für weltweite Vernetzung, Globalisierung und *New Economy* etabliert.

Da IP-Netze traditionell einen auf *Best-Effort* basierenden Netzdienst anbieten, arbeitet auch die IETF an Erweiterungen der Internet-Protokollfamilie, um den Anforderungen an Dienstgüte gerecht zu werden. Auf der anderen Seite haben sowohl das ATM Forum als auch die IETF Protokollarchitekturen spezifiziert, um ATM IP-Anwendungen zugänglich zu machen.

LANE (LAN *Emulation*) [9] emuliert *Shared Medium* LANs mit Hilfe einiger zentraler Server und den standardisierten ATM-Signalisierungsprotokollen zum dynamischen Aufbau von virtuellen Verbindungen. Unter Einbeziehung von *Ethernet* und *Token Ring* LAN Segmenten unterstützt LANE den Aufbau eines ausgedehnten Schicht-2-Netzes.

MPOA (*Multiprotocol over ATM*) [12] kann auch dann noch direkte virtuelle Verbindungen schalten, wenn das Aufkommen an *Broadcast*-Verkehr bei größerer Ausdehnung des Netzes immer kritischer wird und eine Segmentierung in über IP-Router miteinander verbundene Subnetze erfordert. Beide, LANE und MPOA, haben gegenüber reinen IP-Netzen aber eigentlich nur den Vorteil, daß für die Verbindungen, für die *Shortcuts*, welche die Konnektivität herstellen, Bandbreite reserviert und so geeignet dimensionierte virtuelle private IP-Netze konfiguriert werden können.

Bei MPOA nehmen am Netzrand eines emulierten LAN (ELAN) plazierte MPOA *Server* Anfragen von MPOA *Clients* zur Auflösung von ATM-Adressen entfernter MPOA/LANE *Clients* (MPC/LEC) entgegen, die nur unter ihrer IP-Adresse bekannt sind. Die MPOA *Server* leiten die

MPOA-Anfragen unter Verwendung von NHRP (*Next Hop Resolution Protocol*) [142] entlang der von IP vorgegebenen Route zum Zielnetz weiter. Wenn die der IP-Adresse zugeordnete ATM-Adresse in einem NHS (*Next Hop Server*) auf diesem Weg oder dem MPOA-Server im Zielnetz zwischengespeichert ist, kann die Anfrage vorzeitig beantwortet werden. Sobald der MPOA *Client* die von ihm angeforderte ATM-Adresse erhält, baut er eine virtuelle ATM-Verbindung (*Shortcut*) zum Ziel auf, deren Pfad das Routing des ATM-Netzes jetzt autonom bestimmt. Abb. 4.1 zeigt die wichtigsten Elemente der MPOA-Architektur.

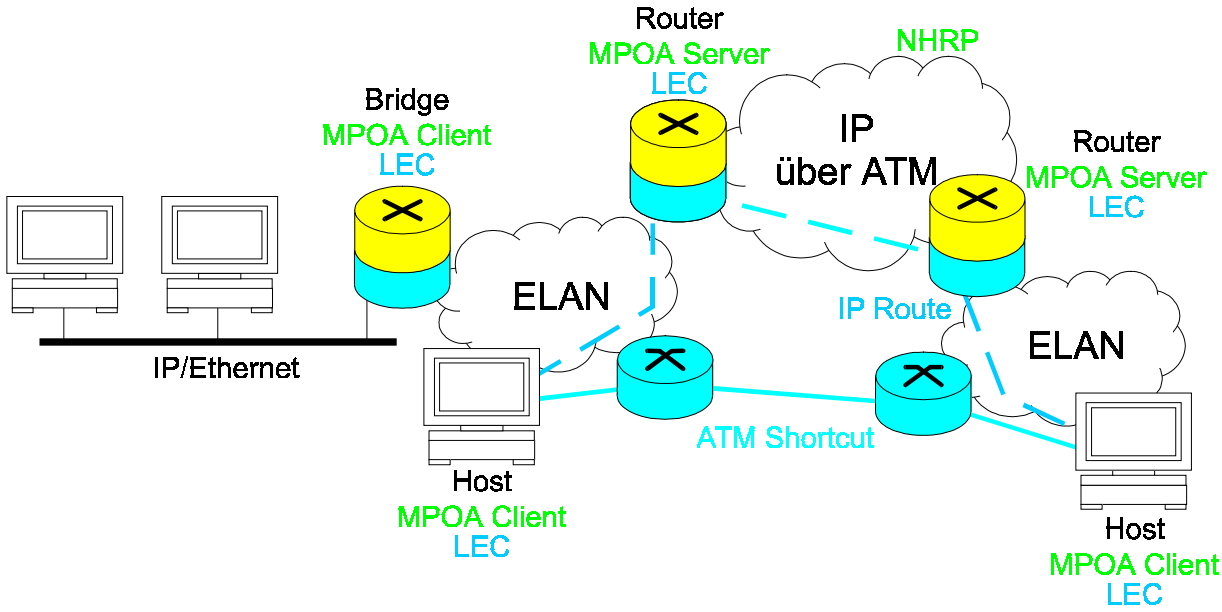

Abb. 4.1: *Auf MPOA basierendes IP über ATM Netz. Nachdem ein MPOA Client auf Anfrage vom MPOA Server die ATM-Adresse des Zielrechners erhalten hat, kann er unter Umgehung der IP-Route eine ATM-Verbindung aufbauen.*

Für den im Rahmen dieser Arbeit untersuchten RSVP über ATM Prototypen genügt auch das auf ein LIS (*Logical IP Subnetwork*) und *Unicast* begrenzte CLIP (*Classical IP over ATM)* [125] zur Auflösung von ATM-Adressen. Dazu registrieren sich alle ATM-Endsysteme im LIS beim ATMARP *Server*, der eine Tabelle mit IP und ATM-Adressen sowie offene virtuelle Verbindungen zentral speichert und auf Anfrage weitergibt.

Als Alternative zur Dienstklasse UBR oder Verbindungen anderer ATM-Dienstklassen mit einer von der einzelnen Anwendung nicht beeinflußbaren Dienstgüte bietet sich *Differentiated* UBR [13] an. Diese Erweiterung von UBR ordnet Verbindungen einen sogenannten *Behaviour Class Selector* Parameter zu. Im Gegensatz zu *Differentiated Services* werden die ATM-Zellen mit diesem Parameter nicht gekennzeichnet und der Verbindungsbezug in der Steuerungsebene beibehalten. Da ein ATM-Knoten allerdings auf der Basis des Zellkopfes einen Verbindungsbezug herstellen und so rasch auf den der entsprechenden Verbindung zugeordneten *Behaviour Class Selector* Parameter

zugreifen kann, können in der Nutzerebene ähnliche Mechanismen wie bei *Differentiated Services* zur differenzierten Behandlung des Verkehrs in Überlastsituationen von auf *Per Hop* Basis zusammengestellten Aggregaten, den *Behaviour* Aggregates, eingesetzt werden. Aus Sicht der Verkehrssteuerung wird in der Nutzerebene also der Verbindungsbezug analog zu *Differentiated Services* aufgegeben. In einem IP über ATM Netz können damit alle Netzknoten in konsistenter Art und Weise zu einem dem *Differentiated Services* Modell entsprechenden Pfad Ende-zu-Ende verknüpft werden.

Wirklich sinnvoll ist eine solche Kombination von IP und ATM aber nur dann, wenn zur gleichen Zeit auch das Alleinstellungsmerkmal von ATM, des bislang einzigen etablierten Standards, der in paketvermittelnden Netzen eine mit leitungsvermittelnden Netzen vergleichbare Dienstgüte Ende-zu-Ende realisiert, nicht der Flexibilität der technologieunabhängigen IP-basierten Netzwerkschicht geopfert, sondern mit ihr gewinnbringend zu einer vor allem in bezug auf Dienstgüte mächtigen Netzarchitektur verknüpft wird.

Eine minimalistische Implementierung von *Integrated Services* und RSVP über ATM [21] – im folgenden in Anlehnung an die verbreitete Sprechweise verkürzend als RSVP über ATM bezeichnet – verlangt von IP-Routern am Rande eines ATM-Netzes nur, auf Anforderung von RSVP virtuelle ATM-Verbindungen mit geeigneten Dienstgüteattributen aufzubauen, diese gegebenenfalls zu modifizieren und schließlich auch wieder abzubauen. Die *Edge Devices*, vgl. Abb. 4.2, genannten Router tauschen RSVP-Nachrichten über gemeinsam mit *Best-Effort* Verkehr benutzte ATM-Verbindungen aus, die zuvor beispielsweise mit CLIP oder einem der anderen Mechanismen eingerichtet worden sind. Die Datenverbindungen sind davon unabhängig und werden im einfachsten Fall für jeden *Integrated Services* Verkehrsstrom getrennt mit Hilfe der Signalisierungsprozeduren des UNI 4.0 [10] aufgebaut. Die dazu notwendige Abbildung der *Integrated Services* auf ATM-Dienstklassen und alle zur Steuerung von ATM-Verbindungen zwischen IP-Routern notwendigen Parameter werden ausführlich in [79] diskutiert.

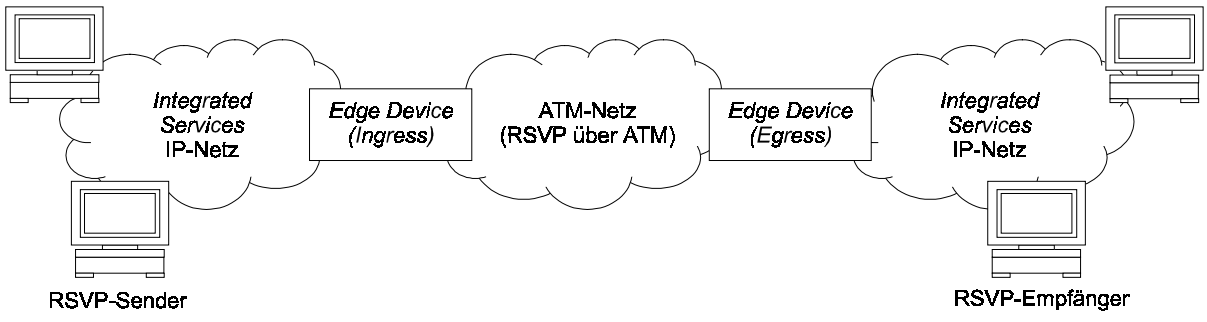

Abb. 4.2: *Referenznetzkonfiguration für RSVP über ATM [79]*

Potentielle Erweiterungen dieses minimalistischen Entwurfs betreffen insbesondere das Einrichten eines vor *Best-Effort* Verkehr geschützten RSVP *Control* VC zum verlustfreien Transport der RSVP-Nachrichten, die möglichst transparente Unterstützung des Mehrpunkt-zu-Mehrpunkt Verbindungsmodells von RSVP durch ATM, Heterogenität in solchen Verbindungen und die dynamische Nachverhandlung der vereinbarten Dienstgüte ohne Unterbrechung des Datenflusses [51, 48, 20, 79, 21, 138]. Anders als die nicht kompatiblen Sitzungsmodelle für *Multicast*, bereitet letzteres seit den entsprechenden Ergänzungen von ATM [14] genauso wenig Probleme wie die in Tabelle 4.1 zusammengestellten Unterschiede bezüglich der Behandlung von Verbindungszuständen oder bezüglich des Spektrums der Dienstklassen.

Tabelle 4.1: *Vergleich von RSVP und ATM-Signalisierung. Ergänzungen von ATM [14] tragen den Anforderungen nach dynamischen Verbindungen, deren Attribute bei ununterbrochener Datenübertragung verändert werden können, Rechnung.*

RSVP [32]	ATM UNI 4.0 [10]
Reservierung für unidirektionale Verbindung	Bidirektionale Verbindung und Reservierung
Empfänger steuert Reservierung	Sender steuert Verbindung
Bei Blockierung *Best-Effort* Datenpfad	Bei Blockierung keine Konnektivität
Reservierung dynamisch	Reservierung dynamisch [14]
Mehrpunkt-zu-Mehrpunkt Verbindungsmodell	Punkt-zu-Mehrpunkt Verbindungsmodell
Heterogene Dienstgüte innerhalb Verbindung	Homogene Dienstgüte innerhalb Verbindung
Periodische Nachrichten erneuern *Soft States*	*Hard States*
Guaranteed, *Controlled Load* und *Best-Effort*	CBR, rt/nrt-VBR, GFR, ABR, UBR [11]

Obwohl also einige Konzepte zur Integration von IP und ATM standardisiert sind, die in Ansätzen Dienstgüte auf Ende-zu-Ende-Basis realisieren können, beschränken sich kommerzielle Anwendungen beider Technologien auf die Bereitstellung einer breitbandigen Internet-Infrastruktur oder virtueller privater Netze bestehend aus Pfaden mit relativ statischen Dienstgüteattributen.

Das mag zwar zum Teil an der meist unzureichenden Infrastruktur im Zugangsnetzbereich und damit an der geringen Attraktivität genau der Anwendungen liegen, die hohe, aber unerfüllbare Dienstgüteanforderungen an die Netze stellen, ist aber wohl vor allem darauf zurückzuführen, daß ein Ende-zu-Ende-Dienst nur dann zustande kommt, wenn sich alle Instanzen ausgehend von den Nutzern, ihren Anwendungsprogrammen bis hin zu jedem einzelnen Netzelement unter Umständen in Netzen mehrerer Betreiber über die Attribute dieses Dienstes verständigen und diese dann umsetzen können. In zunehmend heterogener werdenden Netzen mit hinsichtlich Dienstgüte mehr oder weniger ausgeprägter Funktionalität können über Konnektivität hinausgehende Dienstattribute

daher kaum verwirklicht werden. Vor diesem Hintergrund stellt das in der Vermittlungsschicht verbindungslose, auf IP basierende Internet, das bis heute praktisch ausschließlich Dienste auf *Best-Effort* Basis – diese aber mit großem Erfolg – in einem extrem heterogenen Umfeld anbietet, die aufwendigen Steuerungsfunktionen und Signalisierungsprotokolle der Vermittlungsschicht verbindungsorientierter Netze zur Bereitstellung von Dienstgüte sogar grundsätzlich in Frage. In Netzen, in denen IP und ATM gemeinsam betrieben werden, sind Steuerungsfunktionen der Vermittlungsschicht, insbesondere die Wegesuche, jedenfalls nicht integriert, so daß sich ATM aus IP-Sicht zwar als leistungsfähige, aber funktionell nicht besonders ausgezeichnete Schicht-2-Technologie darstellt: ATM-Verbindungen zwischen IP-Routern, an denen unter Umständen sehr viele ATM-Knoten beteiligt sind, werden zu Links abstrahiert. Man spricht in diesem Zusammenhang auch von einem *Overlay* Modell, das, wie beispielsweise in [188] ausgeführt wird, auf IP-Ebene neue Nachbarschaften erzeugt mit der Folge, daß IP-Router unzureichend aggregierte Topologieinformation austauschen.

Dagegen könnten in einem integrierten Ansatz Routing-Protokolle Informationen über alle Links im Netz verbreiten, aus denen sich Router und Switches ein besseres Bild von der Topologie des Netzes machen und daraus mit geeigneten Algorithmen beste Pfade zu einem Ziel errechnen. Da jeder Router die Information über von ihm aus erreichbare Adressen so weit wie möglich zu Adressenpräfixen aggregiert und nur diese weiter verbreitet, wird um so mehr von Details der Topologie abstrahiert, je weiter das Ziel vom betrachteten Router entfernt ist. Dennoch bauen Router unter Umständen sehr große Tabellen auf, die für jedes IP-Paket durchsucht werden müssen, um den am besten auf die Zieladresse passenden Eintrag zu finden (*Longest Prefix Match*). Diese entsprechend aufwendige Operation wird in IP-Netzen in jedem Router für jedes Paket immer neu und ohne Verbindungsbezug ausgeführt (*Hop-by-Hop Datagram Routing*), eine insbesondere für verbindungsbezogenes Verkehrsmanagement kritische Verfahrensweise [39].

Aus diesen Gründen sind in den letzten Jahren auch mehrere durchgängig auf IP basierende und so auch in bezug auf Routing integrierte Architekturen vorgeschlagen worden, welche die Flexibilität von Routing ohne Verbindungsbezug mit der Effizienz der Paketvermittlung über virtuelle Verbindungen und mit den damit einhergehenden Möglichkeiten des Verkehrsmanagements zu verknüpfen suchen. Genau diese Ziele beabsichtigt eine Arbeitsgruppe der IETF mit *Multiprotocol Label Switching* (MPLS) [171] zu erreichen.

MPLS-Router sind in der Lage, Pakete sogenannten *Forwarding Equivalence Classes* (FEC) zuzuordnen, die jeweils einen oder mehrere Hops lang denselben Weg nehmen und auch sonst dieselbe Behandlung erfahren, und diese Zuordnung als Label in einem neuen Paketkopf kenntlich zu machen. Angefordert werden diese Labels unter Verwendung von *Label Distribution* Protokollen

entlang des gemeinsamen Weges des FEC. Vom letzten Router ausgehend werden dem FEC daraufhin abschnittsweise Labels zugewiesen und dem jeweiligen Vorgängerknoten bekannt gemacht. Der abhängig vom Ort im Netz unterschiedlichen Abstraktion der Topologieinformation, aber auch den Erfordernissen zur Aggregation von Verkehrsströmen entsprechend, können im Netzinnern FECs zu neuen, größeren FECs in einer nächst höheren Ebene aggregiert werden. An die Stelle eines einzelnen Labels im Paketkopf tritt dann ein Label-Stack. Abhängig davon, ob ein MPLS-Router für das FEC einer bestimmten Ebene Durchgangs-, Aggregations- oder Deaggregationsknoten ist, ersetzt er das Label oben im Stack (*Label Swap*), fügt außerdem eines oder mehrere neue hinzu oder entfernt ein Label oben vom Stack, ehe er das Paket zum Nachfolgeknoten übermittelt.

Es gibt durchaus Anzeichen, daß eine vollständige und für jedes Paket neue Auswertung des IP-Paketkopfes ähnlich effizient sein kann wie *Label Switching* [7]. Unabhängig davon eignet sich MPLS jedoch wegen seiner Mechanismen zum Aufbau virtueller Pfade, den sogenannten *Label Switched Paths (LSP),* verbunden mit der Option, diese Pfade mit Dienstattributen zu verknüpfen, für den Einsatz in zukünftigen Aggregationsarchitekturen.

4.2 Verkehrssteuerungsarchitektur für einen auf der Basis von RSVP über ATM aggregierenden Router

Ein Ende-zu-Ende-Dienst kann nur dann mit garantierter Dienstgüte angeboten werden, wenn sich alle an der Erbringung des Dienstes beteiligten Instanzen über die Dienstattribute verständigen und diese dann umsetzen können. Bezogen auf Netzknoten bedeutet dies, daß sie in der Lage sein müssen, Verkehrsströme zu unterscheiden und diesen auch noch in Überlastsituationen die im Verkehrsvertrag zugesicherte Dienstgüte bereitzustellen. Selbst wenn die Anzahl und die Bandbreite von Verbindungen, die sehr harte und präzise Anforderungen an die Dienstgüte haben, nicht so schnell wachsen sollte wie die Anzahl und die Bandbreite im Kernnetz insgesamt, wird eine auf Einzelverbindungen bezogene Differenzierung mit hoher Wahrscheinlichkeit in Kernnetzen nicht beherrschbar sein. Das Zusammenfassen von Einzelverbindungen zu einem Verbindungsaggregat ist deshalb eine sehr naheliegende Lösung. Um sicherzustellen, daß unter dieser Aggregation die Dienstgüte der Einzelverbindungen nicht leidet, müssen Funktionen zur Berechnung der vom Aggregat benötigten Ressourcen und geeignete Bedienstrategien entwickelt werden.

4.2.1 Kopplung der Steuerungsebene bei RSVP über ATM

Die Implementierung eines auf RSVP über ATM basierenden Aggregationsknotens bietet sich vor allem wegen der Verfügbarkeit von RSVP und ATM an. So kann auch heute schon eine interoperable, skalierbare Lösung für die Dienstgüteproblematik evaluiert und demonstriert werden. Von einigen wenigen Szenarien abgesehen [138], bereitet die Kopplung der Steuerungsebene trotz der in Tabelle 4.1 zusammengefaßten Unterschiede zwischen RSVP und ATM UNI 4.0 keine größeren Schwierigkeiten.

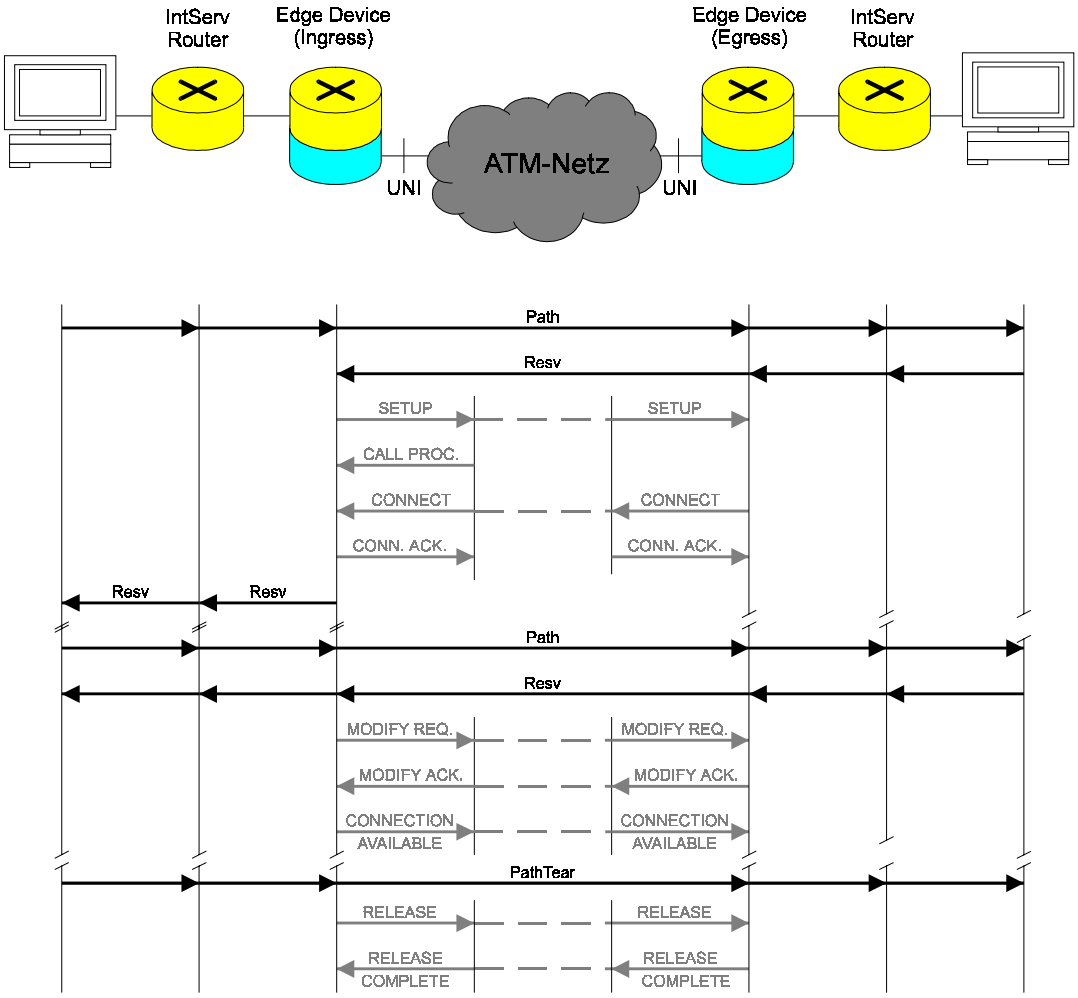

Abb. 4.3: *Kopplung der RSVP- und ATM-Signalisierung. In diesem Beispiel wird die ATM-Verbindung beim Eintreffen der ersten Reservierung synchron aufgebaut. Danach kann die Nachverhandlung meist asynchron erfolgen, wenn das dynamische Bandbreitemanagement entsprechend konfiguriert ist. Die ATM-Verbindung wird asynchron abgebaut.*

Der Aufbau (und ebenso die Modifikation) einer Reservierung im *Integrated Services* Netz mit RSVP [32] läuft zunächst über ATM hinweg wie über gewöhnliche Schicht-2-Übertragungsabschnitte. Erst in Rückrichtung, siehe Abb. 4.3, wird der Meldungsfluß im Aggregationsknoten

(*Ingress Edge Device*) unterbrochen, um zu prüfen, ob die Reservierung einer bestehenden ATM-(Aggregations-)Verbindung zugeordnet werden kann. Wenn dies nicht der Fall ist, muß zunächst der erfolgreiche Verbindungsaufbau auf der ATM-Ebene abgewartet werden, bevor der Aufbau der Reservierung fortgesetzt werden kann. Sollte der Verbindungsaufbau nicht gelingen, sendet der Aggregationsknoten wie jeder andere der Router im Falle des Scheiterns einer lokalen Reservierung eine RSVP-Meldung des Typs *ResvErr* in Richtung des Empfängers. Besteht dagegen bereits eine passende ATM-Aggregationsverbindung, ist zunächst lediglich zu prüfen, ob die für diese Verbindung reservierte Rate noch ausreicht, um auch die neue Reservierung noch aufnehmen zu können. Wenn die Rate nicht mehr ausreicht, muß die Prozedur zur Modifikation der ATM-Aggregatsverbindung eingeleitet werden [154]. Erst dann läuft die Signalisierung auf der RSVP-Ebene weiter. Um die damit verbundene Verzögerung beim Aufbau der Reservierung zu vermeiden, sollte das dynamische Bandbreitemanagement Nachverhandlungen bereits einleiten, wenn zwar im Prinzip die Rate der Aggregatsverbindung noch ausreicht, aber die verbleibende ungenutzte Rate (*Buffer Zone*) allmählich knapp wird und sich so die nächste Nachverhandlung abzeichnet. Wie in Abb. 4.3 angedeutet, können Reservierungen auf der RSVP-Ebene auf diese Weise von wenigen Ausnahmen abgesehen von den notwendigen Modifikation der ATM-Aggregatsverbindung abgekoppelt werden.

4.2.2 Verkehrsteuerungsarchitektur des Aggregationsknotens

Die eingangs erwähnten Verkehrssteuerungsfunktionen müssen auf Anforderung der Steuerungsebene aktiviert werden und einen Pfad durch den Knoten in ein Aggregat so einrichten, daß der Pfad selbst, aber insbesondere das Aggregat den Dienstgüteanforderungen der Verbindung gerecht wird.

Dazu ist für die im Rahmen des Projektes DIANA [38] entworfene RSVP über ATM Architektur ein sogenanntes *Flow-to-VC Mapping Control Module* (F2VM) entwickelt worden [139]. Wie schon sein Name suggeriert, stellt dieses Modul alle Funktionen und Datenstrukturen bereit, die mit der Zuordnung von RSVP-Verbindungen auf ATM-Verbindungen zu tun haben. Es ist die zentrale Einheit des in DIANA und entsprechend in Abb. 4.4 als *ATM Traffic Control Demon* (ATMTCD) bezeichneten Programmes, das darüber hinaus lediglich den Nachrichtenaustausch mit dem RSVP-Dämon mit Mitteln der Interprozeßkommunikation [183] abwickelt. Die für einen Aggregationsknoten notwendige Funktionalität ist in das F2VM ausgelagert. Dazu gehört die Zuordnung von einzelnen Reservierungsanforderungen zu Aggregaten, die Berechnung des Ressourcenbedarfs dieser Aggregate, ein dynamisches Bandbreitemanagement und die dynamische Konfiguration der Elemente der Verkehrssteuerungsarchitektur der Nutzerebene, die von Šironja [181] in die generi-

sche Verkehrssteuerungsarchitektur des Betriebssystemkerns von Linux [5] integriert worden sind. Außer dieser sehr generischen Funktionalität des Aggregationsknotens steuert das F2VM natürlich auch die speziell von einem *Edge Device* erwarteten Funktionen zum *Interworking* zwischen IP und ATM, also das Auflösen von IP- in ATM-Adressen sowie die Umsetzung der RSVP-Nachrichten auf Signalisierungsmeldungen des UNI 4.0.

Die internen Strukturen des F2VM, Zustandsübergangsdiagramme und sein API werden in [139] detaillierter beschrieben. Die der Leistungsuntersuchung gewidmeten Abschnitte weiter unten stellen die implementierte Verkehrssteuerungsfunktionalität zusammen, natürlich mit Verweisen auf Kapitel 3. Die wesentlichen Steuerungsfunktionen, die das F2VM in der Verkehrssteuerungsarchitektur nach Abb. 4.4 erbringt, können wie folgt zusammengefaßt werden:

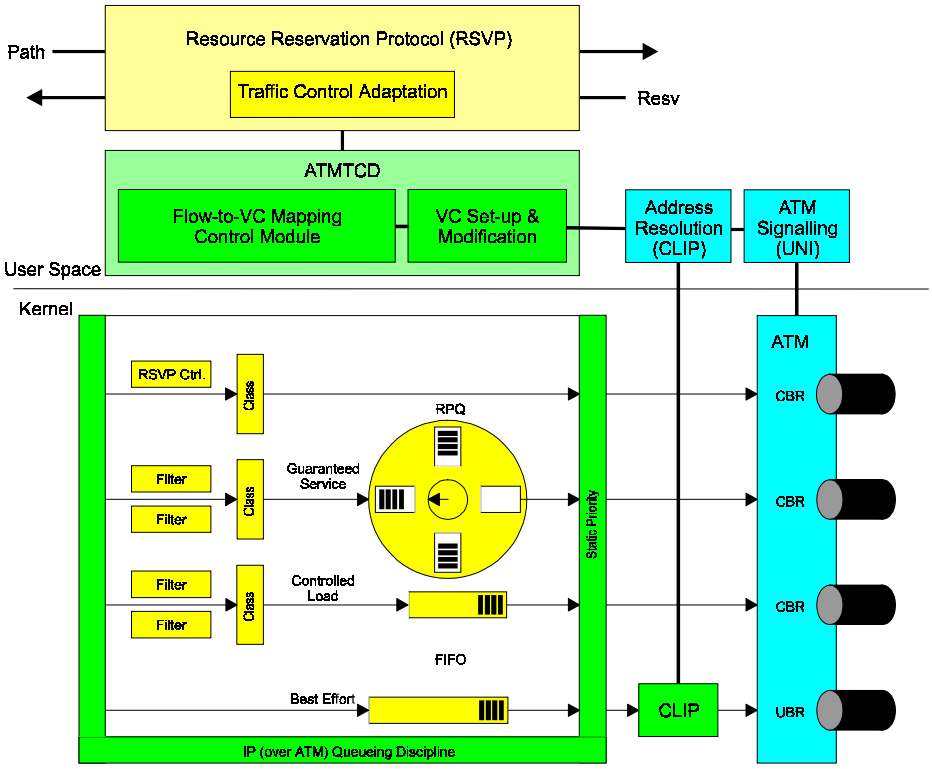

Abb. 4.4: *Verkehrssteuerungsarchitektur des Aggregationsknoten für RSVP über ATM*

Aggregationsfilter und Tabelle der Aggregate

Das F2VM verwaltet die für die Steuerung der Verbindungsaggregate notwendigen Informationen in einer Tabelle. In dieser Tabelle wird für jedes Aggregat eine Liste mit den zur Steuerung der Einzelverbindungen notwendigen Daten und Zustandsvariablen geführt. Um die Berechnung des Ressourcenbedarfs zu beschleunigen, werden die Einzelverbindungen aggregatsintern dynamisch definierten Typen zugeordnet. Die Berechnung des Ressourcenbedarfs des Aggregates, die

Annahme oder Ablehnung einer Reservierungsanforderung und das Hinzufügen oder Löschen von Filtern des *Classifiers* gehen mit einer Verarbeitung der in der Liste verwalteten Daten einher. Andere Operationen, wie etwa die Auflösung der Adresse des Deaggregationsknotens, die Steuerung der Aggregatsverbindung, das dynamische Bandbreitemanagement und die dynamische Konfiguration der Elemente des Bediensystems, also die typischen Aufgaben eines Aggregationsknotens erfordern zusätzliche, das Aggregat als Ganzes beschreibende Datenstrukturen. Zu diesen neuen Datenstrukturen gehört auch eine Liste von generischen Aggregationsfiltern, aus der für jedes neue Aggregat einer ausgewählt und initialisiert wird. Dieser legt die Bedingungen, die Aggregationskriterien fest, unter denen eine neue Reservierungsanforderung dem Aggregat zugeordnet werden kann.

Adressenauflösung und Umsetzung der Signalisierungsnachrichten

Wenn das F2VM eine eintreffende Reservierungsanforderung nicht einem der bestehenden Aggregate zuordnen kann, erzeugt es ein neues Aggregat mit einem zur Verbindung passenden neuen Aggregationsfilter, ermittelt die Adresse des Deaggregationsknotens, bestimmt die für den Aufbau der Aggregationsverbindung notwendigen Parameter und startet die Signalierungsprozedur. Im vorliegenden Falle werden mit der Adressenauflösung und der Signalisierung die entsprechenden Server für CLIP und UNI 4.0 der Implementierung von ATM für Linux [4] beauftragt.

Asynchroner Auf- und Abbau sowie Modifikation von Verbindungen

Um den Aufwand für die Signalisierung für Einzelverbindungen mit möglicherweise kleinem Bandbreitebedarf und kurzer Haltedauer zu reduzieren, steuert wie in Kapitel 3 diskutiert ein dynamisches Bandbreitemanagement die Nachverhandlung der für die Aggregatsverbindung reservierten Bandbreite. Ein heuristisches Verfahren stellt sicher, daß durch Überreservierung die Nachverhandlungsrate so weit reduziert wird, wie dies aufgrund der Kosten für die Nachverhandlungen und ungenützte Bandbreite gerechtfertigt erscheint. Wenn wie in [192] vorgeschlagen oberhalb der *Working Zone*, des eigentlich erlaubten Arbeitsbereichs, eine Buffer Zone angeordnet wird, eine Nachverhandlung aber schon dann ausgelöst wird, wenn der Arbeitsbereich verlassen wird, kann die Reservierung des Aggregates in vielen Fällen asynchron nachverhandelt werden. Das von Burgstahler [1] für ATM entwickelte *VC Setup and Modification Module* (VSMM) erweitert das API von ATM für Linux so, daß das F2VM durch Nachverhandlungen der ATM-Verbindungen nicht blockiert werden kann. Es werden grundsätzlich ATM-Verbindungen der Dienstklasse CBR aufgebaut.

Verkehrssteuerungsarchitektur der Nutzerebene

Das F2VM konfiguriert die Elemente der Verkehrssteuerungsarchitektur der Nutzerebene so, daß die Pakete durch den Knoten in die passende Aggregatsverbindung geleitet, wenn nötig am Eingang des Aggregates zwischengespeichert und schließlich in einer Reihenfolge bedient werden, daß für die Nutzer der Eindruck einer isolierten Ende-zu-Ende-Netzverbindung entsteht, welche die gewünschten Dienstgüteattribute unter allen Umständen beibehält.

Auf der Grundlage der generischen Verkehrssteuerungsarchitektur des Betriebssystems Linux [5] hat Široniја [181] die dazu notwendigen Mechanismen implementiert und eine geeignete Schnittstelle zum F2VM entwickelt: Nachdem der Router den richtigen Ausgangsport bestimmt hat, stellt der diesem Port zugeordnete *Classifier* den Verbindungsbezug her. Dazu fügt das F2VM auf Anfrage des RSVP-Dämons für jede neue Reservierungsanforderung einen der RSVP-Spezifikation entsprechenden Filter (siehe auch Abb. 4.4) in dessen *Hash*-Tabelle ein. Paßt einer dieser Filter, liegt aufgrund der Datenfelder des Filters zur internen Steuerung fest, wie weiter mit dem Paket zu verfahren ist. Vor Aggregaten aus Verkehrsströmen der Klasse *Guaranteed Service* plaziert das F2VM als Bediensystem das in Kapitel 3 ausführlich erörterte RPQ. Auf diese Weise können auch Verbindungen mit unterschiedlichen Verzögerungsanforderungen zu einem Aggregat zusammengefaßt werden. Pakete von Verbindungen der Klasse *Controlled Load*, für die keine maximale Ende-zu-Ende-Verzögerung angegeben werden kann, durchlaufen dagegen eine einfache FIFO-Warteschlange.

Außer Verbindungen für Aggregate dieser beiden Dienstklassen werden bei Bedarf auch virtuelle ATM-Verbindungen für die RSVP-Signalisierungsmeldungen und für *Best-Effort* Verkehr aufgebaut. Zum *Best-Effort* Verkehr gehören all die Pakete, die nicht auf einen der Filter im *Classifier* passen. Für diese werden mit der bereits in ATM für Linux verfügbaren Implementierung von CLIP Verbindungen ohne Beteiligung des F2VM eingerichtet. Es sollte an dieser Stelle vielleicht betont werden, daß sich CLIP – oder allgemeiner ausgedrückt – die Verwendung von virtuellen Verbindungen für in der Vermittlungsschicht eigentlich verbindungslosen *Best-Effort* Verkehr nicht mit dem vom Autor vertretenen Gesamtkonzept zur Verkehrssteuerung in *Integrated Services* Netzen verträgt. Verbindungsloser *Best-Effort* Verkehr kann und sollte nach Möglichkeit ohne Verbindungszustände in Netzelementen vermittelt werden. Insofern ist CLIP hier lediglich als Vervollständigung der praktischen Implementierung von IP über ATM anzusehen und keineswegs Teil des in der vorliegenden Arbeit behandelten Verkehrssteuerungskonzeptes.

Obwohl nur die in der *Integrated Services* Spezifikation [197] vorgesehenen Verkehrsklassen *Guaranteed Service* und *Controlled Load* vollständig implementiert sind, ist das Gesamtkonzept

bestehend aus F2VM und der Verkehrssteuerungsarchitektur der Nutzerebene so angelegt, daß eine praktisch unbegrenzte Zahl verschiedener Verkehrsklassen, Aggregate und Aggregationsstrategien unterstützt werden könnte. Russ [172] zeigt mit seiner Implementierung eines aggregationsfähigen H.323 *Gateways* außerdem, daß die Verkehrssteuerungsarchitektur keineswegs auf RSVP-fähige Router beschränkt ist.

Parallel zum DIANA-Projekt haben auch andere Projekte das Thema RSVP über ATM aufgegriffen. Hier sind vor allem die ACTS-Projekte ELISA [58] und *Peter Pan* [44, 168] zu nennen. Beide Projekte behandeln und implementieren nur Teilaspekte der Mechanismen zur verbindungsorientierten Aggregation in der Steuerungsebene.

4.2.3 Gegenstand und Ziele der Leistungsuntersuchung

Am Beispiel der Verkehrssteuerungsarchitektur eines RSVP über ATM Routers soll demonstriert werden, daß das Zusammenfassen von Einzelverbindungen zu Aggregaten, geeignete Bedienstrategien am Eingang des Aggregates sowie die Funktionen zur Berechnung der Bandbreite des Aggregats gemeinsam mit einem dynamischen Bandbreitemanagement den Aufwand für die Verbindungssteuerung drastisch reduzieren, ohne daß darunter die Dienstgüte einzelner Verbindungen leidet. Dazu wird zunächst demonstriert, wie mit Hilfe von Signalisierung der Aufbau der Aggregate und die Bedieneinheiten so gesteuert werden können, daß die Verbindungen am Eingang des Aggregates entsprechend ihrer Anforderungen eine differenzierte Behandlung erfahren. Diese ist nicht zuletzt auch davon abhängig, ob eine Verbindung *Guaranteed Service* oder *Controlled Load* anfordert. Die Ressourcen werden dementsprechend entweder mit der deterministischen oder mit statistischen Methoden berechnet. So liegt ein Vergleich der Leistungsfähigkeit beider Methoden nicht nur unter Berücksichtigung des errechneten Bedarfs an Bandbreite und Wartespeicher wie in Kapitel 3, sondern auch hinsichtlich der für die Berechnung notwendige Bearbeitungszeit nahe. Im Falle der statistischen Methode hängen diese Leistungsgrößen von der Größe des Aggregates ab. Schließlich werden all diese Mechanismen in Zusammenhang mit einem dynamischen Bandbreitemanagement gebracht, so daß der gesamte Knoten ganzheitlich betrachtet wird.

Deshalb wird in diesem Kapitel auch besonderer Wert auf Messung als Methode zur Leistungsuntersuchung gelegt. Da in den letzten Jahren die Diskrepanz zwischen Theorie und Praxis wie in sonst kaum einem Bereich der Kommunikationstechnik stetig zugenommen hat, ist Dienstgüte ein inzwischen zu sensibles Thema, als daß noch simulative und analytische Methoden allein als glaubwürdiger Nachweis der Machbarkeit und Leistungsfähigkeit einer Architektur akzeptiert würden. Prototypische Implementierungen und Messungen unter realistischen Randbedingungen erfassen die

Gesamteffizienz eines Systems in seinem Umfeld nun einmal besser und sagen so weit mehr aus als die Nähe zu einer theoretisch möglichen optimalen Lösung, die noch dazu nur mit meist beträchtlicher rechnerischer Komplexität erzielt werden kann. Auch die Beschaffung und Verteilung von Information ist in der Regel systembedingten oder protokolltechnischen Einschränkungen unterworfen. Insofern begünstigen prototypische Entwicklungen und begleitende Messungen die Entwicklung von Algorithmen und Protokollen, die bei beherrschbarer Komplexität die theoretisch optimale Lösung gut annähern.

Darüber hinaus durchläuft eine Meßanordnung beispielsweise beim Ein- und Ausschalten häufig unbeabsichtigt Arbeitspunkte, die unter Umständen bei Simulationen relativ statischer Szenarien nicht in Betracht gezogen werden. Wie am Beispiel der Berechnung des Ressourcenbedarfs eines Aggregats für *Controlled Load Service* mit der Methode der effektiven Bandbreite von Elwalid et al. im folgenden gezeigt werden wird, treten dabei mitunter Effekte auf, welche die Effizienz signifikant beeinflussen und dennoch nicht in der Literatur dokumentiert sind.

Im übrigen sind im vorliegenden Fall alle Elemente der Verkehrssteuerungsarchitektur, die in dieser Arbeit untersucht werden, nur im Prototypen integriert. Da die Implementierung auf stabil standardisierte Protokolle und Technologien wie RSVP und ATM setzt, dient der Prototyp neben wissenschaftlichen Zwecken auch der Demonstration skalierbarer Verkehrssteuerungsmechanismen in einem realen Umfeld.

Wenn notwendig, werden einzelne Komponenten der Architektur ergänzend mit simulativen oder seltener mit analytischen Methoden evaluiert, sei es weil die Meßkonfiguration nicht immer im notwendigen Maßstab Verkehr erzeugen kann, der ideale Meßort nicht zugänglich ist, die Messungen selbst oder andere Randbedingungen die Meßgrößen zu sehr beeinflussen oder die Meßgrößen nicht mit der notwendigen Genauigkeit erfaßt werden können, beispielsweise weil die Uhren der an der Messung beteiligten Rechner nicht vollkommen synchronisiert oder präzise genug sind.

Aus genau diesen Gründen sind Messungen mit insbesondere gegenüber Simulation beträchtlichem Mehraufwand verbunden. So ist es nicht weiter verwunderlich, daß sich Untersuchungen zu verbindungsorientierter Aggregation bisher auf Teilprobleme beschränkt haben.

Die ihm Rahmen des ACTS-Projektes Peter Pan ebenfalls an RSVP über ATM Prototypen durchgeführten Messungen [168] untersuchen fast ausschließlich Mechanismen der Nutzerebene, vor allem unterschiedliche Bedienstrategien, die nicht in direktem Zusammenhang mit Aggregation stehen. Einige qualitative Betrachtungen über den in *Integrated Services* Routern notwendigen Steuerungsaufwand leiten zu Simulationsstudien über. Dabei werden Ergebnisse für dynamisches Bandbrei-

temanagement am Beispiel eines einfachen Algorithmus mit konstanter und einer mit nicht näher spezifizierter variabler Überreservierung vorgestellt. Die Messungen der Leistungsfähigkeit der Mechanismen der Nutzerebene werden in [44] in verfeinerter Form fortgeführt.

Umgekehrt zeigen die vom Autor für das ACTS-Projekt DIANA durchgeführten Messungen [146] an einigen ausgewählten einfachen Beispielen im wesentlichen nur die Ersparnis bezüglich des Steuerungsaufwandes, die mit einem dynamischen Bandbreitemanagement grundsätzlich möglich ist. Erst die in [140] vorgestellten Ergebnisse demonstrieren die Kombination einer auf der effektiven Bandbreite beruhenden Berechnung des Ressourcenbedarfs eines Aggregates mit einem heuristischen Algorithmus zum dynamischen Bandbreitemanagement für ein Aggregat unterschiedlicher Typen von Verbindungen.

Eine frühere sich zum Teil auf Messungen stützende Studie [173] vergleicht die Zeiten, die für den Aufbau von Reservierungen benötigt werden, wenn eine neue ATM-Verbindung aufgebaut, nur der Ressourcenbedarf auf IP-Ebene neu berechnet oder zusätzlich die Rate der bestehenden Verbindung verändert werden muß.

4.2.4 Meßanordnung

Den Kern des in Abb. 4.5 abgebildeten lokalen Netzes bilden zwei über eine ATM-Strecke mit 155,52 Mbit/s Übertragungskapazität verbundene Router. Diese erfüllen die Rolle der *Edge Devices* der Referenzkonfiguration für RSVP über ATM gemäß Abb. 4.2. Dazu sind auf den als Router konfigurierten PCs das Betriebssystem Linux, Kernel-*Release* 2.2.9, inklusive der generischen Verkehrssteuerungsumgebung [5], die dazu kompatible *Release* 0.59 der Implementierung von ATM für Linux [4] sowie die von Kuznetsov für Linux ausgebaute *Release* 4.2a4 der Implementierung von RSVP des *USC Information Sciences Institute* installiert. Um den Meßaufgaben gerecht zu werden, aber vor allem zum Aufruf der ins F2VM ausgelagerten Funktionalität wird letztere jedoch in modifizierter Form eingesetzt. Hinzu kommen natürlich die Softwarekomponenten zur Realisierung der Verkehrsteuerungsarchitektur des oben beschriebenen Aggregationsknotens, also das F2VM [139], der im Linux-Kernel implementierte Teil des Aggregationsknotens [181] und schließlich noch das VSMM [38], das den Verbindungsauf- und -abbau sowie die Nachverhandlung von ATM-Verbindungen steuert.

Da zur Zeit der Messungen kein ATM-Vermittlungsknoten mit der notwendigen Funktionalität zur Modifikation des Verkehrsdeskriptors aktiver Verbindungen [14] zur Verfügung gestanden hat, diese aber zur Demonstration der Vorteile einer Aggregationsarchitektur unentbehrlich ist, ist kein ATM-Vermittlungsknoten in die Leistungsuntersuchungen einbezogen worden. Auch das in den

ATM-Übertragungsabschnitt eingeschleifte Meßgerät des Typs interWATCH 95000 von GN Nettest dient lediglich der Überwachung der Messungen, spielt also keine aktive Rolle. Es kann zur Verifikation der Datenquellen auch an das *Fast Ethernet* Segment angeschlossen werden.

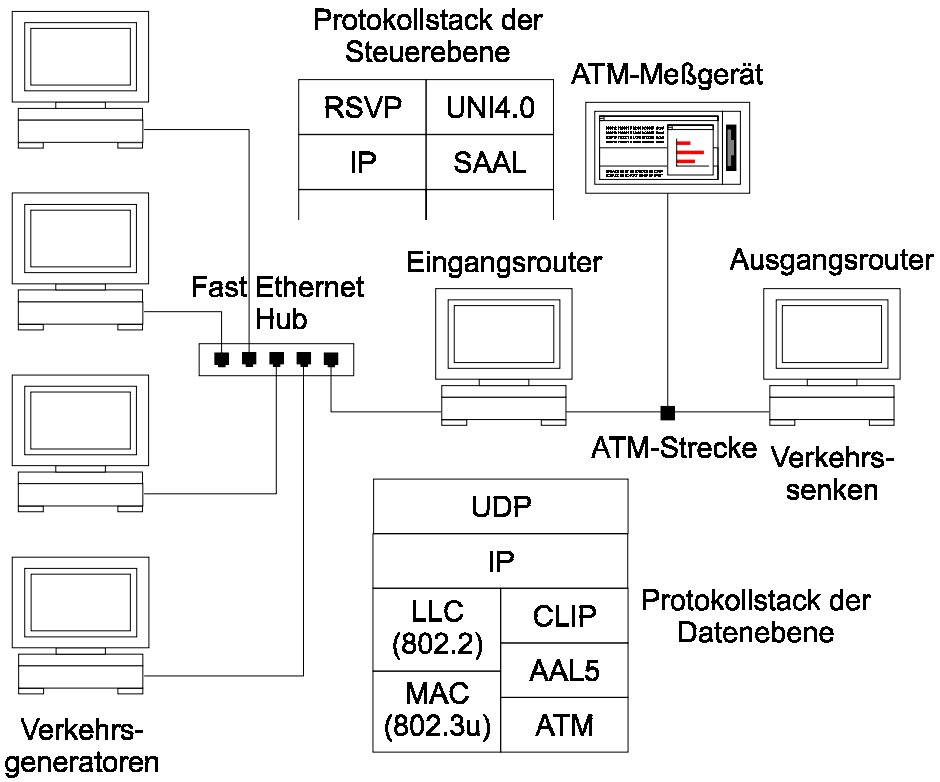

Abb. 4.5: *Konfiguration für die experimentellen Untersuchungen an der Verkehrssteuerungsarchitektur der Nutzerebene*

Die Subnetze auf beiden Seiten (Abb. 4.6) der Aggregationsknoten bestehen nur aus zur *Integrated Services* Spezifikation [197] konformen Endsystemen. Zusätzliche Router wären nur dann sinnvoll, wenn so viel Verkehr erzeugt werden könnte, daß mehrere Verzweigungspunkte oder gar ein vermaschtes Netz zumindest zeitweise ausreichend ausgelastet werden könnte. Die *Fast Ethernet* Segmente mit 100 Mbit/s Übertragungskapazität stellen sicher, daß in allen hier betrachteten Quellenszenarien nur das ratenbegrenzte Aggregat zum Engpaß werden kann. Was die *Integrated Services* Funktionalität angeht, sind die Endsysteme und Aggregationsknoten gleich konfiguriert. Natürlich fehlen in den Endsystemen alle Softwarekomponenten, die zur Aggregation bzw. zur Anbindung von ATM beitragen.

Der Transport der relativ kurzen IP-Pakete über CLIP und AAL5-Rahmen, aber noch viel mehr die Segmentierung der Rahmen in ATM-Zellen beansprucht signifikant mehr Übertragungskapazität als beispielsweise für die Übertragung von IP-Paketen über ein Schicht-2-Protokoll notwendig wäre, das Rahmen variabler Länge übertragen kann. Selbstverständlich muß dies bei der Berechnung der

Verbindungsparameter der ATM-Verbindung aus der Rate des Aggregates auf IP-Ebene berücksichtigt werden [79]. AAL5 [102] wird hier im Grunde auch nur deshalb verwendet, weil es bislang noch das einzige weit verbreitete ATM *Adaptation Layer* ist. Da für alle im Rahmen dieser Arbeit untersuchten Szenarien die Übertragungskapazität ausreichend ist, ist diese Ineffizienz ohnehin für die Leistungsuntersuchung der Verkehrssteuerungsmechanismen zur Aggregation in der IP-Ebene nicht von Bedeutung. Über UDP könnte sogar noch das eigens für verzögerungskritische Anwendungen entwickelte Transportprotokoll RTP [174] verwendet werden. Die zur Messung eingesetzten künstlichen Quellen profitieren jedoch nicht von seiner zusätzliche Funktionalität.

Gleichwohl weist die in Abb. 4.5 dargestellte Meßanordnung, in der ja Paketverzögerungszeiten gemessen werden sollen, einige Schwächen auf und muß deshalb simulativ unterstützt werden.

Insbesondere ist die Genauigkeit, mit der in Linux Ereignisse wie Signale [184] erzeugt werden können, an die Frequenz gekoppelt, mit welcher der *Scheduler* des Betriebssystems getaktet wird [18]. Davon ist in den Verkehrsgeneratoren [181] das von Signalen gesteuerte Senden von Paketen betroffen. Die Quellen können auf keinem Fall Verkehrsmuster mit einer höheren zeitlichen Auflösung erzeugen, d. h. Pakete mit einer höheren Rate senden, als durch die Taktperiode vorgegeben.

Zur Messung der Verzögerungszeiten müssen die Uhren der an der Messung beteiligten Rechner mit Hilfe der in das Betriebsystem integrierten Implementierung des *Network Time Protocol* synchronisiert [147] werden. *Scheduling* kann aber auch dann noch die genaue Feststellung der Empfangszeiten und damit die Messung der Verzögerungszeiten in den empfangenden Anwendung stören. Um die negativen Auswirkungen der Betriebssystemumgebung etwas zu mildern, wird deswegen der *Scheduler* des Betriebssystems mit 400 Hz statt wie sonst üblich mit 100 Hz getaktet. Darüber hinaus ist für jede Kombination von Quellen zu prüfen, ob die Fehler bei der Erzeugung der Verkehrsmuster innerhalb tragbarer Toleranzen liegen.

Immerhin beeinträchtigt das Multitasking des Betriebssystems die Verkehrserzeugung nicht in einer meßbaren Größenordnung, weil Signale und die anschließenden mit Systemaufrufen verbundenen Operationen zum Senden von Paketen in den Generatoren in den sonst unbelasteten Rechnern dominieren. Trotzdem ist zu beachten, daß grundsätzlich immer nur ein Prozeß gleichzeitig auf die CPU oder andere Ressourcen des Rechners zugreifen kann. In den Rechnern, aber natürlich ebenso beim Zugriff auf das *Fast Ethernet*, wird der Verkehr demzufolge serialisiert. Aus diesem Grunde muß für alle Messungen verifiziert werden, daß die daraus resultierenden Verzögerungen deutlich kleiner als die minimale Zwischenankunftszeit der Quellen sind.

Inzwischen werden die für IP erhältlichen spezialisierten Meßgeräte mit Funktionalität angeboten, die auch wissenschaftlichen Ansprüchen an die Erzeugung von Verkehr gerecht wird und so zukünftige Untersuchungen von größeren IP-Netzen erleichtern wird.

Schon heute unproblematisch gestaltet sich die Messung der Aggregationsmechanismen der Steuerungsebene, denn hier können sich die Generatoren auf die Erzeugung von Signalisierungsmeldungen beschränken. Die Last in Quellen und Senken, zwischen denen Verbindungen durch den Austausch von RSVP-Meldungen aufgebaut werden, unterscheidet sich kaum noch und so werden sowohl die Prozesse gleichmäßig auf die Rechner als auch die Rechner selbst entsprechend ihrer Rolle als Sender oder Empfänger auf die beiden *Ethernet* Subnetze verteilt. Abb. 4.6 zeigt diese Anordnung.

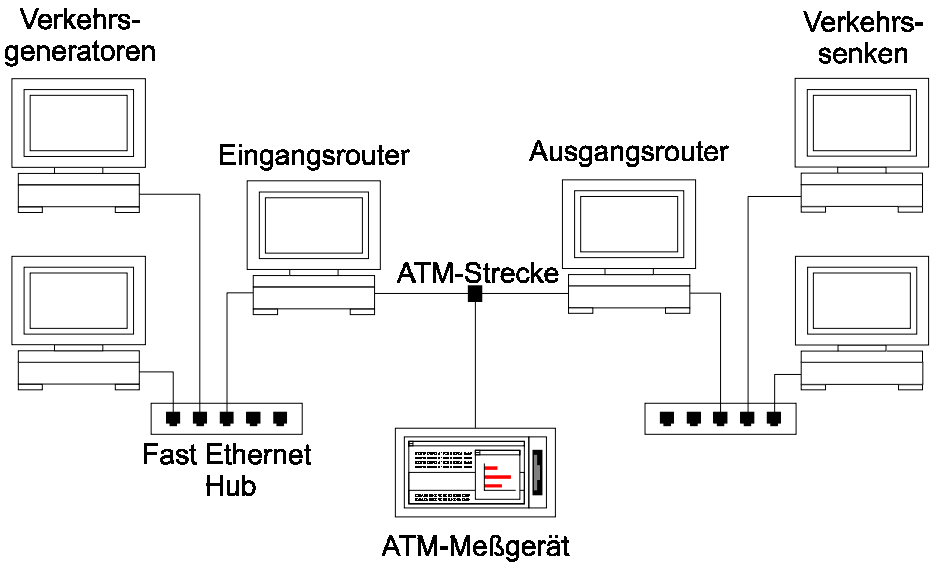

Abb. 4.6: *Konfiguration für die experimentellen Untersuchungen der Verfahren zur Berechnung des Ressourcenbedarfs und des dynamischen Bandbreitemanagements in der Steuerungsebene*

4.3 Dienstklassenabhängige Verkehrssteuerungsfunktionen

Ein Großteil der Verkehrssteuerungsfunktionen eines Aggregationsknotens kann völlig unabhängig von den angebotenen Dienstklassen betrachtet werden. Dazu gehören die Klassifizierung von Paketen, ihre Zuordnung zu Verbindungen, das Zusammenfassen dieser Verbindungen zu Aggregaten, das Einrichten von (virtuellen) Aggregationsverbindungen entweder als ATM-VCs, MPLS-LSPs oder IP-Tunnel, ja sogar das dynamische Bandbreitemanagement und die diese Konfigurationsschritte unterstützende Signalisierung. Die Auswahl der Bedienstrategie am Eingang eines

Aggregates und damit verbunden das Verfahren zur Berechnung des Ressourcenbedarfs prägt dagegen den Beitrag des Knotens zur Erbringung des Netzdienstes ganz entscheidend und muß daher konform zu dessen Spezifikation ausgelegt sein. Wichtige Parameter dieser Spezifikation sind beispielsweise der Verkehrsdeskriptor, zumindest wenn wie in dieser Arbeit auf seiner Basis der Ressourcenbedarf ermittelt werden soll, deterministisch oder statistisch zu interpretierende Dienstgüteparameter, wie z. B. maximale Verzögerungszeit oder Verlustquote, sowie das Modell, auf dessen Grundlage der Netzdienst Ende-zu-Ende erbracht werden soll. Die Verkehrsklassen der *Integrated Services, Guaranteed Service* und *Controlled Load*, sind sicherlich ein gutes Beispiel zur Veranschaulichung all dieser Einflußfaktoren. In diesem Abschnitt steht die meßtechnische Evaluierung der Verkehrssteuerungsfunktionen im Mittelpunkt. Dazu werden zunächst technische Aspekte der Integration der in Kapitel 3 besprochenen theoretischen Konzepte diskutiert. Bisweilen stehen der exakten Umsetzung praktische Erfordernisse entgegen. Die Ergebnisse zeigen aber, daß auch in der Praxis mit vertretbarem Steuerungsaufwand der Aufbau der Aggregate und die Bedieneinheiten so gesteuert werden können, daß die Verbindungen am Eingang des Aggregates entsprechend ihrer Anforderungen behandelt werden können.

4.3.1 *Guaranteed Service*

Der zur *Guaranteed Service* Spezifikation [176] konforme Teil der Verkehrssteuerungsarchitektur des Aggregationsknotens realisiert eine Implementierung von RPQ und des dafür in Kapitel 3 ausführlich diskutierten Verfahrens zur Berechnung der Rate von Aggregaten. Mit Hilfe von RPQ lassen sich alle Verbindungen zwischen zwei Aggregationsknoten, die *Guaranteed Service* nachfragen, zusammenfassen, selbst wenn sie unterschiedliche Anforderungen in bezug auf die maximale Verzögerung haben. Dazu muß lediglich das Aggregat konstanter Rate im Aggregationsbereich in geeigneter Weise, etwa durch Prioritätsbehandlung, von Verbindungen variabler Rate, die beim Multiplexen störende Verzögerungen verursachen können, isoliert sein. Diese bedingungslose Aggregation verspricht die größtmögliche Ersparnis beim Aufwand für die Verbindungssteuerung im Aggregationsbereich.

Darüber hinaus zeichnet sich RPQ durch seine geringe Komplexität aus. Vorausgesetzt man verzichtet darauf, das Rotationsintervall Δ und die Anzahl N_c der Warteschlangen eines bestehenden Aggregates zu modifizieren, muß nur die Länge der mit null indizierten Warteschlange an die neu errechnete Rate angepaßt werden, wenn sich die Zusammensetzung des Aggregates ändert.

Die Berechnung des Ressourcenbedarfs ist gegenüber dem in Kapitel 3 beschriebenen Verfahren etwas vereinfacht worden. Die Einzelbeiträge der Verbindungen in den Ungleichungen zur Verifi-

zierung der *Schedulability Condition* (3.18) werden so berechnet, als ob immer eine Verbindung im Aggregat ist, welche die minimal mögliche Verzögerungszeit angefordert hat. Auf diese Weise müssen nicht jedes Mal alle Ungleichungen zur Verifizierung der *Schedulability Condition* (3.18) neu berechnet werden, wenn sich durch den Aufbau oder das Ende einer Verbindung die kleinste tatsächlich vorhandene Verzögerungspriorität im Aggregat ändert. Statt dessen genügt es, die Liste der kritischen Zeiten zu aktualisieren und in allen schon bestehenden Ungleichungen den Beitrag der Verbindung zu addieren bzw. zu subtrahieren. Die Berechnung wird so beschleunigt, die Bandbreiteeffizienz natürlich etwas reduziert.

Da auf der Ebene des Aggregates der Verkehr in einen zur reservierten Rate C_A konformen Strom umgeformt wird, muß am Ausgang der RPQ-Bedieneinheit kein feingranulares Shaping stattfinden. Andererseits darf der Verkehr auch nicht unkontrolliert an die Ebene des Aggregates weitergereicht werden, weil sich sonst dort Warteschlangen aufbauen können, die nicht entsprechend ihrer Verzögerungspriorität sortiert sind. Die Herleitung der *Schedulability Condition* (3.18) zeigt, daß die maximalen Verzögerungszeiten aber auf jeden Fall dann eingehalten werden können, wenn zwischen zwei Rotationszeitpunkten nie mehr als $C_A \Delta$ Dateneinheiten an die Ebene des Aggregates weitergereicht werden. Dies entspricht dem maximal erlaubten Füllstand der mit null indizierten Warteschlange im System. Aus diesem Grunde blockiert die vorliegende Implementierung, wenn diese Grenze überschritten wird. Erst die nächste Rotation löst die Blockierung. Das hat zur Folge, daß die Pakete nur dann entsprechend ihres Sendetermins sortiert werden, wenn sich aufgrund dieses Stop-and-Go-Verhaltens auch auf der IP-Ebene eine Warteschlange aufgebaut hat.

In der Implementierung wäre von möglichen Verlusten nur die mit null indizierte Warteschlange betroffen. Um solche Verluste gänzlich auszuschließen, wird – wie in Kapitel 3 vorgeschlagen – Paketlängen $L_i > 0$ implizit Rechnung getragen durch eine intern modifizierte Spitzenrate p_i. Das reduziert die Zahl der zu untersuchenden kritischen Zeiten und steigert die Effizienz der Verbindungsannahmesteuerung weiter.

4.3.2 *Controlled Load Service*

Zur *Controlled Load* Service Spezifikation [198] konforme Netzknoten müssen in der Lage sein, Verbindungen Ressourcen so zuteilen zu können, daß ihre Dienstgüte unter allen Umständen der in einem unbelasteten Netz entspricht. Ganz im Gegensatz zur *Guaranteed Service* Spezifikation [176] schließt diese sehr unbestimmte Forderung die Zuteilung von Ressourcen unter Ausnutzung von statistischem Multiplexen nicht aus, auch wenn so gelegentlich Paketverluste auftreten können. Da Anwendungen keine obere Schranke für die Verzögerungszeit festlegen können und ohnehin das in

der Literatur [182] vorgestellte Verfahren zur Berechnung des Ressourcenbedarfs von Verbindungen mit unterschiedlichen Verzögerungsprioritäten in einem EDF-Bediensystem nicht zu einer praktikablen Lösung führt, werden an den Eingängen zu *Controlled Load* Aggregaten FIFO-Bediensysteme plaziert. Der Ressourcenbedarf kann dann beispielsweise mit dem zwar sehr konservativen, dafür aber effizient lösbaren Verfahren von Elwalid et al. [59] berechnet werden.

Dazu werden zunächst die Verbindungen gemäß ihrer durch das virtuelle deterministische Bediensystem modifizierten Spitzenrate e_i und Aktivität w_i mit einer gewissen Toleranz in Verbindungstypen eingeteilt. Anschließend ist anhand von Gleichung (3.86) zu prüfen, ob die Verbindungen ausreichen, um die Bandbreite des Aggregats C_A unter Ausnutzung statistischen Multiplexens berechnen zu können. Wenn die Prüfung positiv ausfällt und aus der letzten Berechnung bereits ein Arbeitspunkt $s*$ bekannt ist, dient dieser als Startpunkt für die Lösung von (3.85) mit dem Newton-Verfahren. Ohne eine solchen guten Startpunkt konvergiert das Newton-Verfahren meist nicht. Deshalb wird beim Übergang vom deterministischen zum statistischen Multiplexen heuristisch ein Startpunkt bestimmt.

Ausgehend von einem homogenen Aggregat wird ein auf einen einzelnen Verbindungstyp bezogener, hypothetischer Wert von s_j berechnet, aus dem die effektive Bandbreite von $e_i - 0{,}1(e_i - r_i)$, also etwas weniger als die deterministische effektive Bandbreite, resultieren würde. Da statistisches Multiplexen – wie im Zusammenhang mit Gleichung (3.86) erläutert – erst ab einer kritischen Anzahl von Verbindungen ausgenutzt werden kann, wird mit dieser kritischen Anzahl und dem hypothetischen Wert von s_j die linke Seite der für die Berechnung des Ressourcenbedarfs entscheidenden linken Gleichungsseite von (3.85) errechnet, mit der festgesetzten oberen Schranke verglichen und s_j entsprechend des Verhältnisses des Ergebnisses und der oberen Schranke linear nachskaliert. Wenn auf diese Weise für alle Verbindungstypen die entsprechenden Werte von s_j bestimmt worden sind, wird als Startwert für das Newton-Verfahren die Summe der s_j gebildet, wobei die Summanden auf der Basis der tatsächlichen Zusammensetzung des Aggregats jeweils mit ihrem Anteil $N_j \log w_j^{-1}$ an der deterministisch bestimmten Gesamtrate $\sum N_j \log w_j^{-1}$ gewichtet werden.

Der in Abb. 3.9 dargestellte charakteristische Verlauf der linken Seite von Gleichung (3.85) läßt noch weitere Maßnahmen zur Stabilisierung der Berechnung zu. Wird im Verlaufe der Iteration eine Zwischenlösung für $s*$ negativ, so ist aufgrund des Krümmungsverhaltens zwischen den beiden Asymptoten der Kurve davon auszugehen, daß die Tangente im Kurvenpunkt der Ausgangslösung

zu flach ist, also zu weit rechts liegt. Zur Korrektur kann dann beispielsweise die Ausgangslösung halbiert werden, um (wieder) näher an die Wendestelle der Kurve heranzukommen.

4.3.3 Leistungsuntersuchung

Mit Hilfe von RPQ lassen sich alle Verbindungen zwischen zwei Aggregationsknoten, die *Guaranteed Service* nachfragen, zusammenfassen, selbst wenn sie unterschiedliche Anforderungen in bezug auf die maximale Verzögerung haben.

Während aber die Berechnung des Ressourcenbedarfs unter Ausnutzung statistischen Multiplexens einen um so größeren Multiplexgewinn ermöglicht, je größer das Aggregat ist, bleibt bei der Berechnung des Ressourcenbedarfs mit der deterministischen Methode der Multiplexgewinn konstant. Die Pakete von Verbindungen unterschiedlicher Verzögerungspriorität erfahren durch die RPQ-Bedieneinheit am Eingang eines *Guaranteed Services* Aggregates eine differenzierte Behandlung, allerdings auf einem von der Größe des Aggregates abhängigen Niveau.

Tabelle 4.2: *Verkehrsparameter der periodischen Ein-Aus-Quellen für die in Abb. 4.7-4.10 dargestellte Leistungsuntersuchung einer RPQ-Bedieneinheit. Die Quellen der Typen 1-3 sind büschelförmiger als die der Typen 4-6. Letztere unterscheiden sich lediglich hinsichtlich ihrer vereinbarten maximalen Verzögerungszeit* D_{max}.

	Szenario 1			Szenario 2		
Typ	1	2	3	4	5	6
Anzahl	24	10	6	15	15	15
$r / \frac{\text{byte}}{s}$	4000	4000	4000	4000	4000	4000
$p / \frac{byte}{s}$	40000	40000	40000	8000	8000	8000
$b / byte$	360	1080	1800	400	400	400
$L / byte$	100	100	100	100	100	100
D_{max} / ms	20	60	100	20	60	100

Dieser Effekt läßt sich sowohl simulativ als auch experimentell mit den unterschiedlich büschelförmigen Quellen aus Tab. 4.2 belegen. Dazu werden in den Abb. 4.7-4.10 zunächst relativ kleine Aggregate untersucht, danach in den Abb. 4.11 und Abb. 4.12 Aggregate variabler Größe.

Vor dem Hintergrund der angesprochenen Schwierigkeiten bei der Messung von Verzögerungszeiten überrascht es doch etwas, daß sich Messung und Simulation in den Abb. 4.7-4.10 so voneinander unterscheiden, wie man dies erwartet: Die Pakete der niedrigsten Verzögerungspriorität

profitieren zu Lasten der höheren Prioritätsstufen, wenn wie in der vorliegenden Implementierung die Pakete nur dann entsprechend ihres Sendetermins sortiert werden, wenn sich aufgrund des Stop-and-Go-Verhaltens auf der IP-Ebene eine Warteschlange aufgebaut hat.

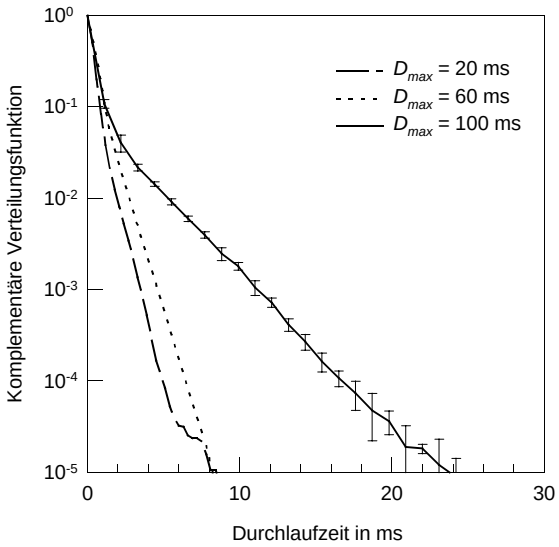

***Abb. 4.7**: Experimentell für Szenario 1 aus Tab. 4.2 ermittelte komplementäre Verteilungsfunktion des variablen Anteils der Durchlaufzeit durch die RPQ-Bedieneinheit*

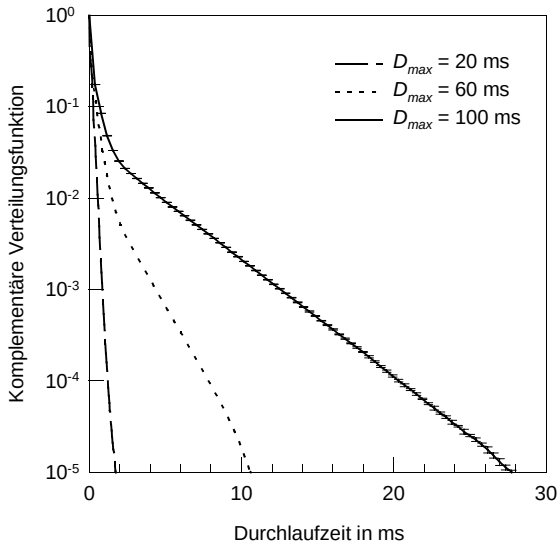

***Abb. 4.8**: Simulativ (mit dem Phasensprungverfahren [42]) für Szenario 1 aus Tab. 4.2 ermittelte komplementäre Verteilungsfunktion der Durchlaufzeit durch die RPQ-Bedieneinheit*

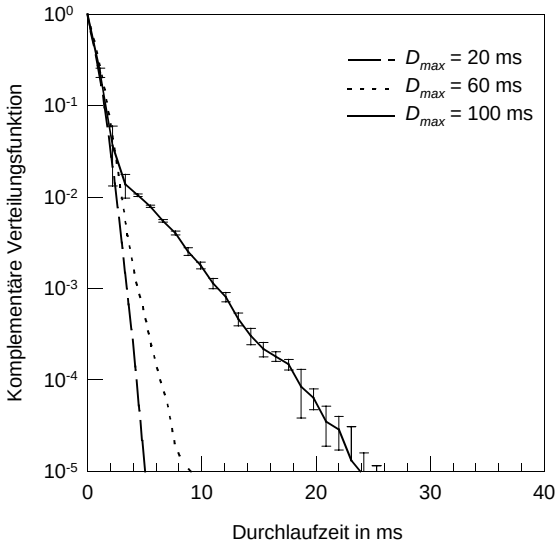

***Abb. 4.9** Experimentell für Szenario 2 aus Tab. 4.2 ermittelte komplementäre Verteilungsfunktion des variablen Anteils der Durchlaufzeit durch die RPQ-Bedieneinheit*

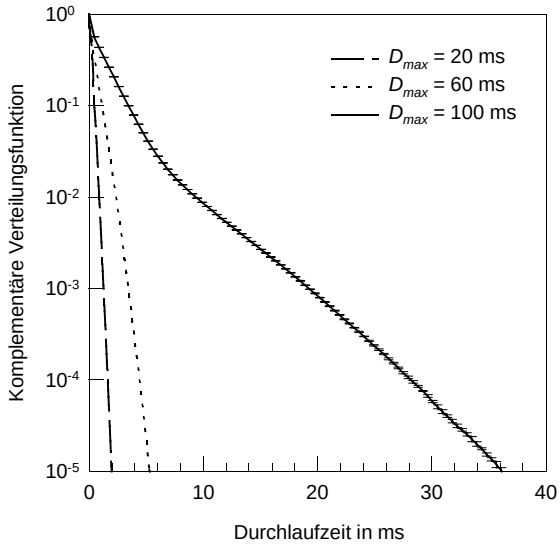

***Abb. 4.10**: Simulativ (mit dem Phasensprungverfahren [42]) für Szenario 2 aus Tab. 4.2 ermittelte komplementäre Verteilungsfunktion der Durchlaufzeit durch die RPQ-Bedieneinheit*

Die aus dem deterministischen *Network Calculus* resultierenden *Schedulability Conditions* (3.18) garantieren völlige Verlustfreiheit und Verzögerungsobergrenzen. Solche Garantien müssen aller-

dings durch eine im Vergleich zu statistischem Multiplexen schlechte Auslastung der Ressourcen erkauft werden. Insbesondere größere Aggregate verfügen über Ressourcen im Überfluß. Eine in Form eines statistischen Maßes, eines Quantils, zugesicherte Verzögerungsobergrenze würde sich schon bei relativ kleinen Aggregaten von der für *Guaranteed Services* im schlimmsten Fall möglichen Verzögerung so sehr entfernen, daß der Verkehrsvertrag zu einem realitätsfernen abstrakten Konstrukt mutiert. Die Kurven der unterschiedlichen Prioritätsstufen in Abb. 4.11 gruppieren sich um das in Abb. 4.12 zum Vergleich mit dem *Many Sources Asymptotic* berechnete 10^{-5}-Quantil einer FIFO-Bedieneinheit. Leider verschieben sich die Kurven auch relativ zueinander.

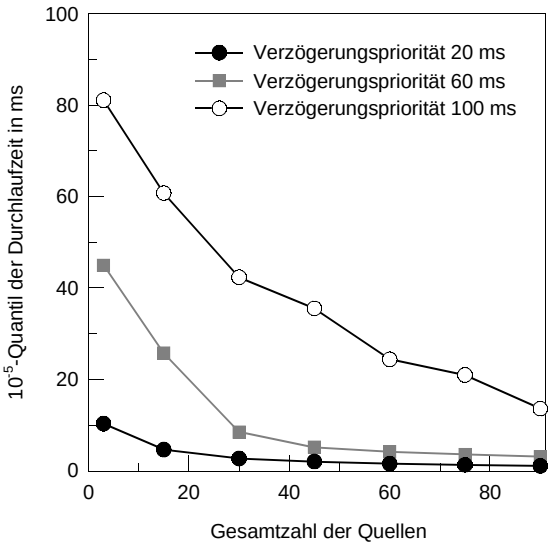

***Abb. 4.11**: Simulativ für die Quelltypen aus Szenario 2 in Tab. 4.2 ermitteltes 10^{-5}-Quantil der Durchlaufzeit in Abhängigkeit von der Gesamtzahl der Quellen bei RPQ*

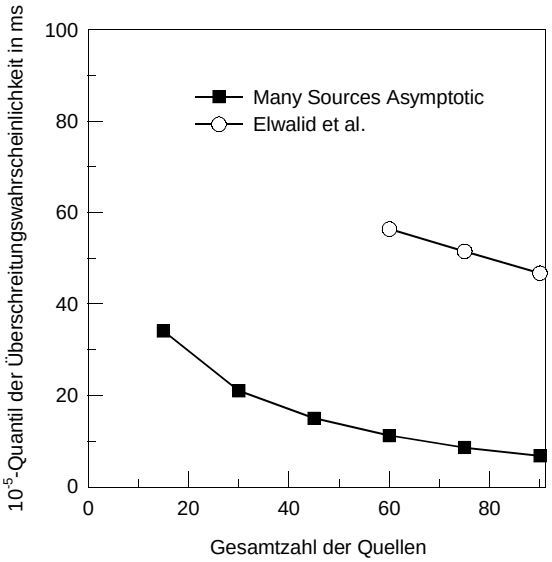

***Abb. 4.12:** In einem FIFO-Bediensystem bei statistischem Multiplexen mögliches 10^{-5}-Quantil der Durchlaufzeit, wenn die gleichen Raten wie in Abb. 4.11 reserviert würden.*

Neben der effizienten Nutzung der für das Aggregat reservierten Ressourcen ist schon mehrfach die Bedeutung der Komplexität des Verfahrens zur Berechnung des Ressourcenbedarfs betont worden. Der Vergleich der in dieser Arbeit berücksichtigten Verfahren zur Berechnung des Ressourcenbedarfs unter Ausnutzung statistischen Multiplexens in Kapitel 3 macht deutlich, daß eine gute Ausnutzung der Ressourcen der Nutzerebene mit einem hohem Rechenaufwand in der Steuerungsebene verbunden sein kann.

Die Abb. 4.13 und 4.14 zeigen jedoch, daß deterministisches Multiplexen auf dieser Ebene kaum für seine Ineffizienz in der Nutzerebene entschädigen kann. Ein Netzdienst *Guaranteed Service* kann also nicht technisch, sondern wirklich nur aufgrund seiner Paradigmen, der vollständigen Verlustfreiheit und der Garantie einer deterministischen Grenze für die Verzögerung, motiviert werden. Er wird ein sehr exklusiver Dienst bleiben. Die in Abb. 4.13 und 4.14 aufgetragenen

Berechnungszeiten erfassen den Aufwand, der zur Ergänzung der Liste der Verbindungstypen und den sich daran anschließenden Prozeduren zur Berechnung der neuen Kapazität des Aggregates benötigt wird, wenn eine neue Verbindung dem Aggregat hinzugefügt wird. Nur am Rande sei bemerkt, daß das Entfernen einer Verbindung mit etwas weniger Aufwand verbunden ist. In der Messung unberücksichtigt bleiben dagegen beispielsweise alle Operationen zur Steuerung der Liste der Verbindungen und – was wahrscheinlich noch mehr ins Gewicht fällt – die Verzögerung aufgrund des Austauschs von Meldungen zwischen den an der Steuerung beteiligten Prozesse ATMTCD, RSVP-Dämon und dem Betriebssystemkern. Dieser zusätzliche Aufwand ist ohnehin unabhängig von den Verfahren zur Berechnung des Ressourcenbedarfs und auf zum Teil strukturbedingte Besonderheiten der Implementierung, wie z. B. die Trennung in Teilprozesse, zurückzuführen.

Damit sich das Bediensystem mit den an sich ja auch auf Verbindungsebene statistisch unabhängigen Verbindungen aus Tabelle 4.4 und 4.5 nicht dauerhaft auf einen Mittelwert einschwingt, werden die Parameter der Verbindungsebene zum Teil statisch manipuliert und zum Teil dynamisch von einem zentralen Server in regelmäßigen Intervallen mit einer abgeschnittenen Normalverteilung modifiziert. Die daraus resultierende Korrelation auf Verbindungsebene erlaubt es, einen relativ großen Bereich aktiver Quellen einigermaßen gleichmäßig abzudecken.

Bei genauerer Betrachtung von Abb. 4.13 fällt außer der statistischen Unsicherheit einiger Meßwerte der sprunghafte Anstieg bei 14 Verbindungen und der anschließende sanfte Abfall der Berechnungszeit auf. An dieser Stelle überschreiten die Verbindungen vom Typ 1 aus Tabelle 4.5, die zum *Controlled Load* Aggregat zusammengefaßt werden, die kritische Anzahl, bei der das statistische Multiplexen einsetzt, vgl. Gleichung (3.86). Mit jeder weiteren Verbindung verbessert sich die Qualität des alten Arbeitspunktes als Anfangslösung für die Berechnung des neuen.

Abb. 4.14 zeigt die entsprechenden Ergebnisse für aus Verbindungen unterschiedlichen Typs zusammengesetzte Aggregate. Das Aggregat der Dienstklasse *Guaranteed Service* besteht aus Verbindungen der acht in Tab. 4.4 aufgeführten Typen, die jedoch auf jeweils drei Verzögerungsprioritäten verteilt werden, so daß die Berechnung letzten Endes sogar 24 Verbindungstypen einzubeziehen hat. Im *Controlled Load* Aggregat werden die Verkehrsparameter der acht Verbindungstypen aus Tab. 4.5 um $\pm 10\,\%$ variiert, um eine mit dem *Guaranteed Service* Aggregat vergleichbare Zusammensetzung aus 24 Verbindungstypen zu erreichen. Wenn nur wenige Verbindungen aktiv sind, sind natürlich nicht jederzeit alle 24 Typen in den Aggregaten vertreten. Da die Zusammensetzung der Aggregate generell statistischen Schwankungen unterworfen ist, kann die kritische Anzahl von Verbindungen, ab der das *Controlled Load* Aggregat von statistischem Multiplexen profitiert, nicht mehr eindeutig beziffert werden. Aus diesem Grunde ist die Meßkurve in

drei Stränge unterteilt, einen für die deterministische Berechnung, einen für die Berechnung unter Ausnutzung statistischen Multiplexens und einen für den Übergang zwischen diesen beiden Modi. In Abb. 4.14 ist dann recht deutlich zu sehen, daß der Übergang um so problematischer ist, je mehr Verbindungen – oder genauer – je mehr Verbindungstypen dazu benötigt werden. Im sprunghaft ansteigenden Teil überschreiten Variablen während der Berechnung ihre durch den Datentyp bedingten Maximalwerte. Eine Begrenzung der zulässigen Berechnungszeit und der vorübergehende Verzicht auf statistisches Multiplexen kann hier aber leicht Abhilfe schaffen.

Demgegenüber ist der Einfluß der Anzahl der Verbindungstypen auf die Berechnung des Ressourcenbedarfs des *Guaranteed Service* Aggregats mit der RPQ-Bedieneinheit am Eingang so gering, daß er sich am Beispiel von Abb. 4.13 und 4.14 nicht nachweisen läßt.

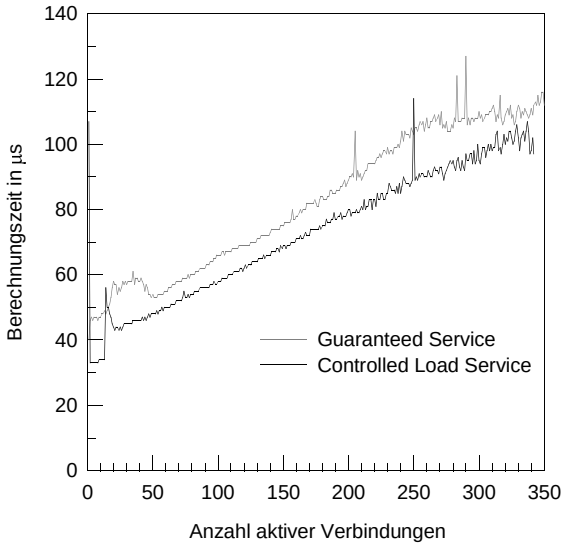

***Abb. 4.13**: Experimentell ermittelte Zeit, die abhängig von der Anzahl aktiver Verbindungen benötigt wird, um den Ressourcenbedarf beim Hinzufügen von Verbindungen zu berechnen. Die beiden Aggregate bestehen jeweils aus Verbindungen vom Typ 1 aus den Tab. 4.4 u. 4.5.*

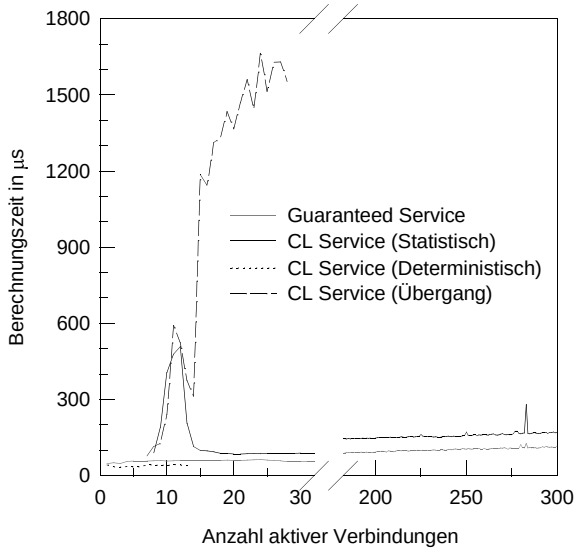

***Abb. 4.14**: Im Ggs. zu Abb. 4.13 enthalten die Aggregate nun alle Verbindungstypen aus den Tab. 4.4 u. 4.5. Durch Variation der max. Verzögerungszeiten bzw. Parameter ist die Zahl der Typen auf 24 erhöht. Die Kurve für Controlled Load ist in Teilstränge aufgespalten.*

Die Messungen belegen durchaus, daß die Komplexität der Implementierung der Algorithmen zur Berechnung des Ressourcenbedarfs nicht den vertretbaren und beherrschbaren Rahmen sprengt. Zwar scheint der Aufwand für die Berechnung des Ressourcenbedarfs sowohl für *Guaranteed Service* als auch *Controlled Load* linear mit der Anzahl der Verbindungen im Aggregat anzusteigen, mit mehreren hundert Verbindungen pro Aggregat ist aber sicherlich schon eine Aggregatsgröße erreicht, die nicht mehr als realitätsfern bezeichnet werden kann. Mit der nach heutigen Maßstäben bereits veralteten CPU (Intel Pentium II, Taktfrequenz 350 MHz) des Aggregationsknotens ist auch das Skalierungspotential wohl bei weitem noch nicht ausgeschöpft.

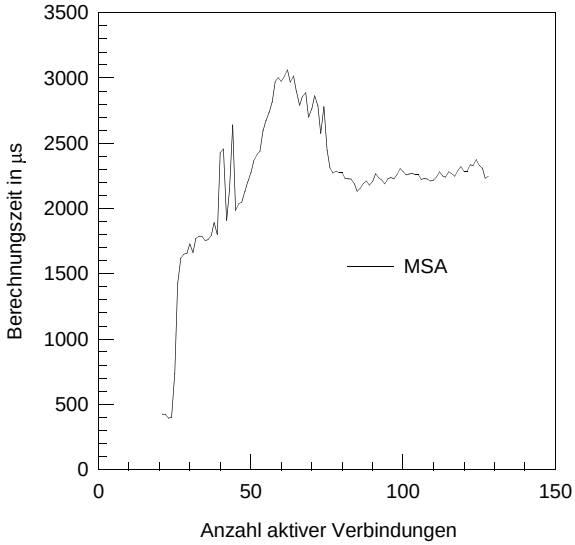

Abb. 4.15: *Experimentell ermittelte Zeit, die für die Berechnung des Ressourcenbedarfs beim Hinzufügen von Verbindungen mit dem Many Sources Asymptotic (MSA) benötigt wird. Das Aggregat besteht aus analog zu Abb. 4.14 variierten Verbindungen der Typen 1-6 aus Tab. 4.5.*

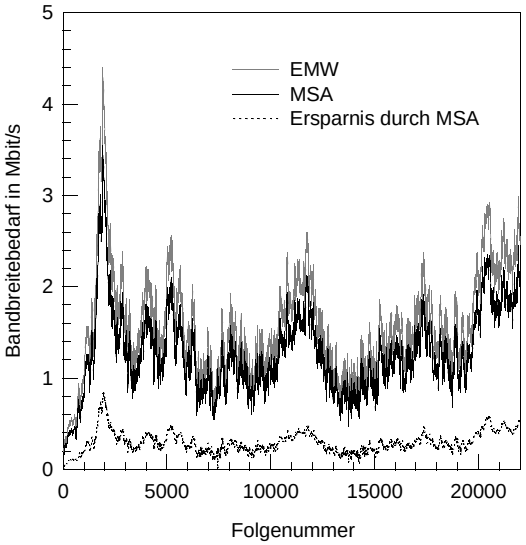

Abb. 4.16: *Vergleich des mit dem Verfahren von Elwalid et al. (EMW) und mit dem Many Sources Asymptotic (MSA) für das Aggregat aus Abb. 4.15 ermittelten Ressourcenbedarfs. Wie in Abb. 4.15 kommt der in Kapitel 3 besprochene praktische Algorithmus zum Einsatz.*

Die in Abb. 4.15 dargestellten Berechnungszeiten geben einen groben Anhaltspunkt für den beim Einsatz des *Many Sources Asymptotic* anfallenden Mehraufwand. Je nachdem welches Maß an Genauigkeit und Zuverlässigkeit gewünscht wird, steigt oder fällt der Berechnungsaufwand. Hierzu darf auf die Diskussion in Abschnitt 3.2.4.4 verwiesen werden. Abb. 4.16 zeigt, daß mit dem dort vorgeschlagenen Algorithmus das *Many Sources Asymptotic* die Ressourcen der Datenebene besser ausgenutzt werden als beim Verfahren von Elwalid et al., ohne daß die Berechnungszeiten ausufern. Im Rahmen von Leinmüllers Arbeit [129] ist anhand einer Reihe von Stichproben verifiziert worden, daß auch bei Berücksichtigung von nur fünf Werten für t im Abstand von einem Vierzigstel des kleinsten gemeinsamen Vielfachen der Periodendauern der Quellen der Algorithmus auf der konservativen Seite bleibt. Ob die so ermittelten Berechnungszeiten tatsächlich den Anforderungen kommerzieller Router standhalten, sich also vertretbare Antwortzeiten ergeben, hängt natürlich zunächst einmal vom Aufkommen der Reservierungsanforderungen und -freigaben und der tatsächlich vorhandenen Rechenleistung ab. Außerdem wird bei Aggregaten mit heterogenerer und noch variablerer Zusammensetzung als in Abb. 4.15 und 4.16 die Vorabschätzung der Bandbreite unter Zuhilfenahme des Verfahrens von Elwalid et al., wie sie der Algorithmus vorsieht, zwangsläufig ungenauer. Je ungenauer aber diese Schätzung ist, desto höher sind aber die Anforderungen an den Wertebereich der für die Darstellung von Gleitkommazahlen verwendeten Datentyps. Das vorliegende System erfüllt diese Anforderungen nicht. Daher konvergieren die beim *Many Sources Asym-*

ptotic notwendigen Berechnungen häufig nicht und der Algorithmus in Leinmüllers Implementierung muß in die Initialisierungsphase zurückkehren. Das hat dann zwei Konsequenzen. Die Ersparnis fällt weitaus geringer aus, und im Bemühen um Konvergenz steigen die Berechnungszeiten deutlich. Die Verwendung einer mathematischen Programmbibliothek, die größere Gleitkommazahlen verarbeitet, kann in diesen Situationen also gleichzeitig die Effizienz des Verfahrens erhöhen und die maximalen Rechenzeiten senken. Sicherlich sind auch dann numerische Probleme nicht völlig auszuschließen, dann kann aber noch auf die mit dem konservativen Verfahren von Elwalid et al. ermittelte Lösung zurückgegriffen werden.

4.4 Dynamisches Bandbreitemanagement

In diensteintegrierenden paketvermittelnden Netzen ist nicht nur auf der Paketebene, sondern auch auf der Verbindungsebene mit sehr heterogenen und wechselnden Verkehrscharakteristiken zu rechnen. Die Aussagen zur Effizienz von Aggregation in Kapitel 3 lassen es wahrscheinlich erscheinen, daß auch innerhalb von Aggregaten diese Heterogenität zu berücksichtigen ist. Insofern ist es problematisch, daß die in Kapitel 3 besprochenen Verfahren für dynamisches Bandbreitemanagement, die heuristischen ebenso wie die analytischen, nur auf homogene Aggregate anwendbar sind. Vordringliches Ziel dieses Abschnittes ist es daher, auf der Grundlage des vielversprechenden Verfahrens von Orda [155] eine heuristische Lösung zu entwickeln, die bei Nachverhandlungen der Rate von Aggregaten die neue Rate mit geringem Berechnungsaufwand so dimensioniert, daß die Gesamtkosten für die Verbindungssteuerung, die sich aus den Kosten für die Überreservierung von Bandbreite und den Kosten für die Nachverhandlungen zusammensetzen, nahezu minimiert werden. Anders als die aus der Literatur bekannten Verfahren soll das neue Verfahren heterogenen Aggregaten mit variablem Verkehr gewachsen sein und mit den Verfahren zur Berechnung des Ressourcenbedarfs kooperieren.

4.4.1 Heuristisches Verfahren

Die Untersuchungen an homogenen Aggregaten am Ende von Kapitel 3 mit dem von Orda vorgestellten Modell [155] lassen folgende Schlußfolgerungen zu:

Bei konstantem Verkehrswert sind *Working Zones* konstanter Größe zumindest nicht ineffizienter als die *Working Zones* variabler Größe, die sich mit diesem Modell errechnen lassen. Wenn sich die Zusammensetzung des Aggregates nicht zu schnell ändert und auch das Verhältnis der Kosten für

Überreservierung auf der einen und Nachverhandlungen auf der anderen Seite über einen längeren Zeitraum konstant bleibt, muß folglich nicht bei jeder Nachverhandlung eine neue optimale *Working Zone* berechnet werden.

Eine zweite Beobachtung kann zu einer weiteren Vereinfachung, einer Verkürzung der Berechnung führen. Da der Geburts-Sterbe-Prozeß der Überzustände im Modell von Orda reversibel ist, gelten für einen Ausschnitt des Prozesses die gleichen lokalen Gleichgewichtsbedingungen wie für das vollständige Modell [113]. Wenn also nur ein Ausschnitt, ein zusammenhängender Teil der Überzustände berücksichtigt wird, gehen diese also wenigstens mit dem relativ zueinander korrekten Gewicht in die Berechnung ein. Da im vorgeschlagenen Markoff-Modell *Working Zones* mit wachsender Entfernung von der *Working Zone* um den Verkehrswert mit schnell abnehmender Wahrscheinlichkeit erreicht werden, sollte es genügen, statt des vollständigen Zustandsmodells nur einen kleinen Ausschnitt durchzurechnen.

Ein Aggregat mit unterschiedlichen Typen von Verbindungen muß im Prinzip durch einen mehrdimensionalen Markoff-Prozeß modelliert werden. Dann – das hat die Diskussion am Ende von Kapitel 3 deutlich aufgezeigt – sind die Zustandsprozesse innerhalb der *Working Zones* aber nicht mehr unabhängig voneinander und als Folge davon auch nicht mehr stationär lösbar. Eine auf dem exakten Zustandsmodell basierende Dimensionierung der *Working Zones* kommt daher nicht in Frage.

Algorithmen, die völlig ohne verkehrstheoretische Modelle arbeiten und so dieses Problem umgehen, können in statischen Verkehrsszenarien mit geringem Aufwand nahezu optimale Ergebnisse erzielen. In [140] wird ein solches Verfahren untersucht, in [168] ein vergleichbares angedeutet. Statt die statistischen Eigenschaften der Verbindungen zu beobachten und auf stochastische Modelle abzubilden, prüfen sie in regelmäßigen Abständen die Kosten für Nachverhandlung auf der einen und Überreservierung auf der anderen Seite und passen abhängig vom Ergebnis die Größe der *Working Zone* an. Der größte Nachteil solcher Verfahren liegt in der direkten Messung der Nachverhandlungsrate, die natürlich erst nach einer gewissen Stichprobengröße ausreichend zuverlässig bestimmt werden kann. Verändert sich also die Verkehrscharakteristik der Verbindungen im Aggregat so, daß bei konstanter Überreservierung die Nachverhandlungsrate kleiner wird, so paßt der Algorithmus die Größe der *Working Zone* (noch) langsamer an als in umgekehrter Richtung.

Verfahren, die sich auf ein stochastisches Modell stützen (selbst wenn dieses nur bedingt der Realität entspricht) und aus der Beobachtung der statistischen Eigenschaften der Verbindungen, des momentanen Systemzustandes usw. die momentan zu erwartende Nachverhandlungsrate prognostizieren, unter Umständen sogar, bevor eine Nachverhandlung stattgefunden hat, sind in der Lage,

sehr viel schneller die Größe der *Working Zone* zeitlichen Schwankungen des Verkehrsangebotes oder Änderungen der Kostenfunktion anzupassen. Aus genau diesem Grunde wird folgender zweistufiger heuristischer Algorithmus vorgeschlagen:

Die erste Stufe des Algorithmus mißt den mittleren Anstieg $\overline{\Delta C_A}$ der Rate C_A des Aggregats, der aus der Ankunft von Reservierungsanforderung resultiert. Ziel dieses Ansatzes ist es, eine Schrittweite, eine Granularität G_A zu gewinnen, um die die Überreservierung für das Aggregat bei Bedarf erhöht oder reduziert werden kann.

Extreme und seltene Abweichungen *Err* der effektiven Bandbreite ΔC_A einer Reservierungsanforderung von $\overline{\Delta C_A}$ werden dabei mit Hilfe des Faktors δ_E in (4.1) herausgefiltert, um zu vermeiden, daß sehr große, aber nur sehr selten eintreffende Reservierungsanforderungen die Überreservierung über einen längeren Zeitpunkt mitbestimmen. Folgende, der Berechnung des *Retransmission Timeout* bei TCP [185] ähnelnden Schritte werden immer dann ausgeführt, wenn eine Reservierungsanforderung eintrifft:

$$Err := \min\{\Delta C_A - \overline{\Delta C_A}, \delta_E \overline{\Delta C_A}\} \tag{4.1}$$

$$\overline{\Delta C_A} := \overline{\Delta C_A} + \delta_C Err \tag{4.2}$$

$$\overline{D_A} := \overline{D_A} + \delta_D \delta_C (|(Err)| - \overline{D_A}) \tag{4.3}$$

$$G_A := \overline{\Delta C_A} + \delta_G \overline{D_A} \tag{4.4}$$

Die mittlere Veränderung des Bandbreitebedarfs wird mit (4.2) als *Exponentially Weighted Moving Average* (EWMA) ermittelt. Dieses Verfahren zur Messung zeitlich variierender Mittelwerte reduziert das Gewicht der Meßwerte ausgehend vom anfänglichen δ_C jedes Mal, wenn ein neuer Meßwert erfaßt wird, um den Faktor $1-\delta_C$. Auf ähnliche Art und Weise wird in (4.3) die mittlere Abweichung $\overline{D_A}$ berechnet. Sie wird bei der Bestimmung der Schrittweite G_A berücksichtigt, damit auch und vor allem bei vergleichsweise kleiner Überreservierung eine größere Anzahl von Verbindungstypen (zumindest alle mit einer effektiven Bandbreite in ähnlicher Größenordnung) von der Erhöhung der Überreservierung profitiert. Um nicht zu früh *Retransmission Timeouts* auszulösen und unter Umständen unnötig Pakete zu wiederholen, setzt TCP sowohl für δ_D (4.3) als auch für δ_G (4.4) relativ große Werte ein und berücksichtigt Meßwerte, die sehr stark vom *Exponentially Weighted Moving Average* abweichen, überproportional. Bei dynamischem Bandbreitemanagement ist dies aber nicht angebracht, weil Überreservierung nur dann die Nachverhandlungsquote signifikant beeinflußt, wenn sie für die große Masse der Verbindungen ausgelegt ist. δ_C, bei TCP eine Konstante, wird hier abhängig von der Überreservierung berechnet:

$$\delta_C := 1 - \exp\left(\frac{\ln \epsilon_C}{K_{\ddot{U}}\, N_\delta}\right) \tag{4.5}$$

$K_{\ddot{U}}\, N_\delta$ in (4.5) ist die Anzahl der Meßwerte, die auf einen ersten Meßwert folgen müssen, bis sich dessen Gewicht auf den Bruchteil ϵ_C seines Anfangswertes verringert hat. $K_{\ddot{U}}$, die Überreservierung in Vielfachen der Schrittweite G_A, stellt in (4.5) die gewünschte Abhängigkeit des Faktors δ_C vom Ausmaß der Überreservierung her: Da $\epsilon_C \leq 1$, erhält man mit Gleichung (4.5) um so größere Werte für δ_C, je kleiner der Überreservierungsfaktor $K_{\ddot{U}}$ ist. Somit reagiert bei kleiner Überreservierung das dynamische Bandbreitemanagement schneller auf Abweichungen von der geschätzten mittleren effektiven Bandbreite $\overline{\Delta C_A}$ als bei großer.

Der zweiten Stufe des Algorithmus obliegt es nun, den Überreservierungsfaktor $K_{\ddot{U}}$ abhängig von der Ankunftsrate von Reservierungsanforderungen und den Haltedauern der Reservierungen so zu bestimmen, daß die Kosten für die Überreservierung und die Nachverhandlung von Bandbreite (annähernd) minimiert werden. Ein mit realisierbarem Aufwand lösbares Modell ist nicht bekannt. Um dennoch eine Lösung zu erhalten, die rasch an zeitliche Schwankungen des Verkehrsangebotes und wechselnde Kostenfunktionen angepaßt werden kann, wird das Aggregat so betrachtet, als ob es sich ausschließlich aus Verbindungen mit einer einheitlichen Rate, der Schrittweite oder Granularität G_A, zusammensetzt. In diesem Modell finden Zustandsübergänge statt, wenn sich der Bandbreitebedarf des Aggregats durch Reservierungsanforderungen oder die Freigabe von Reservierungen so ändert, daß die neue Bandbreite ein höheres oder niedrigeres ganzzahliges Vielfaches (Modulo-Division) von G_A ist. Solche Übergänge werden über einen etwas längeren Zeitraum T_B hinweg beobachtet und gezählt. Am Ende dieses Beobachtungszeitraums können dann die Raten λ und μ im Markoff-Modell nach Orda bestimmt werden (vgl. Abb. 3.16). Falls eine einzige Reservierung Sprünge über mehrere Schrittweiten hinweg nach sich zieht, wird sie entsprechend mehrfach gezählt.

Die Berechnung von $K_{\ddot{U}}$ erfolgt mit dem Modell von Orda ($K_{\ddot{U}}$ entspricht $\frac{1}{2}(u_j - l_j)$ in Abb. 3.16), allerdings anknüpfend an die Ergebnisse von Kapitel 3 unter der Annahme von *Working Zones* konstanter Größe über den gesamten Zustandsraum hinweg. D. h. die Ankunfts- und Bedienraten werden als die Raten des entsprechenden Markoff-Modells übernommen. Stufe 2 des Algorithmus beruht also auf einem Modell, das die tatsächlichen Verhältnisse nur sehr unzulänglich abbildet, dafür aber mit geringem Aufwand gelöst werden kann.

Um seine Berechnung weiter zu beschleunigen, werden außer dem aktuellen Wert für $K_{Ü}$ nur jeweils $N_{Ü}$ weitere Werte nach unten und oben durchgerechnet sowie für jeden dieser Werte nur ein kleiner Ausschnitt, N_{WZ} Zustände, des eindimensionalen Zustandsprozesses der *Working Zones*, so wie oben angedeutet. Gegenüber der doch recht drastischen Vereinfachung des eindimensionalen Modells sind diese Näherungen wohl vergleichsweise unbedenklich.

In Tab. 4.3 sind die Parameter des heuristischen Verfahrens zusammengestellt. Da die Parameter der Stufe 1 bei TCP nicht einmal exportiert werden, sondern als Konstanten implementiert sind, und unter den Parametern der Stufe 2 allenfalls bei geringem Verkehrsangebot der Beobachtungszeitraum T_B etwas kritisch sein könnte, kann das Verfahren als in weiten Bereichen adaptiv angesehen werden.

Tabelle 4.3: *Parameter des heuristischen Verfahrens für dynamisches Bandbreitemanagement*

	Parameter	Wert	Bemerkung
Stufe 1	δ_E	1	Zuweisung (4.1)
	δ_D	1	Zuweisung (4.3)
	δ_G	1	Zuweisung (4.4)
	ϵ_C	0,1	Zuweisung (4.5)
	N_δ	10	Zuweisung (4.5), $N_\delta = 5$ in Tab. 4.4 u. 4.5
Stufe 2	T_B	60 s	Beobachtungszeitraum
	$N_{Ü}$	2	Prüfe $K_{Ü} - N_{Ü}, K_{Ü} - N_{Ü} + 1, ..., K_{Ü} + N_{Ü}$
	N_{WZ}	1	Anzahl der berücksichtigten *Working Zones*

Ganz entscheidend für die Dimensionierung der *Working Zones* sind selbstverständlich die Kosten, die für die Überreservierung von Bandbreite und eine Nachverhandlung veranschlagt werden müssen. Die Kosten für (zusätzliche) Bandbreite können sicherlich rein betriebswirtschaftlich ermittelt werden. Die Kosten für die Signalisierung sind sicherlich ebenfalls kalkulierbar. Möglicherweise empfiehlt es sich aber, statt dessen das Verhältnis von maximal verfügbarer Bandbreite zu maximal möglicher Signalierungskapazität der Steuerungsebene auch als Verhältnis der Kosten für Nachverhandlung und Überreservierung zu übernehmen.

4.4.2 Leistungsuntersuchung des heuristischen Verfahrens

Zunächst soll demonstriert werden, daß die Berechnung eines nur kleinen Ausschnittes aus N_{WZ} *Working Zones* und auch die Betrachtung von nur $2N_{Ü}+1$ verschiedenen Werten für den Überreservierungsfaktor $K_{Ü}$ pro Aufruf der Stufe 2 des Verfahrens ausreichen, um eine mit dem vollständigen Modell von Orda vergleichbare Effizienz des dynamischen Bandbreitemanagements zu erreichen. Dazu wird ein Aggregat aus Verbindungen identischen Typs mit einheitlicher Bandbreite 30 kbyte/s und konstantem Verkehrswert $Y=1000$ betrachtet. Verbindungen werden in negativ exponentiell verteilten Abständen erzeugt, und zwar mit der mittleren Rate $\lambda=10\,\mathrm{s}^{-1}$, und terminieren nach einer exponentiell verteilten Belegungsdauer mit dem Erwartungswert $1/\mu=100\,\mathrm{s}$.

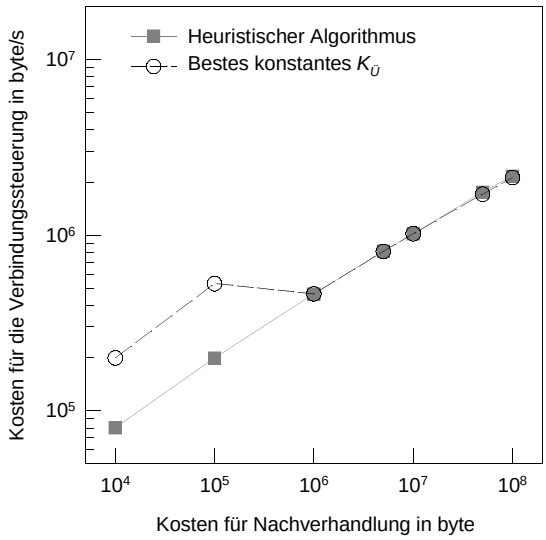

Abb. 4.17: *Kosten der Verbindungssteuerung in Abhängigkeit von den für eine Nachverhandlung veranschlagten, auf die Übertragungsbandbreite bezogenen Kosten*

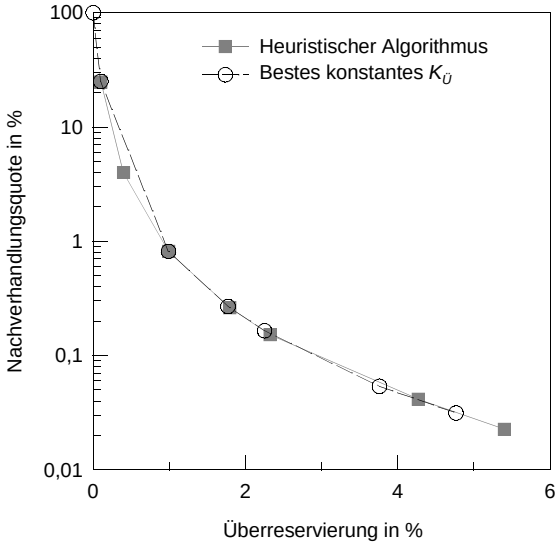

Abb. 4.18: *Nachverhandlungsquote für das homogene Aggregat in Abhängigkeit von der mittleren Überreservierung, wenn die Kosten gemäß Abb. 4.17 variiert werden*

Abb. 4.17 stellt den Zusammenhang zwischen dem Verhältnis der Kosten für eine Nachverhandlung und die Überreservierung von Bandbreite und den entstehenden Kosten für die Verbindungssteuerung dar. Dabei werden die Kosten pro Nachverhandlung in Bandbreite ausgedrückt. Von einem Verfahren für dynamisches Bandbreitemanagement wird erwartet, diese Kosten zu minimieren. Bezogen auf das Modell von Orda, mit dem der optimale konstante Überreservierungsfaktor für das homogene Aggregat berechnet werden kann, gelingt dies dem heuristischen Verfahren sehr gut, obwohl entsprechend der Angaben in Tab. 4.3 die Stufe 2 des Verfahrens bei jedem Aufruf jeweils nur fünf verschiedene Überreservierungsfaktoren $K_{Ü}$ ($N_{Ü}=2$) in Betracht zieht und für jeden dieser Faktoren nur eine *Working Zone* ($N_{WZ}=1$) durchrechnet. Abhängig von den Kosten für eine Nach-

verhandlung in Abb. 4.17 stellen sich die in Abb. 4.18 gegeneinander aufgetragene mittlere Nachverhandlungsquote und Überreservierung ein. Die Ergebnisse des heuristischen Verfahrens stimmen relativ genau mit der vollständigen Berechnung des Modells von Orda überein.

Wesentlich problematischer als die Vereinfachungen bei der Berechnung des Modells ist das Modell an sich, wenn die Aggregate nicht mehr aus identischen Verbindungen, sondern sehr heterogen zusammengesetzt sind. Die Quellen in Tab. 4.4 erzeugen unterschiedliche Verbindungen für ein *Guaranteed Service* Aggregat, die in Tab. 4.5 für ein *Controlled Load* Aggregat. In beiden Fällen dominieren zwar die relativ kleinen Verbindungen des Typs in bezug auf die Anzahl und Ruffrequenz, es sind aber eine Reihe weiterer Verbindungen mit zum Teil recht hoher Bandbreite beigemischt.

Tabelle 4.4: *Verkehrsparameter der Verbindungen, die freie Quellen in negativ exponentiell verteilten Abständen erzeugen (Erwartungswert α^{-1}) und die nach einer ebenfalls negativ exponentiell verteilten Belegungsdauer (Erwartungswert μ^{-1}) terminieren. Alle Verbindungen fordern Guaranteed Service an. Am Aggregationsknoten wird pro Typ für 25% der Verbindungen die Verzögerungspriorität 20 ms, für 50% 60 ms und für 25% 100 ms realisiert.*

Typ	1	2	3	4	5	6	7	8
Anzahl	400	128	64	48	40	100	40	20
$r / \frac{byte}{s}$	5000	15000	22500	30000	37500	50000	50000	50000
$p / \frac{byte}{s}$	12500	37500	60000	75000	100000	125000	250000	625000
$b / byte$	600	1800	2700	3600	4500	6000	16000	46000
$L / byte$	125	125	200	250	250	250	500	1250
$\alpha / \frac{1}{s}$	0,0010	0,0002	0,0002	0,0002	0,0002	0,0002	0,0002	0,0002
$\mu / \frac{1}{s}$	0,010	0,002	0,002	0,002	0,002	0,002	0,002	0,002

Die Abb. 4.19-4.22 stellen jeweils die simulativ und die experimentell gewonnenen Ergebnisse einander gegenüber. Bei der Interpretation dieser Ergebnisse ist zu berücksichtigen, daß nur im Prototypen, also den experimentell gewonnenen Ergebnissen, die Verbindungen mit ihrer effektiven Bandbreite in die Berechnung des Ressourcenbedarfs eingehen. In den Simulationen wird an ihrer Stelle die konstante deterministische effektive Bandbreite der Verbindungen eingesetzt, um die Wirkung der effektiven Bandbreite auf das dynamische Bandbreitemanagement einschätzen zu können. Deterministisches oder statistisches Multiplexen bleibt unberücksichtigt. Aufgrund des Transportes über ATM im Prototypen werden auch bei der Simulation nicht die Raten der IP-Ebene, sondern die auf der ATM-Ebene angesetzt. Bei einer konservativen Berechnung erhöht sich dadurch

die aus den Tab. 4.4 und Tab. 4.5 berechenbare deterministische effektive Bandbreite der Verbindungen um den Faktor 1,66.

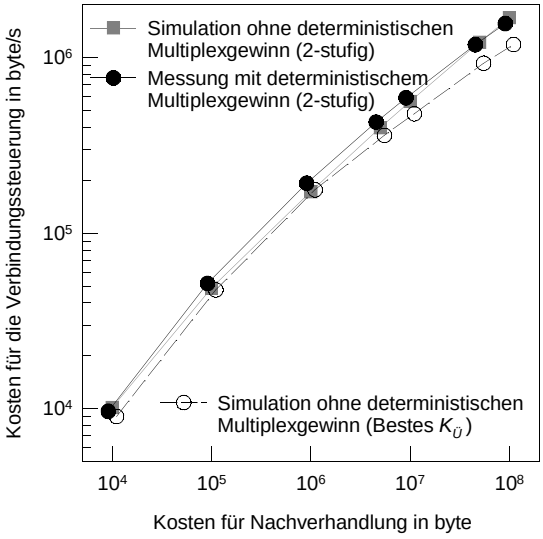

Abb. 4.19: *Kosten der Verbindungssteuerung in Abhängigkeit von den für eine Nachverhandlung veranschlagten, auf die Übertragungsbandbreite bezogenen Kosten*

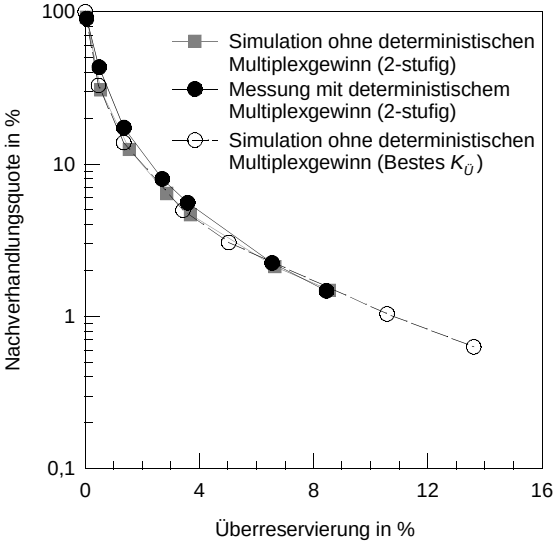

Abb. 4.20: *Nachverhandlungsquote für das Guaranteed-Service-Aggregat in Abhängigkeit von der mittleren Überreservierung, wenn die Kosten gemäß Abb. 4.19 variiert werden*

Der vergleichsweise kleine Multiplexgewinn beim deterministischen Multiplexen beeinflußt, wenn überhaupt, dann doch jedenfalls nicht in einem signifikanten Ausmaß das dynamische Bandbreitemanagement (Abb. 4.19). Etwas anders verhält sich die Kombination einer auf der Methode der effektiven Bandbreite basierenden Berechnung der Bandbreite des Aggregates mit dem heuristischen Algorithmus zum dynamischen Bandbreitemanagement (Abb. 4.21). Statistisches Multiplexen reduziert die effektive Bandbreite der Quellen, so daß sich mit dem Aufbau oder dem Ende einer Verbindung der Bandbreitebedarf weniger stark als bei deterministischem Multiplexen ändert. Die Wirkung auf die Gesamtkosten für die Verbindungssteuerung bleibt jedoch gering. Stärker wirkt sich der Multiplexgewinn auf den Zusammenhang zwischen Überreservierung und Nachverhandlungsquote aus. Beim statistischen Multiplexen in Abb. 4.22 weicht die gemessene Kurve stärker von der simulativ (und ohne Berücksichtigung der Reduktion der effektiven Bandbreite der Verbindungen) ermittelten Kurve ab als in Abb. 4.20.

Tabelle 4.5: *Verkehrsparameter der Verbindungen, die freie Quellen in negativ exponentiell verteilten Abständen erzeugen (Erwartungswert λ^{-1}) und nach einer negativ exponentiell verteilten Belegungsdauer (Erwartungswert μ^{-1}) terminieren. Alle Verbindungen fordern Controlled Load Service an. Der Wartespeicher S_A wird dynamisch der Rate des Aggregats angepaßt: $S_A := C_A \cdot 0.1\,\mathrm{s}$*

Typ	1	2	3	4	5	6	7	8
Anzahl	400	128	64	48	40	100	40	20
$r / \frac{\mathrm{byte}}{s}$	5000	15000	22500	30000	37500	50000	50000	50000
$p / \frac{byte}{s}$	12500	37500	56250	75000	93750	125000	250000	625000
$b / byte$	3000	9000	13500	18000	22500	30000	80000	230000
$\lambda / \frac{1}{s}$	0,0010	0,0002	0,0002	0,0002	0,0002	0,0002	0,0002	0,0002
$\mu / \frac{1}{s}$	0,010	0,002	0,002	0,002	0,002	0,002	0,002	0,002

In beiden Fällen verursacht das heuristische Verfahren aufgrund der Ungenauigkeit des ihm zugrundeliegenden Modells höhere Kosten für die Verbindungssteuerung als mit dem durch Simulation ermittelten besten konstanten Überreservierungsfaktor $K_Ü$ möglich wären. Dies belegt der Vergleich der simulativ gewonnenen Kurven in den Abb. 4.19 und 4.21. Die Differenz ist durchaus signifikant und steigt von kleineren hin zu größeren Überreservierungsfaktoren $K_Ü$ an.

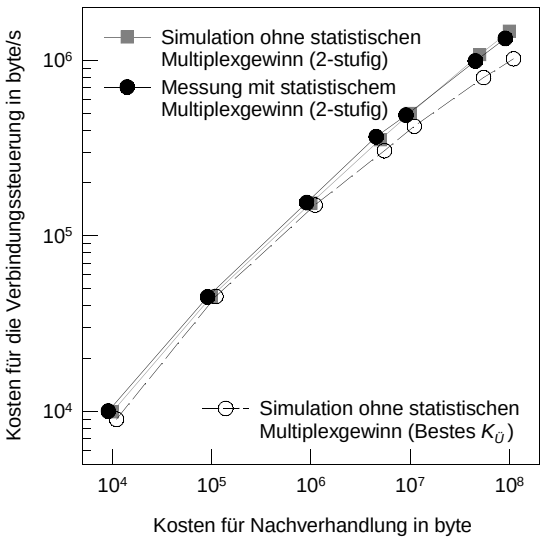

Abb. 4.21: *Kosten für die Verbindungssteuerung in Abhängigkeit von den für eine Nachverhandlung veranschlagten, auf die Übertragungsbandbreite bezogenen Kosten*

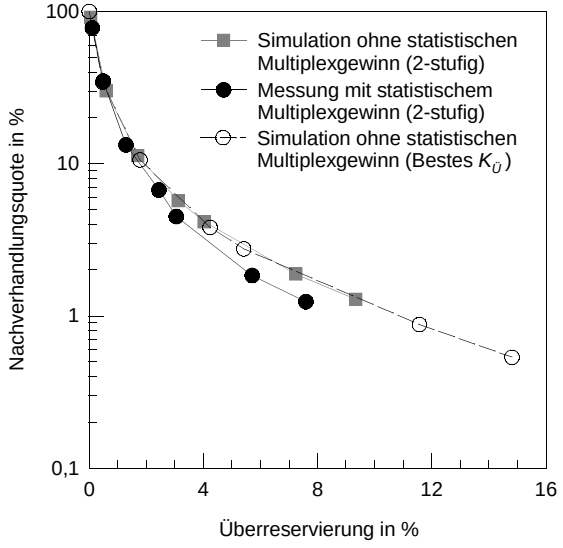

Abb. 4.22: *Nachverhandlungsquote für das Controlled-Load-Service-Aggregat in Abhängigkeit von der mittleren Überreservierung, wenn die Kosten gemäß Abb. 4.21 variiert werden*

Bei der Abwägung der Vor- und Nachteile des nur bedingt tauglichen eindimensionalen Markoff-Modells gegenüber Verfahren, die völlig ohne Verkehrsmodell auskommen, ist letzten Endes die

Forderung, das dynamische Bandbreitemanagement zeitlichen Schwankungen des Verkehrsangebotes anzupassen, ausschlaggebend für die Entscheidung zugunsten des modellbasierten Verfahrens gewesen. Stationäre, voneinander unabhängige Prozesse wie in Tab. 4.4 und 4.5 weisen aber vergleichsweise geringe zeitliche Schwankungen auf. Aus diesem Grunde wird zur Untersuchung des Verhaltens des heuristischen Verfahrens bei höherer Variabilität des Aggregates ein in [63] als CS-DMPP (*Continuous State Deterministically Modulated Poisson Process*) bezeichnetes Quellenmodell eingesetzt. Dieses Modell dient dort zur Modellierung von Korrelationen, die bei der Einwahl ins Internet auftreten, wenn sich unterschiedliche Nutzer beispielsweise abhängig von der Tageszeit ähnlich verhalten.

Als Beispiel wird wiederum ein homogenes Aggregat mit identischen Verbindungen der Bandbreite 30 kbyte/s simuliert. Anders als bei den in Abb. 4.17 und 4.18 dargestellten Untersuchungen sind die Ankunftsabstände nicht mehr negativ exponentiell mit dem konstanten Erwartungswert $\lambda = 10\,\mathrm{s}$ verteilt, sondern der Erwartungswert wird jeweils nach 30 min variiert gemäß einer Normalverteilung mit Standardabweichung 10 s. Negative Werte werden natürlich unterdrückt. Da die Verbindungen weiterhin nach einer exponentiell verteilten Belegungsdauer mit dem konstanten Erwartungswert $1/\mu = 100\,\mathrm{s}$ terminieren, ist der Verkehrswert nun abgeschnitten (nur Werte größer 0) normalverteilt um den Erwartungswert $\mathrm{E}\{Y\} = 1000$.

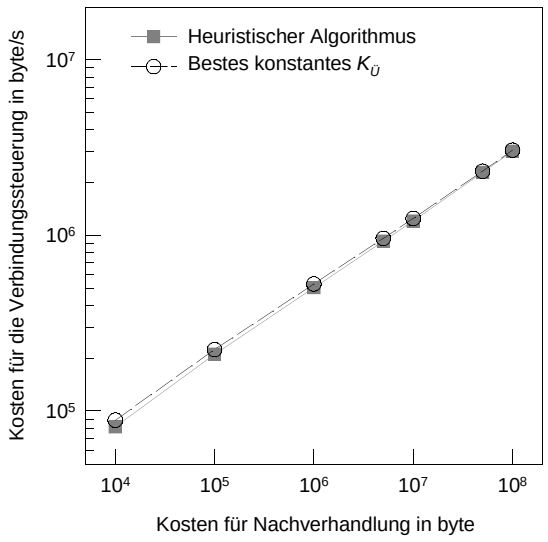

Abb. 4.23: *Kosten für die Verbindungssteuerung in Abhängigkeit von den für eine Nachverhandlung veranschlagten, auf die Übertragungsbandbreite bezogenen Kosten*

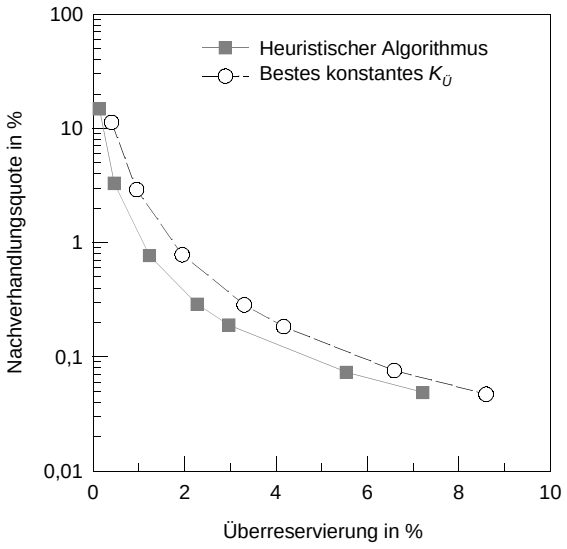

Abb. 4.24: *Nachverhandlungsquote für ein homogenes Aggregat mit normalverteiltem Verkehrswert Y. Die negativen Werte der Normalverteilung werden unterdrückt.*

Bei diesem zeitlich schwankenden Verkehrsangebot erweist sich das zweistufige heuristische Verfahren, das in Stufe 2 den Überreservierungsfaktor $K_Ü$ dynamisch an das Verkehrsangebot

anpaßt, dem einstufigen mit simulativ ermittelten besten konstanten Überreservierungsfaktor als leicht überlegen (Abb. 4.23 und 4.24). Die moderate Ersparnis allein würde aber kaum Stufe 2 rechtfertigen. Man muß aber sehen, daß der in einer relativ langen Simulation ermittelte beste konstante Überreservierungsfaktor in einem Vermittlungsknoten höchstens nach langfristig angelegten Verkehrsmessungen ermittelt werden kann. Das adaptive heuristische Verfahren erspart dem Netzbetreiber solche langwierigen, mit großen Unsicherheiten behaftete Studien und die statische Konfiguration der Aggregationsknoten.

4.5 Weiterführende Aggregationskonzepte

Am Beispiel von RSVP über ATM können sehr gut die Vorzüge einer auf Verkehrsdeskriptoren und Dienstgüteparametern basierenden Verkehrssteuerungsarchitektur demonstriert werden. Da ATM verbindungslose Netzdienste nicht unterstützt, sollte RSVP über ATM nur in Kombination mit IP- und ATM-fähigen Vermittlungsknoten zum Einsatz kommen. Dazu könnten beispielsweise MPLS-Knoten eingesetzt werden, die neben den ATM-Protokollen zur Steuerung von virtuellen Verbindungen auch den verbindungslosen Transport von IP-Paketen unterstützen (*Ships in the Night*). Nur im Zugangsnetzbereich bis zum Aggregationsknoten würden reine *Integrated Services* Router benötigt. In diesem Netz werden die Anforderungen sowohl von verbindungsorientierten als auch von verbindungslosen Netzdiensten berücksichtigt, und zwar beide auf eine effiziente Art und Weise, da die Aggregationsmechanismen den Aufwand für die Verbindungssteuerung reduzieren.

Trotzdem ist immer noch ein zusätzlicher Aufwand für eine letztlich doch unvollständige Integration von unterschiedlichen Signalisierungs- und Routingprotokollen, Adressierungsmechanismen und Paketformaten zu erbringen. Es ist daher erstrebenswert, die oben untersuchten Mechanismen zur Aggregation von Verkehrsströmen und dynamisches Bandbreitemanagement auch in eine homogenere, vollständig IP-basierte Netzarchitektur zu übertragen und dann auch mehrstufige Aggregation zu erlauben.

4.5.1 Erweiterungen von RSVP zur Unterstützung von Aggregation

Erweiterungen von RSVP zur Unterstützung rein IP-basierter Aggregationskonzepte sind in [22], [88] und jüngst in [17] vorgeschlagen worden. Die beiden erstgenannten und auch früheren Arbeiten diskutieren sowohl die Klassifizierung in einige wenige Verkehrsklassen mit Hilfe des *Type-of-Service* (TOS) Feldes des IP-Paketkopfes als auch das Tunneln als Aggregationsmecha-

nismen in der Nutzerebene. Beim Tunneln werden IP-Pakete an einem Aggregationsknoten in ein neues IP-Paket eingepackt, mit einer den Deaggregationsknoten identifizierenden IP-Zieladresse oder einem MPLS-Label versehen und bis zum Deaggregationsknoten übermittelt, ohne daß bis dahin die ursprüngliche IP-Adresse ausgewertet wird. Erst im Deaggregationsknoten wird die Operation am Aggregationsknoten invertiert. Das Tunneln zur Aggregation ähnelt also funktionell dem Transport von IP-Paketen in ATM-Verbindungen, wie er bei RSVP über ATM erfolgt.

Während nun diese beiden Ansätze mit relativ wenigen Änderungen von RSVP zur Unterstützung von Aggregation auskommen, geht der neuere Ansatz [17] weiter, indem er nicht nur die Klassifizierung in Verkehrsklassen mit Hilfe des inzwischen zum DS-Feld umgewidmeten TOS-Feldes im Aggregationsbereich vorschreibt, sondern auch mehrstufige Aggregation, dynamisches Bandbreitemanagement, die protokollgesteuerte Zuordnung von Deaggregationsknoten zu Aggregationsknoten und Routing-Aspekte berücksichtigt.

Die neuen Elemente sind jedoch nicht konsistent mit der *Integrated Services* Architektur verknüpft. Insbesondere die Vorschrift, DS-Feld basiert Pakete zu klassifizieren und so nach dem Vorbild der *Differentiated Services* Architektur in jedem Knoten immer wieder neue Aggregate zu bilden, erschwert die Einführung von Diensten mit garantierter maximaler Verzögerung [41]. Tunnel in Verbindung mit verbindungsbezogener Klassifizierung und Bedienung von Paketen könnten dagegen die Isolation von Aggregaten im Sinne des *Latency Rate* Modells sichern und sollten daher eine Option in einem umfassenden Aggregationskonzept sein.

An der vorliegenden Spezifikation überrascht aber vor allem, daß vom Deaggregationsknoten und nicht vom Aggregationsknoten (wie bei RSVP über ATM) Verbindungen Aggregaten zugeordnet werden, obgleich die Zuordnung von Verbindungen zu Aggregaten und die Neuberechnung der Bandbreite des Aggregats doch von der Verkehrssteuerungsarchitektur des Aggregationsknotens abhängen. Da außerdem die RSVP-Prozeduren im Aggregationsbereich ausgelöst werden, bevor die Reservierung erfolgreich vom Empfänger bis zum Aggregationsknoten außerhalb des Aggregationsbereichs aufgebaut worden ist, kann der Verkehrsdeskriptor (RSVP-*TSpec*) des Aggregats in der vorzeitig ausgelösten *Path* Meldung nicht den tatsächlichen Verkehr widerspiegeln, zumal ja noch gar nicht entschieden ist, welchem Aggregat die neue Verbindung zugeordnet werden wird.

4.5.2 *Overlay*-Modell

Mehrstufige verbindungsorientierte Aggregation und Deaggregation stellt tatsächlich hohe Anforderungen an die Signalisierungsprotokolle. Bei RSVP über ATM, das als Spezialfall eines verallgemeinerten Konzeptes zur mehrstufigen Aggregation angesehen werden kann, wird jede im

Bedarfsfall aufgebaute virtuelle Verbindung aus Sicht der Routing-Mechanismen logisch wie eine Schnittstelle zu anderen Schicht-2-Technologien behandelt. Dies ist der klassische Fall einer wie oben angesprochen unvollständigen Integration im Sinne des *Overlay* Modells. Auf diese Weise liegen am Aggregationspunkt detaillierte Topologieinformation über das IP-Netz außerhalb des ATM-Netzes vor, und insbesondere sind die IP-Adressen der Randknoten als *Next Hop* gespeichert. Da zusätzlich alle Rand- und damit Deaggregationsknoten des ATM-Netzes ihre IP- und ATM-Adresse registrieren lassen, kann der Endpunkt des Aggregates einfach mit den einschlägigen ATM-Adressenauflösungsprotokollen adressiert werden.

Übertragen auf mehrstufige Aggregation in einem durchgängig IP-basierten Netz, können die Protokolle zur Auflösung von IP- zu ATM-Adressen durch ein Protokoll ersetzt werden, das im wesentlichen aus Anfragen zur Ermittlung von Aggregations- und Deaggregationsknotenpaaren und den entsprechenden Antworten besteht. Dazu müssen die Signalisierungsmeldungen Informationselemente mitführen, in denen sich sowohl Aggregations- als auch Deaggregationsknoten registrieren lassen, die auf dem Weg zum Ziel passiert werden, es sei denn, sie sind durch früher aufgebaute Tunnel zum Zeitpunkt der Ausführung der Prozedur bereits unsichtbar geworden [141]. Die virtuellen Verbindungen von ATM werden also durch IP-Tunnel ersetzt, über die dann natürlich auch unaggregierte Routing-Informationen ausgetauscht werden. Dafür kann aber der sendergesteuerte Aufbau des Aggregates von RSVP über ATM übernommen werden.

Zwar läßt sich diese Lösung wohl am schnellsten implementieren, nicht zuletzt weil sie heute gängigen Praktiken am nächsten kommt, als *Overlay* Lösung erhöht sie jedoch aus den oben genannten Gründen den für das Routing notwendigen zusätzlichen Aufwand.

4.5.3 Mehrstufige Aggregation mit RSVP und virtuellen Verbindungen

Dies ist bei MPLS anders, denn über LSPs werden keine Routing-Informationen ausgetauscht [188]. Bei rein topologieorientierter *Label* Verteilung, dem *Topology Driven Label Assignment* [39], markiert nach erfolgreichem Aufbau des LSP *der* Router den Deaggregationspunkt, an dem sich ein FEC aufspaltet. Leider können aus der Verteilung von IP-Adressen nur bedingt Rückschlüsse auf die tatsächliche Topologie des Netzes gezogen werden. Außerdem wird im allgemeinen die Granularität, der Grad der Aggregation, wie er aufgrund der Routing-Protokolle zustande kommt, wenn überhaupt, dann nur gelegentlich den Anforderungen mehrstufiger verbindungsorientierter Aggregation genügen. Daher ist damit zu rechnen, daß bei zusätzlich oder ausschließlich QoS basierter Aggregation die notwendigen Zusatzinformationen auf andere Weise beschafft werden, wenn eine aus Sicht der Verkehrssteuerung sinnvolle mehrstufige Aggregation, unter Umständen auch noch

über den MPLS-Netzbereich hinaus, erfolgen soll. Um wie in der vorliegenden Arbeit für verschiedene Verkehrsklassen getrennte Aggregate parallel zwischen einem Aggregations-Deaggregations-Knotenpaar aufzubauen, muß neben dem topologiebasierten Aufbau von LSPs auch ein Aufbau nach expliziter Aufforderung möglich sein, ein *Request Driven Label Assignment* [39].

Die jüngste, oben ausführlich diskutierte Erweiterung von RSVP zur Unterstützung eines rein IP-basierten Aggregationskonzeptes [17] kann durchaus diese Aufgabe erfüllen. Man könnte sie mit RSVP-TE [16], einer Erweiterung von RSVP zum Aufbau von LSPs, kombinieren und so abwandeln, daß unbedingt Tunnel bzw. MPLS LSPs für die Aggregate verwendet werden, d. h. nicht mehr nur die im DS-Feld kodierten *Behaviour Aggregates* unterschieden werden. Wenn ferner diese Aggregate immer vom Aggregationsknoten aufgebaut werden, und zwar erst dann, wenn der Aufbau der Reservierung für die nächst tiefere Aggregationsebene bis zu diesem Knoten fortgeschritten ist (wie bei RSVP über ATM, vgl. Abb. 4.3), erhält man Prozeduren, die denen von RSVP über ATM ähneln, allerdings unter Umgehung der Nachteile dieses *Overlay* Modelles. RSVP-TE ist für diesen Ansatz insofern bedeutsam, als es neue Objekte zur Zuordnung von *Labels* zu Verbindungen in RSVP-Meldungen einzufügen erlaubt, die beim Aufbau einer Reservierung für ein Aggregat auch gleichzeitig für den Aufbau des entsprechenden LSP sorgen.

Sobald eine RSVP *Path* Meldung einen Aggregationsknoten passiert, kann in Anlehnung an [17] die Meldung so im IP-Paketkopf gekennzeichnet werden, daß sie bis zum Deaggregationsknoten nicht beachtet wird, beispielsweise indem die Aggregationsebene in einem bislang unbenutzten Feld der *Router Alert* Option eingetragen wird. Auf dem Rückweg wird mit der *Resv* Meldung genauso verfahren. Das hat zur Folge, daß der Aggregationsknoten den Deaggregationsknoten, den Absender dieser *Resv* Meldung, aus dem IP-Paketkopf der *Resv* Meldung auslesen kann. Dies genügt ihm, um nach Prüfung der maßgeblichen Parameter seine Aggregationsentscheidung zu treffen, mit RSVP-TE einen LSP für den Aggregationsbereich bis zum Deaggregationsknoten zu schalten und die notwendigen Ressourcen zu reservieren. Unter der Voraussetzung, daß jeder an der Aggregation und Deaggregation beteiligte Knoten um seine Aggregationsebene weiß, kann ebenso eine mehrstufige Aggregationsarchitektur realisiert werden, ohne daß dazu Aggregations- und Deaggregationsknoten unaggregierte Topologieinformation zwischen den Netzen außerhalb des Aggregationsbereiches austauschen.

Als Alternative zu solchen Architekturen, die Aggregations- und Deaggregationsknoten in verschiedenen Ebenen mindestens in der Steuerungsebene über Verbindungsaggregate miteinander verknüpfen, haben Pan et al. [158] jüngst das aggregationsfähige *Border Gateway Reservation Protocol* (BGRP) vorgeschlagen, das alle Reservierungen zu einem Ziel(bereich) unabhängig von ihrem Ausgangspunkt zu Aggregaten zusammenfaßt. Sie argumentieren (allerdings ohne mehrstu-

fige Aggregation zu berücksichtigen), daß diese Aggregationsstrategie am besten skaliert. Auch in diesem Konzept sind natürlich die in der vorliegenden Arbeit untersuchten Verkehrssteuerungsmechanismen unentbehrlicher Bestandteil.

4.5.4 Aggregation von *Multicast* Verbindungen

Die IP-Mechanismen zur Unterstützung von *Multicast* werden sich wohl nur sehr schwer in eine Architektur zur verbindungsorientierten Aggregation einbetten lassen. *Multicast* (Gruppenruf) wird in [101] definiert als ein Ruf, bei dem dieselbe Information an eine Teilmenge aller Benutzer bzw. Endgeräte übermittelt wird. Im verbindungslosen IP werden dazu die Datagramme mit einer Gruppenadresse (*Multicast Address*) versehen und entlang der Zweige eines Baumes, der zuvor durch den Austausch von Nachrichten der *Group Membership* [50, 68] und *Multicast Routing* [145] Protokolle von den Empfängern aus aufgebaut worden ist, von Knoten zu Knoten geroutet. Die Router assoziieren dabei mit der Gruppenadresse keineswegs eine Liste von Mitgliedern der Gruppe, sondern lediglich eine Liste von lokalen Ports, die zum Baum der *Multicast* Sitzung gehören. Auf diese Weise muß sich jeder Knoten für die Sitzung nur seinen Vater und seine Kinder im entsprechenden Baum merken.

Wendet man die oben beschriebenen Prozeduren des Tunnelns von RSVP-Meldungen durch den Aggregationsbereich an, werden beim Aggregationsknoten *Resv* Meldungen von jedem Deaggregationsknoten ankommen, da diese nicht wie sonst in Routern im Aggregationsbereich verarbeitet und in Verzweigungspunkten zusammengeführt werden können. Der Aggregationsknoten müßte also letztlich für jeden Deaggregationsknoten im Baum die Datagramme kopieren und über neue oder bestehende Aggregate verteilen. Diese Lösung ist ineffizient, so daß man wahrscheinlich besser ganz auf IP-*Multicast* verzichtet und statt dessen die Pakete mehrfach Punkt-zu-Punkt zustellt oder *Multicast* Verbindungen einfach nicht aggregiert.

5 Verfahren zur Regelung elastischen Verkehrs bei Überlast

Ihrem einfachen verbindungslosen *Best-Effort* Netzdienst entsprechend, transportieren IP-Netze heute noch vorwiegend Datenströme von Anwendungen, die variable Verzögerungszeiten und Datenraten verkraften. Ein Großteil dieser elastischen Anwendungen setzt zur Übertragung der Daten über IP-Netze das Transportprotokoll TCP ein, das nicht nur den Datenstrom gegen Verluste und Übertragungsfehler absichert, sondern auch die Datenrate an die Auslastung des Netzes anpaßt. Als Protokoll der Transportschicht wird TCP nur in den Endsystemen eingesetzt. Dort arbeitet TCP verbindungsorientiert.

Die Verkehrscharakteristik elastischer Anwendungen läßt sich nicht mit den von diensteintegrierenden Netzen bekannten Verkehrsdeskriptoren beschreiben. Ebenso wenig sind die Dienstgüteanforderungen quantifizierbar. Außerdem ist der Datenaustausch im Internet nicht selten von kurzen Transaktionen geprägt. Damit scheiden wichtige Argumente für verbindungsorientiertes Verkehrsmanagement aus. Aus diesem Grunde sind eine Reihe von Aktivitäten zu beobachten, die über Verbesserungen der auf die Endsysteme beschränkten Überlastregelung von TCP hinaus das Verkehrsmanagement um verbindungslos arbeitende Mechanismen in Netzknoten erweitern.

Ziel der vorliegenden Arbeit ist es, unter systematischem Einsatz einheitlicher und theoretisch fundierter Leistungsmaße und ausgereifter Simulationsmeßtechnik das Potential der unterschiedlichen, zum Teil in den Endsystemen, zum Teil in den Netzelementen ansetzenden Vorschläge zu vergleichen und die Komponenten zu einem im Netzinnern verbindungslosen und daher auch skalierbaren Verkehrsmanagement zu integrieren.

5.1 Überlastregelung von TCP *Reno*

Die ursprüngliche Spezifikation von TCP, RFC 793 [165], sieht lediglich eine Ende-zu-Ende Flußregelung mit Hilfe eines vom Empfänger gesteuerten Sendefensters (*Receiver Advertised Window*) vor. Die Fehlersicherung beruht im wesentlichen auf der Wiederholung verlorengegangener Segmente des Datenstroms, nachdem eine bestimmte Zeitspanne verstrichen ist, in der der Sender vergeblich auf die Quittierung des Datenstromsegmentes am unteren Ende des Sendefensters gewartet hat (*Retransmission Timeout*, RTO). Aufgrund der Annahme eines nicht nur in bezug auf Übertragungsfehler, sondern auch in bezug auf die Übertragungsreihenfolge unzuverlässigen Netzes unterstützt TCP nur positive Quittierungen. Quittierungen sind bei TCP *Reno* noch dazu kumuliert, d. h. eine Quittierung für ein bestimmtes Byte des Datenstroms bestätigt auch gleichzeitig den fehlerfreien Empfang aller früheren Bytes dieses Datenstroms.

Mitte der achtziger Jahre sind aber offenbar immer häufiger Überlastsituationen beobachtet worden, die zu massiven Einbrüchen beim Nutzdurchsatz geführt haben. Um solche Überlastsituationen zu vermeiden, in denen bei weiter wachsendem Verkehrsangebot der Nutzdurchsatz nicht nur nicht mehr zunimmt, sondern sogar abnimmt (*Congestion Collapse*), hat Jacobson [104] eine Reihe zusätzlicher Algorithmen zur Überlastregelung (*Congestion Control*) entwickelt, die unter der Bezeichnung TCP *Reno* bekannt geworden sind und viele Jahre später nach nur geringfügigen Modifikationen als RFC 2581 [2] standardisiert worden sind.

TCP-*Reno*-Sender passen ihre Senderate an die Lastsituation im Netz an, müssen aber nicht unbedingt einen RTO abwarten, um ein verlorengegangenes Paket zu wiederholen, und sollten – auch wenn dies in RFC 2581 [2] nicht ausdrücklich vorgeschrieben wird – einen von Karn [110] vorgeschlagenen Algorithmus zur Messung der Umlaufzeit (*Round Trip Time*, RTT) einsetzen, der auch in Überlastsituationen zuverlässige Ergebnisse liefert.

Die Algorithmen zur Überlastregelung von TCP *Reno* gehen davon aus, daß Paketverluste Überlast im Netz signalisieren, auf die mit einer Verkleinerung des Sendefensters reagiert werden kann. Dies ist gleichbedeutend mit einer Reduktion der Senderate. Denn das Sendefenster begrenzt die Datenmenge, die gesendet werden darf, ohne daß eine Quittierung für das erste Segment am unteren Ende des Fensters eintrifft, was erst nach dem Verstreichen einer der Umlaufzeit entsprechenden Zeitspanne möglich ist. Die Überlastregelung unterscheidet dazu bei einer aktiven Verbindung die in Abb. 5.1 dargestellten vier Phasen: *Slow Start*, *Congestion Avoidance*, *Fast Retransmit* und *Fast Recovery*. Diese Phasen bestimmen, wie das *Congestion Window* (CW) $cwnd(t)$ beim Empfang von Quittierungen angepaßt wird. Im Rahmen der beim Öffnen des Sockets festgelegten maximalen

Fenstergröße bestimmt nun das Minimum von *Receiver Advertised Window* der Flußregelung und das CW der Überlastregelung das Sendefenster und nicht mehr allein das *Receiver Advertised Window* wie in RFC 793 [165].

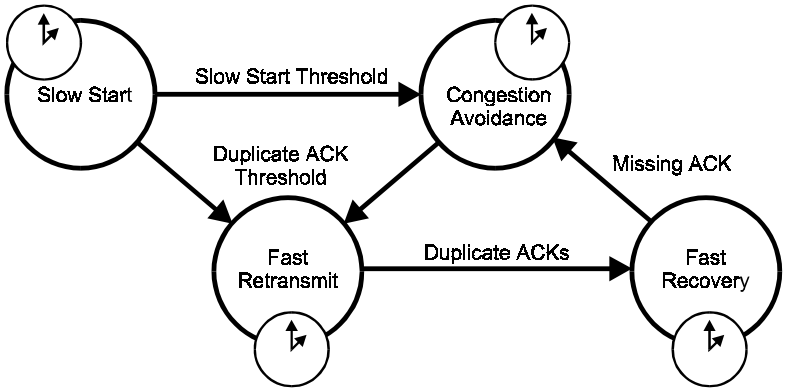

Abb. 5.1: *Die Überlastregelung von TCP Reno unterscheidet vier verschiedene Phasen. Diese Phasen bestimmen, wie das Congestion Window beim Empfang von Quittierungen erhöht wird. Die Verbindung startet in der Phase Slow Start. Sie kehrt nach einem RTO oder bei entsprechend langer Inaktivität dahin zurück.*

Eine TCP-Verbindung startet in der Phase *Slow Start* mit einem CW, dessen Größe durch den Parameter *Initial Window* bestimmt ist. Dieser Parameter muß auf einen Wert kleiner oder gleich dem Zweifachen der MSS (*Maximum Segment Size*) gesetzt werden [2]. Solange das CW nicht die Schwelle *Slow Start Threshold* $ssthresh(t)$ passiert hat, darf das CW beim Empfang einer Quittierung für ein neues Segment des Datenstroms das CW um MSS erhöhen. Wenn die Umlaufzeit konstant und die Quelle immer aktiv ist (*Greedy Source*), führt dies zu einem exponentiellen Anstieg des CW. Wenn die Schwelle $ssthresh(t)$ erreicht wird, geht die Verbindung in die Phase *Congestion Avoidance* über. Der Anstieg des CW verlangsamt sich dann um $MSS^2 * cwnd(t)^{-1}$ pro eintreffender neuer Quittierung, was unter den o. g. Voraussetzungen einem linearen Anstieg entspricht.

Auf diese Weise vergrößert TCP das CW immer weiter, so daß es fast zwangsläufig zu Verlusten kommt. Auf der Empfangsseite führen Paketverluste (ebenso wie fehlerhaft empfangene Pakete) zu Lücken im Datenstrom. Da TCP *Reno* nur positive und nur kumulierte Quittierungen kennt, kann der Empfänger auch dann keine neuen Quittierungen senden, wenn auf Lücken weitere, völlig korrekt empfangene Daten folgen. Statt dessen sendet TCP beim Empfang von *Out-of-Order* [31] Segmenten, also Segmenten, die sich nicht nahtlos an den bereits lückenlos empfangenen Datenstrom anschließen, duplizierte Quittierungen (*Duplicate Acknowledgements*) zurück zum Sender. Bezüglich der *Acknowledgement Number* [165], der Sendefolgenummer des letzten Bytes des lückenlos empfangenen Teils des Datenstroms plus eins, und dem Empfangsfenster sind sie exakte Kopien früherer Quittierungen. Deshalb kann ein Sender aus drei aufeinanderfolgenden duplizierten

Quittierungen relativ sicher schließen, daß mit hoher Wahrscheinlichkeit Pakete auf dem Wege zum Empfänger verloren oder verfälscht worden sind und wiederholt ein Segment des Datenstroms ab dem jetzt schon mehrfach quittierten Byte (*Fast Retransmit*). Aufgrund der Annahme, daß Paketverluste Überlast im Netz signalisieren, setzt der Sender außerdem die Schwelle $ssthresh(t)$ auf die Hälfte der aktuellen Anzahl der gesendeten, aber noch nicht bestätigten Bytes (*Flight Size*) und das CW $cwnd(t)$ auf den Wert $\min\{ssthresh(t)+3\,MSS, 2\,MSS\}$, um den drei Paketen Rechnung zu tragen, die offensichtlich inzwischen noch am Ziel angekommen sind und die duplizierten Quittierungen ausgelöst haben.

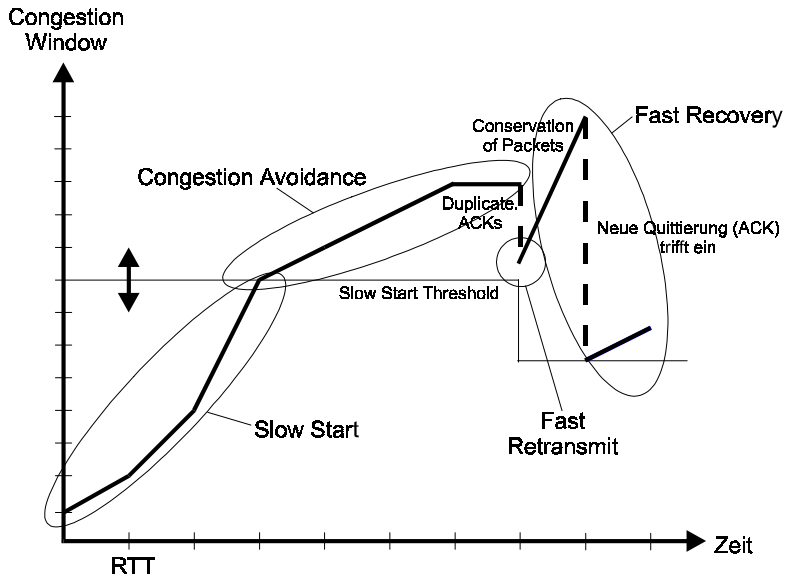

Abb. 5.2: *Bei konstanter Umlaufzeit (RTT) und einer immer aktiven Quelle steigt das CW im Slow Start exponentiell und in Congestion Avoidance linear an. Nach einem Fast Retransmit hält zunächst ein linearer Anstieg des CW die Anzahl unquittierter Pakete konstant. Wenn das wiederholte Paket schließlich quittiert wird, fährt der Sender in Congestion Avoidance fort.*

Da sich bis zum Eintreffen einer neuen Quittierung (frühestens nach einer Umlaufzeit nach der Wiederholung) der untere Rand des Fenster nicht verschiebt, aber weitere duplizierte Quittierungen signalisieren, daß immer noch Pakete unversehrt das Ziel erreichen, wird in der Folge das CW um MSS pro duplizierter Quittierung erhöht. Wenn die Quelle in dieser Phase aktiv ist oder auch so noch genug Daten im Sendepuffer zwischengespeichert sind, löst dann jede duplizierte Quittierung die Übertragung eines neuen Segments aus (*Conservation of Packets*). In Abb. 5.2 sieht man den auffälligen steilen Anstieg auch deshalb so deutlich, weil beim Eintreffen einer neuen Quittierung das CW auf die Schwelle $ssthresh(t)$ zurückgesetzt wird. Nach dieser Phase vom Feststellen des Verlustes bis zur Bestätigung der erfolgreichen Wiederholung des Segmentes, der Phase *Fast Recovery*, kehrt TCP in *Congestion Avoidance* zurück. Die Abb. 5.3 zugrundeliegenden Simulationen bestätigen diesen charakteristischen Verlauf.

Unter Umständen, insbesondere wenn nur wenige Daten zur Übertragung anstehen, ist das CW so klein, daß auch beim Verlust von nur einem Paket häufig nicht drei duplizierte Quittierungen eintreffen können. Bei *Limited Transmit* gemäß RFC 3042 [3] darf TCP deshalb beim Empfang von zwei duplizierten Quittierungen auch dann zwei weitere Pakete senden, wenn das *Congestion Window* dies eigentlich verbietet. Dies soll die Wahrscheinlichkeit eines *Retransmission Timeout* bei Verlusten im Vergleich zu den Varianten *Reno* und *New Reno* und SACK (siehe unten) weiter reduzieren.

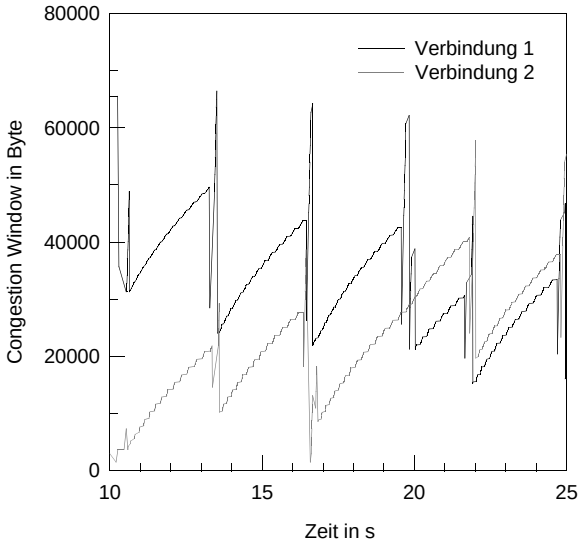

Abb. 5.3: *Der zeitliche Verlauf des CW von zwei konkurrierenden TCP-Verbindungen, deren Datenquellen zeitlich versetzt starten und danach immer aktiv sind. Bei ungefähr 13,3 s stellen beide Verbindungen einen Verlust fest, den sie durch einen Fast Retransmit korrigieren. Das CW von Verbindung 1 fällt dabei deutlich stärker. Mit jedem neuen Verlust rücken die Kurven zusammen. In Congestion Avoidance laufen die Kurven dagegen fast parallel.*

Abb. 5.3 zeigt auch, daß Verbindungen nach dem Start, nach einem RTO (Verbindung 2 ungefähr bei 16,6 s) oder entsprechend langer Inaktivität sehr schnell den *Slow Start* durchlaufen und zumindest bei moderaten Verlusten die meisten Daten in *Congestion Avoidance* (die längeren fast linearen Zuwächse) übertragen werden. *Congestion Avoidance* aber ist gekennzeichnet durch kleine, konstante Zuwachsraten des CW beim regelmäßigen Zustrom neuer Quittierungen. Dagegen wird das CW bei Verlusten sehr drastisch um den Faktor zwei reduziert. In der englischsprachigen Literatur, u. a. in [104], bezeichnet man *Congestion Avoidance* deshalb auch als einen *Additive Increase Multiplicative Decrease* (AIMD) Algorithmus. In Abb. 5.3 ist gut zu erkennen, wie AIMD das CW und folglich den Durchsatz zweier konkurrierender Verbindungen nivelliert.

Wie im Zusammenhang mit Abb. 5.1 erwähnt, kehrt TCP in den *Slow Start* zurück, wenn eine bestimmte Zeitspanne verstrichen ist, in der der Sender vergeblich auf die Quittierung des Daten-

stromsegmentes am unteren Ende *SND.UNA* [165] des Sendefensters gewartet hat. Die Schwelle $ssthresh(t)$ wird wie beim *Fast Retransmit* halbiert, das CW jedoch auf ein MSS (*Loss Window*) reduziert. Anschließend wird das Segment wiederholt und der RTO mit einem Faktor multipliziert, der zunächst auf zwei gesetzt und bei weiteren vergeblichen Versuchen bis zu einer gewissen Grenze weiter verdoppelt wird (*Exponential Back-off*). Wenn das Segment schließlich quittiert wird, wird der Faktor auf eins zurückgesetzt.

Bezüglich der Frage, welches Segment beim Eintreffen einer Bestätigung für dieses Segment, aber nicht für den ganzen bereits gesendeten Teil des Datenstroms übertragen werden soll, ist die Literatur uneinheitlich. Ausschlaggebend dafür ist, ob beim RTO die interne Variable *SND.NXT* [165] auf *SND.UNA* zurückgesetzt wird oder nicht. Für RFC 793 [165] ist eine Quittierung nur dann akzeptabel, wenn sie ein Byte zwischen *SND.UNA* und *SND.NXT* bestätigt. Diese Anforderung macht nur dann wirklich Sinn, wenn beim RTO *SND.NXT* nicht verändert wird und daher gleich der höchsten übertragenen Sendefolgenummer plus eins ist. Comer [45] bestätigt diese Interpretation. Nach einem RTO wird dann nur ein Segment wiederholt. Bei dieser Vorgehensweise werden keine korrekt übertragenen Segmente unnötig wiederholt, es sei denn der RTO selbst ist zu früh ausgelöst worden (*False Retransmit*), dafür pausiert der Sender aber normalerweise bis zur Bestätigung eines Segments in der Nähe des oberen Randes des Fensters oder bis zum nächsten RTO. Um mehrere Paketverluste schneller behandeln zu können, setzen viele Implementierungen beim RTO *SND.NXT* auf *SND.UNA* und machen im *Slow Start* weiter, so als ob die Segmente bis zu dem ursprünglichen *SND.NXT* noch nicht übertragen worden wären. Quittierungen bis hoch zu dem ursprünglichen *SND.NXT* werden zwar akzeptiert, dennoch werden zugunsten einer schnelleren Fehlerbehandlung unnötige Wiederholungen billigend in Kauf genommen. Belege für diese Praxis sind in Abschnitt 5 von [73] und in [94] zu finden[3]. Der Autor schlägt daher als Kompromiß vor, beim Eintreffen von Quittierungen, die nicht alle Segmente bestätigen (*Partial Acknowledgements* [73]), die bereits zum Zeitpunkt des Eintreffens der dritten duplizierten Quittierung unterwegs gewesen sind, unabhängig von der tatsächlichen Größe des CW nur ein weiteres Segment zu wiederholen. Die im Rahmen der vorliegenden Arbeit entstandene Simulationsumgebung unterstützt alle drei Optionen.

Zur Bestimmung einer angemessenen Wartezeit bis zum Auslösen eines RTO wird die Umlaufzeit mit einer Stoppuhr stichprobenartig [185] oder mit Hilfe von Zeitstempeln [105] gemessen. Unter Einbeziehung der mittleren Abweichung wird nach jeder Messung ein neuer Wert festgesetzt [185], letztlich in Abwägung der Vorteile einer schnellen Reaktion auf Paketverluste und der Gefahr von unnötigen Wiederholungen.

3 Keine der dem Autor bekannten Spezifikationen beschreibt das Verhalten von TCP *Reno* nach einem RTO. Eine diesbezügliche Anfrage an das Diskussionsforum end2end-interest@postel.org hat keine Klarheit gebracht.

Paketverluste können die Messung der Umlaufzeit stören. Bei massiven Verlusten geht so viel Zeit für die Wiederholung von Segmenten verloren, daß aufgrund der kumulierten Quittierungen die Messung der Umlaufzeit eines Paketes auch dann zu falschen Ergebnissen führt, wenn das Paket selbst problemlos übertragen worden ist. Bereits vor der Einführung der Überlastregelung von TCP *Reno* hat Karn als Gegenmaßnahme einen Algorithmus [110] entwickelt, den der Autor folgendermaßen auf TCP *Reno* übertragen hat:

Sobald ein Segment wiederholt werden muß (gleichgültig ob die Wiederholung durch einen RTO, *Fast Retransmit* oder auf andere Weise ausgelöst wird), wird die laufende Messung der Umlaufzeit für ungültig erklärt, die zu diesem Zeitpunkt höchste Sendefolgenummer gespeichert und keine Messung mehr durchgeführt, bis diese Sendefolgenummer quittiert wird. Als RTO wird der alte Wert beibehalten, der unter Umständen aufgrund des *Exponential Back-off* nach einem RTO weiter ansteigt. Wenn die Wiederholung eines Segmentes aufgrund eines RTO quittiert wird, wird wie bisher auch das *Exponential Back-off* wieder aufgehoben, der RTO jedoch erst dann aktualisiert, wenn auch eine neue Messung der Umlaufzeit erfolgreich ist. Dies ist notwendig, um in Phasen massiver Verluste, wenn viele Messungen ausfallen, statt dessen mit Hilfe des *Exponential Backoff* noch einen konservativen RTO zu erhalten.

RFC 1122 [31] erlaubt Empfängern, auf die Quittierung jedes einzelnen Paketes zu verzichten (*Delayed Acknowledgement*). Eine Quittierung sollte jedoch nicht länger als 0,5 s hinausgezögert werden. Außerdem wird empfohlen, nicht mehr als zwei MSS Byte unquittiert anstehen zu lassen. Das Verzögern von Quittierungen beeinflußt die Überlastregelung spürbar.

TCP *Reno* gemäß RFC 2581 [2] setzt anders als frühere Varianten von TCP konsequent die schon bei ABR [11] verfolgte Idee des *Use-it-or-loose-it* um. Verbindungen verlieren bei Inaktivität ganz und bei Verlusten immerhin den ungenutzten Teil des Sende-Kredites, als den man das CW interpretieren kann.

5.2 Optionale Ergänzungen der Überlastregelung

Die im Rahmen der Fehlersicherung nach Paketverlusten notwendigen Sendewiederholungen sind besonders aufwendig, wenn sie wie bei TCP nur in den Endpunkten der Verbindung ausgeführt werden. Da der Sender aus dem Rückfluß der kumulierten Quittierungen nicht sehr präzise über die bereits fehlerfrei empfangenen Teile des Datenstroms informiert wird, wiederholt er unter Umständen Teile des Datenstroms unnötig. Mit *Fast Retransmit* kann auch nicht mehr als ein Paket

pro Umlaufzeit wiederholt werden, da mit der Wiederholung eines weiteren Paketes bis zur Rückkehr der Quittierung dieses Paketes gewartet werden muß. Bei gehäuft auftretenden Verlusten sind die Durchsatzeinbußen deshalb unverhältnismäßig hoch.

Es liegt daher nahe, in Quittierungen durch Einfügen der Option *Selective Acknowledgement* (SACK) [143] im Paketkopf mehr Informationen auch über einzelne erfolgreich übertragene Datenblöcke zu übertragen bzw. kumulierte Quittierungen besser zu interpretieren, wie dies in der als TCP *New Reno* [73] standardisierten Variante geschieht.

Statt auf die bei Pufferüberlauf meist gehäuft auftretenden Verluste zu warten und zu reagieren, interveniert aktives Puffermanagement durch kontrolliertes vorzeitiges Verwerfen von Paketen. Verluste in Folge von Pufferüberläufen sind schließlich das Ergebnis einer bereits eine gewisse Zeit andauernden Überlast. Die Einwirkung des Netzes auf die Sender soll so weniger abrupt sein und die Rate der Datenströme weniger dynamisch auf- und abschwingen. Beispiele sind *Random Early Detection* (RED) [71] und adaptives RED [76].

Effizienter ist es natürlich, in dieser Phase der Überlast die Pakete nicht zu verwerfen, sondern wenn möglich nur zu markieren, den Empfänger so zu modifizieren, daß er die Überlast zurückmeldet, und den Sender zu veranlassen, ähnlich wie bei Paketverlusten das CW zu reduzieren. Dann entfällt der an sich überflüssige Aufwand für die Wiederholung von Paketen, die noch Platz im Wartespeicher gehabt hätten. *Explicit Congestion Notification* (ECN) [166] verfolgt diese Strategie.

5.2.1 *Selective Acknowledgement* (SACK)

RFC 2018 [143] schafft die protokolltechnischen Voraussetzungen, daß in den weiterhin kumulierten Quittierungen zusätzlich der Empfang von bis zu drei durch Lücken voneinander bzw. vom unteren Rand des Empfangsfensters getrennten Datenblöcken angezeigt werden kann. Eine Reihe von Regeln stellt sicher, daß diese zusätzliche Information nicht nur aktuell und vollständig, sondern auch redundant übertragen wird. Denn TCP sichert Quittierungen, die nicht gleichzeitig Datenpakete sind, nicht gegen Fehler.

Endgültig quittiert sind die mit SACK zusätzlich rückgemeldeten Datenblöcke jedoch nicht. Der Empfänger ist nicht verpflichtet, alle diese Datenblöcke unbefristet im Empfangspuffer zu belassen, auch wenn er dies natürlich im eigenen Interesse tun sollte. Dann nämlich kann der Sender – mit einer gewissen Verzögerung – sehr detaillierte Informationen über die Situation im Empfangspuffer sammeln, die er für frühzeitige und gezielte Wiederholungen nutzen kann. Falls dennoch ein RTO

nicht vermieden werden kann, schließt der Sender, daß möglicherweise sein Abbild des Empfangspuffers nicht mehr korrekt ist und löscht alle zuvor gesammelten Informationen.

Weder RFC 2018 [143] noch RFC 2833 [75] standardisieren die Integration der Überlastregelung von TCP mit SACK. Der Autor hat daher auf der Grundlage von [61] einen letztlich etwas einfacheren, mitunter geringfügig aggressiveren Ansatz gewählt. Wie in [61] verwaltet der Sender ein *Score Board*, das die vom Empfänger in Form von Quittierungen und zusätzlich durch SACK bereitgestellten Informationen sammelt und auf Anfrage die Datensegmente identifiziert, die wiederholt werden müssen. Zunächst ruft der Sender diese Information aber noch nicht ab. Beim Eintreffen der dritten duplizierten Quittierung wird zunächst auch nur einer neuen internen Variablen, die in [73] ähnlich verwendet und dort *Recover* genannt wird, die höchste bisher übertragene Sendefolgenummer zugewiesen und ein *Fast Retransmit* durchgeführt. Wenn der Empfänger über das mit *Fast Retransmit* wiederholte Segment hinaus keine Lücken gemeldet hat, bleibt anschließend sogar die Phase *Fast Recovery* gegenüber TCP *Reno* unverändert. Wenn dagegen weitere Lücken bekannt sind und in der Folge duplizierte oder partielle Quittierungen (*Duplicate/Partial Acknowledgements*) eintreffen, werden nicht wie bei TCP *Reno* lediglich neue Segmente ab *SND.NXT* übertragen (sobald das CW das zuläßt), sondern in aufsteigender Reihenfolge zusätzlich Datensegmente wiederholt, um die vom *Score Board* registrierten Lücken zu schließen. Eine nicht duplizierte Quittierung wird partiell genannt, wenn sie nicht alle Daten bis zur Sendefolgenummer *Recover* abdeckt.

Auf die vorübergehende Anpassung des CW bis zum Abschluß der *Fast Recovery* wird verzichtet (*Conservation of Packets* in Abb. 5.2). Statt dessen wird aufgrund der im *Score Board* gespeicherten und auf Anfrage bereitgestellten Information geprüft, ob Daten wiederholt oder neu übertragen werden können. Ein Segment pro Quittierung wird wiederholt, wenn das *Score Board* eine Lücke im Bereich von *SND.UNA* und $SND.UNA + cwnd(t)$ plus der Anzahl der mit SACK bestätigten Daten findet. Bei der Übertragung neuer Daten muß auch noch explizit die durch das (vom Empfänger gesteuerte) Sendefenster gegebene Grenze respektiert werden. Wenn schließlich das Byte mit der Sendefolgenummer *Recover* quittiert wird, geht der Sender zur *Congestion Avoidance* über.

5.2.2 TCP *New Reno*

Auch ohne die zusätzlichen Informationen, die bei SACK vom Empfänger bereitgestellt werden, kann ein Sender aus dem Empfang einer partiellen Quittierung während der Phase *Fast Recovery* schließen, daß mit hoher Wahrscheinlichkeit das auf die quittierte Sendefolgenummer folgende

Datensegment ebenfalls verworfen worden ist. In der Spezifikation von TCP *New Reno*, RFC 2582 [73], wird dazu ein Vorschlag von Hoe [94] aufgegriffen und die *Fast Recovery* von TCP *Reno* entsprechend modifiziert. Die folgende Darstellung beschränkt sich auf die dort als *Slow-but-Steady* bezeichnete Variante.

Ebenso wie bei der oben erläuterten Implementierung von SACK wird beim Eintreffen der dritten duplizierten Quittierung die bis zu diesem Zeitpunkt höchste übertragene Sendefolgenummer in der internen Variablen *Recover* abgelegt. Abgesehen davon ändert sich bis zum Eintreffen neuer (also nicht duplizierter) Quittierungen gegenüber TCP *Reno* nichts. Dann aber wird geprüft, ob es sich bei der neuen Quittierung um eine partielle handelt oder ob sie alle Daten bis zur Sendefolgenummer *Recover* abdeckt. Ist die Quittierung partiell, wird das erste unquittierte Segment wiederholt, das CW um die Anzahl der neu bestätigten Bytes reduziert und dann noch um eine MSS erhöht. Soweit das CW dies danach zuläßt, wird auch ein neues Segment übertragen. Der Sender bleibt in *Fast Recovery*. Erst wenn die Sendefolgenummer *Recover* quittiert wird, setzt er das CW entweder auf die Schwelle $ssthresh(t)$ oder auf die aktuellen Anzahl der gesendeten, aber noch nicht bestätigten Bytes (*Flight Size*) plus eine MSS, je nachdem welcher der beiden Werte kleiner ist. Der Sender verläßt die *Fast Recovery* und wechselt in *Congestion Avoidance*.

5.2.3 *Random Early Detection* (RED)

RED [71] ist ein Verfahren aktiven Puffermanagements, das die mittlere Belegung des Wartespeichers an den Ausgängen eines Routers beobachtet und kontrolliert, einen Bruchteil der Pakete markiert oder verwirft, wenn die mittlere Belegung bestimmte Schwellwerte überschreitet. In Kooperation mit der Überlastregelung der Sender soll RED so längeren Überlastperioden frühzeitig entgegenwirken. Die Wahrscheinlichkeit, mit der Pakete markiert oder verworfen werden, ist eine Funktion der mittleren Belegung $\overline{Q}$, die beim Eintreffen eines Paketes aus der aktuellen Belegung Q als *Exponentially Weighted Moving Average* (EWMA)

$$\overline{Q} := (1-\delta_Q)\overline{Q} + \delta_Q Q \tag{5.1}$$

berechnet wird. Zeiten, in denen der Wartespeicher unbelegt ist, werden gesondert, aber ebenfalls unter Einbeziehung des Faktors $\delta_Q \in (0,1)$ berücksichtigt [71]. Die Wahrscheinlichkeit $P_M(\overline{Q})$, mit der abhängig von der Belegung des Wartespeichers Pakete markiert oder verworfen werden, steigt zwischen den Schwellen S_{min} und S_{max} linear von null bis auf einen konfigurierbaren Wert $P_M'(S_{max})$ an. Um die Reaktion auf eine sich abzeichnende Überlast zu beschleunigen, wird $P_M(\overline{Q})$ ausgehend von

$$P_M'(\overline{Q}) = P_M'(S_{max}) \frac{\overline{Q} - S_{min}}{S_{max} - S_{min}} \tag{5.2}$$

unter Berücksichtigung der Anzahl *n* der seit dem letzten Eingriff des Puffermanagements eingetroffenen Pakete schließlich auf

$$P_M(\overline{Q}) = \frac{P_M'(\overline{Q})}{1 - n P_M'(\overline{Q})} \tag{5.3}$$

korrigiert. Im *Gentle Mode* [76] steigt $P_M(\overline{Q})$ auch bei einer mittleren Belegung $\overline{Q}$ über S_{max} hinaus weiter linear bis auf eins an.

Die Bestimmung geeigneter Werte für die zur Berechnung von $P_M(\overline{Q})$ benötigten Parameter δ_Q, S_{min}, S_{max} und $P_M(S_{max})$ erweist sich als sehr problematisch und hängt insbesondere von der Anzahl der aktiven Verbindungen ab [66]. Aus diesem Grunde werden in [76] die Empfehlungen zur Bestimmung geeigneter Werte für δ_Q, S_{min} und S_{max} verfeinert und RED um einen Algorithmus (und weitere Parameter) erweitert, der $P_M(S_{max})$ periodisch (in Abständen von 0,5 s) dynamisch so anpaßt, daß die mittlere Belegung des Wartespeichers $\overline{Q}$ innerhalb des Zielkorridors

$$(S_{min} + 0{,}4 \cdot (S_{max} - S_{min}), S_{min} + 0{,}6 \cdot (S_{max} - S_{min}))$$

bleibt. Die Anpassung erfolgt in relativ kleinen Schritten: Wenn $\overline{Q}$ größer als die durch den Korridor gegebene Grenze ist, wird $P_M(S_{max})$ um die relativ kleine Konstante $\xi = \min\{0{,}01\ ,\ \frac{1}{4} P_M(S_{max})\}$ auf $\min\{P_M(S_{max}) + \xi\ ,\ 0{,}5\}$ erhöht. Umgekehrt, wenn $\overline{Q}$ kleiner als die untere Grenze ist, wird $P_M(S_{max})$ um den Faktor υ (= 0,9 [76]) reduziert auf $\max\{P_M(S_{max}) \cdot \upsilon\ ,\ 0{,}01\}$. Die Werte von S_{min} und S_{max} sollten sich nach Auffassung der Autoren von [76] an der gewünschten Verzögerungszeit orientieren. Für δ_Q schlagen sie dagegen einen von der Bandbreite des Übertragungsabschnittes *C* (in Pakete pro Sekunde) abhängigen Wert von $\delta_Q = 1 - \exp(-1/C)$ vor, mit dem die Belegung der Warteschlange über eine Zeitspanne in der Größenordnung von 1 s gemittelt wird, was in der Regel ein Vielfaches der Umlaufzeit ist.

5.2.4 Explizite Überlastanzeige (*Explicit Congestion Notification*)

Statt durch kontrolliertes Verwerfen von Paketen die Überlastregelung von TCP anzusprechen, markiert aktives Puffermanagement in Verbindung mit ECN Pakete, wenn Überlast festgestellt wird. RED wird damit sozusagen in REM, *Random Early Marking*, umgewandelt. Sowohl für RED als auch für REM sind unterschiedliche Verfahren zum Feststellen von Überlast und zum

Verwerfen oder Markieren von Paketen denkbar [71, 76, 15, 96]. Diese Algorithmen stellen keinen Verbindungsbezug her und unterscheiden sich in dieser Beziehung grundlegend von denen im Zusammenhang mit der Ratenregelung von ABR [11] eingesetzten.

RFC 2481 [166] definiert für ECN zwei Bits im IP-Paketkopf und zwei weitere im TCP-Paketkopf. Bereits beim Verbindungsaufbau bestimmen die Endpunkte, ob sie beide ECN unterstützen. Ist dies der Fall, wird bei der Übertragung eines Paketes in dessen IP-Kopf immer das Bit ECT (ECN-*Capable Transport*) gesetzt. Dieses Bit signalisiert Routern, daß die Endpunkte auch eine Markierung des Paketes und nicht wie sonst nur dessen Verlust als Zeichen von Überlast werten und entsprechend ihr CW reduzieren werden. Zur Markierung ist das zweite Bit im Paketkopf, das Bit CE (*Congestion Experienced*) vorgesehen. Wenn TCP ein Paket empfängt, in dem das Bit CE gesetzt ist, setzt es in den TCP-Paketköpfen aller nachfolgenden Quittierungen das Bit ECN-Echo. Router im Netz interpretieren den TCP-Paketkopf natürlich nicht. Die Gegenseite erfährt aber auf diese Weise auch dann von der Überlast, wenn die ein oder andere Quittierung mangels Daten ungesichert übertragen wird und unter Umständen verlorengeht. Wenn der Sender daraufhin das CW reduziert, setzt er bei der nächsten Übertragung eines Pakets das Bit CWR (*Congestion Window Reduced*). Wenn dieses Paket beim Empfänger eintrifft, stellt dieser das Setzen des Bits ECN-Echo in abgehenden Quittierungen ein, bis einer der Router entlang des Pfades erneut Überlast feststellt und das Bit CE setzt.

Die Autoren von [72] und [166] haben sich beim Entwurf einer geeigneten Reaktion des Senders von dem Grundgedanken leiten lassen, daß die Reaktion auf die Anzeige von Überlast mit Hilfe von ECN der Reaktion auf Paketverluste ähneln sollte. Aus diesem Grunde setzt der Sender bei einer Überlastanzeige die Schwelle $ssthresh(t)$ wie TCP *Reno* auf die Hälfte der aktuellen Anzahl der gesendeten, aber noch nicht bestätigten Bytes (*Flight Size*) und das CW $cwnd(t)$ auf den Wert $\min\{ssthresh(t), 2\,MSS\}$. Andererseits sollte der Sender auch nicht häufiger als einmal pro Umlaufzeit das CW reduzieren. Denn das tun TCP *Reno* und *New Reno* auch nicht. Um dieser Forderung gerecht zu werden, merkt sich TCP in der vorliegenden Implementierung den Stand der Variable *SND.NXT* zum Zeitpunkt der Reaktion. Dort startet das Segment, das als nächstes übertragen und dann natürlich frühestens eine Umlaufzeit später quittiert werden wird. Wenn in der Zwischenzeit weitere Quittierungen mit gesetztem Bit ECN-Echo eintreffen, wird das CW zwar nicht erneut reduziert, aber auch nicht erhöht, ganz gleichgültig ob sich die Verbindung in der Phase *Slow Start* oder *Congestion Avoidance* befindet. Bis zur Bestätigung des Bytes mit der Sendefolgenummer *SND.NXT* zum Zeitpunkt der Reaktion wird selbst auf Verluste hin das CW nicht noch einmal reduziert, es sei denn es geht ein Paket verloren, das bereits aufgrund eines RTO oder des Empfangs von drei duplizierten Quittierungen wiederholt worden ist.

Die *Fast Recovery* bleibt von den für ECN notwendigen Änderungen unberührt. ECN kann daher in Verbindung mit TCP *Reno*, SACK oder TCP *New Reno* aktiviert werden. In dieser Phase werden Quittierungen mit gesetztem ECN-Echo bis auf weiteres ignoriert, obwohl ein Sender durchaus länger als eine Umlaufzeit in *Fast Recovery* verweilen kann. Denn TCP *Reno*, SACK oder TCP *New Reno* reduzieren das CW in diese Phase auch nicht noch einmal.

5.3 Einbindung von TCP in die Betriebssystemumgebung

Die Entwicklung eines Simulationsmodells für TCP setzt die genaue Kenntnis der Abläufe beim Austausch von Daten und Quittierungen voraus. Die folgende (vereinfachende) Darstellung stützt sich außer auf frei zugängliche Implementierungen vor allem auch auf die Erläuterungen in [151, 195, 181, 183] und bezieht sich überwiegend auf die Betriebssysteme Linux und BSD-Unix.

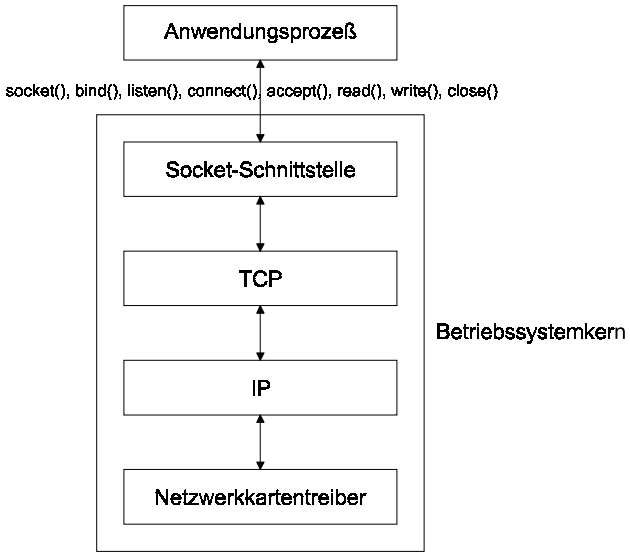

Abb. 5.4: *Sockets stellen die Schnittstelle zur TCP/IP-Implementierung des Betriebssystemkerns bereit. Diese schematische Darstellung zeigt die beteiligten Schichten, wenn die Anwendung einen TCP-Socket geöffnet hat.*

Die heute am weitesten verbreiteten Betriebssysteme erlauben Anwendungen den Zugriff auf die TCP/IP-Implementierung mit einer Reihe von Betriebssystemaufrufen. Sie sind Teil der Socket-Schnittstelle, siehe Abb. 5.4. Der Socket selbst ist eine Datenstruktur, mit deren Hilfe die Kommunikation zwischen Endpunkten gesteuert wird, im Falle von TCP u. a. durch das Speichern von Verbindungszuständen, durch das Bereitstellen von Speicher zum Zwischenspeichern von Daten, durch Variablen für Fehlersicherung und Fluß- sowie Überlastregelung und nicht zuletzt durch Zeiger auf die für TCP maßgeblichen Prozeduren zum Senden und Empfangen von Daten. Die in

den Abschnitten oben erwähnten Algorithmen zur Überlastregelung werden in diesen Prozeduren kodiert.

Bei verbindungsorientierten Protokollen baut eine *Client*-Anwendung eine Verbindung zu einer *Server*-Anwendung auf dem Zielsystem auf. Um solche Verbindungsaufbauwünsche entgegennehmen zu können, hat dieser bereits einen Socket geöffnet (*socket()*), mit einer zuvor vereinbarten Portnummer beim System registriert (*bind()*) und seine Bereitschaft (*listen()*) angezeigt, fortan Verbindungen anzunehmen (*accept()*). Häufig läuft zu diesem Zwecke auf dem Zielsystem ein *Superserver*-Prozeß, der *inetd* [183], der auf Verbindungsaufbauwünsche auf mehreren Portnummern wartet und erst beim Empfang von entsprechenden Anforderungen die zur Erbringung des mit der Portnummer verknüpften Dienstes benötigte Anwendung als Kindprozeß startet. Diese Weitergabe der Anforderung bleibt für die *Client*-Anwendung transparent. Insbesondere adressiert sie die *Server*-Anwendung weiter mit der vereinbarten bzw. standardisierten (*well-known*) Portnummer, so daß zur Identifikation eines Datenstroms die Zieldresse und die mit einem bestimmten Dienst verknüpfte Portnummer nicht genügen, sondern auch die Absenderadresse und die eindeutige Portnummer der *Client*-Seite herangezogen werden müssen. Die Abfolge von Systemaufrufen bis zum Zustandekommen der Verbindung, die Lese- und Schreibeoperationen zum Austausch von Daten und das Beenden der Verbindung zeigt Abb. 5.5.

Die Einzelheiten des Aufbaus einer TCP-Verbindung sind bei Stevens [185] zu finden. Nach erfolgreichem Verbindungsaufbau haben beide Endpunkte der TCP-Verbindung einen Socket eineindeutig zugeordnet, auf den die Anwendungen auf beiden Endpunkten wie auf eine Datei über Bibliotheksfunktionen lesend und schreibend zugreifen können. Die Modellierung und Untersuchungen zur Überlastregelung von TCP der vorliegenden Arbeit beschränken sich auf diese Phase einer Verbindung.

Der Aufruf von Bibliotheksfunktionen zum Lesen oder Schreiben führt zwangsläufig zum Aufruf der Betriebssystemroutinen *read()* oder *write()*, die im Zusammenhang mit *Stream Sockets*, zu denen die TCP-Sockets gehören, einige Besonderheiten aufweisen. Im Prinzip kann der Programmierer bis zu 64 kB Daten zum Schreiben übergeben bzw. einen Lesepuffer dieser Größe angeben. Es ist allerdings nicht garantiert, daß in einem einzigen Aufruf so viele Bytes gelesen oder geschrieben werden können.

Selbst wenn ein Socket wie üblich im blockierenden Modus genutzt wird, kehrt ein Aufruf von *read()* vorzeitig zurück, wenn auch nur ein Teil der erwarteten Daten gelesen werden können. Die Kapazität des Lesepuffers wird häufig also gar nicht ausgeschöpft. Nur wenn noch überhaupt keine Daten verfügbar sind, blockiert der Aufruf die Anwendung, bis das Betriebssystem schließlich

Daten empfängt, die für diesen Socket bestimmt sind. Wenn das *End of File* (EOF) erreicht ist, kehrt *read()* auch bei einem blockierenden Socket sofort zurück. Bei TCP-Sockets ist dies dann der Fall, wenn das letzte übertragene Byte einer bereits abgebauten Verbindung gelesen worden ist.

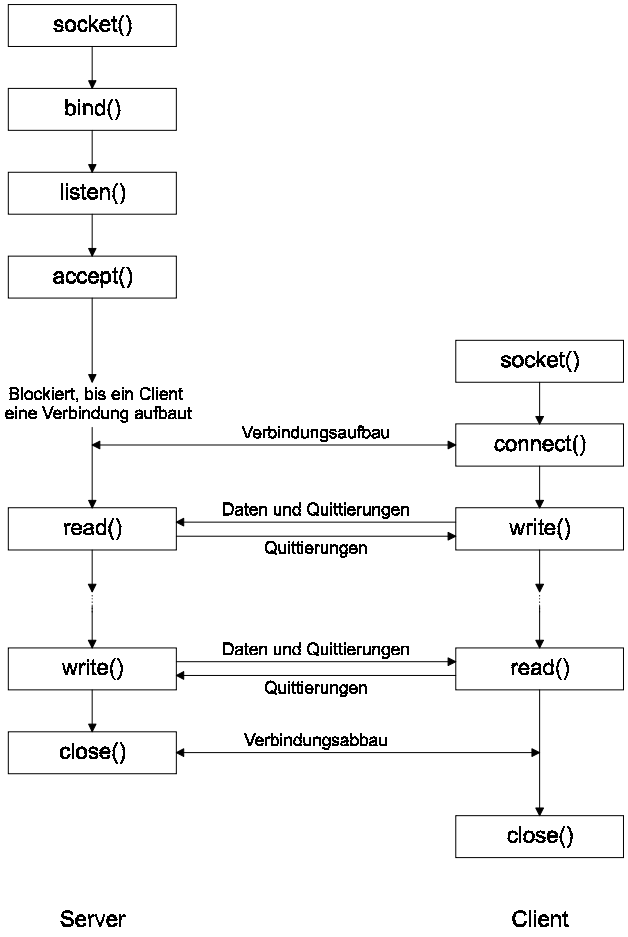

Abb. 5.5: *Eine Folge von Systemaufrufen auf Server- und Client-Seite führt schließlich zum Aufbau einer TCP-Verbindung in den beteiligten Endpunkten, über die Daten und Quittierungen zuverlässig ausgetauscht werden können. Die Datenfluß- und Überlastregelung von TCP begrenzen die Menge der Daten, die zwar bereits gesendet, aber noch nicht quittiert worden sind.*

In ähnlicher Weise kann auch ein Aufruf von *write()* vorzeitig zurückkehren. Der Anwendungsentwickler muß diesem Verhalten Rechnung tragen und prüfen, wie viele Bytes des zum Schreiben übergebenen Datenblocks in den Sendepuffer des Sockets kopiert werden konnten. In der Regel gleichen die Abläufe nach dem Aufruf von *write()* aber den in [151] für SunOS beschriebenen. Diese führen dazu, daß die Anwendung in der Funktion *write()* verharrt, bis tatsächlich alle Daten kopiert sind. Um gegebenenfalls wieder freien Platz im Sendepuffer des Sockets zu schaffen, werden zwischenzeitlich die Sendefunktionen der Netzwerkschichten aufgerufen. Falls zuvor der Sendepuffer nicht alle zum Schreiben übergebene Daten hat aufnehmen können, kopiert der Betriebssystemkern weitere Daten aus dem Sendepuffer der Anwendung in den Sendepuffer des Sockets, sobald ein neuer Teil der über das Netz übertragenen Daten vom Empfänger bestätigt wird

und deswegen aus dem Sendepuffer des Sockets gelöscht werden kann. Erst wenn alle Daten in den Sendepuffer des Sockets kopiert worden sind, ist aus Sicht der Anwendung die Behandlung des Betriebssystemaufrufs abgeschlossen, so daß die Anwendung fortfahren kann.

Auf diese Weise wirkt die Flußregelung auf die von Anwendungen erzeugten Datenströme ein, ohne daß dies der Anwendungsentwickler unbedingt berücksichtigen muß. Alternativ kann ein Socket auch im Modus nicht-blockierend verwendet werden. Statt zu blockieren, kehren Betriebssystemaufrufe auch dann zurück, wenn sie normalerweise blockieren würden. Eine entsprechende Fehlermeldung gibt dem Anwendungsentwickler die Möglichkeit, eine sinnvolle Reaktion zu implementieren. Beschleunigen kann er die Übertragung allerdings auch in diesem Modus nicht.

Der Aufruf einer Betriebssystemroutine geht mit einem Softwareinterrupt einher. Dieser Interrupt wird vom Betriebsystemkern behandelt. Damit wird die Ausführung des Programmcodes der Anwendung unterbrochen und der mit dem Betriebssystemaufruf verknüpfte Programmcode des Betriebssystems gestartet. Unter Umständen kann die Verarbeitung von Betriebssystemaufrufen nicht zu Ende geführt werden, weil zunächst noch auf bestimmte Ereignisse gewartet werden muß. Dazu zählen bei TCP beispielsweise der Empfang von Daten oder Quittierungen (bei blockierendem *read()* oder *write()*). In diesen Fällen wird der Prozeß in eine Warteschlange eingetragen. Er scheidet dann als Kandidat für die Zuteilung des Prozessors aus, bis das zur Weiterbearbeitung erforderliche Ereignis eingetroffen ist.

Darüber hinaus können Systemaufrufe jederzeit von Interrupts unterbrochen werden. Um die Zeit für die Behandlung von Interrupts zu verkürzen und den Weg für neue Interrupts frei zu machen, werden nur die dringendsten Aufgaben als Teil der Interruptbehandlung erledigt. Aber auch die weniger zeitkritischen Aufgaben werden im Zuge einer Nachbearbeitung in einem gegenüber Prozessen im Nutzermodus und Systemaufrufen (abgesehen von Betriebssystemcode, der durch spezielle Kommandos geschützt wird und dann nur noch durch Interrupts unterbrochen werden kann [181]) noch privilegierten Modus ausgeführt (*Bottom-Half* Mechanismus in Linux [181, 195]). Auch der Großteil der im Anschluß an den Empfang von Paketen (Daten oder Quittierungen) in den TCP/IP-Protokollschichten anfallenden Aufgaben ist auf diese Weise aus der eigentlichen Interruptbehandlung ausgegliedert [195].

Wenn die Anwendung auf der Empfangsseite nach dem Aufruf der Betriebssystemroutine *read()* auf den Empfang von Daten wartet, folgt auf die Behandlung des von der Netzwerkkarte ausgelösten Interrupts die Nachbearbeitung über den *Bottom-Half* Mechanismus (bei Linux). Diese Nachbearbeitung schließt insbesondere auch die TCP-Protokolloperationen auf der Empfangsseite ein, die unter anderem zum Senden von Quittierungen führen. Nur wenn im Empfangspuffer des TCP-

Sockets Daten gespeichert sind, die im Strom unmittelbar auf die bereits der Anwendung übergebenen folgen, wird außerdem der Prozeß aus der Prozeßwarteschlange entfernt und der Systemaufruf *read()* der Anwendung abschließend bearbeitet, sobald dem jetzt wieder arbeitsbereiten Prozeß ein Prozessor zugeteilt wird. Natürlich sind jederzeit weitere Unterbrechungen durch neue Interrupts und die Bearbeitung von *Bottom-Half Flags* möglich.

Wie bereits angesprochen, wird beim Senden mit *write()* der von der Anwendung zum Schreiben übergebene Datenblock in den Sendepuffer des Sockets kopiert. Obwohl der Sendepuffer des Sockets als Liste von Datenblöcken implementiert ist, kann man sich die Daten im Sendepuffer durchaus als Bytestrom vorstellen. Denn die begrenzte Kapazität des Sendepuffers, die Fehlersicherung sowie Fluß- und Überlastregelung und das Bestreben durch Ausschöpfen der *Maximum Segment Size* (MSS) von TCP den Protokollaufwand zu minimieren, wirkt sich letztlich so aus, daß TCP aus dem Bytestrom, der auf die Liste der Datenblöcke verteilt zwischengespeichert wird, einen Ausschnitt, ein sogenanntes Segment, in ein Paket einpackt, das der IP-Schicht als *Service Data Unit* (SDU) übergeben wird. Da einerseits die Kapazität des Sendepuffers des Sockets begrenzt ist und anderseits die Quittierungen des Empfängers das Senden von TCP-Segmenten steuern, lassen sich die Abläufe beim Senden anders als die beim Empfang von Paketen nicht als synchrone Abfolge von Funktionsaufrufen darstellen.

5.4 Simulationsmodell

Die im Rahmen der vorliegenden Arbeit und der Diplomarbeit von Grévent [86] entstandene Simulationsumgebung modelliert TCP-Verbindungen zwischen zwei Anwendungen, denen eindeutig und für die Gesamtdauer die Rolle des Senders und des Empfänger zugewiesen wird. Nutzdaten werden nur in einer Richtung übertragen (unidirektional). Bidirektional genutzte Verbindungen hätten einen Einfluß auf den Strom der Empfangsbestätigungen. Wenn nämlich Nutzdaten auch in Gegenrichtung gesendet werden sollen, kann (und muß) eine aufgrund des *Delayed Acknowledgement* noch nicht übertragene Empfangsbestätigung im Datenpaket in Gegenrichtung eingetragen werden [185]. Insbesondere in Verbindungen mit sehr symmetrischem Datenaufkommen werden dann deutlich mehr Empfangsbestätigungen in beiden Richtungen ausgetauscht (und zwar anders als eigenständige Quittierungen dank der Fehlersicherung für die Datenpakete zuverlässig) und der Anstieg des *Congestion Window* auf beiden Seiten beschleunigt. Solche bidirektionalen Szenarien sind jedoch in der Praxis (derzeit) eher selten zu beobachten und spielen aus diesem Grunde auch in den nachfolgenden Untersuchungen keine Rolle.

Praktische Messungen, u. a. auch unter Beteiligung des Autors in lokalen ATM-Netzen durchgeführte [108], belegen, daß schon vergleichsweise unbedeutende Abweichungen der Implementierung von der Spezifikation von TCP das Verhalten signifikant beeinflussen können. Eine Arbeitsgruppe der IETF hat aus diesem Grunde vor einiger Zeit sogar einen RFC herausgegeben, in dem häufige Fehler bei der Implementierung von TCP dokumentiert werden [161]. Es steht daher außer Frage, daß vor allem die Interaktion zwischen Anwendung und Socket beim Lesen und Schreiben, der endliche Wartespeicher und die Bytestromorientierung des Sockets sehr exakt modelliert und die Algorithmen zur Fluß- und Überlastregelung auch im Simulationsmodell ihrer Spezifikation entsprechend implementiert werden müssen, wenn man zu realistischen Ergebnissen kommen möchte.

Neben der in RFC 793 [165] spezifizierten Basisfunktionalität zur vor Datenverlusten geschützten, flußgeregelten Übertragung eines Datenstroms sind im Simulationsmodell die in RFC 1122 [31] ausführlich diskutierten Mechanismen zur *Silly Window Avoidance* auf der Sende-und Empfangsseite sowie *Delayed Acknowledgement* aus RFC 1122 [31], der Algorithmus von Karn [110] zur Verbesserung der Messung der Paketumlaufzeit in der oben besprochenen Weise, die *Timestamp* und *Window Scale Option* von RFC 1323 [105] und die als TCP *Reno* bekannte und in RFC 2581 [2] standardisierte Überlastregelung implementiert. Von den jüngsten Vorschlägen zur Modifikation von TCP sind *New Reno* aus RFC 2582 [73], die in RFC 2018 spezifizierte *Selective Acknowledgement* (SACK) Option [143] mit einer sich weitgehend an [61] anlehnenden *Score Board* Implementierung, *Explicit Congestion Notification* (ECN) gemäß [166] und der in RFC 3042 [3] beschriebene *Limited Transmit* berücksichtigt worden.

Der Autor hat auf eine Implementierung der für TCP spezifizierten Bitfehlererkennung verzichtet, weil in Festnetzen Bitfehler weit seltener auftreten als Paketverluste durch Überlast. Solche Verluste führen zu den Lücken im Datenstrom im Empfangspuffer, die in Abb. 5.7 dargestellt sind. Die Empfangsseite sendet dann abhängig von der eingestellten TCP-Variante duplizierte oder selektive Quittierungen, um die Sendeseite über diese Lücken zu informieren und so die Voraussetzung für die Wiederholung der fehlenden Segmente zu schaffen, möglichst unter Umgehung von zeitraubenden RTOs. Wie eine mit dem Befehl *read()* auf einen TCP-Socket lesend zugreifende Anwendung kann die im Simulationsmodell nach Abb. 5.7 aus dem Empfangspuffer lesende Bedieneinheit nur Datenblöcke entnehmen, die sich ohne Lücke an die bereits verarbeiteten Segmente des Datenstroms anschließen. Außerdem erkennt und verwirft der Empfangspuffer mehrfach übertragene Bytes des Datenstroms.

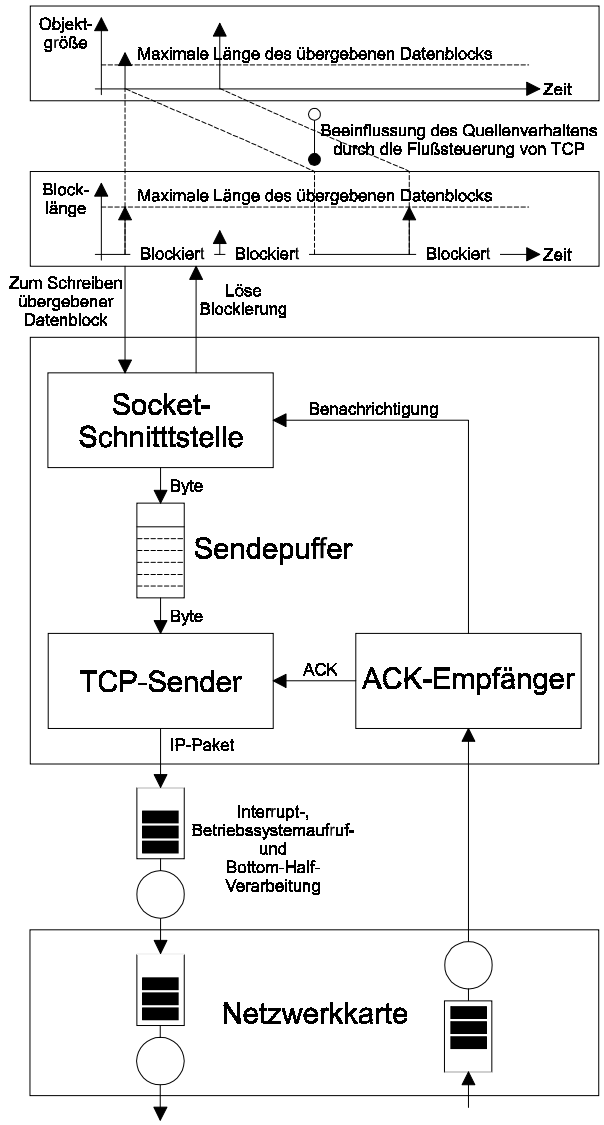

Abb. 5.6: *Das Modell der Sendeseite berücksichtigt insbesondere die Interaktion der Anwendung mit der Socket-Schnittstelle beim Schreiben und die Bytestromorientierung von TCP. Die Algorithmen zur Fluß- und Überlastregelung werden exakt implementiert.*

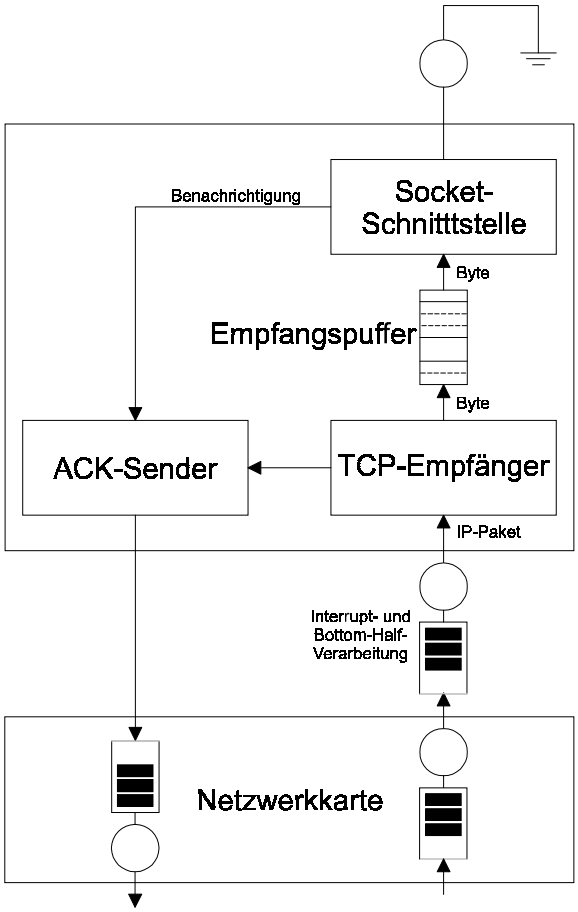

Abb. 5.7: *Das Modell der Empfangsseite ist bei der Betrachtung von Datenströmen in nur einer Richtung einfacher, v. a. weil die Interaktion mit einer Quelle entfällt. Der ACK-Sender steuert mit Quittierung den TCP-Sender aus Abb. 5.6, wie dies die TCP-Algorithmen vorsehen.*

Die Einzelheiten dieser Mechanismen zur Fehlersicherung, der Algorithmen zur Fluß- und Überlastregelung sowie weitere Besonderheiten bei deren Implementierung sind bereits in früheren Abschnitten erläutert worden. Daher soll im folgenden insbesondere auf die Möglichkeiten der Modellierung von Anwendungen und ihre Interaktion mit TCP eingegangen werden.

Die Simulationsumgebung unterstützt Quellen, die eine Folge von Objekten in beliebig verteilter Größe an einen TCP-Socket zur Übertragung übergeben, jeweils die vollständige Übernahme eines Objektes durch den Socket abwarten und anschließend in einem beliebig verteilten Zeitabstand ein

neues erzeugen und übergeben. Jedes Objekt kann mit einer EOF-Markierung (*End of File*) versehen werden, die die TCP-Sockets auf der Sende- und Empfangsseite dazu veranlaßt, am Ende der Übertragung des Objektes alle Verbindungsvariablen zurückzusetzen, so als ob die Verbindung zwischenzeitlich abgebaut würde. Auf diese Weise kann ein Socket-Paar für mehrere aufeinanderfolgende TCP-Verbindungen verwendet werden. Der zum Auf- und Abbau von TCP-Verbindungen über ein Netz eigentlich notwendige Austausch von Meldungen ist jedoch nicht implementiert. Zur Modellierung ungeduldiger Benutzer können Objekte auch noch mit einer INT- (Interrupt) Markierung gekennzeichnet werden. Dann wird die Übertragung des Objektes und die TCP-Verbindung abgebrochen, wenn ein bestimmter Durchsatz nicht erreicht wird.

Die Socket-Schnittstelle in Abb. 5.6 ist so gestaltet, daß die Rückwirkung der Fluß- und Überlastregelung von TCP bei der Spezifikation von Quellenmodellen unberücksichtigt bleiben kann. Zunächst wird – anders als bei den Betriebssystemaufrufen *read()* und *write()* – die Länge des zum Schreiben übergebenen Datenblocks nicht innerhalb der Anwendung, die der Quelle im Simulationsmodell entspricht, sondern innerhalb der Socket-Schnittstelle festgelegt. Ein Objekt, das die Quelle der Socket-Schnittstelle übergibt, wird erst auf dieser Ebene in Datenblöcke zerlegt. Diese werden einzeln und jeder Datenblock unter Umständen auch noch in mehreren Schritten in den Sendepuffer des Sockets kopiert. Wenn nicht frühere Daten zur Übertragung anstehen, kann der TCP-Sender in Abb. 5.6 bereits mit der Datenübertragung beginnen, wenn das Objekt oder sogar die einzelnen Datenblöcke noch gar nicht vollständig in den Sendepuffer des Sockets übernommen worden sind. Die Quelle bleibt aber blockiert, bis das Objekt vollständig in den Sendepuffer kopiert worden ist. Wie lange das dauert, hängt vom Füllstand des Sendepuffers bei der Übergabe des Objektes, der Länge des Objektes, der Größe der zum Schreiben übergebenen Datenblöcke, der für die Kopie der Daten angesetzten Verarbeitungszeiten und der Geschwindigkeit der Datenübertragung über das Netz ab.

Die Verarbeitungszeit für das Kopieren von Daten aus einer Quelle in den Sendepuffer des Sockets wird in der vorliegenden Version der Simulationsumgebung durch Parametrisierung der in [151] sehr detailliert dokumentierten Prozedur bei SunOS festgelegt. Messungen des Durchsatzes in Abhängigkeit von der Größe der von der Anwendung dem Socket zum Schreiben übergebenen Datenblöcken bei immer ausreichender Übertragungskapazität sind die ideale Grundlage dieser Parametrisierung, vgl. S. 86 in [151]. Abb. 5.6 zeigt, daß darüber hinaus zwischen dem TCP-Sender und der Netzwerkkarte ein Bediensystem zur Modellierung von Verarbeitungszeiten eingefügt ist. Damit soll die Zeit für die Verarbeitung von Paketen im Zuge von Betriebssystemaufrufen bei Übertragungen, die direkt durch die Aktivität der Quelle eingeleitet werden, und im Zuge der Interrupt- und *Bottom-Half* Verarbeitung nach dem Empfang von Quittierungen ohne weitere Differenzierung

berücksichtigt werden. Zur Parametrisierung dieses sehr einfachen Modells sollte eine Testanwendung verwendet werden, die so schnell, wie es der Socket erlaubt, Daten zu einer Senke auf einem anderen Rechner über ein Netz überträgt (*Greedy/Persistent Source*). Die Bandbreite des Netzes sollte unbedingt so dimensioniert werden, daß sie in diesem Szenario der Flaschenhals ist und der Sendepuffer des Sockets deshalb nie leer wird. Mit einem Protokollmeßgerät in der Nähe der Quelle kann man dann die Zeit messen, die zwischen dem Empfang einer Quittierung bis zum Senden eines TCP-Segmentes vergeht und diese näherungsweise als die gesuchte Verarbeitungszeit übernehmen.

Die Simulationsumgebung unterstützt für jede Quelle unabhängig durchführbare Messungen einfacher Leistungsmaße wie Durchsatz, Paketverlustwahrscheinlichkeit usw. Darüber hinaus können sich Quellen bei zentralen Einheiten registrieren und danach ein- oder ausbuchen, um z. B. beim Verbindungsanfang oder -ende bestimmte Meßwerte bereitzustellen oder bestimmte Messungen an allen registrierten Quellen auszulösen. Auf diese Weise können Messungen in Szenarien mit variablen und nicht immer aktiven Quellen koordiniert werden, was für manche Leistungsmaße unumgänglich ist. Die Integration in die Simulationsbibliothek des IND [53] eröffnet weitere Möglichkeiten zur statistischen Auswertung von Simulationen zur Laufzeit. Diese und die exakte Modellierung der Interaktion zwischen Anwendung und Socket beim Lesen und Schreiben sind die wesentlichen Punkte, in denen sich die vorliegende Simulationsumgebung beispielsweise vom *Network Simulator* [62] unterscheidet.

5.5 Leistungsmaße

Anwendungen, die TCP, also ein flußsteuerndes Transportprotokoll zur Datenübertragung auswählen, nehmen einen netzlastabhängigen Durchsatz in Kauf. Wenn ihr Datenaufkommen über einen längeren Zeitraum über dem vom Netz realisierten Durchsatz liegt und sich daher der Wartespeicher des Sockets füllt, können sie sogar blockiert, d. h. vorübergehend gestoppt werden. Nach der Klassifikation in [36] (vgl. Kapitel 7) gehören solche Anwendungen sicherlich entweder zu der Klasse der elastischen Anwendungen oder doch wenigstens zu der Klasse der adaptiven. Sie benötigen also allenfalls einen relativ geringen Durchsatz, um zu funktionieren, profitieren aber von mehr Bandbreite bis weit über diesen Minimaldurchsatz hinaus. Ihre Dienstgüteanforderungen lassen sich daher weit schwieriger quantifizieren als die der im Zusammenhang mit *Integrated Services* betrachteten Anwendungen, die Datenströme unabhängig von der Lastsituation im Netz nach Möglichkeit über vorab aufgebaute virtuelle Verbindungen übertragen (rigide Anwendungen).

Algorithmen zur Überlastregelung sollten nach Möglichkeit den Nutzdurchsatz des Netzes steigern. Wenn dennoch die Nachfrage nach Übertragungskapazität nicht befriedigt werden kann und die Quellen im Wettbewerb um die knappen Ressourcen des Netzes stehen, sollte die Zuteilung der Ressourcen doch wenigstens fair erfolgen. Wie in [81] ausgeführt wird, ist im Zusammenhang mit Überlastregelung von Verkehrsströmen die Definition einer Max-Min-Fairneß sehr verbreitet. Ein Netz ist dann max-min-fair, wenn ein Datenstrom mit mindestens der gleichen Rate bedient wird wie andere Datenströme, die auf ihrem Weg denselben Übertragungsabschnitt als den Engpaß des Netzes wahrnehmen. Kelly [115] hingegen favorisiert den Gedanken einer proportionalen Fairneß. Die Zuordnung eines Vektors $\boldsymbol{x}$ von Ressourcen zu einem Vektor von Verbindungen ist dann gewichtet proportional fair, wenn $\boldsymbol{x}$ machbar ist und wenn für jeden anderen machbaren Vektor $\boldsymbol{x}+\Delta\boldsymbol{x}$ gilt:

$$\sum_i g_i \frac{\Delta x_i}{x_i} \leq 0 \tag{5.4}$$

Machbar heißt, daß durch die Zuordnung von $\boldsymbol{x}$ in keinem Übertragungsabschnitt die vorhandenen Ressourcen überschritten werden.

Da in der vorliegenden Arbeit eine differenzierte Behandlung von elastischen oder adaptiven Quellen nicht in Betracht gezogen wird, ergeben sich keine Ansatzpunkte, Verbindungen unterschiedliche Gewichte g_i zuzuordnen.

Kelly setzt in [115] zur Optimierung der Zuteilung der Übertragungskapazität als Zielfunktion die Gesamtzufriedenheit[4] der Nutzer an:

$$\text{maximiere} \sum_{r\in R} U_r(x_r) \quad \text{unter den Randbedingungen } \boldsymbol{A}\,\boldsymbol{x} \leq \boldsymbol{c},\ \boldsymbol{x} \geq 0 \tag{5.5}$$

$U_r(x_r)$ ist die Zufriedenheit eines Nutzers, dessen Verbindung eineindeutig die Route r zugeordnet ist und einen Durchsatz von x_r erzielt. Die Matrix $\boldsymbol{A}$ gibt an, ob ein Übertragungsabschnitt j Teil einer Route $r \in R$, der Menge aller Routen, ist, $A_{jr}=1$, oder nicht, $A_{jr}=0$. $\boldsymbol{c}$ ist der Vektor der Kapazitäten aller Übertragungsabschnitte. Wenn alle $U_r(x_r)$, $r \in R$, konkav und differenzierbar für alle $x_r \geq 0$ sind, ist (5.5) eindeutig lösbar. Vorausgesetzt, es besteht ein logarithmischer Zusammenhang zwischen der Rate, mit der ein Verkehrsstrom bedient wird, und der Zufriedenheit des Nutzers, führt die Optimierungsaufgabe zu einem proportional fairen Lösungsvektor $\boldsymbol{x}$.

4 Der Autor ersetzt den englischen Begriff *Utility* im Deutschen durch *Zufriedenheit*, obwohl oder gerade weil damit eine andere Interpretation des Systemmodells von Kelly suggeriert wird als bei einer wörtlichen Übersetzung.

Es ist recht schwierig, zu verwertbaren Aussagen zur Fairneß zu kommen, wenn Quellen mit nicht identischen statistischen Eigenschaften an verschiedenen Punkten des Netzes Verkehr in das Netz einspeisen. Kellys Netzmodell und eine Zielfunktion der Form (5.5) schafft eine einzige, sogar für den Nutzer nachvollziehbare Zielgröße, die Gesamtzufriedenheit, in der Durchsatz und Fairneß aufgehen.

Aus diesem Grunde wird in den im Rahmen der vorliegenden Arbeit durchgeführten Untersuchungen die mittlere Gesamtzufriedenheit pro Zeit gemessen. Dabei werden die Beiträge einzelner Übertragungen mit einem Gewicht proportional zur übertragenen Datenmenge berücksichtigt. Die so gemessene mittlere Zufriedenheit ist die Zielfunktion des Netzes, die es zu maximieren gilt.

Dies kann natürlich nicht darüber hinwegtäuschen, daß unterschiedliche Definitionen von Zufriedenheit ebenso wie die unterschiedlichen Definitionen von Fairneß, die Auswahl der Quellen, ihre Plazierung im Netz und andere Parameter abhängig von den verwendeten Algorithmen zur Überlastregelung ganz unterschiedliche Auswirkungen auf das Endergebnis haben können.

So führt die Überlastregelung der diversen Varianten von TCP in Verbindung mit den verschiedenen Verfahren des Puffermanagements zwar auf asymptotisch stabile Ruhelagen $\boldsymbol{x}$, diese Ruhelagen sind jedoch Maxima unterschiedlicher Zielfunktionen der folgenden Form [117, 131]:

$$U_r(x_r) - \sum_{j \in r} C_j\Big(\sum_{s:\, j \in s} x_s\Big) \tag{5.6}$$

$C_j(\sum x_s)$ sind die Kosten für die Nutzung einer Ressource j, wenn die Last $\sum x_s$ ist. Anders als die Optimierungsaufgabe (5.5) wird (5.6) von Kelly nicht durch Randbedingungen beschränkt. Die Lösung von (5.5) kann aber mit Hilfe der Lagrange-Funktion [93]

$$\sum_{r \in R} U_r(x_r) - \boldsymbol{P}^{\boldsymbol{T}}(\boldsymbol{A}\,\boldsymbol{x} + \boldsymbol{z} - \boldsymbol{c}) \tag{5.7}$$

gefunden werden. In (5.7) ist der Vektor $\boldsymbol{P}^{\boldsymbol{T}}$ der Lagrange-Multiplikator und der Vektor $\boldsymbol{z}$ eine Schlupfvariable. An einer Extremstelle müssen die partiellen Ableitungen nach allen Komponenten von $\boldsymbol{x}$, $\boldsymbol{P}^{\boldsymbol{T}}$ und $\boldsymbol{z}$ null sein, also insbesondere auch

$$\frac{\partial}{\partial x_r} U_r(x_r) - \sum_{j \in r} p_j = 0 \tag{5.8}$$

Insofern verallgemeinert $C_j(\sum x_s)$ in (5.6) die lineare Kostenfunktion $P_j x_r$ in (5.7).

Dem Ausdruck (5.6) liegt die Annahme zugrunde, daß sich die Kosten für die Nutzung der Ressourcen j auf der Route r zu den Gesamtkosten addieren. Kellys [115] und mehr noch Lows [131] Arbeit machen deutlich, daß mit Kosten weniger eine monetäre Größe, sondern andere

Größen wie Verlustwahrscheinlichkeit, Wahrscheinlichkeit einer (die Priorität eines Paketes reduzierenden) Markierung oder andere zur Rückkopplung geeignete Größen gemeint sind[5]. Insofern sind die Kosten manchmal allenfalls näherungsweise additiv, denn ein einmal verworfenes oder (mit einem Bit) markiertes Paket kann nicht noch in nachfolgenden Knoten verworfen oder markiert werden. Mit dieser Näherung können aber die speziellen Eigenschaften konvexer Optimierungsaufgaben genutzt werden. Dies gilt auch für die von Kelly als Schattenpreis $P_j(y)$ bezeichnete Ableitung

$$\frac{\mathrm{d}}{\mathrm{d}\,y} C_j(y) = P_j(y) \tag{5.9}$$

vorausgesetzt diese existiert überhaupt. Als Last sind natürlich jeweils die Beiträge aller den Übertragungsabschnitt nutzenden Quellen *r* zu berücksichtigen.

Die Zielfunktion (5.6) wird dann maximiert, wenn

$$x_r = \left(\frac{\mathrm{d}}{\mathrm{d}\,x_r} U_r \right)^{-1} \left(\sum_{j \in r} P_j \left(\sum_{s:\, j \in s} x_s \right) \right) \tag{5.10}$$

x_r aus Gleichung (5.10) sollte sich daher als asymptotisch stabile Ruhelage der Differentialgleichung der Überlastregelung ergeben, und $U_r(x_r)$ entsprechend konstruiert werden.

Eine Ruhelage $\boldsymbol{x}_G$ heißt asymptotisch stabil im Sinne von Ljapunov, wenn für alle genügend nahe der Ruhelage gewählten Anfangszustände $\boldsymbol{x}(0)$ die Bewegung $\boldsymbol{x}(t)$ mit wachsender Zeit gegen $\boldsymbol{x}_G$ strebt: $\boldsymbol{x}_G = \lim_{t \to \infty} \boldsymbol{x}(t)$ [107].

Läßt sich für ein System mit der Systemgleichung

$$\frac{\mathrm{d}}{\mathrm{d}\,t} \boldsymbol{x}(t) = f(\boldsymbol{x}(t), t) \tag{5.11}$$

eine positiv definite Funktion $V(\boldsymbol{x}(t), t)$ angeben, deren zeitliche Ableitung in Verbindung mit der Systemgleichung (5.11), $\frac{\mathrm{d}}{\mathrm{d}\,t} V(\boldsymbol{x}(t), t)$, negativ definit ist, so ist die betrachtete Ruhelage asymptotisch stabil. Die Funktion $V(\boldsymbol{x}(t), t)$ wird dann auch Ljapunov-Funktion genannt. Eine beliebige Funktion $V(\boldsymbol{x})$ heißt positiv (negativ) definit, wenn sie für alle Werte $\boldsymbol{x}$ mit Ausnahme von $\boldsymbol{x} = \boldsymbol{0}$ größer (kleiner) null ist und für $\boldsymbol{x} = \boldsymbol{0}$ gleich null ist [107].

5 Die Kostenfunktion richtet sich ausschließlich nach den Erfordernissen der Überlastregelung. Die Vergebührung kann sich ebenfalls daran orientieren. Gibbens und Kelly weisen in [82, 83] nach, daß die Vergebührung von Überlastanzeigen ein Anreiz für die wirtschaftliche Nutzung des Netzes und ein wirksames Mittel zur Differenzierung der Nutzer sein kann. Dann hängt die Wirksamkeit der Überlastregelung nicht mehr so sehr von der Kooperationsbereitschaft der Nutzer ab.

Die Bewegung des Systems ist dann so, daß die Funktion $V(\boldsymbol{x}(t),t)$ immer kleiner wird, bis sie schließlich ihr Minimum 0 bei $\boldsymbol{x}=\mathbf{0}$ erreicht und nicht mehr verläßt. Für die im folgenden als Ljapunov-Funktionen von verschiedenen Verfahren zur Überlastregelung angegebenen Funktionen können nach Vorzeichenumkehr, Verschiebung des Ursprungs des Koordinatensystems in die Ruhelage und Addition einer Konstante die für Ljapunov-Funktionen geforderten Eigenschaften überprüft werden.

Als Beispiel einer Überlastregelung wird in [115] die Differentialgleichung

$$\frac{\mathrm{d}}{\mathrm{d}t}x_r(t)=K_r(w_r(t)-x_r(t)\sum_{j\in r}P_j(\sum_{s:\,j\in s}x_s(t))) \tag{5.12}$$

angegeben, in der das Netz einem permanenten Anheben $K_r w_r(t)$ der Rate durch eine Rückkopplung der Form $K_r x_r(t)\sum P_j(\sum x_s(t))$ entgegenwirkt. Hält man $w_r=w_r(t)$ konstant und setzt in (5.12) $\frac{\mathrm{d}}{\mathrm{d}t}x_r(t)$ gleich null, so sieht man, daß

$$x_r=\frac{w_r}{\sum_{j\in r}P_j(\sum_{s:\,j\in s}x_s)} \tag{5.13}$$

eine asymptotisch stabile Ruhelage der Differentialgleichung sein könnte. Aus dem Ansatz

$$\frac{\partial}{\partial x_r}U(\boldsymbol{x})=\frac{w_r}{x_r}-\sum_{j\in r}P_j(\sum_{s:\,j\in s}x_s) \tag{5.14}$$

kann man die Funktion

$$U(\boldsymbol{x})=\sum_{r\in R}w_r\ln x_r-\sum_{j\in r}C_j(\sum_{s:\,j\in s}x_s) \tag{5.15}$$

gewinnen, die nicht nur ihr Maximum an der vermuteten asymptotisch stabilen Ruhelage einnimmt, sondern auch die Eigenschaften einer Ljapunov-Funktion bezüglich dieser Ruhelage besitzt [115]. x_r aus Gleichung (5.13) ist also tatsächlich eine asymptotisch stabile Ruhelage der durch Differentialgleichung (5.12) beschriebenen Überlastregelung. Diese Ruhelage optimiert gleichzeitig die Zielfunktion des Gesamtsystems (vgl. (5.6) oben)

$$\sum_{r\in R}U_r(x_r)-\sum_{j\in r}C_j(\sum_{s:\,j\in s}x_s) \tag{5.16}$$

wenn $U_r(x_r)=w_r\ln x_r$. Diese Überlastregelung optimiert also ein Netz, wenn zwischen der Rate, mit der ein Verkehrsstrom bedient wird, und der Zufriedenheit des Nutzers ein logarithmischer Zusammenhang besteht. Die Differentialgleichung (5.12) ist ein Beispiel einer Überlastregelung, bei der die negative Rückkopplung $\sum P_j(\sum x_s(t))$ nicht selbst von der Rate der Quelle abhängt, sondern

nur von dem für alle gleichen Schattenpreis. Da die Überlastregelung von TCP aber auf Paketverluste reagiert, diese Paketverluste jedoch nicht unabhängig von der Rate der Verbindung sind, ist (5.12) ganz gewiß kein gutes Modell für TCP [98].

Diese Vorgehensweise läßt sich aber völlig analog auf die Überlastregelung von TCP anwenden [115]. Unter der vereinfachenden Annahme, daß regelmäßig jedes Paket quittiert wird und Paketverluste voneinander unabhängig mit der Wahrscheinlichkeit P_{Loss} auftreten, vergrößert TCP *Reno* in *Congestion Avoidance* beim Eintreffen einer Quittierung das *Congestion Window* $cwnd(t)$ um $\frac{1}{cwnd(t)}$, halbiert es aber, wenn die Quittierung ausbleibt (und statt dessen anschließend ein *Duplicate ACK* empfangen wird). Im Rhythmus der Quittierungen verändert sich das *Congestion Window* folglich um

$$\Delta cwnd(t) = (1 - P_{Loss}) \frac{1}{cwnd(t)} - P_{Loss} \frac{cwnd(t)}{2} \tag{5.17}$$

Wenn jedes Paket bestätigt wird (kein *Delayed Acknowledgement*) und das *Congestion Window* $cwnd(t)$ als Vielfaches der hier als konstant angenommenen Paketgröße angegeben wird, treffen die Quittierungen näherungsweise mit der Rate $\frac{cwnd(t)}{T_r}$ (in Pakete pro Sekunde) ein. Zur Vereinfachung wird die sogenannte *Round Trip Time* T_r, d. h. die Zeit, die vom Senden eines Paketes bis zu seiner Quittierung vergeht, als konstant angenommen. Zur Berechnung des zeitlichen Verlaufs des *Congestion Window* muß man also die rechte Seite von (5.17) mit $\frac{cwnd(t)}{T_r}$ multiplizieren. Wenn man dann noch berücksichtigt, daß die fensterbasierte Flußregelung bei der Fenstergröße $cwnd(t)$ näherungsweise die Rate $x_r(t) = \frac{cwnd(t)}{T_r}$ zuläßt, so ergibt sich die Änderung der Rate pro Zeit $\frac{\Delta x_r}{\Delta t}$. Ersetzt man nun die Verlustwahrscheinlichkeit Ende zu Ende, P_{Loss}, durch $\sum_{j \in r} P_j(\sum_{s:\, j \in s} x_s)$, so als ob sich die Verlustwahrscheinlichkeiten der Übertragungsabschnitte addierten, so erhält man schließlich die Differentialgleichung [115]

$$\frac{\mathrm{d}}{\mathrm{d}t} x_r(t) = \frac{1}{T_r^2} - \left(\frac{1}{T_r^2} + \frac{x_r(t)^2}{2} \right) \sum_{j \in r} P_j\Big(\sum_{s:\, j \in s} x_s \Big) \tag{5.18}$$

Kelly verfährt mit (5.18) wie zuvor mit (5.12) und findet auf diese Weise heraus, daß TCP *Reno* in *Congestion Avoidance* unter den vereinfachenden Annahmen ein Netz optimiert, in dem die Zufriedenheit des Nutzers in Abhängigkeit von der Rate x_r, mit der die Pakete seiner Verbindung *r* übertragen werden, durch die Gleichung

$$U_r(x_r) = \frac{\sqrt{2}}{T_r} \arctan\left(\frac{x_r T_r}{\sqrt{2}} \right) \tag{5.19}$$

beschrieben wird. Die Überlastregelung von TCP ist also dann gut, wenn die Zufriedenheit $U_r(x_r)$ des Nutzers um so langsamer mit wachsendem Durchsatz zunimmt, je größer die Umlaufzeit T_r ist. Anders ausgedrückt: TCP begünstigt Verbindungen mit kurzer Umlaufzeit T_r. Die Ruhelage

$$x_r = \frac{1}{T_r}\sqrt{2\,\frac{1-\sum_{j\in r} P_j\left(\sum_{s:\, j\in s} x_s\right)}{\sum_{j\in r} P_j\left(\sum_{s:\, j\in s} x_s\right)}} \tag{5.20}$$

kann unmittelbar aus (5.18) berechnet ($\frac{\mathrm{d}}{\mathrm{d}t}x_r(t)=0$) werden. Eine Reihe von Autoren [144, 156] kommt mit anderen Methoden ebenfalls auf ähnliche Ausdrücke der Form

$$x_r \sim \frac{1}{T_r}\sqrt{\frac{1}{P_{Loss}}} \tag{5.21}$$

Bei Berücksichtigung weiterer Einzelheiten der Überlastregelung werden die Zusammenhänge deutlich komplizierter [156]. Die Proportionalitätskonstante in (5.21) hängt von der Berechnungsmethode und Details der Überlastregelung ab [144]. Für Verbindungen, die *Delayed Acknowledgement* aktiviert haben, muß man beispielsweise unter der Vereinfachung, daß jede Quittierung Informationen zu zwei Segmenten zurückliefert, statt (5.17)

$$\Delta\, cwnd(t) = (1-P_{Loss})^2 \frac{1}{cwnd(t)} - (P_{Loss}{}^2 + 2\,P_{Loss}(1-P_{Loss}))\frac{cwnd(t)}{2} \tag{5.22}$$

schreiben. Die Multiplikation mit $\frac{cwnd(t)}{2T_r}$, der Rate der Quittierungen, ergibt den zeitlichen Verlauf des *Congestion Window*. Bei moderaten Verlusten kann man außerdem Terme höhere Ordnung vernachlässigen und erhält schließlich statt (5.19)

$$U_r(x_r) = \frac{1}{T_r}\arctan(x_r T_r) \tag{5.23}$$

Diese in Abb. 5.8 auch graphisch dargestellte Abhängigkeit des Durchsatzes von der Umlaufzeit und der Verlustrate kann Nutzer dazu veranlassen, Objekte nach Möglichkeit von nahegelegenen Servern herunterzuladen, und fördert insofern den wirtschaftlichen Umgang mit den Ressourcen des Netzes. Außerdem erscheint angesichts von Gleichung (5.19) der Einsatz von Wartespeicher zur Steigerung des Durchsatzes weniger attraktiv als die Bereitstellung zusätzlicher Bandbreite.

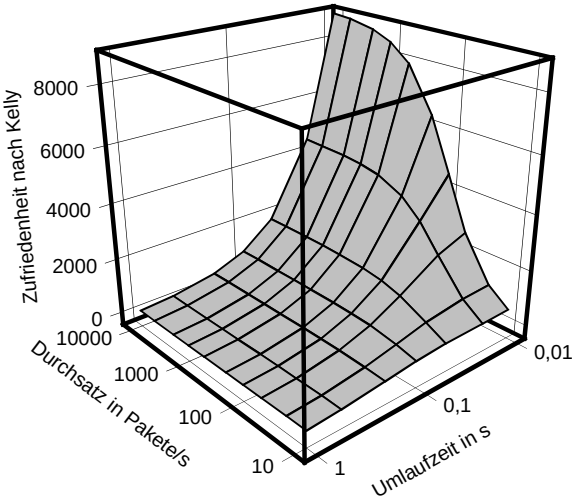

Abb. 5.8: *Graphische Darstellung der Funktion (5.19)*

In den durch die Differentialgleichungen (5.12) und (5.18) beschriebenen Verfahren der Überlastregelung sollten Netzknoten nach Möglichkeit dem Sender ihren Schattenpreis signalisieren, etwa indem sie Pakete mit einer bestimmten Rate markieren [115]. Im Prinzip ist es so, daß ein zusätzliches Paket nichts kostet, wenn es während der Belegtperiode der Bedieneinheit, in dem es ankommt, trotz der zusätzlichen Bedienarbeit nach seiner Ankunft nicht zu Verlusten kommt. Nur wenn es in einer Belegtperiode vor dem Verlust eines Paketes ankommt, erzeugt es zusätzliche Kosten und sollte daher markiert werden. Ob dies der Fall sein wird, kann natürlich beim Eintreffen eines Paketes in einem Netzknoten noch nicht vorhergesagt werden. Daher muß man sich mit heuristischen Verfahren zur Markierung von Paketen behelfen.

Sowohl adaptives RED [76] als auch Athuraliyas und Lows Algorithmus für REM [15] lösen sich bei der Berechnung der Markierungswahrscheinlichkeit von der Verlustwahrscheinlichkeit. Statt dessen verfolgen beide Verfahren das Ziel, bei möglichst geringer Belegung des Wartespeichers die Ankunftsrate im Bereich der Bedienrate zu stabilisieren. Aktives Puffermanagement kann die Zielfunktion des Gesamtsystems verändern [131]. Jeder der Algorithmen führt im Zusammenspiel mit der Überlastregelung von TCP auf eine Ruhelage, die a posteriori als optimal definiert wird. Eine objektive Zielgröße gibt es daher nicht. Allenfalls könnte man messen, wie schnell sich das System auf die jeweilige Ruhelage einschwingt. Bei RED/REM kommt aber erschwerend hinzu, daß die Zielfunktion von Parametern abhängt, die bei adaptivem RED/REM dynamisch sind und sich sicherlich auch von Knoten zu Knoten unterscheiden. Zudem darf nicht übersehen werden, daß Differentialgleichungen wie (5.12) oder (5.18) nur sehr rudimentäre Modelle sind, die weder Einzelheiten der Überlastregelung von TCP (u. a. *Slow Start*, *Retransmission Timeout*, *Fast*

Recovery/Conservation of Packets bzw. bei ECN die Reaktion auf nur eine Überlastanzeige pro Umlaufzeit [144]) noch die Verzögerung der Rückkopplung und den Einfluß des Zustands $\boldsymbol{x}$ auf die Umlaufzeiten T_r darstellen. In der vorliegenden Arbeit wird daher die einfache Funktion (5.19) zur Messung der im Mittel erzielten Gesamtzufriedenheit pro Zeit gemessen. Denn letztlich signalisiert ein Nutzer mit der Wahl von TCP *Reno*, TCP *New Reno* und TCP SACK sein Einverständnis mit dem Verhalten dieser Varianten von TCP und der zugehörigen Zielfunktion so wie er sich mit TCP *Vegas* [35] für ein anderes Verhalten und eine entsprechend andere Zielfunktion [131] entscheiden würde.

In den Simulationen, die in Kapitel 6 vorgestellt werden, wird die Zufriedenheit immer am Ende der Übertragung eines Objektes gemessen. Dazu werden während der Übertragung alle von TCP durchgeführten Messungen der Umlaufzeit T_r übernommen, statistisch erfaßt und am Ende ihr Mittelwert in (5.19) eingesetzt. Dieser Einzelbeitrag multipliziert mit dem Quotienten aus der Objektgröße und der über alle Übertragungen gemittelte Objektgröße wird zur Gesamtzufriedenheit addiert. Am Ende einer Simulation wird die Gesamtzufriedenheit durch die simulierte Zeitspanne dividiert, um zu vergleichbaren Ergebnissen zu kommen.

Um den Einfluß der Umlaufzeiten genauer erfassen zu können, erweitern Hollot et al. [95] Kellys Modell und ersetzen (5.17) durch die miteinander gekoppelten Differentialgleichungen

$$\frac{\mathrm{d}}{\mathrm{d}\,t} cwnd(t) = \frac{1}{cwnd(t)} - \frac{cwnd(t)\,cwnd(t-T_r(t))}{2\,T_r(t-T_r(t))} P_M(t-T_r(t)) \tag{5.24}$$

$$\frac{\mathrm{d}}{\mathrm{d}\,t} Q(t) = \frac{cwnd(t)}{T_r(t)} N(t) - C \tag{5.25}$$

wobei in (5.25) zu berücksichtigen ist, daß zwischen der nun von der Zeit abhängigen Umlaufzeit $T_r(t)$ und dem Füllstand der Warteschlange der einfache Zusammenhang

$$T_r(t) = \frac{Q(t)}{C} + T_{0r} \tag{5.26}$$

angenommen wird. (5.25) und (5.26) beschreiben den Verlauf des Füllstandes des Wartespeichers $Q(t)$ in einem Netz mit einem Knoten mit Bedienkapazität *C* und fester minimaler Umlaufzeit T_{0r}. Gleichung (5.25) ist so nur korrekt unter der Annahme, daß die Fenster $cwnd(t)$ der $N(t)$ Verbindungen synchron laufen und die Umlaufzeit $T_r(t)$ für alle gleich ist. Nach der Linearisierung von (5.24) und (5.25) um den angestrebten Arbeitspunkt und weiteren Näherungen erhält man das Modell einer Regelstrecke mit Blöcken, welche die Umlaufzeit, die Überlastregelung von TCP und das Verhalten des Puffers im Arbeitspunkt modellieren. Seine Übertragungsfunktion stellt den

Zusammenhang zwischen einer kleinen Ablenkung der Markierungswahrscheinlichkeit $\delta P_M(t)$ am Eingang aus dem Arbeitspunkt und der daraus resultierenden Änderung $\delta Q(t)$ des Pufferfüllstandes am Ausgang her. RED schließt den Regelkreis, indem es zunächst aus $\delta Q(t)$ mit Hilfe von Formel (5.1) die mittlere Ablenkung $\delta \overline{Q}(t)$ und in einer zweiten Stufe entsprechend (5.3) die Markierungswahrscheinlichkeit $\delta P_M(t)$ aktualisiert [148, 95]. Die Parameter von RED werden in [95] mit Hilfe dieses Modells so dimensioniert, daß RED das Verhalten des geschlossenen Regelkreises dominiert und gleichzeitig die Anforderungen nach Stabilität und Regelgüte erfüllt sind. Dafür muß aber die Anzahl der Verbindungen und die Umlaufzeit ungefähr bekannt sein.

Um auch unabhängig von einem möglicherweise nicht ganz zutreffenden Modell das dynamische Verhalten der Überlastregelung bei Lastwechseln und ihr Konvergenzverhalten untersuchen zu können, wird innerhalb einer Gruppe von aktiven Verbindungen, die denselben Weg durch das Netz nehmen und deshalb alle den gleichen Durchsatz erzielen sollten, die unfair verteilte Bandbreite

$$\sum_i \max\{0, x_i - \frac{1}{N}\sum_j x_j\} \tag{5.27}$$

in unterschiedlich langen Meßintervallen statistisch erfaßt. In (5.27) ist *N* die Anzahl der während des gesamten Meßintervalls aktiven Verbindungen, die sich zu diesem Zwecke in eine zentralen Meßinstanz einbuchen und am Ende der Aktivität wieder ausbuchen. Eine Quelle und ihre Verbindung zu einer Senke gilt dann als aktiv, wenn sie die Übertragung eines Objektes gestartet hat, dessen letztes Byte aber noch nicht quittiert worden ist. Weitere Maße, wie z. B. der Anteil von Sendewiederholungen am Verkehrsaufkommen oder die mittlere Anzahl gleichzeitig aktiver Verbindungen, zahlreiche Zählstatistiken und Traces können bei Bedarf in die Auswertung einbezogen werden.

Eine zeitliche Mittelung des Durchsatzes hat bei nicht permanent aktiven Quellen wenig Aussagekraft, selbst wenn sich die Messung nur auf die Aktivitätsphasen der Quelle erstreckt. Denn langsame und deshalb lang andauernde Übertragungen werden bei dieser Meßweise überproportional berücksichtigt. Aus diesem Grunde wird bei der Messung des Durchsatzes ähnlich wie bei der Messung der Gesamtzufriedenheit verfahren. Der während der Aktivitätsphase einer Quelle bzw. Verbindung ermittelte Durchsatz wird mit der Größe des übertragenen Objektes multipliziert und am Ende wird zur Mittelung über die Einzelbeiträge nicht durch die Anzahl der Beiträge, sondern die übertragene Datenmenge dividiert. Auf diese Weise wird angegeben, mit welcher Rate eine Dateneinheit im Mittel übertragen wird.

6 Leistungsuntersuchung von Algorithmen zur Überlastregelung mit TCP

Bei der Leistungsuntersuchung der Überlastregelung von TCP ohne oder in Kombination mit verbindungslos arbeitenden Erweiterungen des Verkehrsmanagements in Netzknoten müssen eine ganze Reihe von grundlegenden Problemen berücksichtigt werden. Zum einem lassen sich die Eigenschaften und Anforderungen elastischer Datenströme nur schwer fassen. Aufgrund der Fehlersicherung sowie Fluß- und Überlastregelung von TCP kann nicht einmal das Verkehrsangebot einer Quelle beziffert werden, das die Netzknoten letztlich durchsetzen müssen. Zum anderen kommt es beim Zusammenspiel der unterschiedlichen Algorithmen in den Endsystemen und Netzknoten zu komplexen Wechselwirkungen, die das Systemverständnis erschweren. Außerdem liefern die unbestimmten Dienstgüteanforderungen elastischer Verkehrsströme kaum Ansatzpunkte für einheitliche Leistungsmaße, die eine gute Grundlage für umfassende Studien sein könnten.

Die im Rahmen dieser Arbeit entwickelte und in Kapitel 5 beschriebene Simulationsmeßtechnik erleichtert umfassende Studien, die den Einfluß bisher nur teilweise berücksichtigter Faktoren aufzeigen sollen. Dazu gehören nach Ansicht des Autors umfassende Untersuchungen zum Einfluß der Länge und Aktivität von Verbindungen, mögliche Interaktionen des Nutzers sowie variablere und vor allem nicht immer aktive Quellen. Absolut unumstößliche Schlußfolgerungen lassen indes selbst Ergebnisse nicht zu, die auf einem theoretisch so fundierten Leistungsmaß wie der Zufriedenheit nach Kelly [115] basieren. Dazu sind die Annahmen dieser Modelle zu ideal. Beispielsweise werden Paketverluste, die besonders bei hohem Angebot einen signifikanten Anteil am Verkehrsaufkommen haben, nur unzureichend berücksichtigt. Darüber hinaus steht die Vorgehensweise, a posteriori eine Optimierungsaufgabe zu definieren, für die TCP eine optimale Lösung liefert, unter Umständen im Widerspruch zu den tatsächlichen Anforderungen von Anwendungen und den Erwartungen von Nutzern und Netzbetreibern.

6.1 Ziele der Leistungsuntersuchung

Kellys [115] Untersuchung der Stabilität der Überlastregelung von TCP (siehe Kapitel 5) bezieht sich auf ein Modell, in dem eine feste Anzahl von Verbindungen permanent Daten überträgt mit einer Rate, die sie ausgehend von einem Initialwert entsprechend der Differentialgleichung (5.18) solange anpassen, bis sie den stabilen Arbeitspunkt erreichen. Mit Hilfe der von Hollot et al. vorgestellten Methode [95] ist es sicherlich möglich, den Auf- und Abbau von Verbindungen als Störung zu modellieren und die Systemantwort zu analysieren. Beide Modelle setzen jedoch voraus, daß die Quellen vollkommen elastisch sind und ihren Arbeitspunkt mit Hilfe des für *Congestion Avoidance* festgelegten Algorithmus nachführen, den die Differentialgleichung (5.18) modelliert.

Die Überlastregelung von TCP bei hohem Verkehrsangebot auf Anwendungsebene führt selbst dann zu Verlusten und damit verbunden zu Blindlast für das Wiederholen dieser Pakete, wenn ECN und REM eingesetzt werden. Deshalb interessiert darüber hinaus auch die Frage, wie groß der Nutzanteil der Rate ist, die sich bei Verwendung der Überlastregelung einstellt. Die Überlastregelung sollte dafür sorgen, daß bei wachsendem Angebot im (unter Umständen im Kellyschen Sinne stabilen) Arbeitspunkt der Nutzdurchsatz, die Gesamtzufriedenheit oder andere geeignete Metriken nicht abnehmen. Fredj et al. zeigen jedoch in [77], daß TCP dieser Erwartung nicht gerecht wird und bei steigendem Angebot gleichzeitig der Anteil von Paketwiederholungen steigt und der Durchsatz einbricht. Um weitergehende Schlußfolgerungen für die Gestaltung des Verkehrsmanagements ziehen zu können, bezieht die vorliegende Arbeit Puffermanagement ein und wendet eine über die Betrachtung von Durchsatz und Paketwiederholungen hinausgehende Methodik zur Leistungsbewertung an, die auch die Untersuchung größerer Systeme erlaubt. Eine Schlüsselrolle nimmt dabei die Zufriedenheit nach Kelly gemäß der zur Berücksichtigung von *Delayed Acknowledgement* modifizierten Formel (5.19) ein, die wie in Kapitel 5 erläutert gemessen und bewertet wird.

6.2 Netztopologie und Quellenmodelle

Zur Leistungsuntersuchung werden mehrere Knoten zu einer Kette zusammengeschaltet. Jeder Knoten *i* besteht aus einem Wartespeicher mit oder ohne Puffermanagement, der S_i byte Daten aufnehmen kann, und einer Bedieneinheit, die Pakete in der Reihenfolge ihres Eintreffens mit der Rate C_i bedient. Die Knoten sind durch Übertragungsabschnitte mit Laufzeiten $T_{0r,i}$ miteinander verbunden. Diese Übertragungsabschnitte (mit unendlicher Bedienrate) übertragen die Datenströme

von TCP-Sendern, die entweder den Vorgängerknoten in der Kette passieren oder direkt von einem Endsystem mit einem Netzzugang beschränkter Bandbreite (siehe Tabelle B.1) stammen. Am Ausgang der Bedieneinheiten können die Datenströme entweder zum nachfolgenden Knoten oder zu einem TCP-Empfänger weitergeroutet werden. Quittierungen laufen verzögerungs- und verlustfrei auf direkten Verbindungen vom TCP-Empfänger zum TCP-Sender zurück. Diese Verbindungen sind in Abb. 6.1 und Abb. 6.2 nicht dargestellt.

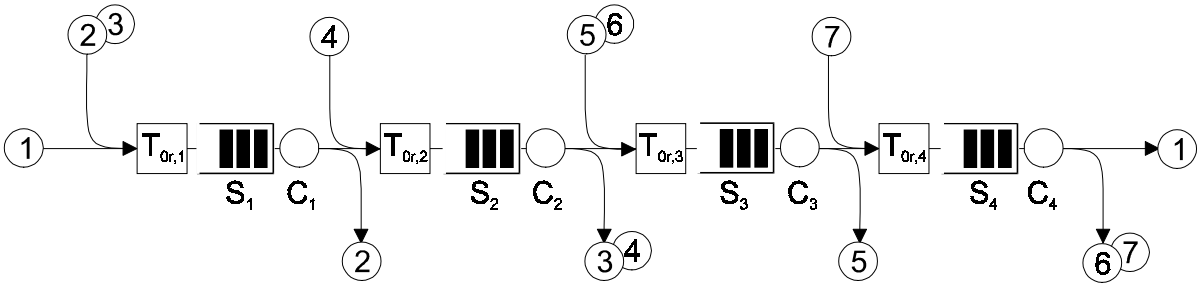

Abb. 6.1: *Das Simulationsmodell besteht aus vier Knoten mit den Übertragungsraten $C_1,...,C_4$ und Wartespeicher der Länge $S_1,...,S_4$. Die Übertragungsabschnitte in Vorwärtsrichtung weisen feste Verzögerungszeiten $T_{0r,1},...,T_{0r,4}$ auf. Nur ein Teil der Quellen (Position 1) passiert alle vier Knoten. Einige (Positionen 3 und 6) werden durch zwei Knoten und wiederum andere (Positionen 2, 4, 5 und 7) nur durch einen Knoten geroutet. Die Quittierungen laufen verzögerungs- und verlustfrei direkt von den Empfängern zu den Sendern.*

Im Modell gemäß Abb. 6.1 sind bei hohem Verkehrsangebot die Bediensysteme 3 und vor allem 1 stärker von Paketverzögerung und -verlust betroffen als die Bediensysteme 2 und 4, in die weniger Quellen von außen eingespeist werden. Aus diesem Grunde wird neben Abb. 6.1 auch ein Modell betrachtet, in dem die Bediensysteme als Ring angeordnet sind. Wie Abb. 6.2 zeigt, können so auch die Quellen, die mehrere Knoten passieren, gleichmäßig von außen eingespeist werden.

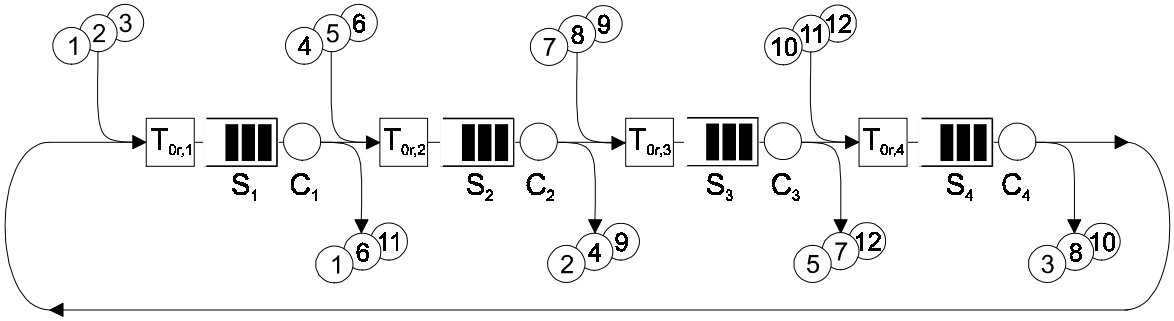

Abb. 6.2: *Das Simulationsmodell gemäß Abb. 6.1 wird zu einem Ring erweitert. Nach wie vor passiert ein Teil der Quellen (Positionen 1, 4, 7, 10) lediglich einen Knoten, andere zwei (Positionen 2, 5, 8, 11) bzw. vier Knoten (Positionen 3, 6, 9, 12). Diese Quellen sind aber nun gleichmäßig im Netz verteilt.*

In diesen beiden Netztopologien werden von Generatoren in negativ exponentiell verteilten Abständen Quellen gestartet, die eine Verbindung eröffnen, anschließend genau ein Objekt über-

tragen, die Verbindung wieder schließen und solange aussetzen, bis sie zu einem späteren Zeitpunkt erneut von einem der Generatoren aktiviert werden. Die Länge des Objektes ist negativ exponentiell oder – wie z. B. in [8, 109] – Pareto verteilt (und auf eine ganze Zahl von Bytes gerundet).

Bei Pareto verteilten Objektgrößen wird neben dem Formparameter α_P statt des Minimalwertes k_P in der Verteilungsfunktion

$$F_X(x) = 1 - \left(\frac{k_P}{x} \right)^{\alpha_P} \tag{6.1}$$

der Erwartungswert

$$\mathrm{E}\{X\} = \begin{cases} \dfrac{\alpha_P}{\alpha_P - 1} k_P, & \text{wenn } \alpha_P > 1 \\ \infty, & \text{wenn } \alpha_P \leq 1 \end{cases} \tag{6.2}$$

angegeben [53]. In den Abbildungen wird oft verkürzend Pareto 1,5 oder Pareto 1,9 für eine Pareto-Verteilungsfunktion mit dem Formparameter $\alpha_P = 1{,}5$ oder $\alpha_P = 1{,}9$ geschrieben. Der Erwartungswert geht jeweils aus dem Zusammenhang hervor.

Die Simulationsumgebung unterstützt zwar auch komplexere Quellenmodelle, etwa die Übertragung mehrerer Objekte in einer einzigen Verbindung bzw. den Neuaufbau weiterer Verbindungen zur Verarbeitung von in das ursprünglich angeforderte Objekt eingebetteten Anforderungen (*Active Off Time* [136]) oder ganz neuer Anforderungen nach einer Denkzeit des Nutzers (*Inactive Off Time, Think Time* [136]). Auch diese komplexeren Quellen sollten jedoch durch einen übergeordneten Zufallsprozeß erzeugt werden, wenn das Modell realistisch sein und das Verkehrsangebot auf der Anwendungsebene eingestellt werden soll. Verzichtet man dennoch auf einen übergeordneten Zufallsprozeß zur Erzeugung von Quellen, empfiehlt es sich zur Dimensionierung des Systems das Bedienmodell von Heyman et al. [92] heranzuziehen, das das Verhalten von Quellen modelliert, in denen sich Übertragungen von Objekten beliebig verteilter Größe und eine ebenfalls beliebig verteilte Denkzeit immer wieder ablösen. In diesem Modell teilen sich die Verbindungen die verfügbare Bandbreite gleichmäßig. Überschreitet das Angebot die Kapazität der Bedieneinheit, wird die Überlastregelung von TCP aktiv. Das Auf und Ab des *Congestion Window* reduziert dann die nutzbare Bedienkapazität. Dieser Reduktion wird im Bedienmodell Rechnung getragen.

Das Verkehrsangebot (auf der Anwendungsebene) wird im folgenden jeweils berechnet als Quotient aus dem von den Quellen pro Zeiteinheit erzeugten Datenvolumen und der auf den einzelnen Übertragungsabschnitten verfügbaren Bandbreite, wenn ohne Unterbrechung nur Pakete in der Länge der *Maximum Segment Size* (vgl. Tabelle B.1) zuzüglich des TCP- und IP-Paketkopfes übertragen

werden würden. Selbstverständlich steuern die Paketwiederholungen in Folge von Verlusten zusätzlichen Verkehr auf den Übertragungsabschnitten bei, der bei der Dimensionierung der Quellen jedoch unberücksichtigt bleibt. Die Beiträge der Quellen auf den verschiedenen Positionen zum Gesamtangebot auf Anwendungsebene werden so eingestellt, daß sie ihrem aufgrund von Gleichung (5.21) zu erwartenden Gesamtdurchsatz entsprechen. Quellen, deren Verbindungen mehrere Knoten passieren erzeugen also weniger Verkehr als Quellen, die nur einen Knoten passieren. Zur Berechnung gemäß Gleichung (5.21) wird angenommen, daß Pakete an jedem Knoten im Mittel die gleiche Verzögerung und Verlustquote erfahren und deren Summe näherungsweise als Gesamtverlustquote entlang des Verbindungspfades gelten kann.

Zum Teil werden zum Vergleich auch Simulationen mit immer aktiven (engl. *greedy* oder *persistent*) Quellen durchgeführt. Auch in diesen Fällen erleichtern kurze, vernachlässigbare Unterbrechungen der Verbindungen (sehr kurze Denkzeiten) die statistische Auswertung.

Mit $C_1 = ... = C_4 = 10$ Mbit/s, $S_1 = ... = S_4 = 100000$ byte, $T_{0r,1} = ... = T_{0r,4} = 10$ ms werden die Parameter des Netzes so gesetzt, daß die Verzögerungen durch die variablen Wartezeiten in den Knoten einen signifikanten Einfluß auf das Verhalten von TCP haben. Die Werte von $C_{1,}...,C_4$ sowie $S_{1,}...,S_4$ beziehen sich dabei auf IP-Pakete. Der zusätzliche Bandbreite- und Speicherbedarf für die Übertragung der Pakete, den Schichten unterhalb von IP erzeugen, wird nicht modelliert. Die maximale Fenstergröße der Sender und Empfänger von 65536 byte stellt sicher, daß gegebenenfalls eine einzige aktive Verbindung ausreicht, um die Bedienkapazität der Knoten, die sie passiert, auszulasten. Dies führt dazu, daß die Regelungsmechanismen von TCP voll zum Tragen kommen. Alle weiteren Parameter der Endsysteme und der Status optionaler Algorithmen von TCP, die während der Leistungsuntersuchung nicht variiert werden, können Tabelle B.1 in Anhang B entnommen werden.

6.3 Verhalten bei variablem Verkehrsangebot

Das Verhalten von TCP ohne oder in Kombination mit adaptivem REM bei variablem Angebot ist in beiden Konfigurationen untersucht worden (Abb. 6.1 und 6.2). In beiden Fällen erhält man qualitativ sehr ähnliche Ergebnisse, so daß sich die folgende Darstellung auf die zum Ring erweiterte Konfiguration nach Abb. 6.2 beschränkt.

Abb. 6.3 zeigt den starken Abfall der Gesamtzufriedenheit gemäß Formel (5.23) bei wachsendem Angebot. Während die Verteilungsfunktion der Objektgröße bei konstantem Erwartungswert von

200000 byte keinen nachweisbaren Einfluß hat, ändert adaptives REM das Systemverhalten deutlich. Wie Abb. 6.5 am Beispiel des ersten Knotens in Abb. 6.2 zeigt, reduziert adaptives REM mit den Schwellen $S_{min} = 12500\,byte$ und $S_{max} = 37500\,byte$ die Paketverzögerung, ohne daß zunächst darunter der Durchsatz spürbar leidet (vgl. Abb. 6.9 und 6.11), und steigert auf diese Weise die Gesamtzufriedenheit bei niedrigem Angebot. Dagegen bricht bei höherem Angebot die Gesamtzufriedenheit früher und deutlicher ein, als dies ohne adaptives REM der Fall ist. Abb. 6.6 zeigt eine starke und regelmäßige Schwingung der Belegung des Wartespeichers.

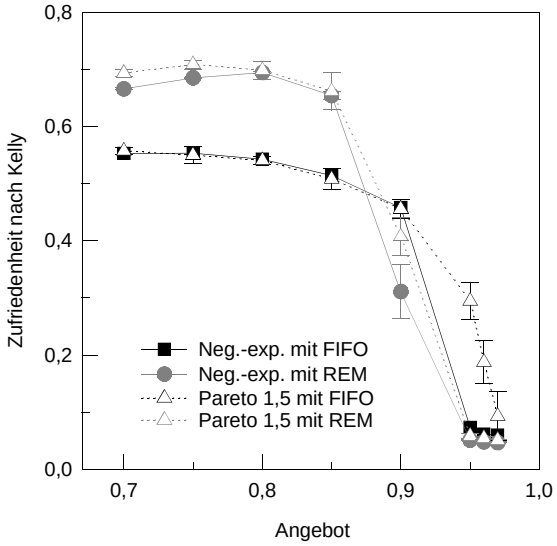

Abb. 6.3: *Gesamtzufriedenheit gemäß Formel (5.23) in Abhängigkeit des Angebots und der Verteilungsfunktion der Objektgröße (Erwartungswert 200000 byte) bei FIFO mit und ohne adaptivem REM.*

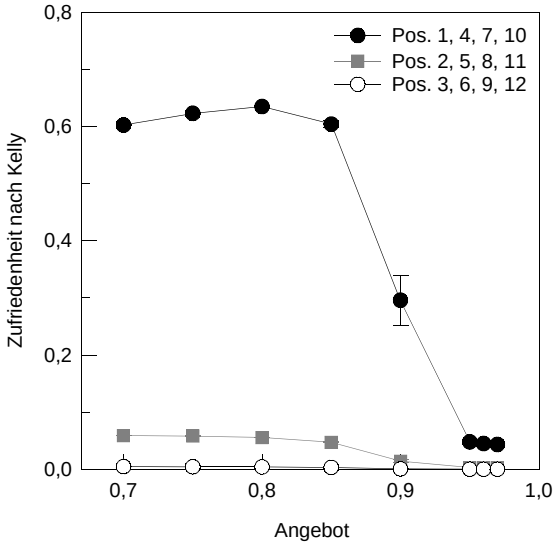

Abb. 6.4: *An den einzelnen Positionen erzielte Gesamtzufriedenheit gemäß Formel (5.23) in Abhängigkeit des Angebots bei neg.-exp. verteilter Objektgröße (Erwartungswert 200000 byte) und FIFO mit adaptivem REM.*

Abb. 6.4 löst die Beiträge der Quellen an den verschiedenen Positionen in Abb. 6.2 am Beispiel einer negativ-exponentiell verteilten Objektgröße und adaptivem REM auf. Einerseits aufgrund der Aufteilung der Bandbreite von TCP und andererseits aufgrund der dieses Verhalten antizipierenden Verteilung des Gesamtangebotes auf die einzelnen Positionen, leisten Verbindungen, die mehrere Knoten passieren, einen eher geringen Beitrag zur Gesamtzufriedenheit.

Bei der Interpretation der Kurvenverläufe in Abb. 6.3, 6.4, 6.5, 6.9, 6.10, 6.11 und 6.12 ist zu beachten, daß die Zahl aktiver Verbindungen auf allen Positionen jeweils auf 200 Quellen begrenzt ist. In den hier betrachteten Fällen setzt daher bei einem Angebot von 0,95 Blockierung von Verbindungen ein, die bei weiter wachsendem Angebot so zunimmt, daß zum Teil die Gesamtzufriedenheit und der entsprechend der Objektgröße gewichtete mittlere Durchsatz aktiver Verbindungen kaum noch abnimmt.

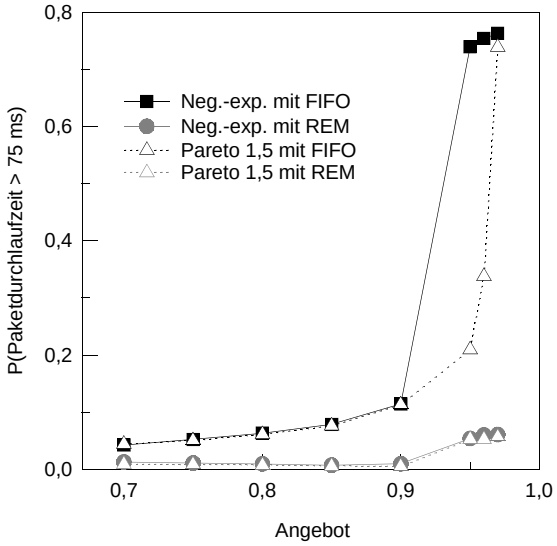

Abb. 6.5: *Wahrscheinlichkeit, daß Pakete im ersten Knoten in Abb. 6.2 länger als 75 ms verweilen, in Abhängigkeit vom Angebot, von der Verteilungsfunktion der Objektgröße (Erwartungswert 200000 byte) und dem Puffermanagement.*

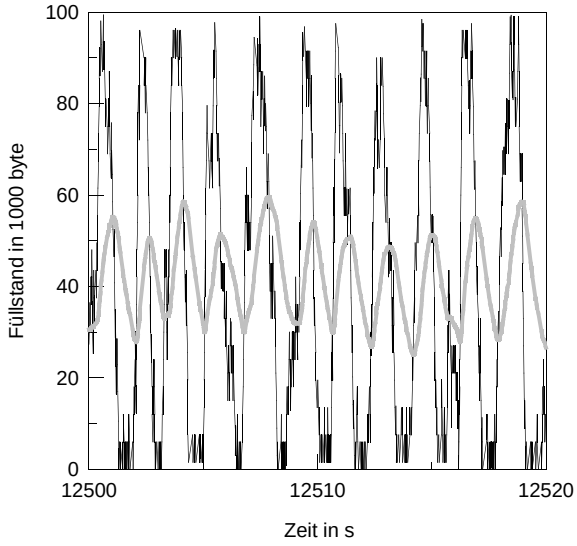

Abb. 6.6: *Aufzeichnung der Belegung des Wartespeichers im ersten Knoten in Abb. 6.2 bei neg.-exp. vert. Objektgröße (Erwartungswert 200000 byte), adapt. REM und einem Angebot von 0,9. Die graue Kurve zeigt die mit dem EWMA ermittelte mittlere Belegung $\overline{Q}$ desWartespeichers.*

Bemerkenswert an diesen Ergebnissen ist, daß die Gesamtzufriedenheit bei wachsendem Verkehrsangebot selbst dann einbricht, wenn eine den Vorschlägen in [76] folgende Implementierung und Parametrisierung von adaptivem REM eingesetzt wird, mit deren Hilfe Verluste und demzufolge Paketwiederholungen weitgehend vermieden und Verzögerungszeiten reduziert werden sollen.

Der beträchtliche Anteil der Paketwiederholungen an dem von den TCP-Sendern ausgehenden Verkehr (Abb. 6.7 und 6.8) ist dafür jedoch nicht allein ausschlaggebend, da der Großteil des Verkehrs von Verbindungen stammt, die lediglich einen Knoten passieren. Wird ein Paket dieser Verbindungen am Eingang eines Knotens verworfen, muß keiner der Knoten dafür Bedienarbeit verrichten. Es entsteht folglich durch den Verlust keine Blindlast an einem der Knoten. In ähnlicher Weise erzeugen Verluste von Verbindungen über mehrere Knoten nur zum Teil Blindlast.

In der Tat sorgt die Überlastregelung von TCP bei höher werdendem Verkehrsangebot und den damit einhergehenden Überlastanzeigen bzw. Paketverlusten für eine Verkleinerung der Sendefenster der aktiven Verbindungen. Die Übertragung der Objekte dauert deshalb länger. Je kleiner aber die Sendefenster sind, desto wahrscheinlicher werden *Retransmission Timeouts (RTOs),* welche die Übertragungszeiten weiter verlängern. Das Systemverhalten wird zunehmend geprägt von einer Vielzahl von Verbindungen (Abb. 6.10 und 6.12) mit so zwangsläufig kleinen Sendefenstern und einer äußerst hohen Quote von Paketwiederholungen, die überwiegend durch *Retransmission Timeouts* ausgelöst werden. Bei adaptivem REM werden die Sendefenster durch Überlastanzeigen früher

heruntergeregelt, so daß dieser Effekt früher einsetzt. Der in Abb. 6.9 und 6.11 zu beobachtende Einbruch des Durchsatzes aktiver Verbindungen kommt also zu einem Großteil durch die Verteilung der Bandbreite auf sehr viel mehr Verbindungen zustande.

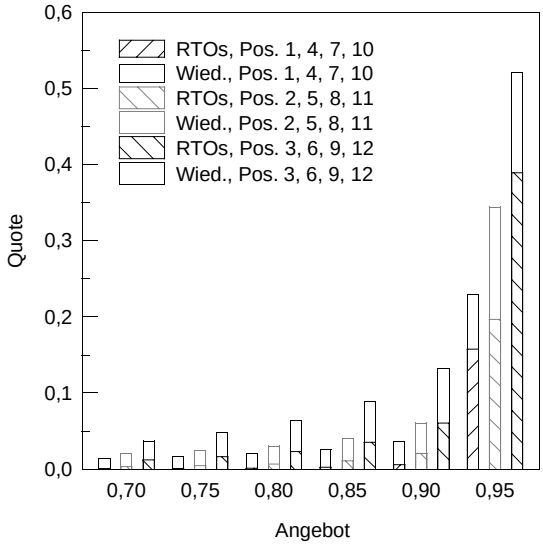

Abb. 6.7 *Anteil der Paketwiederholungen aufgrund von RTOs an den Paketwiederholungen insgesamt in Abhängigkeit des Angebots, bei neg.-exp. verteilter Objektgröße mit Erwartungswert 200000 byte ohne REM.*

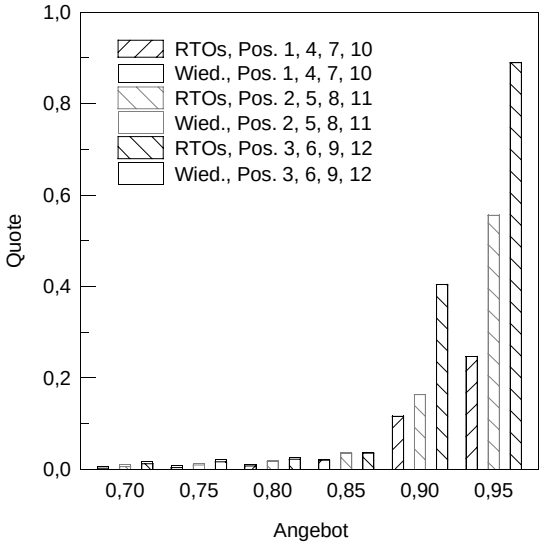

Abb. 6.8: *Anteil der Paketwiederholungen aufgrund von RTOs an den Paketwiederholungen insgesamt in Abhängigkeit des Angebots, bei neg.-exp. verteilter Objektgröße mit Erwartungswert 200000 byte mit adapt. REM.*

Der Verlauf der Gesamtzufriedenheit in Abb. 6.3 zeigt allerdings, daß die Überlastregelung von TCP nicht mit den Zielvorstellungen vereinbar ist, die sich aus den Modellen für TCP in *Congestion Avoidance* ableiten lassen.

Fredj et al. [77] bringen zur Stabilisierung von TCP bei Überlast (Angebot größer als 1) die Verbindungsannahmesteuerung ins Gespräch. Dagegen ist einzuwenden, daß einem Netzknoten nur sehr wenige Informationen zur Verfügung stehen, die ihm als Kriterium dienen könnten, den Meldungsaustausch zum Aufbau einer TCP-Verbindung in den Endsystemen zu unterbinden. Er kennt weder die zu erwartende Verkehrscharakteristik des Datenstromes, die nicht nur von der Aktivität der Datenquelle, sondern auch von der Position von Quelle und Ziel abhängt, noch die Zuordnung von Verbindungen zu einer Sitzung, die in Folge des Blockierens einzelner Verbindungen möglicherweise unter Umständen völlig neu gestartet werden muß. Da erscheint es doch sinnvoller, einerseits auf die Ungeduld der Nutzer zu setzen (wie das durchaus auch in [77] angeregt wird), die ihre Übertragung bei ungenügendem Durchsatz frühzeitig abbrechen, und andererseits dem Nutzer die Möglichkeit zu geben, für besonders kritische Übertragungen durch präventive Verkehrsformung

die Überlastregelung von TCP weitgehend zu neutralisieren und den Aufbau einer Verbindung mit genau spezifizierten Dienstgüteeigenschaften zu ermöglichen.

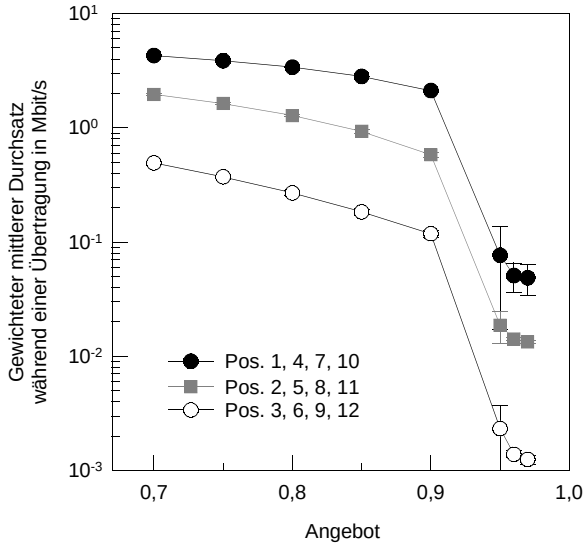

Abb. 6.9: *Entsprechend der Größe des übertragenen Objektes gewichtet gemittelter Durchsatz aktiver Verbindungen in Abhängigkeit von Position und Angebot bei FIFO. Der Erwartungswert der neg.-exp. verteilten Objektgröße ist 200000 byte.*

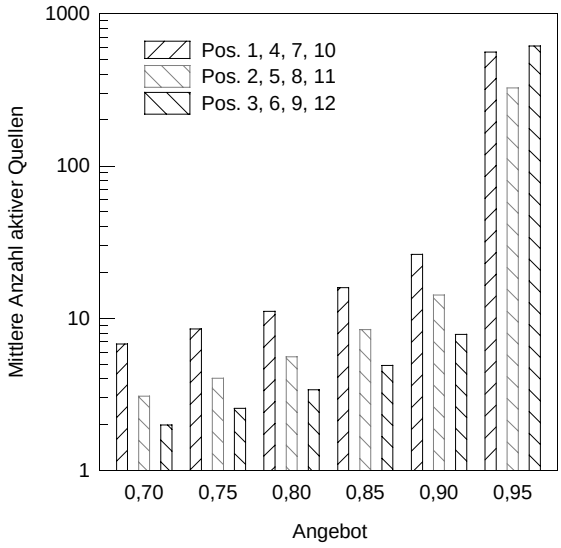

Abb. 6.10: *Mittlere Anzahl aktiver Verbindungen in den Simulationen, die auch Abb. 6.9 zugrunde liegen.*

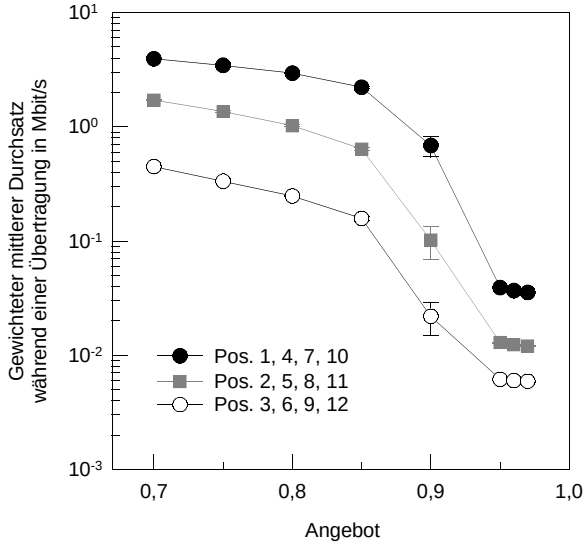

Abb. 6.11: *Entsprechend der Größe des übertragenen Objektes gewichtet gemittelter Durchsatz aktiver Verbindungen in Abhängigkeit von Position und Angebot mit adaptivem REM. Der Erwartungswert der neg.-exp. verteilten Objektgröße ist 200000 byte.*

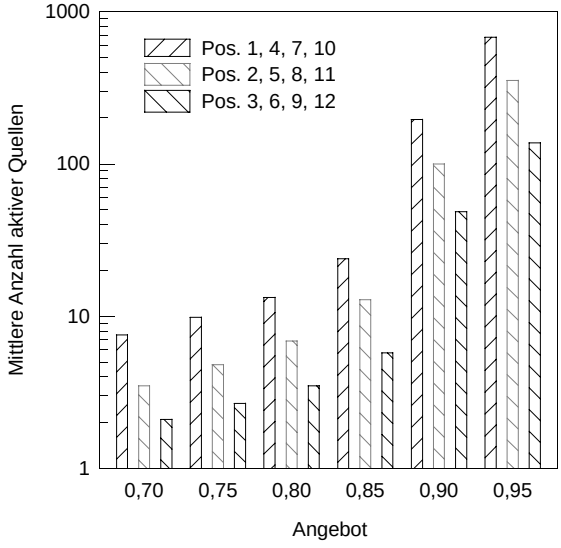

Abb. 6.12: *Mittlere Anzahl aktiver Verbindungen in den auch Abb. 6.11 zugrunde liegenden Simulationen.*

6.4 Verbindungsabbrüche durch den Nutzer

Ohne präzise Dienstgüteanforderungen und verbindliche Dienstgütezusagen des Netzes gewinnt also eine neue Instanz an Bedeutung: Der Nutzer, und zwar entweder direkt interaktiv oder indirekt durch eine ihn intelligent vertretende Anwendung. Weit häufiger als in leitungsvermittelnden Netzen, bricht er als Reaktion auf einen unbefriedigenden Durchsatz bestehende Verbindungen ab, um sie durch neue zu ersetzen.

Benutzerabbrüche sind nach Kenntnisstand des Autors zum ersten Mal in [77] am Beispiel eines Netzes mit nur einem Knoten untersucht worden. Dort basiert die Leistungsbewertung auf Messungen des Durchsatzes und der Blindlast. Eine Übertragung wird abgebrochen, wenn sie länger als eine konkav von der Objektgröße abhängige Zeit benötigt. Das im folgenden verwendete Quellenmodell ist noch etwas einfacher, aber ebenfalls hypothetisch. In regelmäßigen Zeitabständen (und zwar alle 5s) prüft die Quelle die Rate, mit der das Objekt seit dem Start übertragen wird, und setzt die Übertragung nur fort, wenn sie oberhalb einer konfigurierbaren Schwelle liegt. Damit Abbrüche nicht nur bei den aufgrund ihrer Position benachteiligten Quellen in Abb. 6.1 auftreten, wird die Schwelle jeweils auf den mit der Übertragungsdauer gewichtet gemittelten Durchsatz gesetzt. Dieser Wert liegt normalerweise deutlich unter dem mit der Objektgröße gewichtet gemittelten Durchsatz.

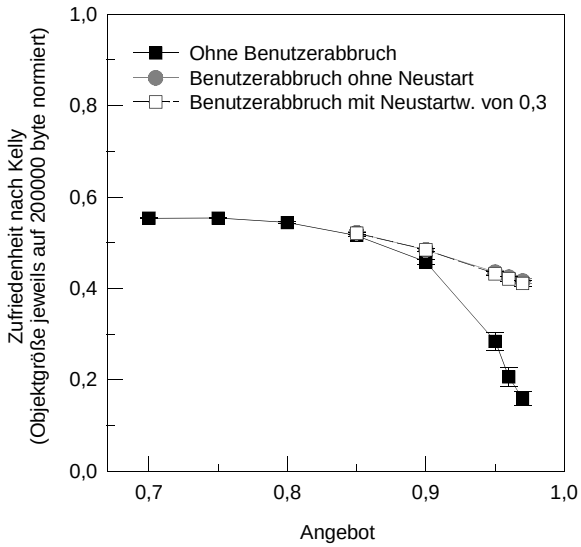

Abb. 6.13: *Gesamtzufriedenheit gemäß Formel (5.23) in Abhängigkeit des Angebots bei neg.-exp. verteilter Objektgröße (Erwartungswert 200000 byte) und FIFO ohne adaptives REM.*

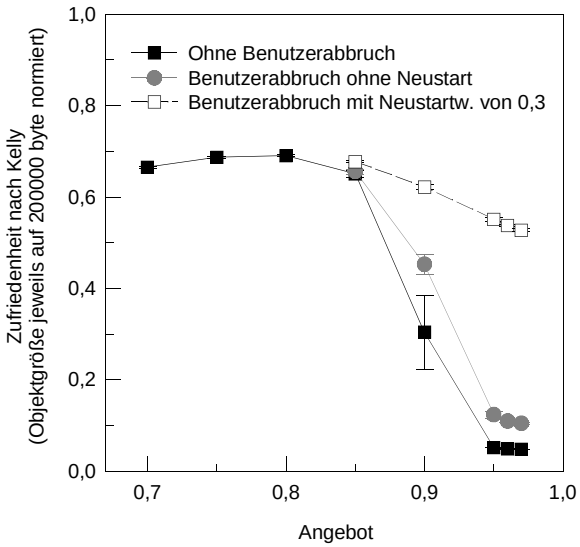

Abb. 6.14: *Gesamtzufriedenheit gemäß Formel (5.23) in Abhängigkeit des Angebots bei neg.-exp. verteilter Objektgröße (Erwartungswert 200000 byte) und FIFO mit adaptivem REM.*

Die Abb. 6.13 und 6.14 vergleichen die Gesamtzufriedenheit bei variablem Angebot, wie sie ohne Abbrüche erzielt wird, mit der Gesamtzufriedenheit bei Benutzerabbrüchen ohne und mit anschließenden neuen Versuchen, dasselbe Objekt noch einmal zu übertragen. Übertragungen, die vorzeitig abgebrochen werden, tragen nichts zur Gesamtzufriedenheit bei und erhöhen also die Blindlast. Dennoch können Benutzerabbüche einen durchaus positiven Einfluß auf die Gesamtzufriedenheit haben, wenn sie das von den Bedienelementen zu bewältigende Verkehrsaufkommen reduzieren.

6.5 Einfluß der Objektgröße

Heyman et al. [92] haben mit der Analyse des oben erwähnten Bedienmodells nachgewiesen, daß der mittlere Durchsatz von TCP-Verbindungen in *Congestion Avoidance* nicht von der Verteilungsfunktion der Objektgröße und Denkzeit abhängt. Dies ist auch dann noch der Fall, wenn ein übergeordneter Poisson-Prozeß die Quellen erzeugt [77]. Da Übertragungen kleinerer Objekte vergleichsweise kurz im Bediensystem verweilen und deshalb den Systemzustand nur kurze Zeit beobachten, ist der Durchsatz, den sie erzielen, allerdings sehr viel variabler als bei größeren Objekten.

Die Ergebnisse in Abb. 6.3 unterstützen diese theoretischen Modelle. Andererseits wird in [77] auch darauf hingewiesen, daß solche Modelle *Retransmission Timeouts* und *Slow Start* nicht adäquat berücksichtigen. Übertragungen kleinerer Objekte sind aber weit mehr von der anfänglichen *Slow Start* Phase und einem größeren Anteil der durch *Retransmission Timeouts* ausgelösten Paketwiederholungen geprägt.

6.5.1 Abhängigkeit von der Verteilungsfunktion der Objektgröße

Wenn alle Quellen kleinere Objekte senden, deren Länge um den Erwartungswert 20000 byte statt um 200000 byte verteilt ist, zeigen die für das Modell aus Abb. 6.1 gewonnenen Ergebnisse einen durchaus signifikanten Einfluß der Verteilungsfunktion auf die Gesamtzufriedenheit (Abb. 6.15) und die Verzögerung in den Wartespeichern (Abb. 6.17 gegenüber Abb. 6.16).

Die auf den Verbindungsaufbau folgenden *Slow Start* Phasen beeinflussen das Verhalten der Datenströme kurzer Übertragungen besonders stark. Dementsprechend wirken sich der Ankunftsprozeß der Verbindungen stärker und die bezüglich der Verteilung der Objektgröße invariante Regelung der Datenströme in *Congestion Avoidance* weniger stark auf das Verkehrsverhalten aus.

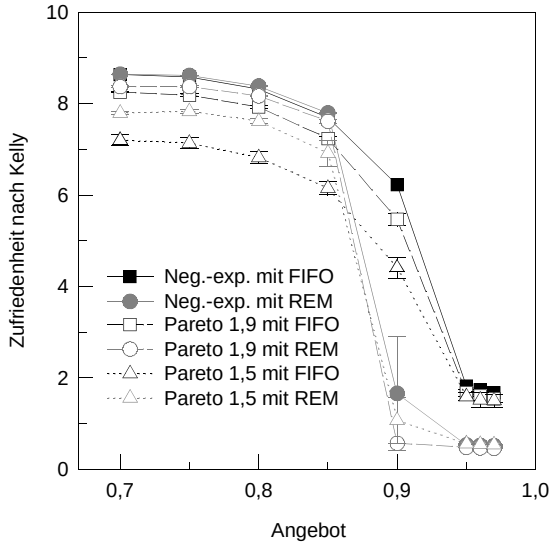

Abb. 6.15: *Gesamtzufriedenheit gemäß Formel (5.23) in Abhängigkeit vom Angebot und der Verteilungsfunktion der Objektgröße (Erwartungswert 20000 byte) bei FIFO mit und ohne adaptivem REM.*

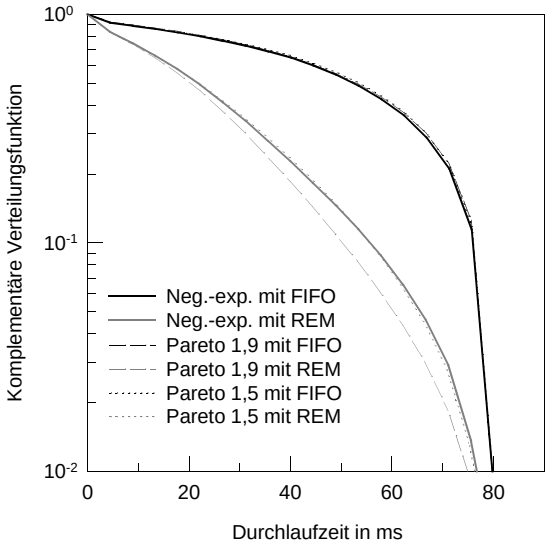

Abb. 6.16: *Komplementäre Verteilungsfunktion der Durchlaufzeit der Pakete an Knoten 1 in Abb. 6.1 bei einer mittleren Objektgröße von 200000 byte und einem Angebot von 0,9.*

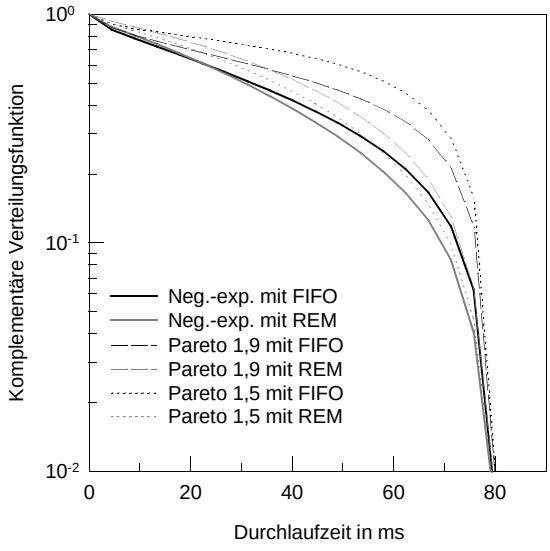

Abb. 6.17: *Komplementäre Verteilungsfunktion der Durchlaufzeit der Pakete an Knoten 1 in Abb. 6.1 bei einer mittleren Objektgröße von 20000 byte und einem Angebot von 0,9.*

Abb. 6.18 und 6.19 zeigen die Varianz-Zeit-Graphen der Ankunftsprozesse an Knoten 1 aus Abb. 6.1 für Quellen mit negativ-exponentiell und mit Pareto (α_P=1,5) verteilten Objektgrößen bei einem Angebot von 0,9.

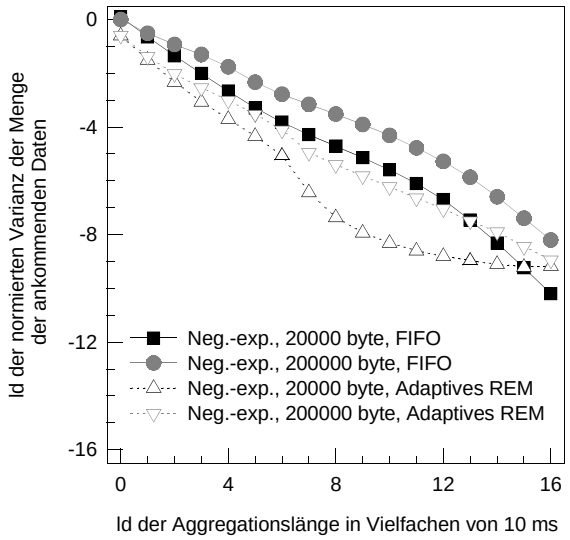

Abb. 6.18: *Varianz-Zeit-Graph des Ankunftsprozesses an Knoten 1 aus Abb. 6.1 bei neg.-exp. vert. Objektgröße und einem Angebot von 0,9. Die Varianzen sind normiert auf die Varianz bei einer mittleren Objektgröße von 200000 byte und FIFO bei der Aggregationslänge 10 ms.*

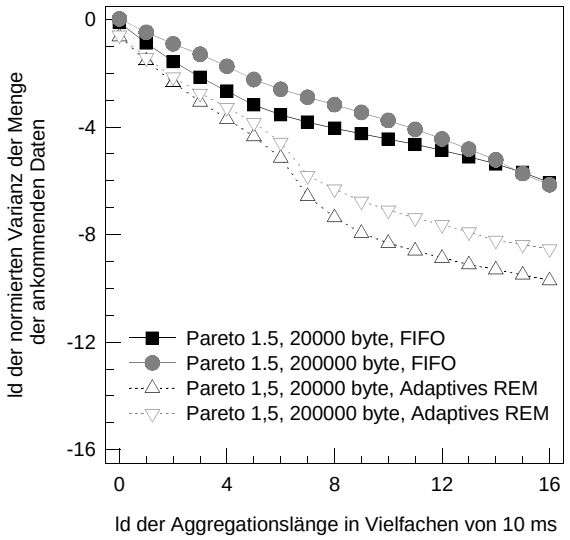

Abb. 6.19: *Varianz-Zeit-Graph des Ankunftsprozesses an Knoten 1 aus Abb. 6.1 bei Pareto ($\alpha_P = 1{,}5$) verteilten Objektgröße und einem Angebot von 0,9. Die Varianzen sind wie in Abb. 6.18 normiert.*

Ein Varianz-Zeit-Graph vollzieht eine Abbildung des Logarithmus der Länge von Meßintervallen (Aggregationslänge) auf den Logarithmus der empirischen Varianz der in Meßintervallen dieser Länge ankommenden Datenmenge $A(t)$. Einzelheiten zu dieser Methode sind in [60] nachzulesen. Wenn man aber zu Formel (3.71) zurückkehrt, sieht man auch ohne weitere Lektüre, daß die Steigung der Kurven im Varianz-Zeit-Graphen für hinreichend große Mittelwertstichproben ein Maß für die Langzeitkorrelation der Stichprobe des Ankunftsprozesses sein muß.

Der flache Kurvenverlauf im Bereich großer Aggregationslängen in Abb. 6.19 bestätigt, daß der Verkehr auch unter dem Einfluß der Überlastregelung von TCP langzeitkorreliert bleibt. Dieses Ergebnis korrespondiert mit [8]. Darüber hinaus zeigen sich die größeren Unterschiede zwischen Abb. 6.18 und 6.19 in den Simulationen, in denen kleinere Objekte ohne adaptives REM übertragen werden. Außerdem belegen Abb. 6.18 und 6.19, daß REM die Varianz des Verkehrs reduziert und sich diesbezüglich den Erwartungen entsprechend verhält.

6.5.2 Abhängigkeit vom Erwartungswert der Objektgröße

Charakteristisch für die Übertragung kurzer Objekte ist der besonders hohe Anteil von durch *Retransmission Timeouts* ausgelösten Paketwiederholungen. Dieses Verhalten belegen die Abb. 6.26 mit 6.27 im nachfolgenden Abschnitt beispielhaft. Bei konstantem Angebot, das in

Abb. 6.20 und 6.21 0,9 beträgt, steigt zwangsläufig die Anzahl aktiver Verbindungen. Wegen der überwiegend kleinen Sendefenster kommt kein kontinuierlicher Datenstrom mehr zustande. Die stabilisierende Regelung gemäß Differentialgleichung (5.18) kommt dann nicht mehr zum Tragen, so daß besonders bei adaptivem REM die Gesamtzufriedenheit einbricht. Wenn größere Objekte übertragen werden, funktioniert die Regelung weitaus besser. Bei FIFO, vgl. Abb. 6.20, erreicht die Gesamtzufriedenheit sehr schnell ein Maximum. Anschließend werden die Störungen durch Verbindungsauf- und abbau immer weniger, was dazu führt, daß die Überlastregelung bei geringen Verlusten die Raten der Verbindungen auf ein im Mittel etwas höheres Niveau einstellt. Damit verbunden sind aber auch etwas höhere Verzögerungszeiten, die letztlich die Gesamtzufriedenheit reduzieren.

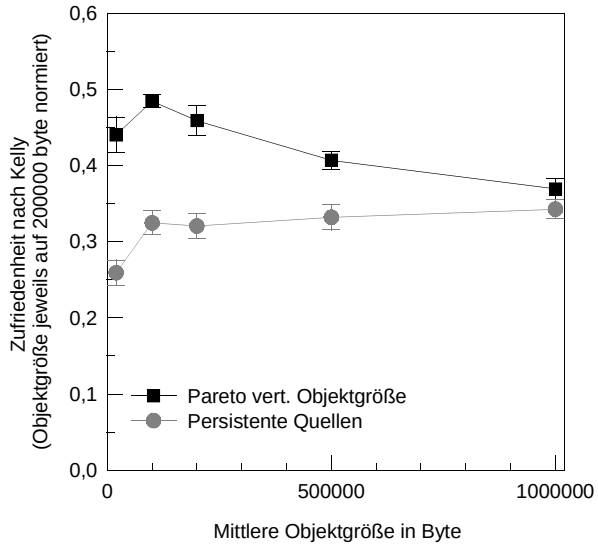

Abb. 6.20: *Vergleich der Gesamtzufriedenheit von Quellen mit Pareto (α_P=1,5) verteilter Objektgröße und persistenten Quellen bei FIFO.*

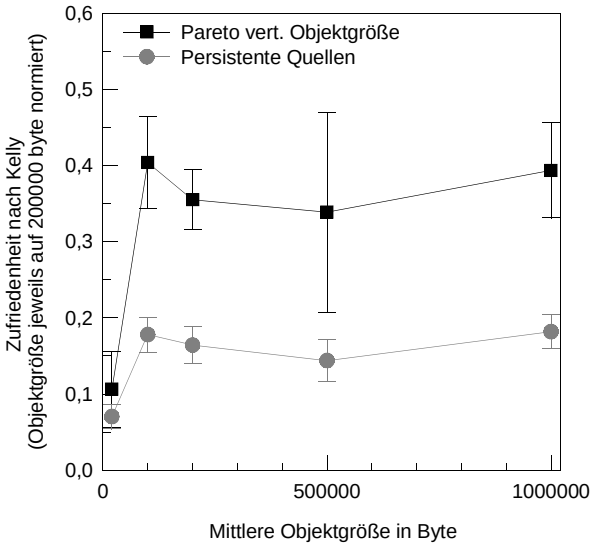

Abb. 6.21: *Vergleich der Gesamtzufriedenheit von Quellen mit Pareto (α_P=1,5) verteilter Objektgröße und persistenten Quellen bei REM.*

Bei adaptivem REM ändern sich von Objektgröße zu Objektgröße die Verlustquote, die Anzahl aktiver Quellen und der Durchsatz auf den einzelnen Positionen zum Teil recht drastisch, was die Gesamtzufriedenheit in nicht vorhersehbarer Weise beeinflußt.

Abb. 6.20 und 6.21 zeigen auch einen Vergleich der Quellen mit Pareto verteilten Objektgrößen mit persistenten Quellen. Deren Anzahl wird jeweils so eingestellt, daß sie der mittleren Anzahl aktiver Quellen im Falle Pareto verteilter Objektgrößen beim auf der Ordinate aufgetragenen Mittelwert entspricht. Die persistenten Quellen erzielen durchweg eine niedrigere Gesamtzufriedenheit, weil sie die Systemressourcen immer maximal auszulasten versuchen und damit insbesondere die Verzögerungszeiten nach oben treiben.

6.5.3 Heterogenes Quellenszenario

Da adaptives REM den Anteil der von *Retransmission Timeouts* ausgelösten Paketwiederholungen an den Paketwiederholungen insgesamt erhöht (vgl. Abb. 6.26 mit 6.27), sorgt es für eine etwas fairere Behandlung (vgl. Abb. 6.24 und 6.25) von Übertragungen kleiner Objekte, die oft nicht genug Pakete auf den Weg bringen, um von den schnelleren Mechanismen zum Auslösen von Paketwiederholungen zu profitieren.

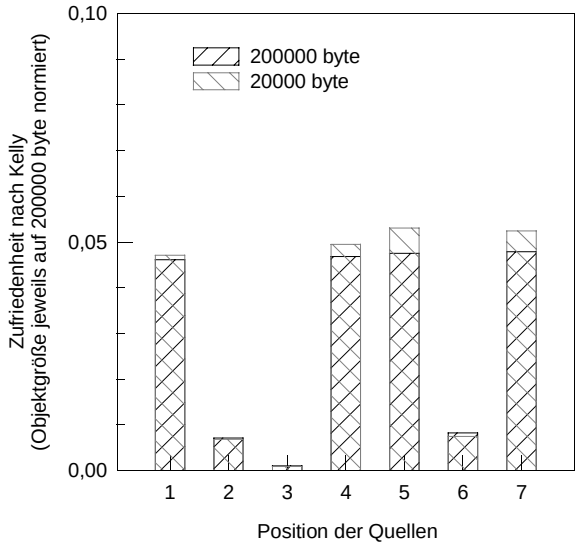

Abb. 6.22: *Gesamtzufriedenheit in Abhängigkeit von der Position der Quellen in Abb. 6.1 für Pareto (α_P=1,5) verteilte Objektgrößen mit den Erwartungswerten 20000 und 200000 byte. Adaptives REM kommt nicht zum Einsatz. Das Angebot beträgt 0,9.*

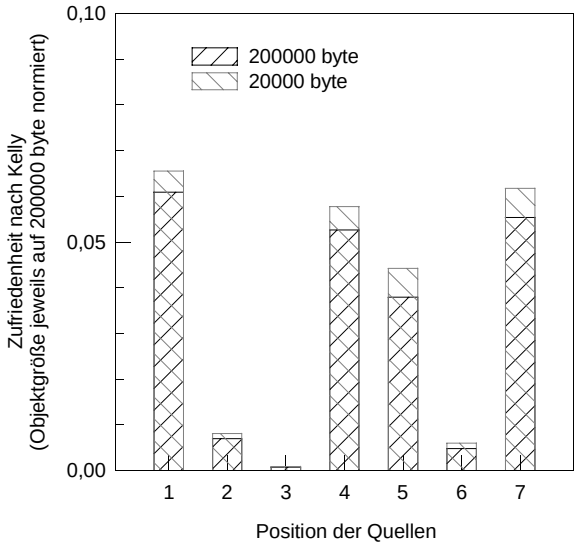

Abb. 6.23: *Gesamtzufriedenheit in Abhängigkeit von der Position der Quellen in Abb. 6.1 für Pareto (α_P=1,5) verteilte Objektgrößen mit den Erwartungswerten 20000 und 200000 byte und adaptivem REM. Das Angebot beträgt 0,9.*

In bezug auf die Gesamtzufriedenheit, die an den einzelnen Positionen in Abb. 6.1 erzielt wird, liegen die kürzeren Übertragungen in Abb. 6.22 und 6.23 sogar vorne, obwohl ihr Durchsatz niedriger ist. Eine genauere Untersuchung zeigt, daß während der längeren Übertragungen im Mittel auch eine etwas höhere Umlaufzeit gemessen wird. Vermutlich beeinflußt die Übertragung einzelner längerer Objekte die Umlaufzeit bzw. deren Messung durch die TCP-Sender so stark, daß diese einen mit ihrer Aktivität korrelierten Mittelwert ermitteln. Die Steigung der Gesamtzufriedenheit (vgl. Abb. 5.8) mit wachsenden Durchsatz ist im fraglichen Bereich von etwa 100 Paketen/s so gering, daß dies den Ausschlag zugunsten der Übertragungen kleinerer Objekte gibt. Diese Beobachtungen können durchaus als Beispiel dafür dienen, daß die Messung der Gesamtzufriedenheit auf der Basis von Formel (5.23) allein für eine objektive Leistungsbewertung nicht ausreicht.

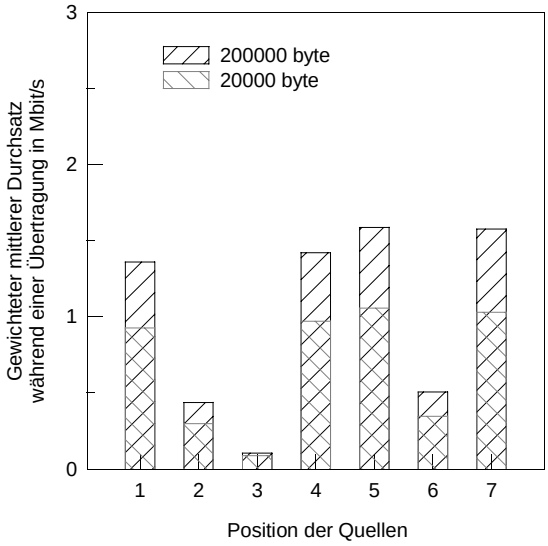

Abb. 6.24: *Mit der Objektgröße gewichtet gemittelter Durchsatz aktiver Quellen in Abhängigkeit von ihrer Position für das Szenario aus Abb. 6.22.*

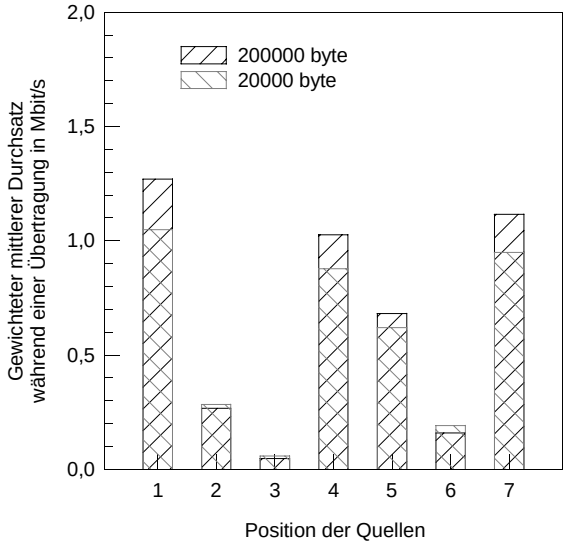

Abb. 6.25: *Mit der Objektgröße gewichtet gemittelter Durchsatz aktiver Quellen in Abhängigkeit von ihrer Position für das heterogene Quellenszenario aus Abb. 6.23.*

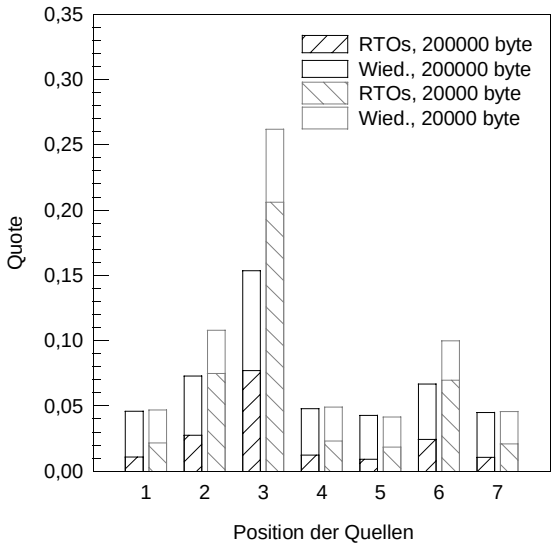

Abb. 6.26: *Anteil von Paketwiederholungen aufgrund von Retransmission Timeouts an den Paketwiederholungen der Quellen auf den verschiedenen Positionen im Szenario aus Abb. 6.22.*

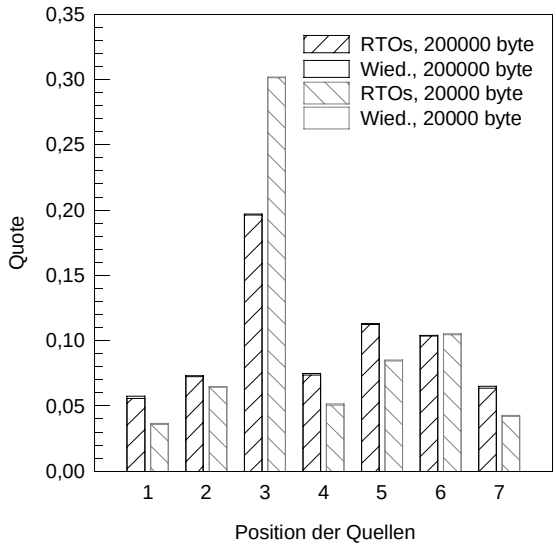

Abb. 6.27: *Anteil von Paketwiederholungen aufgrund von Retransmission Timeouts an den Paketwiederholungen der Quellen auf den verschiedenen Positionen im Szenario aus Abb. 6.23.*

6.6 Vergleich des Einflusses des Puffermanagements und der Varianten *Reno*, *New Reno* und SACK

Angesichts des hohen Anteils der durch *Retransmission Timeouts* ausgelösten Paketwiederholungen bei einem Angebot von 0,9 in Abb. 6.26 mit 6.27 überrascht es, daß bei der Übertragung größerer Objekte (Tabelle 1) SACK und bei der Übertragung kleinerer Objekte (Tabelle 2) sowohl SACK und *NewReno* deutlich schlechter abschneiden als *Reno*, wenn adaptives REM eingesetzt wird.

Tabelle 1: *Mit der Objektgröße gewichtet gemittelte Gesamtzufriedenheit gemäß Gleichung (5.23) pro Zeit. Das Angebot beträgt konstant 0,9, die mittlere Objektgröße 200000 byte.*

		Reno	*New Reno*	SACK
FIFO	Neg.-exp.	0,464 (±0,006)	0,458 (±0,005)	0,390 (±0,004)
	Pareto 1.5	0,461 (±0,026)	0,459 (±0,020)	0,407 (±0,022)
Adaptives REM [76]	Neg.-exp.	0,302 (±0,061)	0,315 (±0,058)	0,272 (±0,049)
	Pareto 1.5	0,443 (±0,079)	0,355 (±0,039)	0,463 (±0,070)

Die im Mittel höhere Anzahl aktiver Quellen bei adaptivem REM verstärkt bei dem hohen Angebot von 0,9 die negative Wirkung, welche die aggressiveren Algorithmen zur Wiederholung von Paketen haben können. Abb. 6.29 zeigt den Anstieg des Anteils der Paketwiederholungen am Gesamtverkehrsaufkommen. Als Folge steigt die (Blind-)Last, die die Knoten bedienen müssen. Daraus resultieren die Unterschiede der Gesamtzufriedenheit, die in Abb. 6.28 zu sehen sind.

Tabelle 2: *Mit der Objektgröße gewichtet gemittelte Gesamtzufriedenheit gemäß Gleichung (5.23) pro Zeit. Das Angebot beträgt konstant 0,9, die mittlere Objektgröße 20000 byte.*

		Reno	*New Reno*	SACK
FIFO	Neg.-exp.	6,267 (±0,064)	6,198 (±0,045)	5,954 (±0,037)
	Pareto 1.5	4,439 (±0,302)	4,404 (±0,230)	4,335 (±0,229)
Adaptives REM [76]	Neg.-exp.	3,989 (±1,742)	1,494 (±1,118)	1,351 (±0,818)
	Pareto 1.5	1,883 (±1,313)	1,064 (±0,502)	0,754 (±0,080)

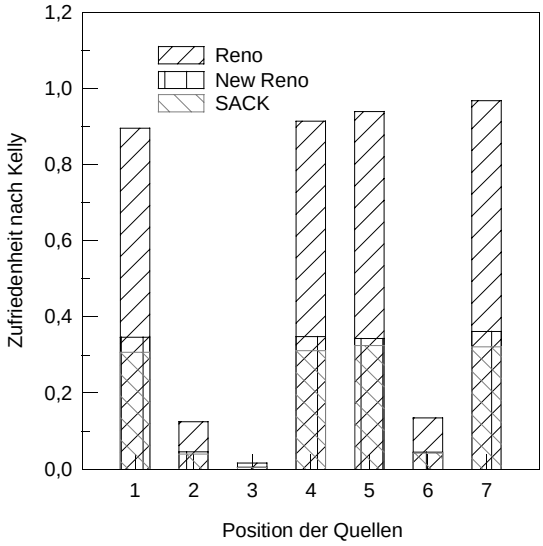

Abb. 6.28: *Gesamtzufriedenheit bei unterschiedlichen TCP-Versionen in Kombination mit adaptivem REM für die Quellen auf den verschiedenen Positionen in Abb. 6.1. Die Objektgröße ist neg.-exp. verteilt um den Erwartungswert 20000 byte. Das Angebot beträgt 0,9.*

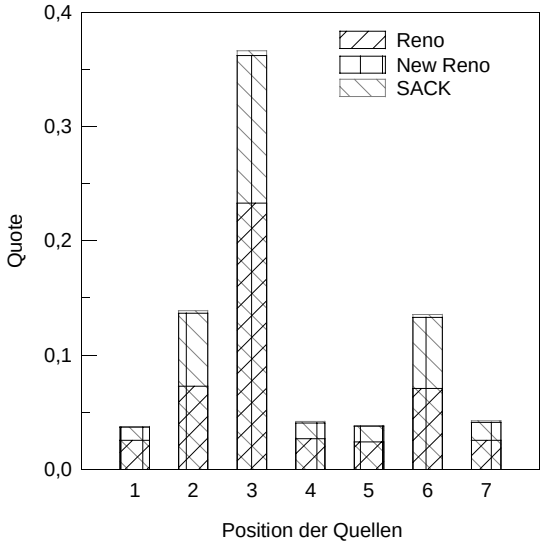

Abb. 6.29: *Paketwiederholungsquote bei unterschiedlichen TCP-Versionen in Kombination mit adaptivem REM für die Quellen auf den verschiedenen Positionen in Abb. 6.1. Die Objektgröße ist neg.-exp. verteilt um den Erwartungswert 20000 byte. Das Angebot beträgt 0,9.*

Der Einfluß von adaptivem REM auf das Systemverhalten ist in den vorangegangenen Abschnitten eingehend untersucht worden. Bei moderatem Angebot kann mit adaptivem REM durchaus eine höhere Gesamtzufriedenheit erzielt werden. Bei hohem Angebot dagegen wirkt sich das vorzeitige Markieren von Paketen so aus, daß die Anzahl gleichzeitig aktiver Quellen zunimmt und die Regelung außer Kraft gesetzt wird. Dies ist ein zu Systemen mit Verbindungsannahmesteuerung gegenläufiges Verhalten und eigentlich unerwünscht.

Bei adaptivem RED ist dieser Effekt noch stärker ausgeprägt. Ohne die Möglichkeit des Markierens von Paketen ist natürlich auch der Anteil von Paketwiederholungen am Gesamtverkehr deutlich höher, vgl. Abb. 6.30. Wie Abb. 6.31 zeigt, werden dann auch die Netzressourcen weniger fair zugeteilt. Beide bleiben hier deutlich gegenüber FIFO ohne Puffermanagement zurück. Bei RED führen äußerst hohe Verluste dazu, daß von Knoten zu Knoten immer weniger Last zu bewältigen ist. Besonders die Quellen auf den Positionen 4, 5 und 7 profitieren davon und gleichen den Ausfall der anderen Quellen mehr als aus. Das positive Bild, das Tabelle 3 für adaptives RED besonders bei der Übertragung kleinerer Objekte zeichnet, täuscht also über diesen sicherlich unerwünschten Umstand hinweg. Die Gesamtzufriedenheit gemäß Formel (5.23) reicht für die Leistungsbewertung nicht aus.

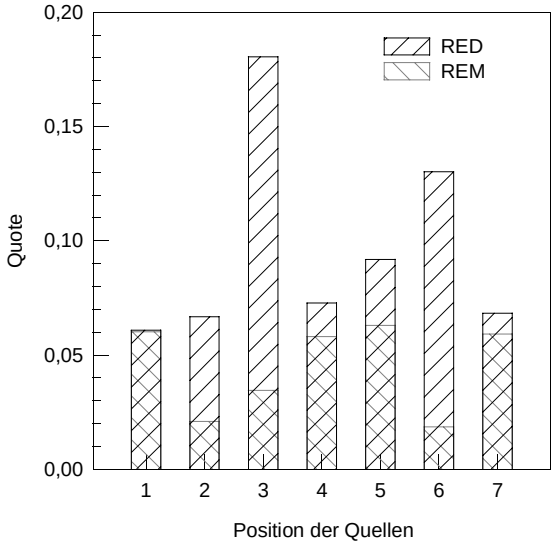

Abb. 6.30: *Paketwiederholungsquote bei adaptivem RED und REM in Abhängigkeit von der Position der Quellen in Abb. 6.1. Die Objektgröße ist neg.-exp. verteilt um den Erwartungswert 200000 byte. Das Angebot beträgt 0,9.*

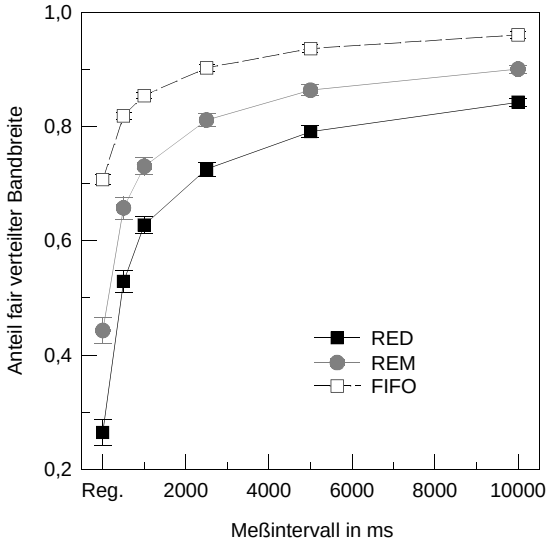

Abb. 6.31: *Anteil fair verteilter Bandbreite in Phasen ohne Verbindungsauf- und -abbau (Reg.) und Meßintervallen fester Länge für die Quellen auf Position 1 in Abb. 6.1 in Abhängigkeit vom Puffermanagement. Die Objektgröße ist neg.-exp. verteilt um den Erwartungswert 200000 byte. Das Angebot beträgt 0,9.*

Tabelle 3: *Mit der Objektgröße gewichtet gemittelte Gesamtzufriedenheit gemäß Gleichung (5.23) pro Zeit in Abhängigkeit vom Puffermanagement für TCP New Reno. Das Angebot beträgt konstant 0,9.*

		Mittlere Objektgröße	
		200000 byte	20000 byte
FIFO	Neg.-exp.	0,458 (±0,005)	6,198 (±0,045)
	Pareto 1.5	0,459 (±0,020)	4,404 (±0,230)
Adaptives RED [76]	Neg.-exp.	0,445 (±0,016)	4,379 (±0,139)
	Pareto 1.5	0,428 (±0,040)	3,760 (±0,283)
Adaptives REM [76]	Neg.-exp.	0,315 (±0,058)	1,494 (±1,118)
	Pareto 1.5	0,355 (±0,039)	1,064 (±0,502)

7 Zusammenfassung und Ausblick

Im Rahmen der vorliegenden Arbeit ist ein skalierbares Konzept für das Verkehrsmanagement in paketvermittelnden Netzen entwickelt und untersucht worden, deren Knoten sowohl virtuelle Verbindungen und Verbindungsaggregate als auch verbindungslose Paketvermittlung unterstützen.

Das Einrichten von virtuellen Verbindungen ist besonders dann gerechtfertigt, wenn die Kommunikationsbeziehung zwischen den Endpunkten der Verbindung vergleichsweise lange aufrechterhalten wird, wenn sich die erforderliche Dienstgüte und das Datenaufkommen mit Hilfe eines Verkehrsdeskriptors und Dienstgüteparametern darstellen lassen oder ihr Datenaufkommen so hoch ist, daß die Vorteile von verbindungsbezogenem Verkehrsmanagement wirklich zum Tragen kommen.

Um dabei den Aufwand für die Verbindungssteuerung zu reduzieren, ist ein Verkehrssteuerungskonzept entwickelt, prototypisch implementiert und analytisch, simulativ und experimentell untersucht worden. Dieses Konzept kombiniert die Zuordnung von Verbindungen zu einem Verbindungsaggregat, die Berechnung des Ressourcenbedarfs des Aggregates, die Bedieneinheit am Eingang des Aggregates und dynamisches Bandbreitemanagement. Aus der Literatur bekannte Verfahren zur Verbindungsannahmesteuerung werden auf die Berechnung des Ressourcenbedarfs von Verbindungsaggregaten angewandt und um Verfahren erweitert, die den Berechnungsaufwand reduzieren. Bei RPQ [133] genügt es so noch, zur Neuberechnung des Ressourcenbedarfs mit dem deterministischen *Network Calculus* [49] bei Änderung der Zusammensetzung eines Aggregates lediglich den Beitrag der betroffenen Verbindung zu addieren oder zu subtrahieren. Beim statistischen Multiplexen ändert sich dagegen immer der Arbeitspunkt, also die Parameter s und (wenn Wartespeicher berücksichtigt wird) t im Ausdruck (3.79) für die effektive Bandbreite nach Kelly, so daß die effektive Bandbreite aller Verbindungen von der Änderung des Aggregates betroffen ist. Die Untersuchungen zeigen jedoch, daß sich der Arbeitspunkt um so weniger ändert, je größer das Aggregat ist. Aus dieser Beobachtung leitet sich das in Kapitel 3 vorgestellte Verfahren für den praktischen Einsatz des *Many Sources Asymptotic* [46, 30] mit periodischen Ein-Aus-Quellen ab. Hierbei wird der für die genaue Berechnung des Ressourcenbedarfs nötige Rechenaufwand begrenzt, indem die Berechnung auf mehrere Änderungen der Zusammensetzung des Aggregates

verteilt und jedes Mal die Berechnung präzisiert wird. Messungen am Prototypen zeigen, daß der Rechenaufwand für ein solches Verfahren zwar deutlich höher als bei den einfacheren deterministischen oder den pufferlosen statistischen Modellen ist, aber durchaus in praktisch beherrschbaren Größenordnungen liegt.

Das Zusammenfassen von Verbindungen zu Aggregaten reduziert den Aufwand für das Speichern von und den Zugriff auf Zustandsinformationen sowie für die Berechnung des Ressourcenbedarfs in den Knoten des Netzes, welche die Verbindungen als Teil eines Aggregates durchlaufen. Dynamisches Bandbreitemanagement sorgt ferner dafür, daß nicht jeder Verbindungsauf- und -abbau oder die Änderung einzelner Verbindungen auch unmittelbar entsprechende Steuerungsaktivitäten auf der Ebene des Aggregates auslösen. Bereits der in dieser Arbeit für die Experimente am Aggregationsknoten entwickelte sehr einfache Algorithmus zeigt die Stärken des auf Aggregation beruhenden verbindungsorientierten Verkehrsmanagements in Weitverkehrsnetzen mit vielen Verbindungen.

Wenn keine der o. g. drei Voraussetzungen für das Einrichten virtueller Verbindungen gegeben ist, ist auch in einem aggregationsfähigen Netz die verbindungslose Vermittlung von Paketen vorzuziehen. Das trifft heute vor allem auf durch TCP geregelte elastische Datenströme zu. Aus diesem Grunde ist auch das Potential der unterschiedlichen, zum Teil in den Endsystemen, zum Teil in den Netzelementen ansetzenden Vorschläge zur Steuerung und Regelung elastischen Verkehrs zur Ergänzung des skalierbaren Verkehrsmanagements eingehend untersucht worden.

Ein nach dem Kenntnisstand des Autors neues Ergebnis dieser Untersuchungen ist, daß bei hohem Verkehrsangebot (im Sinne der in Kapitel 6 gegebenen Definition) die Zufriedenheit nach Kelly [115], die gemäß eines vereinfachten Modells der Überlastregelung von TCP in *Congestion Avoidance* Durchsatz und Umlaufzeit zu einem Leistungsmaß kombiniert, so drastisch einbricht. Dies steht in Widerspruch zu den Anforderungen an eine Überlastregelung. Weiter fällt auf, welch hohen Anteil durch *Retransmission Timeouts* ausgelöste Paketwiederholungen an den Paketwiederholungen insgesamt haben, insbesondere bei der Übertragung kurzer Objekte und beim Einsatz von adaptivem RED oder REM [76] in Kombination mit *Explicit Congestion Notification* (ECN) [166]. Verbindungsabbrüche ungeduldiger Nutzer können bei anhaltender Überlast in den Netzknoten durchaus eine ähnliche Rolle wie die Verbindungsannahmesteuerung für virtuelle Verbindungen spielen. Allerdings sind beim Einsatz von TCP *Reno* [2] und davon abgeleiteten Varianten der Überlastregelung von TCP die Kommunikationsbeziehungen über mehrere überlastete Knoten besonders von Abbrüchen betroffen.

Die mindestens in dieser Kombination erstmals eingesetzten Simulationstechniken zur Messung der Zufriedenheit nach Kelly für alle oder einzelne Verbindungen, der Anzahl gleichzeitig aktiver

Verbindungen, die Beobachtung der die Paketwiederholungen auslösenden Algorithmen, die Messung der Fairneß unter gleichartigen Verbindungen in Abhängigkeit des Beobachtungsintervalls und Statistiken zur Charakterisierung von Verkehrsströmen zeigen, wie sich in Abhängigkeit dieser Faktoren das Verhalten des Systems grundsätzlich ändert.

Die vorliegenden Untersuchungen bestätigen, daß sowohl verbindungsorientierte als auch verbindungslose Netzdienste in einem diensteintegrierenden IP-Netz effizient unterstützt werden können. Dennoch stellt sich angesichts der dramatisch wachsenden Kapazität der Transportnetze trotz wachsender Bandbreite der Zugangsnetze und bis auf weiteres exponentiell steigendem Verkehrsaufkommen die Frage, ob Probleme bei der Bereitstellung von Dienstgüteanforderungen statt durch Reservierung durch Überdimensionierung und bzw. oder durch sehr einfache Prioritätsmechanismen vermieden werden können.

Breslau und Shenker gehen dieser Frage nach, indem sie wie Kelly [115] die Zufriedenheit des Nutzers als Funktion der Rate modellieren, um so den Wert darzustellen, den ein Nutzer aus einer Anwendung in Abhängigkeit von der verfügbaren Rate schöpft [36]. Die Untersuchung bezieht drei Typen von Funktionen ein. Das sind zum einen Funktionen, die in der Nähe des Koordinatenursprungs, also für kleine Raten, konvex sind, bei einer bestimmten Rate einen Wendepunkt haben und danach konkav weiterlaufen. Solche Funktionen sollen Anwendungen repräsentieren, die bei geringen Raten keine akzeptable Dienstgüte erzielen, sich aber jenseits der Wendestelle relativ gut anpassen und von da an sinnvoll eingesetzt werden können, mit streng monoton, aber immer langsamer steigender Dienstgüte. Diese Anwendungen werden adaptiv genannt. Zum anderen werden als rigide bezeichnete Anwendungen betrachtet, die bei Raten unterhalb einer Schwelle nutzlos sind und darüber gleich ihren maximalen Wert erreichen. Dagegen kann die Zufriedenheit von Nutzern elastischer Anwendungen in Abhängigkeit von der Rate durch durchgehend konvexe Funktionen nachgebildet werden. Diese Anwendungen profitieren aber in keinem Fall von Reservierung. Wie erwartet ist Reservierung um so gewinnbringender, je rigider die Anwendungen sind und je kostengünstiger der zusätzliche Aufwand für die notwendigen Verkehrssteuerungsmechanismen ist. Je günstiger Bandbreite wird, desto kleiner ist der Vorteil, aber Reservierung kann selbst dann noch günstiger sein, wenn die Kosten für die Bandbreite gegen null tendieren.

Da heute noch niemand mit Gewißheit sagen kann, welche Typen von Anwendungen in zukünftigen Breitbandnetzen dominieren werden, bleibt die Antwort auf die Frage, welche Rolle Überdimensionierung, Prioritätsmechanismen und Reservierung in Zukunft spielen werden, unbeantwortet. Wie in anderen Bereichen auch wird letzten Endes die Nachfrage des Kunden entscheiden.

Anhang A: Momentenerzeugende Funktion periodischer Ein-Aus-Quellen

Zur Berechnung der *Large Deviation Rate Function* (3.87) im durch die Formeln (3.72)-(3.76) beschriebenen *Many Sources Asymptotic* für periodische Ein-Aus-Quellen wird die momentenerzeugende Funktion $\mathrm{E}\{\mathrm{e}^{sA(0,t)}\}$ solcher Quellen benötigt. In [85] wird lediglich ein entsprechender Ansatz und in [180] ein Ausdruck angegeben, der nicht ganz korrekt ist. Außerdem betrachten beide Arbeiten nur den Fall, daß die Quelle während ihrer Periodendauer T_P länger pausiert als sie sendet: $T_{AUS} \geq T_{EIN}$.

Da bei der Überlagerung periodischer Ein-Aus-Quellen die Phasenlage der Quellen zu einem Bezugspunkt die einzige Zufallsgröße ist, bietet es sich an, im Gegensatz zur oben erwähnten Literatur diese Phasenlage zum Ausgangspunkt der Berechnung zu machen. Im Einklang mit [85] und [180] wird zur Vereinfachung ein Flüssigkeitsmodell zugrunde gelegt: Die Ein-Aus-Quellen erzeugen während einer Ein-Phase nicht etwa Pakete in konstanten Abständen, sondern einen kontinuierlichen Datenstrom (Flüssigkeitsstrom) mit konstanter Rate p.

Im folgenden wird die in einem Intervall $[0,t)$ ankommende Datenmenge $A(0,t)$ abhängig von der Phasenlage der Quelle zu diesem Intervall berechnet.

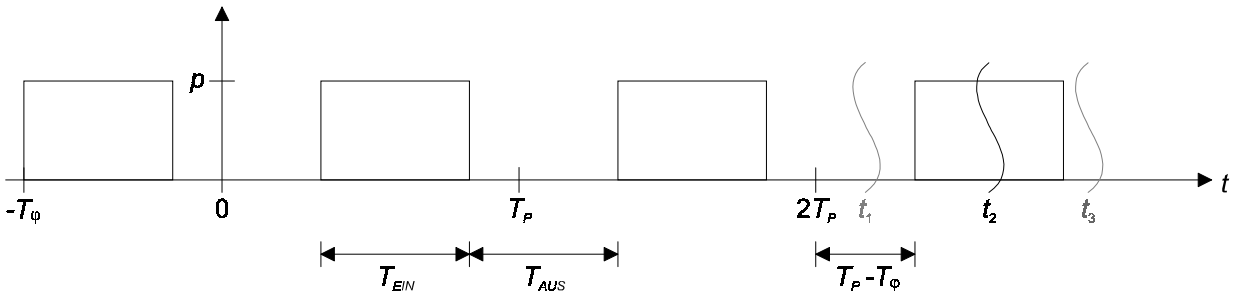

Abb. A.1: *Fall 1 ($T_{\varphi} \geq T_{EIN}$)*

Dazu werden zunächst zwei Fälle unterschieden. Im Fall 1, vgl. Abb A.1, fällt der Zeitpunkt 0 in eine Inaktivitätsphase der Quelle. $-T_\varphi$, der Zeitpunkt des Beginns der letzten Aktivitätsphase, liegt schon länger als die konstante Dauer T_{EIN} einer Aktivitätsphase zurück. Dagegen ist im Fall 2 die Quelle zum Zeitpunkt 0 aktiv. Abb. A.2 stellt diesen Fall dar.

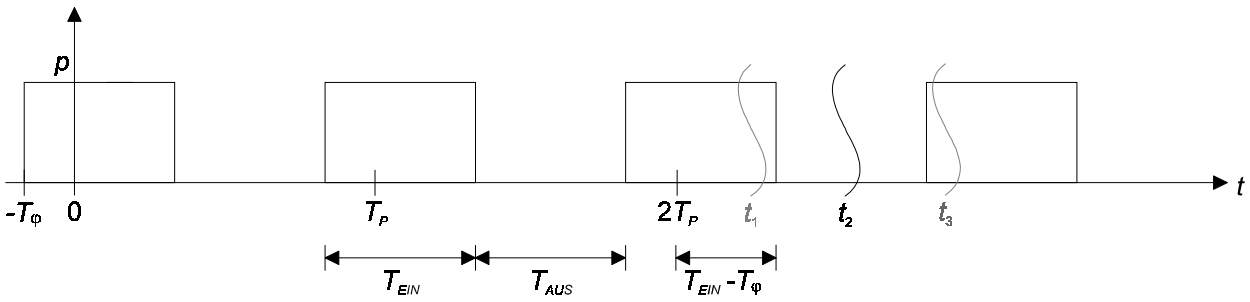

Abb. A.2: *Fall 2 ($T_\varphi < T_{EIN}$)*

Während der im Intervall $[0,t)$ vollständig enthaltenen Perioden kommt jeweils die Datenmenge $r\,T_P = p\,T_{EIN}$ an. Abhängig von der Phasenlage ist lediglich der Anteil, der auf die Zeit nach der letzten vollständig in $[0,t)$ enthaltenen Periode der Quelle entfällt. Um diesen Anteil zu berechnen, sind sowohl in Abb. A.1 als auch in Abb. A.2 drei Fälle für t skizziert: $t=t_1, t=t_2, t=t_3$. Entsprechend unterschiedlich ist die im Intervall $[0,t)$ ankommende Datenmenge, nämlich für

Fall 1.1: $t_1 - \left\lfloor \frac{t_1}{T_P} \right\rfloor T_P < T_P - T_\varphi \;\Leftrightarrow\; T_\varphi < T_P - (t_1 - \left\lfloor \frac{t_1}{T_P} \right\rfloor T_P)$

$$A(0,t_1) = \left\lfloor \frac{t_1}{T_P} \right\rfloor T_{EIN}\, p \qquad \text{(A.1)}$$

Fall 1.2: $T_P - T_\varphi \le t_2 - \left\lfloor \frac{t_2}{T_P} \right\rfloor T_P < T_P - T_\varphi + T_{EIN}$

$$\Leftrightarrow\; T_P - (t_2 - \left\lfloor \frac{t_2}{T_P} \right\rfloor T_P) \le T_\varphi < T_P + T_{EIN} - (t_2 - \left\lfloor \frac{t_2}{T_P} \right\rfloor T_P)$$

$$A(0,t_2) = \left\lfloor \frac{t_2}{T_P} \right\rfloor T_{EIN}\, p + \left(t_2 - \left\lfloor \frac{t_2}{T_P} \right\rfloor T_P - (T_P - T_\varphi) \right) p \qquad \text{(A.2)}$$

Fall 1.3: $t_3-\left\lfloor \frac{t_3}{T_P} \right\rfloor T_P \geq T_P - T_\varphi + T_{EIN} \;\Leftrightarrow\; T_\varphi \geq T_P + T_{EIN} - (t_3 - \left\lfloor \frac{t_3}{T_P} \right\rfloor T_P)$

$$A(0,t_3) = \left\lfloor \frac{t_3}{T_P} \right\rfloor T_{EIN}\; p + T_{EIN}\; p \tag{A.3}$$

Fall 2.1: $t_1-\left\lfloor \frac{t_1}{T_P} \right\rfloor T_P < T_{EIN} - T_\varphi \;\Leftrightarrow\; T_\varphi < T_{EIN} - (t_1 - \left\lfloor \frac{t_1}{T_P} \right\rfloor T_P)$

$$A(0,t_1) = \left\lfloor \frac{t_1}{T_P} \right\rfloor T_{EIN}\; p + (t_1 - \left\lfloor \frac{t_1}{T_P} \right\rfloor)\, p \tag{A.4}$$

Fall 2.2:
$$T_{EIN} - T_\varphi \leq t_2 - \left\lfloor \frac{t_2}{T_P} \right\rfloor T_P < T_{EIN} - T_\varphi + T_{OFF}$$
$$\Leftrightarrow\; T_{EIN} - (t_2 - \left\lfloor \frac{t_2}{T_P} \right\rfloor T_P) \leq T_\varphi < T_{EIN} + T_{OFF} - (t_2 - \left\lfloor \frac{t_2}{T_P} \right\rfloor T_P)$$

$$A(0,t_2) = \left\lfloor \frac{t_2}{T_P} \right\rfloor T_{EIN}\; p + (T_{EIN} - T_\varphi)\, p \tag{A.5}$$

Fall 2.3: $t_3-\left\lfloor \frac{t_3}{T_P} \right\rfloor T_P \geq T_{EIN} - T_\varphi + T_{OFF} \;\Leftrightarrow\; T_\varphi \geq T_{EIN} + T_{OFF} - (t_3 - \left\lfloor \frac{t_3}{T_P} \right\rfloor T_P)$

$$A(0,t_3) = \left\lfloor \frac{t_3}{T_P} \right\rfloor T_{EIN}\; p + (t_3 - \left\lfloor \frac{t_3}{T_P} \right\rfloor T_P - T_{OFF})\, p \tag{A.6}$$

Die in den Gleichungen A.1-A.6 angegebenen Ausdrücke für das gesuchte $A(0,t)$, die jeweils unter bestimmten Bedingungen für die Phasenlage T_φ der Quelle bezüglich des Nullpunktes gelten, müssen nun noch entsprechend der Wahrscheinlichkeit des Auftretens der unterschiedlichen Bedingungen gewichtet zu einem endgültigen Ergebnis für $A(0,t)$ summiert werden. Dabei ist zu berücksichtigen, daß die Fälle 1.1-1.3 unter der Bedingung $T_\varphi \geq T_{EIN}$ und die Fälle 2.1-2.2 unter der Bedinung $T_\varphi < T_{EIN}$ zustande gekommen sind. Zur Vereinfachung der Schreibweise wird im folgenden $T_{REST} := t - \lfloor \frac{t}{T_P} \rfloor T_P$ gesetzt. Dann erhält man folgende Wahrscheinlichkeiten:

Fall 1.1:

$$\mathrm{P}\{T_\varphi < T_P - T_{REST} | T_\varphi \geq T_{EIN}\} = \frac{\mathrm{P}\{T_\varphi < T_P - T_{REST} \wedge T_\varphi \geq T_{EIN}\}}{\mathrm{P}\{T_\varphi \geq T_{EIN}\}}$$
$$= \begin{cases} \dfrac{T_P - T_{EIN} - T_{REST}}{T_{AUS}}, \text{wenn} & T_P - T_{REST} > T_{EIN} \\ 0 & \text{sonst} \end{cases} \qquad \text{(A.7)}$$

Fall 1.2:

$$\mathrm{P}\{T_P - T_{REST} \leq T_\varphi < T_P + T_{EIN} - T_{REST} | T_\varphi \geq T_{EIN}\}$$
$$= \frac{\mathrm{P}\{T_P - T_{REST} \leq T_\varphi < T_P + T_{EIN} - T_{REST} \wedge T_\varphi \geq T_{EIN}\}}{\mathrm{P}\{T_\varphi \geq T_{EIN}\}}$$
$$= \begin{cases} \dfrac{\mathrm{P}\{T_{EIN} \leq T_\varphi < T_P + T_{EIN} - T_{REST}\}}{\mathrm{P}\{T_\varphi \geq T_{EIN}\}} & , \text{wenn} \quad T_{EIN} \geq T_P - T_{REST} \wedge T_{EIN} \leq T_{REST} \\ \dfrac{\mathrm{P}\{T_P - T_{REST} \leq T_\varphi < T_P + T_{EIN} - T_{REST}\}}{\mathrm{P}\{T_\varphi \geq T_{EIN}\}} & , \text{wenn} \quad T_{EIN} < T_P - T_{REST} \wedge T_{EIN} \leq T_{REST} \\ \dfrac{\mathrm{P}\{T_{EIN} \leq T_\varphi\}}{\mathrm{P}\{T_\varphi \geq T_{EIN}\}} & , \text{wenn} \quad T_{EIN} \geq T_P - T_{REST} \wedge T_{EIN} > T_{REST} \\ \dfrac{\mathrm{P}\{T_P - T_{REST} \leq T_\varphi < T_P\}}{\mathrm{P}\{T_\varphi \geq T_{EIN}\}} & , \text{wenn} \quad T_{EIN} < T_P - T_{REST} \wedge T_{EIN} > T_{REST} \end{cases} \qquad \text{(A.8)}$$
$$= \begin{cases} \dfrac{T_P - T_{REST}}{T_{AUS}} & , \text{wenn} \quad T_{EIN} \geq T_P - T_{REST} \wedge T_{EIN} \leq T_{REST} \\ \dfrac{T_{EIN}}{T_{AUS}} & , \text{wenn} \quad T_{EIN} < T_P - T_{REST} \wedge T_{EIN} \leq T_{REST} \\ 1 & , \text{wenn} \quad T_{EIN} \geq T_P - T_{REST} \wedge T_{EIN} > T_{REST} \\ \dfrac{T_{REST}}{T_{AUS}} & , \text{wenn} \quad T_{EIN} < T_P - T_{REST} \wedge T_{EIN} > T_{REST} \end{cases}$$

Fall 1.3:

$$\mathrm{P}\{T_P + T_{EIN} - T_{REST} \leq T_\varphi < T_P | T_\varphi \geq T_{EIN}\} = \frac{\mathrm{P}\{T_\varphi \geq T_P + T_{EIN} - T_{REST}\}}{\mathrm{P}\{T_\varphi \geq T_{EIN}\}}$$
$$= \begin{cases} 0 & , \text{wenn} \quad T_{EIN} > T_{REST} \\ \dfrac{-T_{EIN} + T_{REST}}{T_{AUS}} & \text{sonst} \end{cases} \qquad \text{(A.9)}$$

Fall 2.1:

$$\mathrm{P}\{T_\varphi < T_{EIN} - T_{REST} | T_\varphi < T_{EIN}\} = \frac{\mathrm{P}\{T_\varphi < T_{EIN} - T_{REST}\}}{\mathrm{P}\{T_\varphi < T_{EIN}\}}$$
$$= \begin{cases} \frac{T_{EIN} - T_{REST}}{T_{EIN}}, \text{wenn} & T_{EIN} > T_{REST} \\ 0 & \text{sonst} \end{cases} \quad \text{(A.10)}$$

Fall 2.2:

$$\mathrm{P}\{T_{EIN} - T_{REST} \le T_\varphi < T_P - T_{REST} | T_\varphi < T_{EIN}\}$$
$$= \begin{cases} \frac{\mathrm{P}\{T_{EIN} - T_{REST} \le T_\varphi < T_{EIN}\}}{\mathrm{P}\{T_\varphi < T_{EIN}\}} & \text{, wenn } T_{EIN} \le T_P - T_{REST} \wedge T_{EIN} \ge T_{REST} \\ \frac{\mathrm{P}\{T_{EIN} - T_{REST} \le T_\varphi < T_P - T_{REST}\}}{\mathrm{P}\{T_\varphi < T_{EIN}\}} & \text{, wenn } T_{EIN} > T_P - T_{REST} \wedge T_{EIN} \ge T_{REST} \\ \frac{\mathrm{P}\{T_\varphi < T_{EIN}\}}{\mathrm{P}\{T_\varphi < T_{EIN}\}} & \text{, wenn } T_{EIN} \le T_P - T_{REST} \wedge T_{EIN} < T_{REST} \\ \frac{\mathrm{P}\{T_\varphi < T_P - T_{REST}\}}{\mathrm{P}\{T_\varphi < T_{EIN}\}} & \text{, wenn } T_{EIN} > T_P - T_{REST} \wedge T_{EIN} < T_{REST} \end{cases} \quad \text{(A.11)}$$
$$= \begin{cases} \frac{T_{REST}}{T_{EIN}} & \text{, wenn } T_{EIN} \le T_P - T_{REST} \wedge T_{EIN} \ge T_{REST} \\ \frac{T_{AUS}}{T_{EIN}} & \text{, wenn } T_{EIN} > T_P - T_{REST} \wedge T_{EIN} \ge T_{REST} \\ 1 & \text{, wenn } T_{EIN} \le T_P - T_{REST} \wedge T_{EIN} < T_{REST} \\ \frac{T_P - T_{REST}}{T_{EIN}} & \text{, wenn } T_{EIN} > T_P - T_{REST} \wedge T_{EIN} < T_{REST} \end{cases}$$

Fall 2.3:

$$\mathrm{P}\{T_P - T_{REST} \le T_\varphi < T_P | T_\varphi < T_{EIN}\} = \frac{\mathrm{P}\{T_P - T_{REST} \le T_\varphi < T_{EIN}\}}{\mathrm{P}\{T_\varphi < T_{EIN}\}}$$
$$= \begin{cases} \frac{T_{EIN} - T_P + T_{REST}}{T_{EIN}}, \text{wenn} & T_{EIN} > T_P - T_{REST} \\ 0 & \text{sonst} \end{cases} \quad \text{(A.12)}$$

Mit diesen Formel läßt sich die momentenerzeugende Funktion folgendermaßen berechnen:

$$\mathrm{E}\{e^{sA(0,t)}\} = \mathrm{P}\{T_\varphi \ge T_{EIN}\}\,\mathrm{E}\{e^{sA(0,t)} | T_\varphi \ge T_{EIN}\} + \mathrm{P}\{T_\varphi < T_{EIN}\}\,\mathrm{E}\{e^{sA(0,t)} | T_\varphi < T_{EIN}\} \quad \text{(A.13)}$$

Zur Berechnung der bedingten Erwartungswerte $\mathrm{E}\{e^{sA(0,t)} | T_\varphi \ge T_{EIN}\}$ und $\mathrm{E}\{e^{sA(0,t)} | T_\varphi < T_{EIN}\}$ in (A.13) sind für $T_\varphi \ge T_{EIN}$ die Fälle (A.1)-(A.3), für $T_\varphi < T_{EIN}$ die Fälle (A.4)-(A.6) entspre-

chend der Wahrscheinlichkeit ihres Auftretens zu berücksichtigen. Dabei ist zu beachten, daß in (A.2) ebenso wie in (A.5) die in einem Intervall $[0,t)$ ankommende Datenmenge $A(0,t)$ von der Phasenlage T_φ abhängig sind. Insofern sind die in (A.8) und (A.11) berechneten Wahrscheinlichkeiten nicht von unmittelbarem Nutzen. Stattdessen muß innerhalb der dort angegebenen Schranken über T_φ integriert werden. Die Wahrscheinlickeiten (A.7), (A.9), (A.10) und (A.12) können dagegen direkt eingesetzt werden. Damit die zahlreichen Fälle nicht alle einzeln angegeben werden müssen, werden sie im folgenden durch die Symbole $P_{1.1}$, $P_{1.3}$, $P_{2.1}$ und $P_{2.3}$ ersetzt. In dieser vereinfachten Schreibweise ist die momentenerzeugende Funktion dann

$$
\begin{aligned}
\mathrm{E}\{\exp(s\,A(0,t))\} = \exp\left(s \left\lfloor \frac{t}{T_P} \right\rfloor T_{EIN}\; p \right) \cdot \\
\left(\frac{T_{AUS}}{T_P} P_{1.1} \;+ \right. \\
\int_{\max\{T_P - T_{REST},\, T_{EIN}\}}^{\min\{T_P + T_{EIN} - T_{REST},\, T_P\}} \exp(s\,(T_{REST} - (T_P - T_\varphi))\,p) \frac{1}{T_P} \mathrm{d}\,T_\varphi \;+ \\
\frac{T_{AUS}}{T_P} P_{1.3} \exp(s\,T_{EIN}\; p) \;+ \\
\frac{T_{EIN}}{T_P} P_{2.1} \exp(s\,T_{REST}\; p) \;+ \\
\int_{\max\{0,\, T_{EIN} - T_{REST}\}}^{\min\{T_P - T_{REST},\, T_{EIN}\}} \exp(s\,(T_{EIN} - T_\varphi)\,p) \frac{1}{T_P} \mathrm{d}\,T_\varphi \;+ \\
\left. \frac{T_{EIN}}{T_P} P_{2.3} \exp(s\,(T_{REST} - T_{AUS})\,p) \right)
\end{aligned}
\qquad \text{(A.14)}
$$

Diese Funktion ist nicht differenzierbar nach t.

Anhang B: Liste der Parameter der TCP-Endsysteme

Tabelle B.1: *Aufstellung aller Parameter in den TCP-Endsystemen, die während der Leistungsuntersuchung in Kapitel 6 nicht variiert werden.*

Anwendung auf der Sendeseite	
Maximale Länge der zur Übertragung übergebenen Datenböcke	16384 byte
Socket-Schnittstelle auf der Sendeseite	
Konstanter Anteil der Zeit für das Kopieren von Daten der Anwendung [151]	0,3 ms
Zeit für das Verarbeiten eines 1024 byte großen Speicherblocks [151]	0,1 ms
Zeit für das Verarbeiten eines 128 byte großen Speicherblocks [151]	0,1 ms
Größe des Sendepuffers	65536 byte
TCP-Sender	
Maximales Sendefenster	65536 byte
Initiales *Congestion Window*	2920 byte
Maximum Segment Size	1460 byte
Inititaler *Slow Start Threshold*	65536 byte
Auflösung der Zeitgeber (*Timer Granularity*)	100 ms
RTO bis zur ersten Messung der Umlaufzeit	100 ms
Nagle-Bedingungen der *Silly Window Avoidance* [31]	nein
Limited Transmit [3]	ja
Untere Schichten der Sendeseite (vgl. Abb. 5.6)	
Warteschlangen	unbegrenzt
Bedienzeit für Interrupt-, Betriebssystemaufrufe und *Bottom-Half* konstant	0,5 ms
Bandbreite des Netzzugangs (für IP-Pakete)	10 Mbit/s
Untere Schichten der Empfangsseite (vgl. Abb. 5.7)	
Warteschlangen	unbegrenzt
Bedienzeit für Interrupt- und *Bottom-Half* konstant	0,3 ms

TCP-Empfänger	
Maximum Segment Size	1460 byte
Delayed Acknowledgement [31]	ja
Maximale Verzögerung einer Quittierung (*Delayed Acknowledgement*)	100 ms
Maximales (und initiales) Empfangsfenster	65536 byte
Socket-Schnittstelle auf der Empfangsseite	
Größe des Empfangspuffers	65536 byte
Konstante Bedienzeit für die Kopieroperation in den Lesepuffer der Anwendung	0,5 ms
Anwendung auf der Empfangsseite	
Maximale Länge des zum Empfang bereitgestellten Lesepuffers	16384 byte

Literaturverzeichnis

[1] E. Aarstad, L. Burgstahler and M. Lorang, „Description of the Flow-to-VC Mapping Control Module in DIANA´s RSVP over ATM Architecture", *in QoS Summit´99*, Paris, France, November 16-19, 1999.

[2] M. Allman, V. Paxson and W. Stevens, *TCP Congestion Control*. IETF Request for Comments, RFC 2581, April 1999.

[3] M. Allman, H. Balakrishnan and S. Floyd, *Enhancing TCP's Loss Recovery Using Limited Transmit*. IETF Request for Comments, RFC 3042, January 2001.

[4] W. Almesberger, „ATM on Linux - The 3rd year", in *4. Internationaler Linux-Kongreß*, Würzburg, May 1997.

[5] W. Almesberger, „Linux Network Traffic Control - Implementation Overview", in *Proceedings of the 5th Annual Linux Expo*, Raleigh, NC, May 1999, pp. 153-164.

[6] D. Anick, D. Mitra and M. M. Sondhi, „Stochastic Theory of a Data-Handling System with Multiple Sources", in *The Bell System Technical Journal*, Vol. 61, No. 8, October 1982, pp. 1871-1894.

[7] G. Armitage, „MPLS: The Magic Behind the Myths", in *IEEE Communications Magazine*, Vol. 38, No. 1, January 2000, pp. 124-131.

[8] Å. Arvidsson and P. Karlsson, „On Traffic Models for TCP/IP", in *Teletraffic Engineering in a Competitive World - Proceedings of the International Teletraffic Congress (ITC-16)*, Edinburgh, U. K., June 7-11, 1999, pp. 457-466.

[9] The ATM Forum Technical Committee, *LAN Emulation over ATM Version 1.0*. ATM Forum, af-lane-0021.000, January 1995.

[10] The ATM Forum Technical Committee, *ATM User-Network Interface (UNI) Signalling Specification Version 4.0. ATM Forum*, af-sig-0061.000, July 1996.

[11] The ATM Forum Technical Committee, *Traffic Management Specification Version 4.1*. ATM Forum, af-tm-0121.000, March 1999.

[12] The ATM Forum Technical Committee, *Multiprotocol over ATM Version 1.1*. ATM Forum, af-mpoa-0114.000, May 1999.

[13] The ATM Forum Technical Committee, *Addendum to TM 4.1: Differentiated UBR*. ATM Forum, af-tm-0149.000, July 2000.

[14] The ATM Forum Technical Committee, *Modification of Traffic Descriptor for an Active Connection, Addendum to UNI 4.0, PNNI 1.0, and AINI*. ATM Forum, af-cs-0148.000, July 2000.

[15] S. Athuraliya and S. H. Low, „REM: Active Queue Management“, in *IEEE Network*, Vol. 15, No. 3, May/June 2001, pp. 48-53.

[16] D. O. Awduche, L. Berger, D.-H. Gan, T. Li, G. Swallow and V. Srinivasan, *RSVP-TE: Extensions to RSVP for LSP Tunnels*. Work in Progress, IETF Multiprotocol Label Switching Working *Group*, <draft-ietf-mpls-rsvp-lsp-tunnel-05.txt>, February 2000.

[17] F. Baker, C. Iturralde, F. Le Faucheur and B. Davie, *Aggregation of RSVP for IPv4 and IPv6 Reservation*. Work in Progress, IETF Integrated Services over Specific Link Layers Working Group, <draft-ietf-issll-rsvp-aggr-02.txt>, March 2000.

[18] M. Beck, H. Böhme, M. Dziadzka, U. Kunitz, R. Magnus und D. Verworner, *Linux-Kernel-Programmierung - Algorithmen und Strukturen der Version 2.0*. Bonn, Addison-Wesley, 4. Auflage, 1997.

[19] J. Beran, *Statistics for long-memory processes*. New York: Chapman & Hall, 1994.

[20] L. Berger, *RSVP over ATM Implementation Guidelines*. IETF Request for Comments, RFC 2379, August 1998.

[21] L. Berger, *RSVP over ATM Implementation Requirements*. IETF Request for Comments, RFC 2380, August 1998.

[22] S. Berson and S. Vincent, *Aggregation of Internet Integrated Services State*. Work in Progress, IETF Integrated Services Working Group, <draft-berson-classy-approach-01.ps>, November 1997.

[23] P. Billingsley, *Probability and Measure - Second Edition*. New York: Wiley, 1986.

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

[25] S. Bodamer, „A New Scheduling Mechanism to Provide Relative Differentiation for Real-Time IP Traffic“, in *Proceedings of the IEEE Globecom´00*, San Francisco, Nov. 2000, pp. 646-650.

[26] S. Bodamer and J. Charzinski, „Evaluation of Effective Bandbwidth Schemes for Self-Similar Traffic“, in *Proceedings of the 13th ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management*, Monterey, California, September 18-20, 2000, pp. 21-1-21-10.

[27] O. Bonaventure, „A flexible buffer acceptance algorithm to support the GFR service category in ATM switches“, in *Proceedings of the Sixth IFIP Workshop on Performance Modelling and Evaluation of ATM Networks (IFIP ATM'98)*, Ilkley, West Yorkshire, U.K., July 20-22, 1998, pp.71/1-71/10.

[28] R. Boorstyn, A. Burchard, J. Liebeherr and C. Oottamakorn, „Effective Envelopes: Statistical Bounds on Multiplexed Traffic in Packet Networks“, in *Proceedings of the IEEE Infocom 2000*, Tel Aviv, Israel, March 28-30, pp. 1223-1232.

[29] R. Boorstyn, A. Burchard, J. Liebeherr and C. Oottamakorn, „Statistical Service Assurances for Traffic Scheduling Algorithms“, in *IEEE Journal on Selected Areas in Communications*, Vol. 18, No. 12, December 2000, pp. 2651-2664.

[30] D. D. Botvich and N. G. Duffield, „Large deviations, the shape of the loss curve, and economies of scale in large multiplexers“, in *Queueing Systems*, Vol. 20, 1995, pp. 293-320.

[31] R. Braden (Ed.), *Requirements for Internet Hosts - Communication Layers*. IETF Request for Comments, RFC 1122, October 1989.

[32] R. Braden (Ed.), L. Zhang, S. Berson, S. Herzog and S. Jamin, *Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification*. IETF Request for Comments, RFC 2205, September 1997.

[33] R. Braden, D. Clark and S. Shenker, *Integrated Services in the Internet Architecture: an Overview*. IETF Request for Comments, RFC 1633, July 1994.

[34] R. Braden and L. Zhang, *Resource ReSerVation Protocol (RSVP) – Version 1 Message Processing Rules*. IETF Request for Comments, RFC 2205, September 1997.

[35] L. S. Brakmo and L. L. Peterson, „TCP Vegas: End to End Congestion Avoidance on a Global Intranet“, in *IEEE Journal on Selected Areas in Communications*, Vol. 13, No. 8, October 1995, pp. 1465-1480.

[36] L. Breslau and S. Shenker, „Best-Effort versus Reservations: A Simple Comparative Analysis“, in *Computer Communication Review*, Vol. 28, No. 4, pp. 3-16, October 1998.

[37] J. A. Bucklew, *Large Deviation Techniques in Decision, Simulation and Estimation.* New York: Wiley, 1990.

[38] L. Burgstahler (Ed.), *Prototype Implementation of a RSVP/IP and ATM Network Integration Unit*. ACTS 319 DIANA Deliverable 3, August 1999.

[39] R. Callon, P. Doolan, N. Feldman, A. Fredette, G. Swallow and A. Viswanathan, *A Framework for Multiprotocol Label Switching*. Work in Progress, IETF Multiprotocol Label Switching Working Group, <draft-ietf-mpls-framework-05.txt>, September 1999.

[40] C.-S. Chang, „Stability, Queue Length, and Delay of Deterministic and Stochastic Queueing Networks“, in *IEEE Transactions on Automatic Control*, Vol. 39, No. 5, May 1994, pp. 913-931.

[41] A. Charny and J. Y. Le Boudec, „Delay Bounds in a Network with Aggregate Scheduling“, in *Quality of Future Internet Services - First COST 263 International Workshop (QofIS 2000) Proceedings*, Berlin, Germany, September 25-26, 2000, pp. 1-13.

[42] J. Charzinski, *TDMA-Medienzugriffsverfahren im Rückkanal passiver optischer ATM-Zugangsnetze*. 74. Bericht über verkehrstheoretische Arbeiten, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1999.

[43] J. Choe and N. B. Shroff, „A Central-Limit-Theorem-Based Approach for Analysing Queue Behaviour in High-Speed Networks“, in *IEEE/ACM Transactions on Networking*, Vol. 6, No. 5, October 1998, pp. 659-671.

[44] N. Ciulli and S. Giordano, „Implementation and Experimental Analysis of IP Schedulers in an IntServ-over-ATM Test-bed“, in *Proceedings of the 13th ITC Specialist Seminar IP Traffic Measurement, Modeling and Management*, Monterey, California, USA, September 18-20, 2000, pp. 25-1-25-10.

[45] D. E. Comer, *Internetworking with TCP /IP, Volume I: Principles, prototocols and architecture*. Upper Saddle River, New Jersey: Prentice Hall, 4th Edition, 2000.

[46] C. Courcoubetis and R. Weber, „Buffer Overflow Asymptotics for a Buffer Handling Many Traffic Sources“, in *Journal of Applied Probability*, Vol. 33, 1996, pp. 886-903.

[47] C. Courcoubetis, V. A. Siris and G. D. Stamoulis, „Application of the many sources asymptotic and effective bandwidths to traffic engineering“, in *Telecommunication Systems*, Vol. 12, No. 2,3, November 1999, pp. 167-191.

[48] E. Crawley, *A Framework for Integrated Services and ATM Signalling*. IETF Request for Comments, RFC 2382, August 1998.

[49] R. L. Cruz, „A Calculus for Network Delay, Part I: Network Elements in Isolation“, in *IEEE Transactions on Information Theory*, Vol. 37, No. 1, January 1991, pp. 114-131.

[50] S. Deering, *Host Extensions for IP Multicasting*. IETF Request for Comments, RFC 1112, August 1989.

[51] A. Demirtjis, B. Edwards, B. Braden, S. Berson, M. P. Maher and A. Mankin, *RSVP and ATM Signalling*. ATM Forum Contribution 96-0258, January 1996.

[52] C. Diot and J. Y. Le Boudec, „Control of Best Effort Traffic“, in *IEEE Network*, Vol. 15, No. 3, May/June 2001, pp. 14-15.

[53] K. Dolzer and S. Bodamer, *Introduction to the Concepts of the C++ Simulation Library*. Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, July 2001.

[54] N. G. Duffield and N. O´Connell, „Large deviations and overflow probabilities for the general single-server queue, with applications“, in *Mathematical Proceedings of the Cambridge Philosophical Society*, Vol. 118, 1995, pp. 363-374.

[55] N. G. Duffield, „Economies of scale in queues with sources having power-law large deviation scalings“, in *Journal of Applied Probability*, Vol. 33, 1996, pp. 840-857.

[56] N. G. Duffield, „Economies of scale for long-range dependent traffic in short buffers“, in *Telecommunication Systems*, Vol. 7, Nos. 1-3, 1997, pp. 267-280.

[57] N. G. Duffield and S. H. Low, „The Cost of Quality in Networks of Aggregate Traffic“, in *Proceedings of IEEE Infocom´*98, San Francisco, California, USA, March 31 - April 2, 1998, pp. 525-532.

[58] G. Eichler, H. Hussmann, G. Mamais, I. Venieris, C. Prehofer and S. Salsano, „Implementing Integrated and Differentiated Services for the Internet with ATM Networks

– A Practical Approach“, in *IEEE Communications Magazine*, Vol. 38, No. 1, January 2000, pp. 132-141.

[59] A. Elwalid, D. Mitra and R. H. Wentworth, „A New Approach for Allocating Buffers and Bandwidth to Heterogeneous, Regulated Traffic in an ATM Node“, in *IEEE Journal on Selected Areas in Communications*, Vol. 13, No. 6, August 1995, pp. 1115-1127.

[60] J. Enssle, *Modellierung und Leistungsuntersuchung eines verteilten Video-On-Demand-Systems für MPEG-codierte Videodatenströme mit variabler Bitrate*. 68. Bericht über verkehrstheoretische Arbeiten, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1998.

[61] K. Fall and S. Floyd, „Simulation-based Comparisons of Tahoe, Reno, and SACK-TCP“, in *Computer Communication Review*, Vol. 26, No. 3, July 1996, pp. 5-21.

[62] K. Fall and K. Varadhan, *The ns Manual*. Infomation Sciences Institute, University of Southern California, Februar 2002.

[63] J. Färber, S. Bodamer and J. Charzinski, „Statistical evaluation and modelling of Internet dial-up traffic“, in *Proceedings of SPIE Photonics East´99 Conference on Performance and Control of Network Systems III (SPIE PE 99)*, Boston, September 20-21, 1999, pp. 112-121.

[64] G. Fehér, K. Németh, M. Maliosz, I. Cselényi, J. Bergkvist, D. Ahlard and T. Engborg, „Boomerang - A Simple Protocol for Resource Reservation in IP Networks“, in *Proceedings of the IEEE Workshop on QoS Support for Real-Time Internet Applications*, Vancouver, Canada, June 1, 1999.

[65] W. Feller, *An Introduction to Probability Theory and its Applications – Volume II*. New York: Wiley, 1966.

[66] W. Feng, D. Kandlur, D. Saha and K. Shin, „A Self-Configuring RED Gateway“, in *Proceedings of the IEEE Infocom´99*, New York, USA, March 23-25, 1999, pp. 1320-1328.

[67] W. Feng, D. Kandlur, D. Saha and K. Shin, *BLUE: A New Class of Active Queue Management Algorithms*. Technical Report CSE-TR-387-99, University of Michigan, April 1999.

[68] W. Fenner, *Internet Group Management Protocol, Version 2*. IETF Request for Comments, RFC 2236, November 1997.

[69] T. Ferrari, W. Almesberger and J.-Y. Le Boudec, „SRP: a Scalable Reservation Protocol for the Internet", in *Proceedings of the 6th International Workshop on Quality of Service (IWQoS´98)*, Napa, California, USA, May 18-20, 1998, pp. 107-116.

[70] V. Firoiu, J. Kurose and D. Towsley, „Efficient Admission Control for EDF Schedulers", in *Proceedings of the IEEE Infocom´97*, Kobe, Japan, April 9-11, 1997, pp. 310-317.

[71] S. Floyd and V. Jacobson, „Random Early Detection Gateways for Congestion Avoidance", in *IEEE/ACM Transactions on Networking*, Vol. 1, No. 4, August 1993, pp. 397-413.

[72] S. Floyd, „TCP and Explicit Congestion Notification", in *Computer Communication Review*, Vol. 24, No. 5, October 1994[6], pp. 8-23.

[73] S. Floyd and T. Henderson, *The NewReno Modification to TCP´s Fast Recovery Algorithm*. IETF Request for Comments, RFC 2582, April 1999.

[74] S. Floyd and K. Fall, „Promoting the Use of End-to-End Congestion Control in the Internet", in *IEEE/ACM Transactions on Networking*, Vol. 7, No. 4, August 1999, pp. 458-472.

[75] S. Floyd, J. Mahdavi, M. Mathis and M. Podolsky, *An Extension to the Selective Acknowledgement (SACK) Option for TCP*. IETF Request for Comments, RFC 2883, July 2000.

[76] S. Floyd, *Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management*. Under Submission, August 2001.

[77] S. Ben Fredj, T. Bonald, A. Proutiere, G. Régnié and J. Roberts, „Statistical Bandwidth Sharing: A Study of Congestion at Flow Level", in *Proceedings of the ACM SIGCOMM'01*, S. Diego, California, USA, August 27-31, 2001, pp. 111-122.

[78] K. Funk, *Untersuchung von skalierbaren Protokollen zur Bereitstellung einer garantierbaren Dienstgüte im Internet*. Diplomarbeit Nr. 1602, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, März 1999.

[79] M. W. Garrett and M. Borden, *Interoperation of Controlled Load Service and Guaranteed Service with ATM*. IETF Request for Comments, RFC 2381, March 1998.

6 Auf der Titelseite dieser Zeitschriftenausgabe ist fälschlicherweise das Jahr 1995 abgedruckt.

[80] L. Georgiadis, R. Guérin, V. Peris and R. Rajan, „Efficient Support of Delay and Rate Guarantees in an Internet“, in *Computer Communication Review*, Vol. 26, No. 4, October 1996, pp. 106-116.

[81] P. Gevros, J. Crowcroft, P. Kirstein and S. Bhatti, „Congestion Control Mechanisms and the Best Effort Service Model“, in *IEEE Network*, Vol. 15, No. 3, May/June 2001, pp. 16-26.

[82] R. J. Gibbens and F. P. Kelly, „Resource pricing and the evolution of congestion control“, in *Automatica*, Vol. 35, 1999, pp. 1969-1985.

[83] R. J. Gibbens and F. P. Kelly, „Distributed connection acceptance control for a connectionless network“, in *Teletraffic Engineering in a Competitive World - Proceedings of the International Teletraffic Congress (ITC-16)*, Edinburgh, U. K., June 7-11, 1999, pp. 941-952

[84] S. Golestani, „A Self-Clocked Fair Queueing Scheme for Broadband Applications“, in *Proceedings of IEEE Infocom´94*, Toronto, Ontario, Canada, June 14-16, 1994, pp. 636-646.

[85] M. de Graaf, M. Mandjes and H. v. d. Berg, *On the efficiency of EMW´s Connection Admission Control algorithm*. COST 257 Technical Document cost257td97(34), May 1997.

[86] C. Grévent, *Simulative Leistungsuntersuchung von TCP über ATM*. Diplomarbeit Nr. 1524, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, April 1997.

[87] D. Grossman, *New Terminology for Diffserv*. Work in Progress, IETF Diffserv Working Group, <draft-ietf-diffserv-new-terms-05.txt>, August 2001.

[88] R. Guerin, S. Blake and S. Herzog, *Aggregating RSVP-based QoS Requests*. Work in Progress, IETF Integrated Services Working Group, <draft-guerin-aggreg-rsvp-00.txt>, November 1997.

[89] R. Händel, M. N. Huber and S. Schröder, *ATM-Networks - Concepts, Protocols, Applications*. Harlow: Addison Wesley, Third Edition, 1998.

[90] C. Hauser, *Untersuchung von paketorientiertem Routing in IP-Netzen mit Reservierung*. Diplomarbeit Nr. 1662, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, März 2000.

[91] J. Heinanen, F. Baker, W. Weiss and J. Wroclawski, *Assured Forwarding PHB*. IETF Request for Comments, RFC 2597, June 1999.

[92] D. P. Heyman, T. V. Lakshman and A. L. Neidhardt, „A New Method for Analysing Feedback-Based Protocols with Applications to Engineering Web Traffic over the Internet“, in *Performance Evaluation Review*, Vol. 25, No. 1, June 1997, pp. 24-38.

[93] F. Hillier and G. Lieberman, *Introduction to Operations Research*. New York: McGraw Hill, Fourth Edition, 1986.

[94] J. C. Hoe, „Improving the Start-up Behavior of a Congestion Control Scheme for TCP“, in *Proceedings of the ACM SIGCOMM'96*, Stanford University, California, USA; August 26-30, 1996, pp. 270-280.

[95] C. V. Hollot, V. Misra, D. Towsley and W.-B. Gong, „A Control Theoretic Analysis of RED“, in *Proceedings of the IEEE Infocom 2001*, Anchorage, Alaska, USA, April 22-26, 2001.

[96] C. V. Hollot, V. Misra, D. Towsley and W.-B. Gong, „On Designing Improved Controllers for AQM Routers Supporting TCP Flows“, in *Proceedings of the IEEE Infocom 2001*, Anchorage, Alaska, USA, April 22-26, 2001.

[97] J. Y. Hui, „Resource Allocation for Broadband Networks“, in *IEEE Journal on Selected Areas in Communications*, Vol. 6, No. 9, December 1988, pp. 1598-1608.

[98] P. Hurley, J.-Y. Le Boudec and P. Thiran, „A Note on the Fairness of Additive Increase and Multiplicative Decrease“, in *Teletraffic Engineering in a Competitive World - Proceedings of the International Teletraffic Congress (ITC-16)*, Edinburgh, U. K., June 7-11, 1999, pp. 467-478.

[99] P. Hurley, J.-Y Le Boudec, P. Thiran and M. Kara, „ABE: Providing a Low-Delay Service within Best Effort“, in *IEEE Network*, Vol. 15, No. 3, May/June 2001, pp.60-69.

[100] A. D. Ioffe und V. M. Tichomirov, *Theorie der Extremalaufgaben*. VEB Deutscher Verlag der Wissenschaften, Berlin, 1979.

[101] ITG-Fachgruppe 5.2.2 „Systemtechnik“, *Architekturen und Verfahren der Vermittlungstechnik - Entwurf für die ITG-Empfehlung 5.2-01*. Dezember 1996.

[102] ITU-T, *B-ISDN ATM Adaptation Layer specification: Type 5 AAL*. ITU-T Recommendation I.363.5, August 1996.

[103] ITU-T, *Traffic control and congestion control in B-ISDN*. ITU-T Recommendation I.371 (Prepublished Recommendation), February 2000.

[104] V. Jacobson, „Congestion Avoidance and Control“, in *Computer Communications Review*, Vol. 18, No. 4, August 1988, pp. 314-328.

[105] V. Jacobson, R. Braden and D. Borman, *TCP Extensions for High Performance*. IETF Request for Comments, RFC 1323, May 1992.

[106] V. Jacobson, K. Nichols and J. Poduri, *An Expedited Forwarding PHB*. IETF Request for Comments, RFC 2598, June 1999.

[107] H. Jaschek, *Systemtheorie der Elektrotechnik*. Unterlagen zur Vorlesung, Universität des Saarlandes, 1991/92.

[108] L. Jaussi, M. Lorang and J. Nelissen, „A detailed experimental performance evaluation of TCP over UBR“, in *Telecommunication Systems*, Vol. 11, Nos. 3,4, April 1999, pp. 353-371.

[109] P. Karlsson and Å. Arvidsson, „TCP/IP user Level Modelling for ATM“, in *Proceedings of the Sixth IFIP Workshop on Performance Modelling and Evaluation of ATM Networks (IFIP ATM'98)*, Ilkley, West Yorkshire, U.K., July 20-22, 1998, pp.83/1-83/10.

[110] P. Karn and C. Partridge, „Improving Round-Trip Time Estimates in Reliable Transport Protocols“, in *ACM Transactions on Computer Systems*, Vol. 9, No. 4, November 1991, pp. 365-373.

[111] K. Kilkki, „Simple Integrated Media Acess – a Novel Service Concept for Internet“, in *Proceedings of the EXPERT ATM Traffic Symposium*, Mykonos, Greece, September 1997.

[112] D. Katz, *IP Router Alert Option*. IETF Request for Comments, RFC 2113, February 1997.

[113] F. P. Kelly, *Reversibility and Stochastic Processes*. Chichester: Wiley, 1979.

[114] F. P. Kelly, „Notes on Effective Bandwidth“, in *Stochastic Networks - Theory and Applications (ed. by F. Kelly, S. Zachary and I. Ziedins)*, Oxford, U. K: Oxford University Press, 1996.

[115] F. P. Kelly, „Charging and Rate Control for Elastic Traffic“, in *European Transactions on Telecommunications*, Vol. 8, No. 1, January 1997, pp. 33-37.

[116] F. P. Kelly, „Effective bandbwidths at multi-class queues“, in *Queueing Systems*, Vol. 9, 1999, pp. 5-16.

[117] F. P. Kelly, „Mathematical modelling of the Internet“, in *Proceedings of the 4th International Congress on Industrial and Applied Mathematics*, Edinburgh, Scotland, July 1999, pp. 105-116.

[118] L. Kleinrock, *Queueing Systems - Volume 1: Theory*. New York: Wiley, 1975.

[119] E. W. Knightly and N. B. Shroff, „Admission Control for Statistical QoS: Theory and Practice“, in *IEEE Network*, Vol. 13, No. 2, April/March 1999, pp. 20-29.

[120] H. Kröner, *Verkehrssteuerung in ATM-Netzen – Verfahren und verkehrstheoretische Analysen zur Zellpriorisierung und Verbindungsannahme*. 62. Bericht über verkehrstheoretische Arbeiten, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1995.

[121] H. Kröner, P. J. Kühn und T. Renger, „Management von ATM-Netzen“, in *Informationstechnik und Technische Informatik* 39 (1997) 1, pp. 5-14.

[122] P. J. Kühn, *Teletraffic Theory and Engineering*. Lecture Notes, University of Stuttgart, Edition 2000/2001.

[123] S.-O. Larsson and Å. Arvidsson, „Performance Evaluation of a Local Approach for VPC Capacity Management“, in *IEICE Transactions on Communications*, Vol. E81-B, No. 5, May 1998, pp. 870-876.

[124] S.-O. Larsson and Å. Arvidsson, „An Adaptive Local Method for VPC Capacity Management“, in *Teletraffic Engineering in a Competitive World - Proceedings of the International Teletraffic Congress (ITC-16)*, Edinburgh, U. K., June 7-11, 1999, pp. 561-570.

[125] M. Laubach, *Classical IP and ARP over ATM*. IETF Request for Comments, RFC 1577, January 1994.

[126] T. Laut, „Frame Relay – ein neues Übertragungsprotokoll im Bereich der Datenkommunikation“, in *Unterrichtsblätter Jg. 48 11/1995*, pp. 618-630.

[127] J. Y. Le Boudec, „Network Calculus, Deterministic Effective Bandwidth and VBR Trunks“, in *Proceedings of the IEEE Globecom´97*, Vol. 3, Phoenix, Arizona, USA, November 3-8, 1997, pp. 1349-1354.

[128] J. Y. Le Boudec and A. Ziedinš, „A CAC algorithm for VBR Connections over a VBR Trunk“, in *Teletraffic Contributions for the Information Age - Proceedings of the 15th*

International Teletraffic Congress (ITC-15), Washington, DC, USA, June 22-27, pp. 59-70.

[129] T. Leinmüller, *Experimentelle Untersuchung von Verfahren zur Berechnung des Ressourcenbedarfs von dynamischen Verbindungsaggregaten*. Studienarbeit Nr. 1709, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, September 2001.

[130] S. Löffler, *Verwendung von Flows zur Analyse und Messung von Internet-Verkehr*. Studienarbeit Nr. 1506, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, August 1997.

[131] S. Low, „A Duality Model of TCP Flow Controls“, in *Proceedings of the 13th ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management*, Monterey, California, September 18-20, 2000.

[132] J. T. Lewis and R. Russell, „An Introduction to Large Deviations and its Applications to Teletraffic Engineering“, in *International Conference on Performance Theory, Measurement and Evaluation of Computer and Communication Systems*, Lausanne, Switzerland, October 7-11, 1996.

[133] J. Liebeherr, D. E. Wrege and D. Ferrari, „Exact Admission Control for Networks with a Bounded Delay Service“, in *IEEE/ACM Transactions on Networking*, Vol. 4, No. 6, December 1996, pp. 885-901.

[134] J. Liebeherr and D. E. Wrege, „Priority Queue Schedulers with Approximate Sorting in Output-Buffered Switches“, in *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 6, June 1999, pp. 1127-1144.

[135] N. Likhanov and R. Mazumdar, „Cell loss asymptotics in buffers fed with a large number of independent stationary sources“, in *Proceedings of IEEE Infocom´*98, San Francisco, California, USA, March 31 - April 2, 1998, pp. 339-346.

[136] Z. Liu, N. Niclausse, C. Jalpa-Villanueva and S. Barbier, *Traffic Model and Performance Evaluation of Web Servers*. Rapport de Recherche Numéro 3840, Institut National de Recherche en Informatique et en Automatique (INRIA), December 1999.

[137] M. Lorang (Ed.), *Demonstration of IP and ATM Networking for Real-Time Applications (DIANA)*. ACTS Programme Proposal Number 30069, September 1997.

[138] M. Lorang, „A QoS Guaranteeing Framework for the Integration of IP and ATM in DIANA“, in *EUNICE´98 Open European Summer School on Network Management and Operation*, Munich, Germany, August 31 - September 3, 1998, pp. 178-187.

[139] M. Lorang, *Draft Description of the Flow-to-VC Mapping Control Module in DIANA´s Integration Unit*. ACTS 319 DIANA Contribution AC319_UST_2_002.01_CD_CC, Stuttgart, Germany, December 1998.

[140] M. Lorang, „Connection-oriented Flow Aggregation with a RSVP over ATM Example“, in *Sixteenth UK Teletraffic Symposium on Management of Quality of Service - The New Challenge*, Harlow, UK, May 22-24, 2000, pp. 36/1-36/6.

[141] M. Lorang und C. Macian, *Protokollkonzept zum dynamischen Aufbau von Verbindungsaggregaten*. UNIQUE Projektbeitrag, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, Juli 1999.

[142] J. Luciani, D. Katz, D. Piscitello, B. Cole and N. Doraswamy, *NBMA Next Hop Resolution Protocol (NHRP)*. IETF Request for Comments, RFC 2332, April 1998.

[143] M. Mathis, J. Mahdavi, S. Floyd and A. Romanow, *TCP Selective Acknowledgement Options*. IETF Request for Comments, RFC 2018, October 1996.

[144] M. Mathis, J. Semke, J. Mahdavi and T. Ott, „The Macroscopic Behaviour of the TCP Congestion Avoidance Algorithm“, in *Computer Communication Review*, Vol. 27, No. 3, July 1997, pp. 67-82.

[145] T. Maufer and C. Semeria, *Introduction to IP Multicast Routing*. Work in Progress, IETF, <draft-ietf-mboned-intro-multicast-03.txt>, July 1997.

[146] M. Meuli (Ed.), *Evaluation and Demonstration of Various Approaches towards QoS in IP Networks*. ACTS 319 DIANA Deliverable 5, February 2000.

[147] D. L. Mills, *Network Time Protocol (Version 3) - Specification, Implementation & Analysis*. IETF Request for Comments, RFC 1305, March 1992.

[148] V. Misra, W.-B. Gong and D. Towsley, „Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED“, in *Computer Communication Review*, Vol. 30, No. 4, October 2000, pp. 151-160.

[149] J. Mo, R. La, V. Anantharam and J. Walrand, „Analysis and Comparison of TCP Reno and Vegas“, in Proceedings of the Infocom 1999, New York, USA, March 21-25, 1999, pp. 1556-1563.

[150] U. Mocci, P. Pannunzi and C. Scoglio, „Adaptive Capacity Management of Virtual Path Networks“, in *Proceedings of IEEE Globecom´96*, November 18-22, 1996, paper no. 19-2.

[151] K. Moldeklev, *Performance Analyses and Issues of End Systems Attached to High-Speed Networks*. IDT-rapport 1996:4, Department of Computer Sytems and Telematics, Norwegian University of Science and Technology, Trondheim, April 1996.

[152] J. Moy, *OSPF Version 2*. IETF Request for Comments, RFC 2328, April 1998.

[153] K. Nichols, S. Blake, F. Baker and D. Black, *Definition of the Differentiated Services Field (DS field) in the IPv4 and IPv6 Headers*. IETF Request for Comments, RFC 2474, December 1998.

[154] R. O. Onvural and R. Cherukuri; *Signaling in ATM Networks*. Boston: Artech House, 1997.

[155] A. Orda, G. Pacifici and D. Pendarakis, „An Adaptive Virtual Path Allocation Policy for Broadband Networks“, in *Proceedings of the IEEE Infocom´96*, San Francisco, California, USA, March 26-28, 1996, pp. 329-335.

[156] J. Padhye, V. Firoiu, D. Towsley and J. Kurose, „Modeling TCP Reno Performance“, in *IEEE/ACM Transactions on Networking*, Vol. 8, No. 2, April 2000, pp. 133-145.

[157] P. Pan and H. Schulzrinne, „YESSIR: A Simple Reservation Mechanism for the Internet“, in *Proceedings of the 8th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV´98)*, Cambridge, U.K., July 8-10, 1998, pp. 141-153.

[158] P. Pan, E. Hahne and H. Schulzrinne, „BGRP: A Tree-Based Aggregation Protocol for Inter-domain Reservations“, in *Journal of Communications and Networks*, Vol. 2, No. 2, June 2000, pp. 157-167.

[159] A. Parekh and R. Gallager, „A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks“, in *IEEE/ACM Transactions on Networking*, Vol. 1, No. 3, June 1993, pp.344-357.

[160] A. Papoulis, *Probability, Random Variables and Stochastic Processes*. Tokyo: Mc Graw Hill, 1965.

[161] V. Paxson, M. Allman, S. Dawson, W. Fenner, J. Griner, I. Heavens, K. Lahey, J. Semke and B. Volz, *Known TCP Implementation Problems*. IETF Request for Comments, RFC 2525, March 1999.

[162] L. Peinado Cardona, *Puffermanagement und Bedienstrategien zum Transport von TCP-Verkehr mit der ATM-Dienstklasse GFR*. Studienarbeit Nr. 1648, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, Dezember 1999.

[163] J. Postel, *User Datagram Protocol*. IETF Request for Comments, RFC 768, September 1980.

[164] J. Postel, *Internet Control Message Protocol*. IETF Request for Comments, RFC 792, September 1981.

[165] J. Postel (Ed.), *Transmission Control Protocol*. IETF Request for Comments, RFC 793, September 1981.

[166] K. Ramakrishnan and S. Floyd, *A Proposal to add Explicit Congestion Notification (ECN) to IP*. IETF Request for Comments, RFC 2481, January 1999.

[167] T. Renger, E. Aarstad, H. Pettersen and J. Kroeze, „Experimental Validation of an ATM Traffic Control Framework in the EXPLOIT Testbed“, in *European Transactions on Telecommunications*, Vol. 7, No. 5, September/October 1996, pp. 393-405.

[168] G. Rigolio, A. Casaca, D. Shepherd, S. Giordano, F. Rossi and C. Edwards, „Experience on Quality of Service Support by Exploiting Reservation in IP and ATM Networks“, in *Proceedings of the IEEE Conference on High Performance Switching and Routing (ATM´2000) - Joint IEEE ATM Workshop 2000 and 3rd International Conference on ATM (ICATM´2000)*, Heidelberg, Germany, June 26-29, 2000, pp. 323-331.

[169] J. Roberts, „Virtual Spacing for Flexible Traffic Control“, in *International Journal of Communication Systems*, Vol. 7, 1994, pp. 307-318.

[170] J. Roberts, U. Mocci, J. Virtamo (Eds.), *Broadband Network Teletraffic - Final Report of Action COST 242*. Berlin: Springer, 1996.

[171] E. C. Rosen, A. Viswanathan and R. Callon, *Multiprotocol Label Switching Architecture*. Work in Progress, IETF Multiprotocol Label Switching Working Group, <draft-ietf-mpls-arch-07.txt>, July 2000.

[172] S. Russ, *Erweiterung eines H.323 Gateways um Mechanismen zur Verkehrssteuerung*. Studienarbeit Nr. 1701, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, Juli 2001.

[173] L. Salgarelli, M. De Marco, G. Meroni and V. Trecordi, „Efficient Transport of IP Flows Across ATM Networks“, in *Proceedings of the IEEE ATM´97 Workshop*, Lisboa, Portugal, May 25-28, 1997, pp. 43-50.

[174] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, *RTP: A Transport Protocol for Real-Time Applications*. IETF Request for Comments, RFC 1889, January 1996.

[175] S. Shenker and L. Breslau, „Two Issues in Reservation Establishment“, in *Proceedings of the ACM SIGCOMM´95*, Cambridge, Massachusetts, August 28 – September 1, 1995, pp. 14-26.

[176] S. Shenker, C. Partridge and R. Guérin, *Specification of Guaranteed Quality of Service*. IETF Request for Comments, RFC 2212, September 1997.

[177] S. Shenker and J. Wroclawski, *General Characterization Parameters for Integrated Services Network Elements*. IETF Request for Comments, RFC 2215, September 1997.

[178] S. Shioda and H. Uose, „Virtual Path Bandwidth Control Method for ATM Networks: Successive Modification Method“, in *IECE Transactions*, Vol. E 74, No. 12, December 1991.

[179] G. Siegmund, *ATM - Die Technik des Breitband-ISDN*. Heidelberg: R. v. Decker´s Verlag, 2. Auflage, 1994.

[180] V. A. Siris, *Performance Analysis and Pricing in Broadband Networks*. Department of Computer Science, Universtity of Crete, December 1997.

[181] V. Šironja, *Implementation of a Traffic-Control Architecture for RSVP over ATM*. Diplomarbeit Nr. 1767, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, Dezember 1999.

[182] V. Sivaraman and F. Chiussi, „Providing End-to-End Statistical Delay Guarantees with Earliest Deadline First Scheduling and Per-Hop Traffic Shaping“, in *Proceedings of the IEEE Infocom 2000*, Tel Aviv, Israel, March 28-30, pp. 631-640.

[183] W. R. Stevens, *UNIX Network Programming*. Englewood Cliffs, New Jersey: Prentice Hall, 1990.

[184] W. R. Stevens, *Advanced Programming in the UNIX Environment*. Reading, Massachusetts: Addison-Wesley, 1993.

[185] W. R. Stevens, *TCP/IP Illustrated, Volume 1 – The Protocols*. Reading, Massachusetts: Addison-Wesley, 1994.

[186] D. Stiliadis and A. Varma, „Latency-Rate Servers: A General Model for Analysis of Traffic Scheduling Algorithms“, in *IEEE/ACM Transactions on Networking*, Vol. 6, No. 5, October 1998, pp. 611-624.

[187] I. Stoica and H. Zhang, „Providing Guaranteed Services Without Per Flow Management“, in *Proceedings of the ACM SIGCOMM'99*, Cambridge, Massachusetts, August 31 – September 3, 1999, pp. 81-94.

[188] G. Swallow, „MPLS Advantages for Traffic Engineering“, in *IEEE Communications Magazine*, Vol. 37, No. 12, December 1999, pp. 54-57.

[189] A. S. Tanenbaum, *Computer Networks – Third Edition*. Upper Saddle River, New Jersey: Prentice Hall, 1996.

[190] B. Teitelbaum, S. Hares, L. Dunn, R. Neilson, V. Narayan and F. Reichmeyer, „Internet2 QBone: Building a Testbed for Differentiated Services”, in *IEEE Network*, Vol. 13, No. 5, September/October 1999, pp. 8-16.

[191] G. Urvoy, Y. Dallery and G. Hébuterne, „CAC Procedures for Delay-constrained VBR Sources“, in *Sixth IFIP Workshop on Performance Modelling and Evaluation of ATM Networks (IFIP ATM´98)*, Ilkley, West Yorkshire, U. K., July 20-22, 1998, pp. 39/1-39/10.

[192] M. Verdier, D. Griffin and P. Georgatsos, „Dynamic Bandwidth Management in ATM Networks“, in *EUNICE´98 Open European Summer School on Network Management and Operation*, Munich, Germany, August 31 - September 3, 1998, pp. 204-211.

[193] J. T. Virtamo and S. Aalto, *Blocking probabilities in a transient system*. COST 257 Technical Document cost257td97(14), January 1997.

[194] J. T. Virtamo and S. Aalto, *Remarks on the effectiveness of dynamic VP bandwidth management*. COST 257 Technical Document cost257td97(15), January 1997.

[195] R. Vonier, *Erweiterung einer Testapplikation zur Leistungsuntersuchung von TCP*. Studienarbeit Nr. 1562, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, April 1998.

[196] P. P. White and J. Crowcroft, „The Integrated Services in the Internet: State of the Art“, in *Proceedings of the IEEE*, Vol. 85, No. 12, December 1997, pp. 1934-1946.

[197] J. Wroclawski, *The Use of RSVP with IETF Integrated Services*. IETF Request for Comments, RFC 2210, September 1997.

[198] J. Wroclawski, *Specification of the Controlled-Load Network Element Service*. IETF Request for Comments, RFC 2211, September 1997.

[199] J. Widmer, R. Denda and M. Mauve, „A Survey on TCP-Friendly Congestion Control“, in *IEEE Network*, Vol. 15, No. 3, May/June 2001, pp. 28-37.

[200] H. Zhang, „Service Disciplines for Guaranteed Performance Service in Packet-Switched Networks“, in *Proceedings of the IEEE*, Vol. 83, No. 10, October 1995, pp. 1373-1396.

[201] L. Zhang, S. Deering, D. Estrin, S. Shenker and D. Zappala, „RSVP: A New Resource Reservation Protocol“, in *IEEE Network*, Vol. 7, No. 5, September 1993, pp. 8-18.