

Universität Stuttgart

Institut für Kommunikationsnetze und Rechnersysteme
Prof. Dr.-Ing. habil. Dr. h. c. mult. P. J. Kühn

84. Bericht über verkehrstheoretische Arbeiten

Traffic Shaping-Verfahren in ATM-Netzen für die Übertragung komprimierter Videodaten

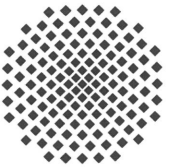
von

Marcel Graf

2002

D 93

© 2002 Institut für Kommunikationsnetze und Rechnersysteme, Universität Stuttgart
Druck: E. Kurz & Co., Druck und Medientechnik GmbH, Stuttgart
ISBN 3-922403-94-8



University of Stuttgart

Institute of Communication Networks and Computer Engineering

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

84th Report on Studies in Congestion Theory

Traffic Shaping in ATM Networks for the Transmission of Compressed Video Data

by

Marcel Graf

2002

Summary

Integration of services is an important requirement for today's wide and local-area networks. It means the support of a broad gamut of applications that have different traffic characteristics and different requirements regarding Quality of Service. Packet switching seems to be best suited to deal with heterogeneous data rates. Two widely deployed network concepts, the family of Internet Protocols (IP) and the Asynchronous Transfer Mode (ATM), are based on packet switching.

The medium video has properties that make it exemplary for other time-dependent media. The encoding process produces a variable amount of data for each picture. Application scenarios impose a set of real-time requirements on the transmission of the data. From these factors stems a traffic characteristic that is difficult to handle: traffic is bursty on several time scales, with the time scales ranging from double-digit milliseconds to several minutes.

This thesis is concerned with traffic shaping as a means to improve the traffic characteristic of a source of time-dependent media. A traffic shaper consists of a rate control and a buffer to temporarily store data units. In periods of high source activity the buffer retains data units and the rate control releases them with a well-defined, possibly variable rate to the network. The amelioration of the traffic characteristic has always a tradeoff in the delay the data units suffer and the expense for the provisioning of sufficient buffer capacity. Some applications are sensitive to these consequences, especially concerning the delay, for example videoconferencing. Others are able to accomodate the disadvantages quite well, for example video-on-demand. This work tries to answer the following questions:

- How can the transmission of time-dependent data in the network and the real-time requirements of applications be formalised in a model?
- What construction principles are there for the rate control of a shaper, for example to achieve a given traffic characteristic?
- What does the tradeoff look like between an improved traffic characteristic on one hand and delay and buffer capacity on the other hand?
- How well are real video sources suited to traffic shaping?
- How do approaches for traffic shapers differ regarding performance and implementation effort?

Chapter 1 motivates the need for traffic shaping in the context of integrated services networks.

Chapter 2 introduces applications that transmit compressed video data. They are first classified based on wich steps are involved in real-time data handling. Two of the three classes impose requirements on the transmission. In one case the production and transmission of video data happen at the same time ("live video"), in the other case video data is produced

offline and transmitted later out of mass storage ("stored video"). Encoding methods for digital video data according to international standards are presented. Processing of an individual picture reduces spatially redundant data. Processing of neighbouring pictures additionally reduces temporally redundant data. Optionally a rate control may dynamically set encoding parameters either to achieve for the encoded data a constant rate or a constant quality. Each processing step has its own impact on the volume characteristic of the compressed video data. The section concludes with buffering strategies a receiver employs to cope with variable transmission delay and to maintain the playback of media synchronised.

Chapter 3 introduces architectural aspects of networks based on the Asynchronous Transfer Mode that are related to integration of services. After introducing the basic specifications of the ITU for this variant of packet switching the chapter focuses on the role of Traffic Management to ensure a reliable Quality of Service. Section 3.4 gives an overview of causes for cell delay jitter and loss at different timescales and the mechanisms of Traffic Management to deal with them. It then takes a closer look at the concept of reserving network resources per connection, which becomes visible at the User-Network Interface as Traffic Contract. In a Traffic Contract the application characterises precisely its traffic and in return the network gives precise guarantees about the Quality of Service. Traffic shaping is one component of Traffic Management with two application areas. First, it is used inside the network to remove cell clumping which makes traffic more bursty and which occurs when several connections are multiplexed. Second, it is used at the traffic source to obtain a desired traffic characteristic. The latter is the focus of this work. Previous approaches to this problem are reviewed.

Chapter 4 is devoted to the mathematical modelling of video transmission in ATM networks. A fluid-flow model is used which describes the amount of data flown through a point in the network by a function over time that is continous in time and value. Cruz's *traffic envelopes* are used to characterise a traffic stream in a deterministic and time-independent way. An envelope $b(\tau)$ gives an upper bound for the amount of data flown during a time interval, the bound being a function of the interval duration τ. This characterisation has the advantage of describing the burstiness on all timescales. It generalizes the concept of Leaky Bucket and Moving Window traffic descriptors. Its effectiveness is demonstrated by a simple network resource allocation scheme which is able to provide deterministic Quality of Service guarantees. Section 4.2 applies the model to the three network elements involved in the transmission: the sender with its traffic shaper, the receiver with its mechanism to remove delay jitter, and the network, which is modeled as a single element. Real-time requirements are translated into bounds for the fill levels of the two buffers. The problem of finding a good rate control algorithm for the shaper is transformed into an geometrical optimisation problem in the data/time plane: the real-time requirements define a region on the plane through which one has to find a path which corresponds to the schedule of the shaper's rate control. The stochastic delay the

traffic stream suffers in the network affects the delay budget available for traffic shaping. Two approaches to deal with the delay jitter are discussed: In the first approach the traffic shaper of the sender accounts for it before transmission by adjusting its schedule. In the second approach the receiver deals with it after transmission by allocating additional buffer resources and further delaying the playout of the data.

Chapter 5 discusses methods for traffic shaping. With the analytical tools of Chapter 4 the performance objective of a traffic shaper can be stated as follows: The shaper has to produce an output traffic for which an envelope has to be known. This envelope is mapped by the application into a Traffic Descriptor (losing some information). The Traffic Descriptor is handed by the application in the framework of the traffic contract in a signalling message to the network, and used there to reserve resources. A good traffic shaper produces an evelope / Traffic Descriptor that enables a network to accept many connections while introducing a delay and a requirement for buffer capacity that are both as small as possible.

Shapers are constructed from two approaches. In the first approach covered in Section 5.2 simple shapers based on the Leaky Bucket mechanism are used. They are called (σ,ρ) shapers because they guarantee an envelope $b(\tau) = \sigma + \rho\tau$. They are concatenated in a series to form a *Multiple Leaky Bucket* shaper. Cruz's calculus of shaper delay gives formulas for the relation between input traffic, shaper parameters and delay. Dual formulas relating buffer capacity with input traffic and shaper parameters are developed and Cruz's theorem of the preservation of output envelopes (the overall output envelope is the minimum of the individual envelopes) is verified. These formulas are used to derive dimensioning guidelines for the selection of shaper parameters for a given input traffic and output envelope. A special case is the On/Off shaper which is a single (σ,ρ) shaper that either emits traffic at rate ρ or not at all ($\sigma = 0$). Two properties of the On/Off shaper are noteworthy: First, it produces an optimal traffic characteristic regarding the peak rate, i.e. other shapers only improve the traffic envelope for $\tau > 0$. Second, the trade-off between decreasing the peak rate and the resulting increase in buffer capacity and delay is best for high rates and deteriorates for lower rates. In the case of a step-formed input traffic there is a critical rate above which the buffer capacity remains constant and the delay is inversely proportional to the rate. Returning to the Multiple Leaky Bucket shaper another analytical result predicts for a given set of parameters and an envelope of the input traffic a much tighter output envelope than the one derived from the preservation of ouptut envelopes. Unmodified a leaky bucket shaper can only be used for live video data, it is shown how an addition to the shaping mechanism enables its use for stored video data as well.

The second approach to shaper construction is covered in Section 5.3. The geometrical optimization problem of Chapter 4 is solved directly. The *shortest path* through the region is constructed and used to drive the rate control of the shaper. Intuitively this Shortest Path

shaper is the optimal shaper because it produces the smoothest path. It can be conjectured that it produces a minimal envelope for the output traffic, however proofs are not available. In the literature it could only be shown that the peak rate is minimal (and the variance of the rate as well). The construction of the shortest path demands much higher implementation complexity than the simple counter mechanisms of the Multiple Leaky Bucket shapers. As the simple On/Off shaper has been shown to produce a minimal peak rate, Shortest Path and On/Off shaper exhibit equal performance when the network performs peak rate allocation. Shortest Path shapers have more degrees of freedom and are able to account for network delay jitter, which gives them an advantage over Multiple Leaky Bucket shapers.

Chapter 6 complements the qualitative analytical comparison of Chapter 5 with a quantitative analysis through simulations with real video data. The datasets contain several different types of scene content typical for television broadcast. Fourteen of them are MPEG-encoded, one is Motion-JPEG-encoded. The Motion-JPEG encoding reduces only spatial redundancy (intra-coding). The MPEG-encoding additionally reduces temporal redundancy (inter-coding) on some pictures which as a result tend to have smaller encodings. Intra- and inter-coded pictures are interleaved in a pattern that is repeated.

First it is investigated how well the selected datasets are suited to traffic shaping. With the MPEG-encoded datasets the peak rate can be reduced without adverse effects far below the biggest picture size divided by the picture period (the critical rate is low), while this is not the case for the Motion-JPEG encoding. This result was to be expected, as in the MPEG picture pattern big intra-coded pictures are followed by inter-coded pictures which are usually smaller. Traffic shaping is able to achieve the biggest effect in the scenario of stored video data, because there are no stringent delay requirements and the buffer capacity of the shaper can be made large. In the case of an 8 MByte buffer there remains in 12 of 15 datasets a burstiness (peak rate divided by mean rate) below 1.5, including the Motion-JPEG dataset. Thus bursty video material can be efficiently transmitted in the case of stored video data, even when the network only supports peak rate allocation.

When the Traffic Descriptor exchanged between application and network is more sophisticated and the network resource reservation is able to account for parts of the traffic envelope beyond the peak rate the prediction of a good traffic envelope becomes important. Regarding the Multiple Leaky Bucket shaper it is furthermore interesting to see how many (σ,ρ) shapers are needed to achieve a good envelope. The simulations show that for all datasets a three-stage Multiple Leaky Bucket shaper is sufficient to achieve 95% of the performance of the ideal (inifinite-stage) Multiple Leaky Bucket shaper. Other simulation scenarios look at the impact of the Traffic Descriptor on the overall performance. If the Traffic Descriptor is restricted to two Leaky Buckets, as in today's ATM standards, the shape of the envelope for larger τ is not accounted for and efficiency suffers. The simulations show that weakly

shaped video streams (delay 80 ms) are not sufficiently described with two Leaky Buckets, while strongly shaped video streams (delay 200 ms) require only two Leaky Buckets: In the case of MPEG-encoding such a description achieves in 11 of 13 datasets 95% performance of an ideal Traffic Descriptor.

The Shortest Path shaper has two advantages over the Multiple Leaky Bucket shaper: it produces the smoothest output traffic and it is able to account for network delay jitter before transmission, which could be beneficial to performance. A simulation scenario which includes significant network delay jitter (up to 400 ms) compares the two shapers and varies the number of stages of the Multiple Leaky Bucket shaper. The result shows that the two shapers are close in performance, and more so when the traffic is strongly shaped with a big buffer. With a 256 KByte buffer a Multiple Leaky Bucket shaper with only two stages is indistinguishable from a Shortest Path shaper.

Chapter 7 summarizes the results and gives an outlook on future developments in distributed multimedia applications in which traffic shaping is to play an important role.

Inhaltsverzeichnis

Abkürzungen · *xi*
Formelzeichen und Notation · *xiii*

1 Einleitung · *1*
1.1 Übertragung von zeitabhängigen Medien und der Bedarf an Traffic Shaping · *1*
1.2 Übersicht über die Arbeit · *4*

2 Komprimierte digitale Videodaten · *7*
2.1 Anwendungen · *7*
2.1.1 Klassifikation der ITU-T · *7*
2.1.2 Echtzeitanforderungen · *8*
2.2 Verfahren zur Redundanzminderung · *10*
2.2.1 Ziele · *10*
2.2.2 Abtastung und Vorbehandlung · *11*
2.2.3 Verminderung der räumlichen Redundanz · *12*
2.2.4 Verminderung der zeitlichen Redundanz · *15*
2.2.4.1 Intrabild- und Interbild-Kodierung · *15*
2.2.4.2 Vorhersage · *15*
2.2.4.3 Verminderung der Abhängigkeiten bei Interbild-Kodierung · *17*
2.3 Eigenschaften · *18*
2.3.1 Variable Rate · *18*
2.3.2 Ratenregelung · *20*
2.4 Anforderungen an die Übertragung · *21*
2.4.1 Ausgleich von Übertragungsschwankungen · *21*
2.4.2 Feste und variable Ausspielverzögerung · *22*
2.4.3 Synchronisation · *24*

3 ATM-Netze · *27*
3.1 Diensteintegration · *27*
3.2 Zeitmultiplexverfahren und Vermittlungstechniken · *27*
3.2.1 Synchrones Zeitmultiplex · *28*

3.2.2 Nachrichtenmultiplex · *29*
3.2.3 Asynchrones Zeitmultiplex mit fester Blocklänge · *31*
3.3 ATM-Netzkonzept · *32*
3.3.1 Referenzmodell · *32*
3.3.2 ATM-Schicht · *33*
3.3.3 ATM-Adaptierungsschicht · *34*
3.4 Verkehrssteuerung · *37*
3.4.1 Ziele · *37*
3.4.2 Verkehrsvertrag · *37*
3.4.3 Dienstgüteparameter · *38*
3.4.3.1 Verzögerungsparameter · *38*
3.4.3.2 Verläßlichkeitsparameter · *39*
3.4.4 Ursachen für Verzögerungsschwankung und Verlust · *40*
3.4.4.1 Verbindungsebene · *40*
3.4.4.2 Burstebene · *41*
3.4.4.3 Zellebene · *41*
3.4.5 Komponenten der Verkehrssteuerung · *43*
3.4.6 Dienstkategorien · *45*
3.4.7 Verkehrsvertrag und Quellflußkontrolle für CBR und rt-VBR · *47*
3.4.7.1 Quellverkehrsdeskriptor · *47*
3.4.7.2 Verbindungsverkehrsdeskriptor · *48*
3.4.7.3 GCR-Algorithmus · *49*
3.4.7.4 Konformitätsdefinition · *51*
3.5 Traffic Shaping · *53*
3.5.1 Netz-Reshaping · *53*
3.5.2 Quellshaping · *55*
3.5.3 Stochastische analytische Leistungsuntersuchung des Quellshapings · *55*

4 Modellierung der Videoübertragung in ATM-Netzen · *59*
4.1 Charakterisierung von Verkehrsströmen durch Verkehrshüllen · *59*
4.1.1 Leaky Bucket · *60*
4.1.2 Verkehrshülle · *62*
4.1.2.1 Definition · *62*
4.1.2.2 Anwendungsbeispiel · *62*
4.1.2.3 Empirische Hülle · *63*
4.1.2.4 Anwendung in der Zuweisung von Netzressourcen · *65*

4.1.2.5 Spezialfälle Leaky Bucket und Moving Window · *67*
4.1.3 Verkehrsdeskriptor · *68*
4.2 Modellierung des Empfängers · *70*
4.2.1 Pufferspeicher · *70*
4.2.2 Dekodier-Zeitplan von MPEG-2 kodierten Strömen · *74*
4.3 Modellierung des Senders · *75*
4.3.1 Echtzeit-Kodierung · *75*
4.3.2 Gespeicherte Videodaten · *77*
4.4 Verzögerungsschwankungen bei der Übertragung · *78*
4.4.1 Problemstellung · *78*
4.4.2 Behandlung im Sender (vor der Übertragung) · *79*
4.4.3 Behandlung im Empfänger (nach der Übertragung) · *79*

5 Traffic Shaping-Verfahren für komprimierte Videodaten · *83*
5.1 Einfache Ansätze · *83*
5.2 Leaky Bucket-basierte Verfahren · *85*
5.2.1 Einfacher Leaky Bucket · *85*
5.2.1.1 Speicher · *85*
5.2.1.2 Verzögerung · *87*
5.2.2 Mehrere Leaky Buckets · *87*
5.2.2.1 Speicher · *88*
5.2.2.2 Paralleles Ersatzschaltbild · *89*
5.2.2.3 Verzögerung · *90*
5.2.3 MLB-Shaper für Echtzeit-kodierte Videodaten · *91*
5.2.3.1 Leistungsabwägung beim Ein/Aus-Shaper · *91*
5.2.3.1.1 Abhängigkeiten zwischen Leistungsgrößen · *91*
5.2.3.1.2 Kritische Rate · *93*
5.2.3.1.3 Kleinste konkave Hülle · *94*
5.2.3.2 Dimensionierung des MLB-Shapers · *95*
5.2.3.2.1 Gegebene Verzögerung · *95*
5.2.3.2.2 Gegebene Speicherkapazität · *97*
5.2.3.3 Optimalität der maximalen Rate des MLB-Shapers · *99*
5.2.3.4 Speicherkapazität des Empfängers · *99*
5.2.4 Hülle für den Verkehrsstrom am Ausgang eines MLB-Shapers · *100*
5.2.5 MLB-Shaper für gespeicherte Videodaten · *102*
5.2.6 Implementierungsaspekte · *104*
5.2.6.1 Mechanismus des MLB-Shapers · *104*

5.2.6.2 Einsatz bei gespeicherten Videodaten · *105*
5.2.6.3 Einsatz bei Echtzeit-kodierten Videodaten · *105*
5.3 Direkte Verfahren · *106*
5.3.1 Verfahren des kürzesten Wegs · *106*
5.3.1.1 Mechanismus · *106*
5.3.1.2 Implementierungsaspekte · *108*
5.3.2 Shaping für Echtzeit-kodierte Videodaten · *109*

6 Leistungsuntersuchung · *113*
6.1 Simulationen · *113*
6.1.1 Methodik · *113*
6.1.2 Nachgebildete Szenarien · *114*
6.2 Simulationsdaten · *115*
6.2.1 Datensätze · *115*
6.2.2 Empirische Hüllen · *115*
6.3 Ein/Aus-Shaper · *118*
6.3.1 Leistung · *118*
6.3.2 Relevante Zeitebenen · *120*
6.4 MLB-Shaper · *122*
6.4.1 Bewertung der Verkehrshülle · *122*
6.4.2 Einfache Vorhersage · *123*
6.4.3 Vorhersage mit Kenntnis der Eingangshülle und Satz 5.4 · *123*
6.4.4 Einfluß des Verkehrsdeskriptors · *127*
6.5 KW-Shaper · *127*

7 Zusammenfassung und Ausblick · *131*

8 Literaturverzeichnis · *135*

9 Anhänge · *145*
9.1 Beweis von (5.2) · *145*
9.2 Beweis, daß (5.3) hinreichend ist für (5.1) · *145*
9.3 Beweis von (5.5) · *147*
9.4 Beweis von Lemma 5.1 · *148*
9.5 Beweis von Satz 5.1 · *150*
9.6 Beweis von Satz 5.4 · *151*

Abkürzungen

AAL	ATM Adaptation Layer
ABR	Available Bitrate
ATM	Asynchronous Transfer Mode
B-ISDN	Broadband Integrated Services Digital Network
B-picture, B-Bild	bidirectionally predictive-coded picture
BT	Burst Tolerance
CBR	constant bitrate
CAC	Call Admission Control
CDV	peak-to-peak Cell Delay Variation
CDVT	Cell Delay Variation Tolerance
CER	Cell Error Ratio
CLR	Cell Loss Ratio
CMR	Cell Misinsertion Ratio
CPCS	Common Part Convergence Sublayer
CS	Convergence Sublayer
CTD	Cell Transfer Delay
DCT	Discrete Cosine Transform
FCFS	First Come First Served
GCRA	Generic Cell Rate Algorithm
GOP	Group of Pictures
ICI	Inter-Carrier Interface
IP	Internet Protocol
I-picture, I-Bild	intra-coded picture
ISDN	Integrated Services Digital Network
ISO	International Organization for Standardization
ITU	International Telecommunication Union
JPEG	Joint Photographic Experts Group
KW	Kürzester Weg
L	Limit
LB	Leaky Bucket
LCT	Last Compliance Time
maxCTD	maximum Cell Transfer Delay
MBS	Maximum Burst Size

MCR	Minimum Cell Rate
MLB	Mehrfach-Leaky Bucket
MPEG	Moving Pictures Expert Group
MW	Moving Window
nrt-VBR	Non-Real-Time Variable Bitrate
PCR	Peak Cell Rate
P-picture, P-Bild	predictive-coded picture
PPP	Point-to-Point Protocol
QoS	Quality of Service
rt-VBR	Real-Time Variable Bitrate
SAR	Segmentation and Reassembly
SCR	Sustainable Cell Rate
SDH	Synchronous Digital Hierarchy
SECBR	Severely Errored Cell Block Ratio
SSCOP	Service Specific Connection Oriented Protocol
SSCS	Service Specific Convergence Sublayer
TAT	Theoretical Arrival Time
TCP	Transmission Control Protocol
UBR	Unspecified Bitrate
UNI	User-Network Interface
VBR	variable bitrate
VHS	Video Home System
VCI	Virtual Channel Identifier
VPI	Virtual Path Identifier

Formelzeichen und Notation

Folgende Formelzeichen werden verwendet:

b	von einem Moving Window überwachte Verkehrsmenge
$b(\tau)$	Verkehrshülle
c	Bandbreite eines Übertragungsabschnitts
$C(t)$	relativer Dekodier-Zeitplan; Menge der im Intervall $[0,t]$ zu dekodierenden Daten
$d(t)$	momentane Verzögerung
$\hat{d}$	obere Schranke für die Verzögerung
$\underline{d}$	untere Schranke für die Verzögerung
$d_{\sigma,\rho}(R)$	maximal auftretende Verzögerung eines von Verkehrsstrom $R(t)$ gespeisten (σ,ρ) -Shapers
$d_{\eth,\tilde{\rho}}(R)$	maximal auftretende Verzögerung eines von Verkehrsstrom $R(t)$ gespeisten MLB-Shapers mit Parametern $(\eth,\tilde{\rho})$
D_E	Ausspielverzögerung des Empfängers
$I_E(t)$	Verkehrsstrom am Eingang des Pufferspeichers des Empfängers
$I_S(t)$	Verkehrsstrom am Eingang des Pufferspeichers des Senders
$L(t)$	Menge der im Intervall $[0,t]$ verlorengegangenen Daten
$O_E(t)$	Verkehrsstrom am Ausgang des Pufferspeichers des Empfängers
$O_S(t)$	Verkehrsstrom am Ausgang des Pufferspeichers des Senders
$q(t)$	momentaner Quantisierungsskalierungsfaktor
$R(t)$	Verkehrsstrom; im Zeitintervall $[0,t]$ geflossene Datenmenge
$\tilde{r}$	mittlere Rate
$r(t)$	momentane Rate
r_{krit}	kritische Rate
r_{max}	maximal auftretende Rate
$s(t)$	Sprungfunktion
s_i	Größe von Bild i
s_{max}	maximale Bildgröße
t	Variable für die Zeit
T	Bildperiode
$W_\rho(R)(t)$	momentaner Füllstand eines von Verkehrsstrom $R(t)$ gespeisten Leaky Bukket mit Rate ρ
$W_{\rho,max}(R)$	maximal auftretender Füllstand eines von Verkehrsstrom $R(t)$ gespeisten Leaky Bucket mit Rate ρ

$X(t)$	momentaner Füllstand eines Netzelements
$\hat{X}$	obere Schranke für den Füllstand eines Netzelements
X_E	Kapazität des Pufferspeichers des Empfängers
X_S	Kapazität des Pufferspeichers des Senders
$X_{\sigma,\rho}(R)$	maximal auftretender Füllstand eines von Verkehrsstrom $R(t)$ gespeisten (σ,ρ)-Shapers
$X_{\hat{\sigma},\hat{\rho}}(R)$	maximal auftretender Füllstand eines von Verkehrsstrom $R(t)$ gespeisten MLB-Shapers mit Parametern $(\hat{\sigma},\hat{\rho})$
δ	Verzögerungsjitter
$\delta(t)$	Dirac-Impuls
$\varepsilon(R)(\tau)$	Empirische Hülle des Verkehrsstroms $R(t)$
ρ	Rate eines Leaky Bucket
σ	Kapazität eines Leaky Bucket
τ	Zeitdauer

Folgende abkürzende Notation wird verwendet:

$$R(s,t) = R(t) - R(s)$$
$$(a)^+ = \max\{a, 0\}$$

1 Einleitung

1.1 Übertragung von zeitabhängigen Medien und der Bedarf an Traffic Shaping

Von einem modernen Kommunikationsnetz wird heute neben hoher Geschwindigkeit vor allem die *Diensteintegration* verlangt, das heißt die Unterstützung einer breiten Palette von Anwendungen mit ganz unterschiedlichen Anforderungen. Ein Dienst im Sinne des OSI-Referenzmodells ergibt sich aus dem Zusammenwirken der von den Vermittlungsstellen im Netz erbrachten Vermittlungsleistung auf den unteren Protokollschichten bis zur Vermittlungsschicht und den Protokollfunktionen in den Endsystemen, die die höheren Schichten ab der Transportschicht beherbergen. Die asynchrone Paketmultiplextechnik, auf der die Paketvermittlung basiert, scheint am besten geeignet, effizient mit heterogenen Datenraten von Verkehrsströmen zurechtzukommen. Außerdem scheint es vorteilhaft, nur grundlegende Protokollfunktionen im Netz ausführen zu lassen, und alle weiteren in den höheren Protokollen in den Endsystemen abzuhandeln. Damit sind dienstspezifische Funktionen an den Rand des Netzes verlagert, und zur Einführung eines neuen Dienstes brauchen nicht jedesmal die Vermittlungsstellen verändert zu werden. Zwei Netzkonzepte, die diesen Ansätzen folgen, sind der Asynchrone Transfermodus (ATM), und die Familie der Internet-Protokolle (IP).

Netze mit Paketvermittlung wurden traditionellerweise für die Datenkommunikation eingesetzt. Anwendungen, wie z.B. Datensicherung oder Verteilung von Software, stellen hohe Anforderungen an die Fehlerfreiheit der übertragenen Daten, aber keine besonderen Anforderungen an die Übermittlungsdauer. Das Netz wird bei solchen Anwendungen effizient genutzt, wenn keine Verluste durch Überlast auftreten. Ein Paketverlust bedeutet, daß ein Teil der Übertragungskapazität (bis zum Verlust des Paketes) umsonst verbraucht wurde. In Extremfällen können bei hohem Verkehrsangebot die Vermittlungsstellen im Netz zwar einen hohen Durchsatz aufweisen, der nützliche Durchsatz Ende-zu-Ende kann aber wegen zu hohen Verlusten und Wiederholungen zusammenbrechen. Weil die Anwendungen nicht zeitkritisch sind, kann ein dynamischer Flußkontrollmechanismus eingesetzt werden, der auf Überlast im Netz reagiert. Geeignete Protokolle sorgen dafür, daß der Sender den momentanen Lastzustand erfährt und bei zu hoher Last die eingespeiste Verkehrsmenge herunterregelt. Bei ATM leistet das der Available Bitrate Service (ABR) explizit durch Signalisierung der Vermittlungsstellen, bei IP das Transmission Control Protocol (TCP) implizit durch Erkennen von Paketverlusten aus ausbleibenden Quittierungen der Gegenstelle. Unter Verwendung solcher Protokolle reagieren die Anwendungen *elastisch* auf eine Überlast durch Reduzierung ihres Verkehrsaufkommens.

Mit einem diensteintegrierenden Netz kommen Anwendungen hinzu, die vom Netz zeitabhängige Medien transportieren lassen, wie z.B. beim Abruf von Fernsehsendungen von einem Massenspeicher oder bei Telefongesprächen, möglicherweise mit Bewegtbild. Zu den zeitabhängigen Medien gehören die kontinuierlichen Medien Audio und Video, jedoch auch die diskreten Medien Text, Grafik und Bild, wenn ihre Darstellung von der Zeit abhängt, wie es z.B. bei Untertiteln eines Films oder den einen Vortrag begleitenden Schautafeln der Fall ist. Die Medien werden als kodierte Dateneinheiten übertragen, die vom Empfänger für eine kontinuierliche Wiedergabe nach einem bestimmten Zeitplan dekodiert und präsentiert (ausgespielt) werden müssen. Diese Anwendungen stellen hohe Anforderungen an das rechtzeitige Eintreffen von Daten beim Empfänger, können aber gelegentliche Verluste oder Fehler mehr oder weniger überdecken, weil eine einzelne fehlende Dateneinheit rasch von der nächsten ersetzt wird. Der Verkehr dieser Anwendungen ist das Produkt aus dem erwähnten Zeitplan der Dateneinheiten und ihrer Größe. Zwar können spezielle Techniken der adaptiven Kodierung die Größe der Dateneinheiten beeinflussen, jedoch erkaufen sie dies mit Qualitätseinbußen. Der Verkehr ist hauptsächlich *unelastisch*. Er eignet sich nicht für die geschlossene Regelschleife, wie sie bei der Datenkommunikation eingesetzt wird. Es müssen andere Maßnahmen zur Verkehrssteuerung gefunden werden.

Für Netze basierend auf ATM wurde deshalb ein Konzept für die Verkehrssteuerung entwickelt, das auf dem Ansatz der verbindungsindividuellen Reservierung von Ressourcen basiert. Netzressourcen werden beim Vermitteln von Paketen in Vermittlungsstellen und beim Übertragen auf Übertragungsabschnitten beansprucht. Die Anwendung muß beim Aufbau einer Verbindung die Charakteristik des Verkehrs deklarieren, damit das Netz entlang des Wegs Reservierungen so vornehmen kann, daß die geforderte Dienstgüte auf jeden Fall erbracht wird. Wenn an einer Stelle des Vermittlungsweges keine freien Ressourcen mehr vorhanden sind, lehnt das Netz den Verbindungswunsch ab. Die deklarierten Verkehrscharakteristiken der Verbindungen haben entscheidenden Einfluß darauf, wieviele Verbindungen das Netz annehmen kann. Eine Anwendung kann einen Vorteil erzielen, z.B. einen geringeren Preis für die Verbindung, wenn sie eine günstigere Verkehrscharakteristik produziert und deklariert.

Die modernen Verfahren zur Kodierung von Video, wie z.B. die Standards der Moving Picture Experts Group (MPEG)-Familie, sind sehr effektiv im Entfernen von räumlicher und zeitlicher Redundanz aus den abgetasteten Rohdaten. Zwar erreichen sie damit im Mittel eine sehr starke Verringerung der Rate, die die Verarbeitung und Übertragung von Video überhaupt erst praktikabel macht, jedoch ist der von einem Empfänger konsumierte Verkehr nicht gleichmäßig, sondern stark büschelförmig (engl. *bursty*). Die Büschelförmigkeit rührt sowohl von bestimmten Eigenschaften der Kodierung her, als auch aus dem im Szenenmaterial enthaltenen Anteil an nicht-redundanter Information. Es zeigt sich, daß die Perioden erhöhter Aktivi-

tät auf mehreren Zeitebenen gleichzeitig auftreten. Die Dauern reichen von einzelnen Bildperioden, d.h. einige zehn Millisekunden, bis zu mehreren Minuten. Dies erschwert die Aufgabe der Verkehrssteuerung erheblich.

Verglichen mit der Videokodierung erzielen die Verfahren zur Audiokodierung keine so großen Kompressionsgewinne. Das ist auch nicht nötig, weil die Rohdaten in der Regel eine weitaus geringere Rate aufweisen. Um die Probleme mit büschelförmigem Verkehr von vornherein zu vermeiden, weil schon früh herkömmliche, durchschaltevermittelnde Netze zu unterstützen waren, sind die heute verbreitetsten Verfahren so konzipiert, daß sie eine konstante Ausgangsrate produzieren. Abgesehen von Verfahren zur Pausenunterdrückung sind Audioströme also meistens konstantratig.

Ganz anders verhalten sich die diskreten Medien Text, Grafik und Bild, wenn sie zeitabhängig sind. Ihre Verkehrsströme weisen eine sehr hohe Büschelförmigkeit auf wegen des Zeitplans, mit dem sie beim Empfänger präsentiert werden (plötzliche Bildwechsel gefolgt von langen Ruheperioden). Genauere Untersuchungen über die Charakteristik liegen aber noch nicht vor.

Das in dieser Arbeit untersuchte Medium Video kann als Musterbeispiel für zeitabhängige Medien mit schwierig zu handhabender Verkehrscharakteristik betrachtet werden.

Traffic Shaping ist ein Mittel, um die ungünstige Verkehrscharakteristik einer Quelle zu verbessern. Ein Traffic Shaper besteht aus einer Ratensteuerung und einem Pufferspeicher zum Zwischenspeichern von Dateneinheiten. Der Pufferspeicher fängt in Perioden erhöhter Aktivität der Quelle Dateneinheiten auf und die Ratensteuerung gibt sie dann mit einer wohldefinierten, möglicherweise variablen Rate an das Netz ab. Die Verbesserung der Verkehrscharakteristik wird immer erkauft mit einer Verzögerung der Dateneinheiten und dem Aufwand für die Bereitstellung genügender Kapazität im Pufferspeicher. Manche Anwendungen reagieren darauf empfindlich, insbesondere was die Verzögerung angeht, für andere stellt es keinen gravierenden Nachteil dar. In dieser Arbeit wird unter anderem den Fragen nachgegangen,

- wie sich die Übertragung von zeitabhängigen Daten in einem Netz und die Echtzeit-Anforderungen der Anwendungen in einem Modell formalisieren lassen,
- welche prinzipiellen Konstruktionsprinzipien es für die Ratensteuerung eines Shapers gibt, z.B. um eine gegebene Verkehrscharakteristik zu erzielen,
- wie die Abwägung zwischen verbesserter Verkehrscharakteristik einerseits und Verzögerung und Speicherkapazität andererseits sich prinzipiell darstellt,
- wie gut sich reale Videoquellen für das Traffic Shaping eignen, und
- wie sich verschiedene Ansätze für Traffic Shaper in Leistung und Implementierungsaufwand unterscheiden.

1.2 Übersicht über die Arbeit

Kapitel 2 stellt die Grundlagen der Komprimierung und Übertragung von digitalem Video vor. Die Anforderungen verschiedener Anwendungen an die Echtzeit-Übertragung von Video werden untersucht und klassifiziert. Es ergeben sich zwei grundlegende Szenarien, bezeichnet als "gespeicherte Videodaten" und "Echtzeit-kodierte Videodaten", die unterschiedlich empfindlich auf die Ende-zu-Ende Verzögerung reagieren. Dann wird auf die Kodierung bewegter Bilder eingegangen mit ihren verschiedenen Verfahren zur räumlichen und zeitlichen Redundanzminderung. Die Darstellung orientiert sich an den von ISO und ITU gemeinsam spezifizierten MPEG-Standards (Moving Pictures Expert Group), deren Einsatz heute weit verbreitet ist. Aus der Kenntnis dieser Verfahren können schon im voraus einige qualitative Aussagen über die Charakteristik des kodierten Videostroms gemacht werden. Am Schluß des Kapitels werden speziell die Mechanismen näher beleuchtet, die ein Empfänger einsetzt, um einen Videostrom gleichmäßig ohne Unterbrechungen wiederzugeben. Aus diesen werden dann detaillierte Anforderungen an die Vermittlungsleistung des Netzes abgeleitet.

Im Kapitel 3 werden die Prinzipien der Architektur von ATM-Netzen dargestellt unter dem Blickwinkel der Diensteintegration. Nach der Beschreibung der spezifischen Festlegungen der ITU für diese spezielle Variante der Paketvermittlung wird ausführlich auf die Rolle der Verkehrssteuerung zur Sicherstellung einer verläßlichen Dienstgüte eingegangen. Es wird gezeigt, wie auf verschiedenen Zeitebenen verschiedene Ursachen für Verzögerungsschwankung und Verlust wirken, und wie entsprechende Komponenten der Verkehrssteuerung jeweils eingreifen. Genauer betrachtet wird das Konzept der verbindungsindividuellen Reservierung von Netzressourcen, die an der Benutzer/Netz-Schnittstelle als Verkehrsvertrag sichtbar wird, bei dem die Anwendung ihren Verkehr genau charakterisiert und im Gegenzug das Netz genaue Garantien über die Dienstgüte geben kann. Schließlich wird näher auf das Traffic Shaping als eine Komponente der Verkehrssteuerung eingegangen und zwei Einsatzorte unterschieden, den im Innern des Netzes, und den an der Quelle zur Erzielung einer gewünschten Verkehrscharakteristik. Letzterer ist Gegenstand dieser Arbeit.

In Kapitel 4 wird eine verkehrstheoretische Modellierung der Videoübertragung in ATM-Netzen eingeführt, die mit zeit- und wertkontinuierlichen Verkehrsmengenfunktionen operiert. Insbesondere wird der Ansatz von Cruz aufgegriffen, einen Verkehrsstrom mit einer deterministischen, zeitunabhängigen oberen Schranke für die während bestimmter Zeitdauern geflossenen Verkehrsmengen zu charakterisieren. Diese Charakterisierung hat den Vorteil, auf allen Zeitebenen gültig zu sein. Die Leistungsfähigkeit der Charakterisierung wird anhand eines einfachen Verfahrens zur Zuweisung von Netzressourcen demonstriert, das deterministischen Garantien für die Dienstgüte geben kann. In einem weiteren Abschnitt wird der Modellierungsansatz der Verkehrsmengenfunktionen auf die drei bei der Übertragung beteiligten

Elemente angewendet, den Sender mit seinem Traffic Shaper, den Empfänger mit seinem Mechanismus zum Ausgleich von Verzögerungsschwankungen, und das Netz, das insgesamt als ein Element modelliert wird. Es wird gezeigt wie Echtzeitanforderungen sich als Schranken für Füllstände von Zwischenspeichern ausdrücken lassen und schließlich das Problem des Traffic Shaping sich als zweidimensionales Optimierungsproblem in der Menge/Zeit-Ebene darstellen läßt. Dabei beeinflußt die Verzögerung, die der Verkehrsstrom im Netz erfährt, die Randbedingungen, und bei der Abwesenheit geeigneter Maßnahmen der Verkehrssteuerung kann die Verzögerung stochastisch schwanken. Es werden daher zwei verschiedene Ansätze diskutiert: Berücksichtigung der Verzögerungsschwankungen schon im voraus im Traffic Shaper des Senders, oder erst im nachhinein im Empfänger.

Kapitel 5 widmet sich den Verfahren für das Traffic Shaping. Das Ziel ist die Erzielung einer für das Netz möglichst günstigen Verkehrscharakteristik bei möglichst geringer Verzögerung und Speicherkapazität für den Shaper. Die Verfahren werden aus zwei sehr verschiedenen Ansätzen entwickelt. Der erste Ansatz baut komplexe Shaper als Hintereinanderschaltung einfacher, auf dem Leaky Bucket-Mechanismus basierendender Shaper auf. Dazu wird die von Cruz entwickelte Theorie der Shaper-Verzögerungen um entsprechende Formeln für den Speicherbedarf erweitert. Der zweite Ansatz geht vom erwähnten Optimierungsproblem aus und löst es direkt mit geometrischen Mitteln. Die sich aus beiden Ansätzen ergebenden Verfahren werden qualitativ in ihrer Leistung bezüglich Verkehrscharakteristik, Verzögerung und Speicherkapazität verglichen und es wird jeweils auf die Implementierungskomplexität eingegangen.

In Kapitel 6 wird der qualitative Vergleich durch eine quantitative Untersuchung mittels Simulation mit realen Videodaten ergänzt. Zunächst wird untersucht, wie gut sich verschiedene Datensätze für das Traffic Shaping eignen und welche Zeitebenen im Videomaterial die ausschlaggebende Rolle für die Leistung des Traffic Shaping spielen. Bei den aus mehreren Leaky Bucket-Stufen gebildeten Shapern wird der Frage nachgegangen, wie die Leistung von der Anzahl der Stufen und damit vom Implementierungsaufwand abhängt. Ebenfalls Untersuchungsgegenstand ist die Frage wie präzise die Beschreibung des Verkehrsstroms im Verkehrsvertrag sein sollte und wann sich eine Verbesserung nicht mehr lohnt. Schließlich werden die Shaper, die aus dem direkten Ansatz hervorgehen und wesentlich komplexer sind, bezüglich ihrer Leistung mit den Leaky Bucket-Shapern verglichen.

Kapitel 7 faßt die Ergebnisse der Arbeit zusammen und gibt einen Ausblick auf die zukünftige Entwicklung von verteilten Multimedia-Anwendungen, bei denen Traffic Shaping eine wichtige Rolle spielen wird.

2 Komprimierte digitale Videodaten

Die Entwicklung redundanzmindernder Kompressionsalgorithmen [Stei93] und leistungsfähiger Prozessorhardware hat die Speicherung und den Transport von Videodaten in digitaler Form möglich gemacht. Die Behandlung von Video als (fast) normale Computerdaten bringt einen wesentlichen Zugewinn an Flexibilität im Umgang mit diesem Medium, der bestehenden Anwendungen zugute kommt (z.B. kostengünstige Videokonferenzen, individualisiertes Fernsehen), und völlig neue Anwendungen ermöglicht (Integration von Video mit Text und Bildern in Multimedia-Anwendungen).

2.1 Anwendungen

2.1.1 Klassifikation der ITU-T

Die ITU-T hat in [ITU-T I.121] Teledienste mit Breitband-Anforderungen klassifiziert. In diesen Diensten spielt Video eine wichtige Rolle, so daß damit auch eine Einteilung verschiedener Anwendungen von komprimiertem Video gegeben ist. Die ITU-T unterscheidet zunächst zwischen interaktiven Diensten (engl. *Interactive Services*) und Verteildiensten (engl. *Distribution Services*). Die interaktiven Dienste werden unterteilt in

- *Dialogorientierte Dienste* (engl. *Conversational Services*): Austausch von in Echtzeit generierten Audio-, Video- oder Dokument-Daten zwischen zwei oder mehreren individuellen Benutzern. Dazu zählen die Videotelefonie oder Videokonferenzen.
- *Nachrichtenorientierte Dienste* (engl. *Messaging Services*): Kommunikation zwischen individuellen Benutzern nach dem Speichervermittlungsprinzip (engl. *store and forward*). Dazu zählt die elektronische Post (E-Mail).
- *Abrufdienste* (engl. *Retrieval Services*): Zugriff auf gespeicherte, allgemein zugängliche Informationen auf individuelle Anfrage eines Benutzers. Dazu zählen Video-Abrufdienste (engl. *video on demand*).

Die Verteildienste werden unterteilt in

- *Dienste ohne individuelle Präsentationssteuerung* (engl. *Broadcast Services Without User Individual Presentation Control*): Verteilung eines Informationsstroms an eine unbegrenzte Zahl autorisierter Benutzer, ohne daß der Benutzer in den Ablauf eingreifen kann. Diese Dienste sind dem bekannten Rundfunk und Fernsehen ähnlich.

- *Dienste mit individueller Präsentationssteuerung* (engl. *Distribution Services With User Individual Presentation Control*): Hier werden Informationseinheiten periodisch verteilt. Der Benutzer kann in Grenzen den Anfang und evtl. die Reihenfolge der Informationseinheiten bestimmen.

Im folgenden werden diese Dienste auf ihre Echtzeitanforderungen hin untersucht.

2.1.2 Echtzeitanforderungen

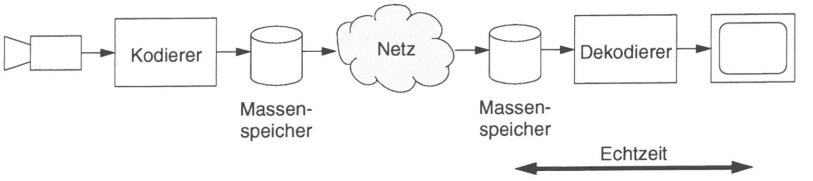
Video ist ein kontinuierliches Medium, im Gegensatz zu diskreten Medien wie Text, Grafiken oder Bilder. Bei seiner Präsentation enstehen Echtzeitanforderungen an Endgeräte und Netze. Diese fallen unterschiedlich aus als Funktion des von der Anwendung gesetzten Kontexts, und zwar abhängig davon, ob die Präsentation an die Übertragung oder Generierung der Daten gekoppelt ist. Bezüglich der Echtzeitanforderungen lassen sich die Teledienste in drei Klassen einteilen (siehe Bild 2.1):

1. Nur Echtzeitanforderungen an das präsentierende Endgerät bei interaktiven, nachrichtenorientierten Telediensten mit Speichervermittlung (Bild 2.1 a). Ein Massenspeicher im Empfänger entkoppelt die Übertragung von der Präsentation. Da an das Netz keine Echtzeitanforderungen gestellt werden, wird dieser Fall nicht weiter betrachtet.
2. Echtzeitanforderungen an das Netz und das präsentierende Endgerät bei Verteildiensten und interaktiven Telediensten für den Zugriff auf gespeicherte Informationen (Bild 2.1 b). Hier werden die Daten in Echtzeit von einem Massenspeicher im Sender ausgelesen, über das Netz transportiert und im Endgerät präsentiert. Die Generierung der Daten unterliegt keinen Echtzeitanforderungen.
3. Echtzeitanforderungen an das erzeugende Endgerät, das Netz und das präsentierende Endgerät bei dialogorientierten Telediensten (Bild 2.1 c). Hier werden die Daten in Echtzeit von einem Endgerät generiert, über das Netz transportiert und in einem anderen Endgerät präsentiert.

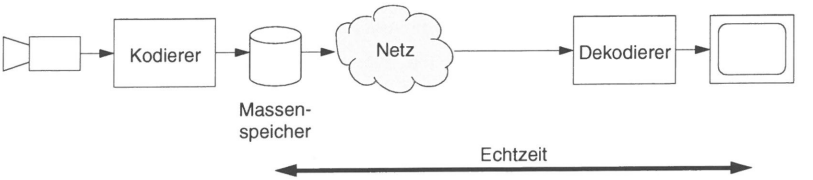
Damit verbleiben zwei Klassen von Telediensten mit Echtzeitanforderungen an das Netz, die im weiteren näher untersucht werden.

Für den Benutzer ist die Ende-zu-Ende-Verzögerung ein wichtiges Leistungsmerkmal. Eine zu große Verzögerung kann den Dienst für den Benutzer unbrauchbar machen. Im Falle der zweiten Klasse (Bild 2.1 b) ist die Ende-zu-Ende-Verzögerung die Zeitspanne zwischen der Anforderung von Videomaterial durch den Benutzer (das Drücken eines Knopfs) und der Präsentation des ersten Bilds. Diese Zeitspanne sollte möglichst klein sein. Um den Eindruck einer unmittelbaren Reaktion zu geben, muß sie etwa im Bereich einer halben Sekunde liegen.

a) interaktive Teledienste, nachrichtenorientiert



b) interaktive Teledienste für Zugriff auf gespeicherte Information; Verteildienste



c) interaktive Teledienste, dialogorientiert

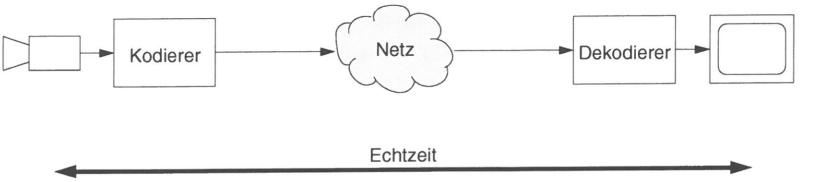


Bild 2.1: Reichweite der Echtzeitanforderungen

Die Toleranz des Benutzers gegenüber längeren Zeitspannen nimmt jedoch mit der Länge des dargebotenen Materials zu. Zum Beispiel wäre bei der Anwahl eines zweistündigen Kinofilms eine Verzögerung von vier Sekunden durchaus angemessen.

Im Falle der dritten Klasse (Bild 2.1 c) ist die Ende-zu-Ende-Verzögerung die Zeitspanne zwischen der Aufnahme eines Bild/Tonsignals am einen Endgerät und seiner Präsentation am anderen Endgerät. Bei einem Dialog zwischen Personen wird die zulässige Verzögerung vom menschlichen Reaktionsvermögen bestimmt. Die strengsten Anforderungen stellt ein Gespräch zwischen nur zwei Personen. Hier liegen Erfahrungen aus der klassischen Telefonie vor [ITU-T G.114], die zu folgenden Empfehlungen für die Planung von Telekommunikationsnetzen führen:

- Verzögerung zwischen 0 und 150 ms: akzeptabel für die meisten Anwendungen.
- Verzögerung zwischen 150 ms und 400 ms: akzeptabel unter Einschränkungen.
- Verzögerung über 400 ms: für die Planung im allgemeinen nicht mehr akzeptabel.

Die Empfindlichkeit gegenüber der Verzögerung ist in manchen Sonderfällen herabgesetzt, z.B. bei moderierten Diskussionen oder Vorträgen, bei denen durch einen expliziten Mechanismus das Wort erteilt wird und es nicht auf eine schnelle Reaktion ankommt. Hier zeigt die Erfahrung, daß die Verzögerung im Bereich einer Sekunde liegen kann.

Zwischen der Kodierung und Dekodierung wird eine Asymmetrie deutlich: während die Dekodierung immer in Echtzeit erfolgen muß, ist das in den ersten beiden Fällen für die Kodierung nicht der Fall. Hier kann der Kodierer z.B. mit zwei Massenspeichern arbeiten, einem für das kodierte und einem für das unkodierte Material, und damit das Kodieren in einer Stapelverarbeitung organisieren. Da zudem in vielen Anwendungen das Material nur einmal kodiert aber sehr oft dekodiert wird, lohnt es sich, asymmetrische Verfahren zu entwikkeln, die viel Aufwand in die Kodierung stecken, aber nur wenig in die Dekodierung.

2.2 Verfahren zur Redundanzminderung

In diesem Abschnitt werden die von der ISO und ITU normierten Verfahren zur Redundanzminderung vorgestellt. Dies ist der JPEG-Standard für Standbilder [ISO 10918-1], der MPEG-1 Standard für Bewegtbilder [ISO 11172-1] und sein Nachfolger MPEG-2 [ISO 13818-1] (JPEG steht für *Joint Photographic Experts Group*, MPEG steht für *Moving Pictures Experts Group*). Bei der Entwicklung von MPEG-1 stand die Compact Disc mit einer Datenrate von etwa 1.4 Mbit/s als Wiedergabemedium für Fernsehgeräte im Vordergrund und es wurde von Kinofilmen als Quellenmaterial ausgegangen. MPEG-2 enthält Erweiterungen, die die zeilenversetzte Halbbild-Abtastung von Videomaterial berücksichtigen, sowie eine Syntax für sogenannte skalierbare Kodierungen (darauf wird hier nicht weiter eingegangen). Außerdem werden hochauflösende Formate mit Datenraten bis zu 80 Mbit/s definiert.

2.2.1 Ziele

Wenn Videodaten in digitaler statt analoger Form gespeichert und transportiert werden sollen, erfordert dies eine digitale Redundanzminderung. Ohne sie wären die anfallenden Datenmengen nicht ökonomisch zu bewältigen. So erzeugt z.B. die Analog/Digital-Wandlung eines Fernsehsignals bei einer Auflösung von 352x288 Pixeln mit 4:2:0-Abtastung (siehe weiter unten) und 25 Hz Bildwiederholfrequenz ein Verkehrsaufkommen von 60,8 Mbit/s. Der überwiegende Teil der Daten ist aber redundant, weil sich die Bildinformation z.B. sowohl im

Raum als auch in der Zeit wenig ändert. Allgemein ausgedrückt weist das Datenmaterial typischerweise starke räumliche und zeitliche Korrelationen auf. Diese Korrelationen können für eine informationssparende Kodierung ausgenutzt werden.

Der Begriff Redundanz hat mehrere Bedeutungen. In der Informationstheorie bezeichnet Redundanz die Differenz zwischen der minimal benötigten Menge an Informationseinheiten (Bits) zur Kodierung einer Information und der dafür tatsächlich verwendeten Menge. Ein Verfahren, das diese Redundanz eliminiert, arbeitet verlustfrei, denn es kann nach dem Dekodieren der Kodewörter die ursprüngliche Information wieder zurückgewinnen. Dagegen bezeichnet redundante Information im Zusammenhang mit der menschlichen Wahrnehmung diejenige Information, die vom Sehzentrum nicht wahrgenommen wird. Ein Verfahren, das einen Teil dieser Information eliminiert, arbeitet im informationstheoretischen Sinne verlustbehaftet, jedoch nimmt der Mensch den Verlust bis zu einem gewissen Grad nicht wahr. Die im folgenden vorgestellten Kompressionsverfahren reduzieren das Volumen des kodierten Datenstroms durch verlustfreie und durch verlustbehaftete Bearbeitungsschritte. Mit dem Begriff Redundanzminderung seien hier kollektiv beide Arten von Redundanz gemeint.

Wird bei der Kodierung der Verlust an Information zu groß, schlägt die Redundanzminderung in eine wahrnehmbare Qualitätsminderung um. Wird z.B. das Verkehrsaufkommen des erwähnten Beispiels durch die im folgenden beschriebenen Verfahren auf eine konstante Rate von 1.4 Mbit/s reduziert, so ergibt sich eine Qualität vergleichbar der eines Video Home System (VHS)-Videorekorders.

2.2.2 Abtastung und Vorbehandlung

Die Datenreduktion setzt schon bei der Abtastung des analogen Videosignals ein, indem die zeitliche und räumliche Auflösung der Signalkomponenten dem menschlichen Sehvermögen angepaßt wird. Das Videosignal besteht aus einer Luminanzkomponente (Helligkeitsinformation) und zwei Chrominanzkomponenten (Farbinformation). Für die Wahrnehmung eines flüssigen Bewegungsablaufs werden in zeitlich konstanten Abständen Bilder (engl. *picture*) dieser drei Komponenten abgetastet (siehe Bild 2.2). Dabei wird die zeitliche Auflösung üblicherweise so gewählt, daß der Abtasttakt zwischen 24 und 30 Bilder pro Sekunde beträgt. Die Bilder wiederum werden horizontal und vertikal in räumlich konstanten Abständen abgetastet. Ein Abtastwert wird *Pixel* (von engl. *picture element*) genannt. Für Fernseh-Anwendungen beträgt eine typische räumliche Auflösung der Luminanzkomponente 352x288 Pixel. Die Chrominanzkomponenten werden üblicherweise mit einer verminderten Auflösung abgetastet, weil das menschliche Auge für Farbänderungen weniger empfindlich ist als für Hellig-

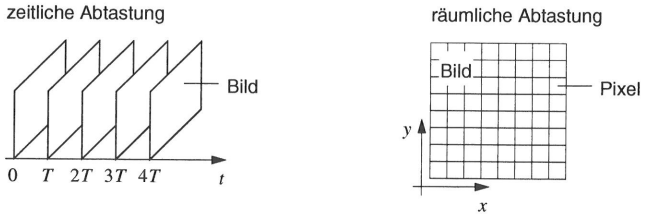


Bild 2.2: Zeitliche und räumliche Abtastung eines Videosignals

keitsänderungen. Bei der am häufigsten verwendeten 4:2:0-Abtastung ist die Auflösung der Chrominanzkomponente in horizontaler und vertikaler Richtung halbiert, bei der 4:2:2-Abtastung ist die Auflösung nur in horizontaler Richtung halbiert, und bei der 4:4:4-Abtastung ist die Auflösung identisch zur Luminanzkomponente. Die höheren Auflösungen werden meist dann gewählt, wenn das digitale Videosignal noch weiterverarbeitet werden soll. Die verminderte Auflösung könnte sonst zu Artefakten führen.

Wenn die Videoquelle Bildrauschen aufweist, ist es nach der Abtastung vorteilhaft, das typischerweise hochfrequente Rauschen in den einzelnen Bildern z.B. mit einem zweidimensionalen Tiefpaß zu dämpfen. Dadurch geht allerdings auch Bildschärfe verloren. Diese Filterung entfernt irrelevante Information und reduziert damit den Aufwand an Kodeworten in den nachfolgenden Bearbeitungsstufen.

2.2.3 Verminderung der räumlichen Redundanz

Räumliche Redundanz tritt sowohl bei festen als auch bei bewegten Bildern auf. Deshalb können in beiden Fällen dieselben Verfahren zur Redundanzminderung eingesetzt werden. Es gibt daher korrespondierende Abschnitte in den Spezifikationen des JPEG-Standards für Festbilder und den MPEG-Standards für Bewegtbilder.

Ein Verfahren für die Verminderung der räumlichen Redundanz ist die ausschnittsweise Transformation eines Bildsignals in einen anderen Raum. Eine geeignete Transformation ermöglicht zum einen die platzsparende Kodierung der Koeffizienten im transformierten Raum und zum anderen die Unterdrückung von Informationen, die für die menschliche Wahrnehmung nicht relevant sind. Neben zahlreichen anderen Verfahren hat sich die diskrete Cosinus-Transformation (DCT) [IEEE 1180] mit anschließender Quantisierung, Zick-Zack-Ordnung und Lauflängenkodierung als effizient und praktikabel herausgestellt. Das Verfahren kodiert jeweils getrennt Luminanz- und Chrominanzsignale.

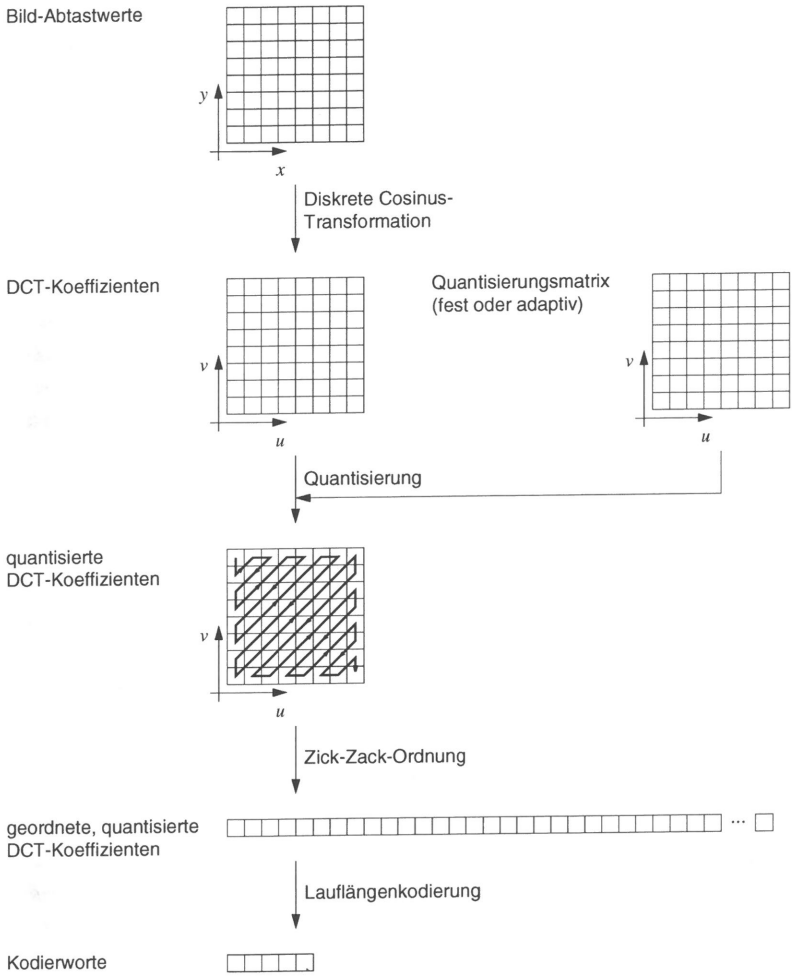


Bild 2.3: Verminderung der räumlichen Redundanz durch DCT-Transformation, Quantisierung, Zick-Zack-Ordnung und Lauflängenkodierung

Das in den Standards festgelegte Verfahren ist in Bild 2.3 dargestellt. Die Bildinformation, sei es Helligkeits- oder Farbinformation, wird in Blöcke von 8x8 Pixel aufgeteilt, wobei ein Pixel einen Wert zwischen -256 und 255 annehmen kann. Die diskrete Cosinus-Transfor-

mation transformiert diesen Block in den Frequenzbereich in eine Matrix von 8x8 DCT-Koeffizienten, die Werte zwischen -2048 und 2047 annehmen können. Eine Sequenz von aufeinanderfolgenden gleichbleibenden Werten kann durch eine Lauflängenkodierung besonders platzsparend kodiert werden. Deshalb werden die Koeffizienten in einer Zick-Zack-Ordnung nach steigender Frequenz sortiert und die Lauflängenkodierung auf die geordneten Koeffizienten angewandt. Es sind in den Standards zwei verschiedene Zick-Zack-Ordnungen festgelegt, von denen eine im Bild dargestellt ist.

Zuvor findet jedoch die Quantisierung der DCT-Koeffizienten statt, die einen entscheidenden Einfluß auf den Platzbedarf der Kodierworte hat. Die Quantisierung ist gleichförmig mit gleichgroßen Quantisierungsstufen. Eine *Quantisierungsmatrix* legt für jeden Koeffizienten die Höhe der Stufen fest. Durch die Quantisierung geht Information verloren. Mit größeren Stufen verschlechtert sich die Bildqualität und das Quantisierungsrauschen nimmt zu. Eine geeignet konstruierte Quantisierungsmatrix erlaubt nun den Informationsverlust so zu gewichten, daß vor allem diejenigen Koeffizienten gröber quantisiert werden, die vom menschlichen Auge nicht oder kaum wahrgenommen werden. Dadurch reduziert sich der Platzbedarf erheblich, ohne daß unbedingt eine signifikante Verschlechterung der Bildqualität wahrgenommen wird.

Für die Bestimmung der Quantisierungsmatrix kann der Kodierer einen unterschiedlich großen Aufwand treiben. Im einfachsten Fall bedient er sich einer im JPEG- oder den MPEG-Standards festgelegten Matrix, die empirisch aus allgemeinem Bildmaterial gewonnen wurde. Er kann auch eine eigene Matrix bestimmen, die auf eine bestimmte Anwendung oder eine bestimmte Szene hin zugeschnitten ist. Diese Matrix wird dann als Kontextinformation im kodierten Strom übermittelt. Um zusätzlich einen einfach zu handhabenden Steuerungsparameter für die kodierte Datenmenge zu erhalten, wird der *Quantisierungsskalierungsfaktor* (engl. *quantisation scale factor*) eingeführt, mit dem die Matrix vor der Quantisierung multipliziert wird, und der ebenfalls als Kontextinformation übermittelt wird. Ein großer Faktor vermindert die kodierte Datenmenge und verschlechtert die Bildqualität, während ein kleiner Faktor die Datenmenge erhöht und die Bildqualität verbessert. Der Kodierer hat damit ein sehr feines Steuerungsmittel, denn die MPEG-Syntax erlaubt es, den Faktor in sehr kurzen Abständen zu ändern, bis herab zu einzelnen Bildteilen.

Die Lauflängenkodierung bildet aus den geordneten DCT-Koeffizienten eine Folge von (Wert, Lauflänge)-Paaren. Diese Paare werden dann mittels variabel langer Kodes, ähnlich einer Huffman- oder Entropie-Kodierung, kodiert. Dabei werden den am häufigsten vorkommenden Werten die kürzesten Kodes zugewiesen. Die Standards definieren dafür eine einheitliche, feste Kode-Tabelle.

2.2.4 Verminderung der zeitlichen Redundanz

Die Verminderung der bei bewegten Bildern im Vergleich zu festen Bildern zusätzlich vorhandene zeitliche Redundanz erfordert spezielle Verfahren. Diese erlauben eine zusätzliche Kompression um etwa den Faktor 3 [LeGa91].

2.2.4.1 Intrabild- und Interbild-Kodierung

Intrabild-Kodierung (engl. *intraframe coding*) bezeichnet die Kodierung eines Bildes, bei der keine Bezüge zu anderen Bildern gemacht werden, oder allgemein zu anderer Information als der im Bild enthaltenen. Ein intrakodiertes Bild verwendet nur die Verfahren aus Abschnitt 2.2.3 und kann daher immer aus sich selbst heraus vollständig dekodiert werden. Die Interbild-Kodierung (engl. *interframe coding*) dagegen enthält Verweise auf Bilder, die in der Übertragung vorangegangen sind.

Beide Kodierungsarten können nebeneinander eingesetzt werden. Ihr Einsatz stellt einen Kompromiß zwischen sich widersprechenden Anforderungen dar. Eine starke Reduktion der Datenmengen erhält man durch Interbild-Kodierung. Eine ausschließliche Verwendung der Interbild-Kodierung ist jedoch aus zwei Gründen problematisch:

- Der wahlfreie Zugriff auf beliebige Bilder wird erschwert, weil zur Dekodierung des Bildes alle Verweise auf andere Bilder verfolgt und diese ebenfalls dekodiert werden müssen.
- Etwaige Übertragungsfehler in einem Bildbereich pflanzen sich in allen darauf verweisenden Bildern fort.

Der Kodierer kann beide Probleme dadurch mildern, daß er die Abhängigkeiten zwischen den Bildern einschränkt. Dazu verwendet er überwiegend die Interbild-Kodierung, fügt aber in regelmäßigen Abständen Intrabilder ein. Diese sind dann Startpunkte für den wahlfreien Zugriff und gleichzeitig Barrieren für die Fortpflanzung von Fehlern. Ein ausgeklügeltes Schema dafür wird in Abschnitt 2.2.4.3 vorgestellt.

2.2.4.2 Vorhersage

Das am weitesten verbreitete Verfahren zur Verminderung der zeitlichen Redundanz ist die *bewegungskompensierte Vorhersage* (engl. *motion-compensated predicition*). Das Verfahren versucht, für einzelne Bildteile eine Vorhersage zu finden, die aus Bildteilen vorhergegangener oder nachfolgender Bilder besteht. Die Vorhersage kann eventuell verschoben sein, um eine zwischen Bildern aufgetretene Bewegung zu kompensieren. Kodiert wird dann nur der Bewegungsvektor der Verschiebung und die Differenz zwischen der Vorhersage und dem tat-

sächlichen Bildteil. Das Differenzbild wird dabei genauso behandelt wie ein normaler Bildteil, indem auf dieses das normale Verfahren zur Verminderung der räumlichen Redundanz aus Abschnitt 2.2.3 angewandt wird[1]. Wenn die Differenz klein ist, werden viele aufeinanderfolgende DCT-Koeffizienten nach der Quantisierung null und in der Lauflängenkodierung durch ein einziges Kodewort dargestellt. Es ist offensichtlich, daß damit Objekte, die sich nur bewegen, aber nicht ihr Aussehen ändern, sehr platzsparend kodiert werden können.

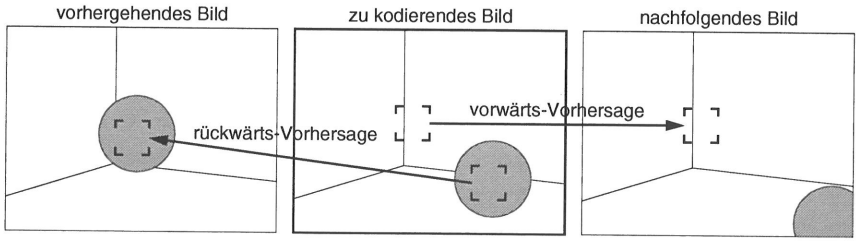


Bild 2.4: Verminderung der zeitlichen Redundanz durch rückwärts- und vorwärts-Vorhersage mit Bewegungsvektoren

Wie im Beispiel von Bild 2.4 dargestellt, läßt sich ein bewegtes Objekt im Vordergrund durch *rückwärts-Vorhersage* (engl. *backward prediction*) von einem vorhergehenden Bild beschreiben. Umgekehrt läßt sich ein durch das bewegte Objekt aufgedeckter Hintergrund durch *vorwärts-Vorhersagen* (engl. *forward prediction*) von einem nachfolgenden Bild beschreiben. Die vorwärts- und rückwärts-Vorhersage kann kombiniert werden in der *bidirektionalen Vorhersage* (engl. *bidirectional prediction*), bei dieser ist die Vorhersage der Mittelwert aus einem vorhergehenden und einem nachfolgenden Bild. Ein Vorteil dieser Methode ist, daß sie z.B. die Auswirkung von Bildrauschen auf den Vorhersage-Fehler vermindert.

Die vorwärts-Vorhersage ist bei Echtzeit-Kodierung problematisch, weil der Kodierer mit dem Kodieren warten muß, bis das referenzierte zukünftige Bild vorliegt. Dieses verursacht eine signifikante Verzögerung um eine oder mehrere Bildperioden. Deshalb wird bei dialogorientierten interaktiven Telediensten typischerweise nur die rückwärts-Vorhersage angewendet.

Die bewegungskompensierte Vorhersage bezieht sich auf einzelne Bildteile. In den MPEG-Standards [ISO 11172-2] [ISO 13818-2] wird ein Bild dafür in 16x16 Pixel große *Makroblöcke* (engl. *macro blocks*) zerlegt. Der Kodierer muß für jeden Makroblock entscheiden, ob dieser ohne Vorhersage (Intra-Kodierung) oder mit Vorhersage (Inter-Kodierung) und

1. Dabei wird jedoch eine spezielle Quantisierungsmatrix für Bild-Differenzen verwendet.

mit welcher Vorhersage kodiert werden soll. Dazu muß er die Bilder, die Kandidaten für eine Referenz sind, nach ähnlichen Bildteilen absuchen. Er muß den oder die Bewegungsvektoren finden, die die Menge der kodierten Bits minimieren, und vergleichen, ob diese Menge geringer ist als bei einer Intra-Kodierung des Makroblocks. Diese Suche nach Bewegungsvektoren ist sehr aufwendig, da üblicherweise systematisch alle möglichen Vektoren getestet werden. Für jeden Vektor muß die Menge der für die Kodierung nötigen Bits abgeschätzt werden. Deshalb schränkt der Kodierer die Suche üblicherweise auf einen bestimmten Radius ein. Insbesondere bei einer Kodierung in Echtzeit ist der Aufwand, der für die Bewegungssuche getrieben werden kann, sehr stark eingeschränkt. Darunter kann natürlich die Effizienz der Kodierung leiden, da der Kodierer schnelle Bewegungen mit großen Vektoren unter Umständen nicht erkennen kann.

2.2.4.3 Verminderung der Abhängigkeiten bei Interbild-Kodierung

Wie schon in Abschnitt 2.2.4.1 erwähnt wurde, ist es sinnvoll, die Abhängigkeiten Interbildkodierter Bilder von anderen Bildern einzuschränken. Die MPEG-Standards [ISO 11172-2] [ISO 13818-2] definieren deshalb drei Bildtypen mit unterschiedlichen Einschränkungen bezüglich der Kodierung von Makroblöcken:

- *I-Bilder* (engl. *intra-coded pictures*) enthalten nur Makroblöcke, die intra-kodiert sind, d.h. das gesamte Bild ist intra-kodiert. I-Bilder sind Startpunkte für einen wahlfreien Zugriff.
- *P-Bilder* (engl. *predictive-coded pictures*) enthalten nur Makroblöcke, die intra-kodiert sind oder inter-kodiert mit rückwärts-Vorhersage. Es dürfen nur Verweise auf das unmittelbar vorhergegangene I- oder P-Bild gemacht werden.
- *B-Bilder* (engl. *bidirectionally predictive-coded pictures*) dürfen alle Arten von Makroblöcken enthalten, also intra-kodierte oder inter-kodierte mit rückwärts- oder vorwärts- oder bidirektionaler Vorhersage. Es dürfen nur Verweise auf das unmittelbar vorhergegangene oder nachfolgende I- oder P-Bild gemacht werden.

Es ist unmittelbar klar, daß dasselbe Bild kodiert als I-Bild die größte und als B-Bild die kleinste Datenmenge benötigt. Um die Anforderungen nach wahlfreiem Zugriff und Robustheit gegenüber Fehlern zu erfüllen, wechselt der Kodierer die drei Bildtypen in einem periodischen Muster. Die MPEG-Standards definieren auf dem Videostrom eine Struktur, die *Group of Pictures* (GOP) genannt wird. Üblicherweise verwendet der Kodierer für jede GOP dasselbe Muster. Auf eine GOP muß direkt zugegriffen werden können, deshalb muß sie mit einem I-Bild beginnen. Die übrigen Bilder sind B-Bilder, in die in regelmäßigen Abständen P-Bilder zur Fehlerrobustheit eingefügt werden. Ein typisches Beispiel für gespeicherte Video-

daten ist in Bild 2.5 dargestellt. Hier umfaßt die GOP neun Bilder. Bei einer Bildfrequenz von 25 Hz erlaubt dies einen wahlfreien Zugriff mit einer Granularität von etwa einer drittel Sekunde. Bei den acht Bildern mit Referenzen folgt jeweils ein P-Bild auf zwei B-Bilder.

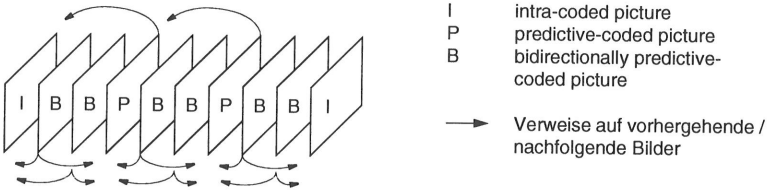


Bild 2.5: Struktur der Verweise bei MPEG-Kodierung

Weil die Dekodierung eines Bildes mit vorwärts-Vorhersage erst beginnen kann, wenn das zukünftige referenzierte Bild vorliegt, spezifizieren die MPEG-Standards eine Vertauschung der kodierten Bilder bei der Übertragung. Alle Bilder, die Ziele von Verweisen sind, müssen *in der Übertragung immer vorangehen*. Der Unterschied zwischen Aufnahme-Reihenfolge und Übertragungs-Reihenfolge ist an einem Beispiel in Bild 2.6 dargestellt.

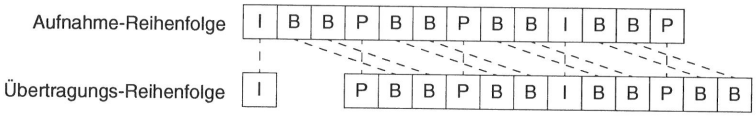


Bild 2.6: Bildreihenfolge bei MPEG-Video

2.3 Eigenschaften

2.3.1 Variable Rate

Die Menge an Bits, die der Kodierer für ein Bild benötigt, ist bei jedem Bild verschieden. Die Rate des Stroms der kodierten Bilder ist deshalb variabel. Folgende Faktoren haben einen Einfluß auf die Größe eines kodierten Bilds:

- Die räumliche Struktur eines Bildes beeinflußt die Größe von intra-kodierten Makroblöcken. Z.B. ergeben gleichmäßige Flächen kleine Makroblöcke, ungleichmäßige Strukturen dagegen große Makroblöcke.

- Die zeitliche Struktur benachbarter Bilder bestimmt, wieviele Makroblöcke mit der platzsparenden inter-Kodierung dargestellt werden können. Z.B. ergeben langsame Bewegungen, bei denen der Bewegungsvektor innerhalb des Suchradius liegt, eine große Zahl solcher Makroblöcke. Bei schnellen Bewegungen oder Schnitten dagegen muß der Kodierer auf die intra-Kodierung zurückgreifen.
- Der Typ eines Bildes, I, P oder B, legt fest, ob der Kodierer überhaupt die inter-Kodierung verwenden darf. Ein I-Bild ist in der Regel größer als ein P-Bild und ein P-Bild größer als ein B-Bild. Der typische Verlauf von Bildgrößen bei stationären Bildstrukturen ist in Bild 2.7 für verschiedene GOP-Muster gezeigt.

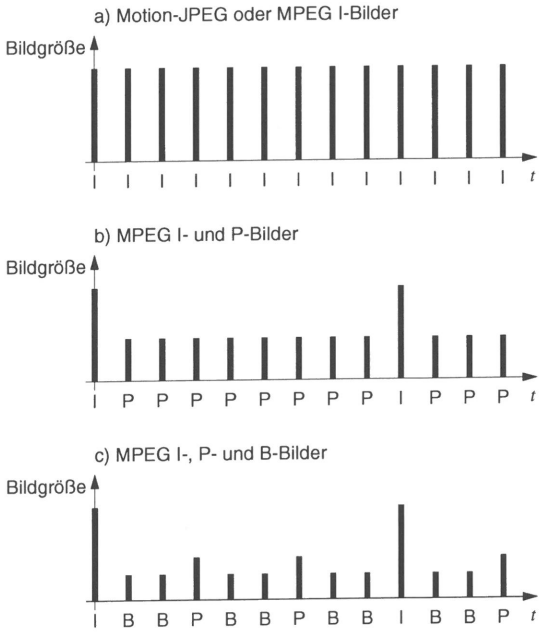


Bild 2.7: Charakteristischer Verlauf der Bildgrößen für a) Videobearbeitung, b) dialogorientierte Teledienste, c) Zugriff auf gespeicherte Information

- Die Größe des Suchradius des Kodierers für Bewegungsvektoren bestimmt, wie "gut" ein Kodierer ist, d.h. ob er schnellere Bewegungen erkennen und inter-kodieren kann. Ein kleiner Suchradius ergibt eher große Bilder, ein großer Suchradius eher kleine Bilder.
- Die Quantisierungsmatrix und damit der Quantisierungsskalierungsfaktor beeinflussen die Größe von intra- und inter-kodierten Makroblöcken. Eine grobe Quantisierung ergibt kleine Makroblöcke, eine feine Quantisierung dagegen große Makroblöcke.

2.3.2 Ratenregelung

Komprimiertes Video weist also selbst bei konstanten Kodierparametern von Natur aus eine variable Rate auf. Weil aber konstante Raten technisch einfacher handhabbar sind, sie passen z.B. besser zur Kanalstruktur von herkömmlichen durchschaltevermittelnden Netzen (siehe Abschnitt 3.2), verwenden die meisten Anwendungen heutzutage Video mit einer konstanten Rate (z.B. Kodierer für Videotelefonie nach [ITU-T H.261]). Diese konstante Rate wird mit einer Regelschleife im Kodierer realisiert, die einen oder mehrere Parameter des Kodierers verändert (siehe Bild 2.8). Ein idealer Kandidat dafür ist der Quantisierungsskalierungsfaktor.

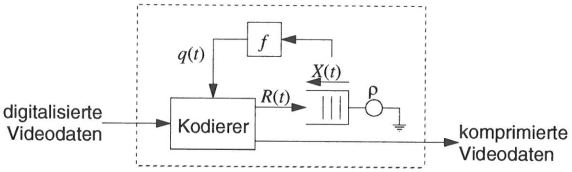


Bild 2.8: Regelung für konstante Ausgangsrate

Die gewünschte Ausgangsrate sei ρ. Der kodierte Strom gelangt in einen virtuellen Zwischenspeicher, welcher mit Rate ρ geleert wird. Der momentane Füllstand des Zwischenspeichers $X(t)$ wird zur Bestimmung des Quantisierungsskalierungsfaktors $q(t)$ herangezogen. Das geschieht über die Funktion $f: X \rightarrow q$. Die Funktion f ist monoton steigend, so daß große Füllstände eine grobe Quantisierung zur Folge haben. Sie muß in Abhängigkeit der Charakteristik des Kodierers so gewählt werden, daß der Zwischenspeicher nicht überläuft. Da es sich nur um einen virtuellen Zwischenspeicher handelt, ist die Rate am Ausgang des Kodierers nicht konstant. Wird der Strom jedoch auf einen Kanal mir Kapazität ρ gegeben, und der Kanal hat am Eingang einen realen Zwischenspeicher, so läuft dieser garantiert nicht über.

Der Nachteil einer solchen Ratenregelung liegt im schwankenden Quantisierungsskalierungsfaktor, der eine schwankende Qualität zur Folge hat. Soll eine minimale Qualität garantiert werden, so muß die Rate ρ erhöht werden, was zu einer schlechten Auslastung der Ressourcen führt.

Ein konträrer Ansatz wird in [DaTo95] verfolgt, wo das Ziel eine gleichbleibende Qualität ist. Qualität ist hier definiert als summierte Fehlerquadrate zwischen Original und dekodiertem Bild. Weil die DCT-Transformation und Quantisierung selbst bei konstanter Quantisierungsmatrix keine konstante Qualität liefert, wird der Quantisierungsskalierungsfaktor in einer Regelschleife verändert. Diesmal ist die bestimmende Größe das Ergebnis einer Qualitätsschätzung, die Original mit dekodiertem Bild vergleicht. Bei abnehmender Qualität wird der Quantisierungsskalierungsfaktor erhöht, und umgekehrt. Es ist zu erwarten, daß diese Ratenregelung eine noch größere Variabilität in der Rate erzeugt als die Kodierung mit konstantem Quantisierungsskalierungsfaktor.

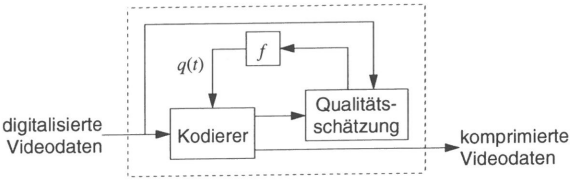


Bild 2.9: Ratenregelung für konstante Qualität

2.4 Anforderungen an die Übertragung

2.4.1 Ausgleich von Übertragungsschwankungen

Der Empfänger einer Videoübertragung stellt Echtzeitanforderungen an die Übertragung. Die Bilder müssen rechtzeitig bei ihm eintreffen, sonst enthält die Wiedergabe Diskontinuitäten, d.h. fehlende Bilder oder ruckhafte Bewegungen. Bei der Übertragung treten jedoch Verzögerungsschwankungen auf. Der Empfänger kann diese Schwankungen in einem gewissen Maß ausgleichen, indem er zwischen dem Empfang und Dekodieren der Bilder eine Verzögerung einführt, die Bilder also zwischenspeichert. Diese Verzögerung ist ein Bestandteil der Ende-zu-Ende Verzögerung und sollte daher möglichst klein gehalten werden.

Die *Ausspielverzögerung* (engl. *playback point*) des Empfängers ist die Zeitspanne, die der Empfänger mit dem Ausspielen, also Präsentieren, eines Bildes wartet. Die Zeitspanne wird relativ zu einem realen oder gedachten Produktionsprozess der Bilder beim Sender gemessen (siehe Bild 2.10). Für eine ruckfreie Wiedergabe muß die Ausspielverzögerung auf die maximal auftretende Bildverzögerung eingestellt werden.

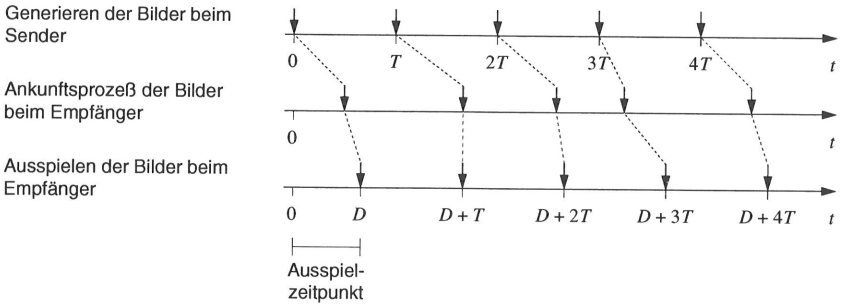


Bild 2.10: Ausspielverzögerung beim Empfänger

2.4.2 Feste und variable Ausspielverzögerung

Falls eine gewisse Verlustwahrscheinlichkeit durch verspätete Ankünfte α toleriert werden kann und die Verteilung der Bildverzögerung bekannt ist, kann der Empfänger für die Dauer der Übertragung eine feste Ausspielverzögerung einstellen. Er setzt ihn gleich dem $(1-\alpha)$ Quantil der Verteilung der Bildverzögerung (siehe Bild 2.11).

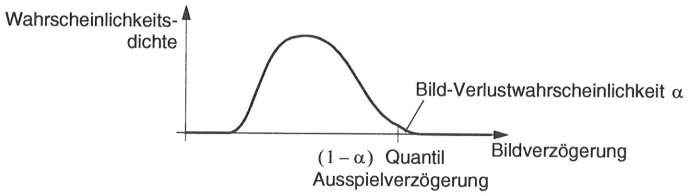


Bild 2.11: Bestimmung der Ausspielverzögerung

Für Netzumgebungen, in denen die Verzögerungsschwankungen gut kontrolliert werden können, ist dies ausreichend. Wenn jedoch die Verzögerung stark schwankt und nicht stationär ist, muß entweder ein pessimistischer, sehr große Ausspielverzögerung gewählt werden, was die Ende-zu-Ende Verzögerung verschlechtert, oder es tritt in Perioden mit großen Verzögerungen ein übermäßiger Bildverlust auf. Eine variable Ausspielverzögerung kann dieses Problem mildern [JSS93], [RKTS94]. Ein sogenannter toleranter Empfänger führt kontinuierlich Messungen der Bildverzögerung durch und schätzt ihre Verteilung. Aus der Schätzung bestimmt er die Ausspielverzögerung wie oben dargestellt. Wenn er eine Änderung in der Schätzung feststellt, verschiebt er die Ausspielverzögerung, um ihn den neuen Gegebenheiten anzupassen (siehe Bild 2.12). Jede Verschiebung hat zwar eine Diskontinuität zur Folge (für eine Verkleinerung muß ein Bild übersprungen werden, für eine Vergrößerung muß ein Bild wiederholt werden), jedoch wird die Ende-zu-Ende Verzögerung auf das kleinstmögliche Maß verringert.

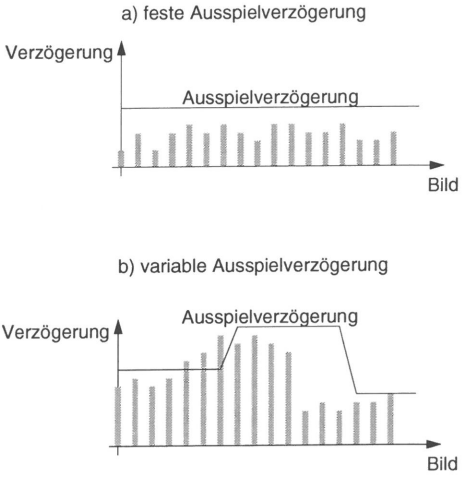


Bild 2.12: Empfänger mit a) festem und b) variabler Ausspielverzögerung

2.4.3 Synchronisation

Eine weitere Voraussetzung für die störungsfreie Wiedergabe beim Empfänger ist die Synchronisation des Empfängers mit dem Sender. Die Geschwindigkeit, mit der der Empfänger Bilder dekodiert, muß mit der Geschwindigkeit des Senders übereinstimmen. Wenn der Takt des Empfängers zu schnell ist, ist dies als ob seine Ausspielverzögerung allmählich verkleinert würde, und es treten Bildverluste auf (siehe Bild 2.13). Umgekehrt wirkt ein zu langsa-

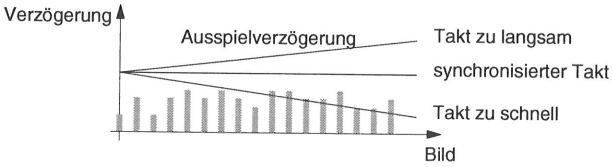


Bild 2.13: Auswirkungen fehlender Synchronisation des Empfängers

mer Takt wie eine Vergrößerung der Ausspielverzögerung und die Ende-zu-Ende Verzögerung nimmt stetig zu. Jeder Taktgeber weist Schwankungen in der Frequenz (engl. *jitter*) und konstante Abweichungen von der Nominalfrequenz (engl. *wander*) auf. Diese Abweichungen können bei den weitverbreiteten Taktgebern aus Quartzkristallen über mehrere Minuten eine Differenz von einer Bildperiode ausmachen.

Manche Netze arbeiten mit einem synchronen Netztakt, der aus einer Hierarchie synchronisierter Uhren abgeleitet wird. Zum Beispiel werden Übertragungsnetze basierend auf SDH *(Synchronous Digital Hierarchy)* [ITU-T G.783] weiträumig synchron betrieben und bestimmte ATM Schicht-1-Protokolle [ATMF94] setzen einen synchronen Netztakt voraus. In diesem Fall können Sender und Empfänger den Netztakt als gemeinsame Referenz benutzen. Der Sender kann z.B. Bilder regelmäßig mit einem Zeitstempel versehen, der sich auf den Netztakt bezieht, und der Empfänger kann dadurch seinen Taktgeber mit dem Sender synchronisieren. Viele Netze sparen jedoch den synchronen Netztakt ein, wenn er zum Betrieb des Netzes nicht zwingend ist. In diesem Fall muß der Empfänger den Takt des Senders aus dem Ankunftsprozeß der Daten regenerieren [ATMF97].

Ein Empfänger mit variabler Ausspielverzögerung kann auch die Synchronisation mit dem Sender durch Verschieben der Ausspielverzögerung realisieren [JSS93]. Der Taktgeber des Empfängers wird freilaufend angenommen. Der Empfänger mißt die Verzögerung zwischen der Ankunft eines Bildes und seiner Dekodierung. Aus den Messungen schätzt er die Verteilung und bestimmt das $(1-\alpha)$ -Quantil. Dann verschiebt er den Ausspielzeipunkt so,

daß sich eine Verlustwahrscheinlichkeit α ergibt. Bei diesem Verfahren werden Verzögerungsschwankungen im Netz nicht mehr von Schwankungen des Sendertakts unterschieden, sondern nur noch in der Summe betrachtet.

3 ATM-Netze

3.1 Diensteintegration

Bei der Entwicklung moderner Kommunikationsnetze steht heute neben der fortschreitenden Erhöhung der Übertragungsraten die Integration der Dienste im Vordergrund. In der Vergangenheit wurden spezialisierte Netzkonzepte entwickelt, um jeweils einen oder wenige miteinander verwandte Dienste zu erbringen, was die technische Realisierung vereinfachte. Damit entstanden Netze, die dediziert z.B. nur Sprache, nur Fernsehbilder oder nur Daten vermitteln konnten. Diese Trennung setzte sich bis in die Endgeräte fort. Mit der Entwicklung leistungsfähiger Übertragungstechnik und der vollständigen Digitalisierung bisher analoger Medien, wie z.B. Telephonie und Fernsehen, ergibt sich der Wunsch, nicht nur neue breitbandige Dienste einzuführen, sondern auch alle Dienste mit einem universellen Kommunikationsnetz abzudecken.

Der Asynchrone Transfermodus (ATM) wurde zu diesem Zweck ursprünglich von der ITU für das öffentliche Weitverkehrsnetz Breitband-ISDN als Nachfolger des Schmalband-ISDN konzipiert [ITU-T I.121]. Inzwischen hat sich der Einsatzbereich von ATM bedeutend erweitert. ATM spielt heute eine wichtige Rolle für den Aufbau privater Firmennetze, wo es wegen seiner Skalierbarkeit von der Weitverkehrsebene in den Bereich lokaler Netze vordringt [LPT93] [BCS93]. Der starke Entwicklungsdruck in diesem Sektor hat zur Gründung des ATM Forum geführt, eines Konsortiums hauptsächlich von Herstellern und Netzbetreibern. Das ATM Forum entwickelt möglichst frühzeitig sogenannte *"implementation agreements"*, die dann den Standardisierungsgremien vorgelegt werden [ATMF94]. Die Diensteintegration, die Flexibilität bei der Nutzung verschiedenster Übertragungsmedien und die Skalierbarkeit sind Eigenschaften, die das Vordringen von ATM in weitere Einsatzbereiche vorantreiben, so z.B. der Einsatz als Bussystem oder in drahtlosen Zugangsnetzen [LMT93].

3.2 Zeitmultiplexverfahren und Vermittlungstechniken

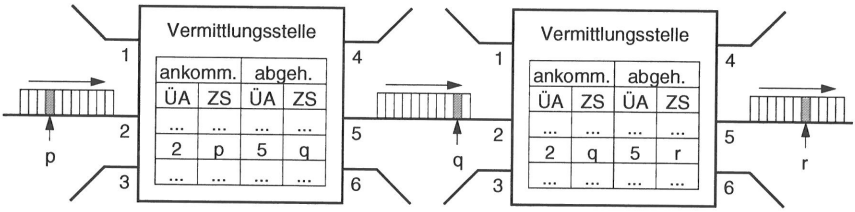
In diesem Abschnitt sollen die prinzipiellen Eigenschaften des von ATM verwendeten Zeitmultiplexverfahrens und der Vermittlungstechnik dargestellt und die wesentlichen Unterschiede zu anderen Verfahren herausgearbeitet werden.

3.2.1 Synchrones Zeitmultiplex

Das synchrone Zeitmultiplex wird bei durchschaltevermittelnden Netzen eingesetzt. Auf dem Übertragungsmedium wird ein sich periodisch wiederholender Pulsrahmen gebildet, der in Zeitschlitze konstanter Länge unterteilt ist. Die Folge gleich numerierter Zeitschlitze bildet einen Zeitkanal. Typischerweise beträgt eine Rahmenperiode 125 µs und die Größe eines Zeitschlitzes ist ein Byte. Einige Zeitschlitze des Rahmens sind reserviert zur Herstellung der Rahmensynchronisation, sowie für Betriebs- und Wartungsaufgaben, während der Rest die Nutzinformation enthält. Durch Zuweisen dieser Zeitschlitze zu Verbindungen erhält man Kanäle mit konstanter Bitrate. Ein Zeitschlitz-Byte ergibt bei einer Rahmenperiode von 125µs die Standardrate eines Basiskanals von 64 kbit/s.

Bei synchronen Übertragungshierarchien wie SDH *(Synchronous Digital Hierarchy)* [ITU-T G.783] ist der Takt aller Netzelemente im ganzen Netz synchronisiert. Dadurch ergeben sich in den Vermittlungsstellen feste zeitliche Beziehungen zwischen den Rahmen der ankommenden und abgehenden Übertragungsabschnitte. Maßnahmen zum Ausgleich eines Schlupfs können damit entfallen.

Auf jedem Übertragungsabschnitt ist eine Verbindung durch die Lage ihres Zeitschlitzes innerhalb des Rahmens gekennzeichnet. Diese Lage kann auf jedem Übertragungsabschnitt verschieden sein. Die Vermittlungsstelle verwaltet daher eine Tabelle, die für jede Verbindung den ausgehenden Übertragungsabschnitt und seinen Zeitschlitz enthält. Indiziert wird diese Tabelle durch den ankommenden Übertragungsabschnitt und seinen Zeitschlitz. Jede Vermittlungsstelle initialisiert den Tabelleneintrag einer Verbindung beim Verbindungsaufbau. Zum Durchschalten der Verbindung wird das Byte aus dem ankommenden Rahmen gelesen und bis zum Auftreten des Ziel-Zeitschlitzes verzögert (Zeitlagenumsetzung), was bis zu eine Rahmenperiode ausmachen kann. Bei mehrstufigen Koppelnetzen kann die Verzögerung auch mehrere Rahmenperioden betragen. (Siehe Bild 3.1.)



ÜA Übertragungsabschnitt
ZS Zeitschlitz

Bild 3.1: Synchrone Durchschaltung

Prinzipiell können auch einer Verbindung mehrere Zeitschlitze zugeordnet werden, um höhere Raten (Mehrfache des Basiskanals von 64 kbit/s) zu erhalten, jedoch steigt der Aufwand, die Zeitschlitze gleichzeitig in derselben Reihenfolge durchzuschalten. Die Unterstützung von sehr verschiedenen Ratenanforderungen wird dadurch schwierig.

Im Vergleich mit den anderen Multiplex- und Durchschalteverfahren bietet das synchrone Zeitmultiplex die geringste Verzögerung. Überdies ist die Verzögerung konstant. Durch die starre Zuteilung von Übertragungskapazität muß aber für eine Verbindung jeweils die Spitzenbitrate zugewiesen werden. Momentan nicht benötigte Kapazität kann nicht von anderen Verbindungen genutzt werden.

3.2.2 Nachrichtenmultiplex

Das Nachrichtenmultiplex ist ein Zeitmultiplex, das auf dem Bitstrom des Mediums Nachrichtenblöcke variabler Länge definiert. Beispiele dafür sind die ITU-Empfehlung X.25 [ITU-T X.25], Frame Relay [ITU-T I.233] und das Point-to-Point Protocol (PPP) [RFC 1661]. Ein Block besteht aus einem Kopffeld mit Adressierungs- und Steuerungsinformation und einem Feld für die Nutzdaten. Das Feld für die Nutzdaten kann verschieden lang sein. Nachrichtenblöcke werden entsprechend der betrachteten Schicht im ISO/OSI-Architekturmodell unterschiedlich bezeichnet: eine Schicht 2-Protokolldateneinheit wird als *Rahmen* (engl. *frame*) bezeichnet, eine Schicht 3-Protokolldateneinheit als *Paket* (engl. *packet*).

Die Vermittlung von Paketen kann z.B. verbindungslos (Internet Protocol (IP) [RFC 791] [RFC 1883] über PPP) oder verbindungsorientiert (X.25, Stream Protocol II (ST-II) [RFC 1819]) stattfinden. Im verbindungslosen Fall enthält das Kopffeld die vollständige Zieladresse und die Vermittlungsstelle bestimmt für jedes Paket den ausgehenden Übertragungsabschnitt aufs neue, indem es jedesmal eine Wegesuche vornimmt. Zur Erhöhung der Leistung wird das Ergebnis der Wegesuche für eine bestimmte (kurze) Zeit zwischengespeichert und für das nächste Paket mit gleicher Zieladresse wiederverwendet.

Im verbindungsorientierten Fall wird das Prinzip der virtuellen Verbindung angewandt (siehe Bild 3.2). Dabei wird im Gegensatz zum verbindungslosen Fall die Wegesuche beim Verbindungsaufbau durchgeführt und der gefundene Weg für alle Pakete der Verbindung festgelegt. Die Adreßinformation kann dann vereinfacht werden, indem ihre Gültigkeit lokal auf den Übertragungsabschnitt eingeschränkt wird. Auf jedem Übertragungsabschnitt wird eine *logische Kanalnummer* vergeben, die unter allen Verbindungen – nur auf diesem Übertragungsabschnitt – eindeutig ist. Eine Tabelle in der Vermittlungsstelle enthält für jede Verbindung den abgehenden Übertragungsabschnitt und die ihr darauf zugeteilte Kanalnummer.

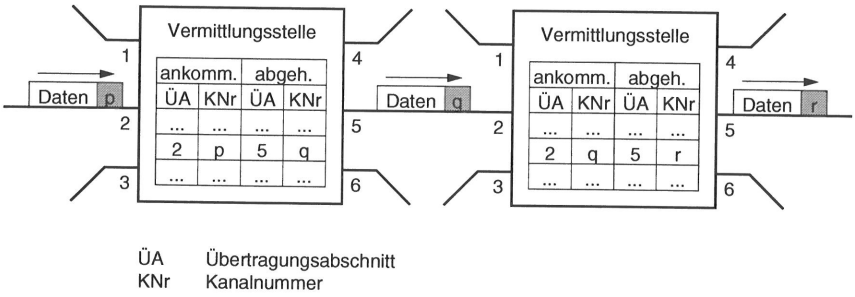


Bild 3.2: Prinzip der virtuellen Verbindung

Indiziert wird diese Tabelle durch den ankommenden Übertragungsabschnitt und die logische Kanalnummer auf diesem Abschnitt. Die Folge der Tabelleneinträge bildet die virtuelle Verbindung. Durch die Festlegung des Weges sowie die FIFO-Warteschlangenorganisation bleibt die Reihenfolge der Pakete erhalten, d.h. eine Vertauschung der Reihenfolge wie im verbindungslosen Fall ist ausgeschlossen.

Beim Vermitteln eines Paketes liest die Vermittlungsstelle die logische Kanalnummer aus dem Kopffeld und ermittelt den ausgehenden Übertragungsabschnitt aus der Tabelle. Dann überschreibt sie die Kanalnummer mit dem neuen Wert aus der Tabelle und ordnet das Paket in die Ausgangs-Warteschlange ein, in der es auf die Übertragung auf dem ausgehenden Übertragungsabschnitt wartet. Wenn der Pufferspeicher voll ist, tritt ein Verlust von Paketen auf.

Beim Nachrichtenmultiplex kann die Übertragungskapazität sehr flexibel genutzt werden. Die von einer Verbindung beanspruchte Kapazität ergibt sich aus dem Produkt der Rate der Nachrichtenblöcke und ihrer Länge. Durch den Einsatz von speziellen Bedienstrategien bei der Vermittlung der Pakete kann die Verzögerung und der Verlust sehr genau gesteuert werden [NaKu92] [Kuro93]. Zu den einfacheren Bedienstrategien gehören statische Prioritäten für die Bestimmung der Bedienreihenfolge (zeitliche Prioritäten) und die Belegung von Speicherplätzen (räumliche Prioritäten). Diese können auf der Verbindungsebene oder der Paketebene eingeführt werden: Auf der Verbindungsebene werden verschiedenen Verbindungen unterschiedliche Prioritäten zugewiesen, um z.B. verschiedene Verzögerungs- und Verlustklassen zu erhalten. Die Priorität einer Verbindung wird beim Verbindungsaufbau in die Tabellen der Vermittlungsstellen eingetragen. Auf der Paketebene werden den Paketen innerhalb einer Verbindung unterschiedliche Prioritäten zugewiesen, um z.B. die Wichtigkeit von Paketen anzuzeigen, die dann in verschiedenen Verlustklassen berücksichtigt wird. Diese Paketprioritäten werden in das Kopffeld der Pakete eingetragen. Daneben sind auch Bedienstrategien möglich, die mit Hilfe von Zeitstempeln im Kopffeld für jedes Paket die im Netz

aufgelaufene Verzögerung berechnen und darauf basierend die Bedienreihenfolge festlegen [virtual clock, predictive service]. Diese Bedienstrategien haben den Vorteil, daß sie eine globale Sicht der Verzögerung haben und damit selektiver die Pakete bevorzugen können, die aufgrund ihrer bisherigen Verzögerung eine schnelle Bearbeitung benötigen.

Ein prinzipieller Nachteil der Paketvermittlung besteht im relativ großen Aufwand beim Vermitteln eines Paketes. Dieser Aufwand begrenzt die Leistung einer Vermittlungsstelle in der Zahl der Pakete, die pro Zeiteinheit vermittelt werden können. Aus Effizienzgründen sind daher möglichst lange Pakete erwünscht. Lange Pakete führen jedoch zu zwei unerwünschten Effekten: (1) Die *Paketierungsverzögerung* (engl. *packetization delay*) im sendenden Endsystem steigt. Das ist die Zeitspanne, bis eine Quelle ein Paket vollständig mit Daten gefüllt hat. Die Paketierungsverzögerung macht sich besonders bei Quellen mit niedrigen Bitraten, wie z.B. Telephonie mit Raten von 64 kbit/s und darunter, bemerkbar. (2) Die Blockierung eines ausgehenden Übertragungsabschnittes einer Vermittlungsstelle durch ein noch nicht vollständig gesendetes Paket nimmt zu. Aus Effizienzgründen ist das Senden eines Pakets nicht unterbrechbar (engl. *non-preemptive*). Solange das Paket übertragen wird, müssen alle anderen Pakete im Pufferspeicher warten, auch das höchstpriore Paket. Die Verzögerung, die einer Verbindung beim Durchlaufen einer Vermittlungsstelle garantiert werden kann, ist damit nach unten beschränkt durch die Zeitspanne, bis das längstmögliche Paket die Vermittlungsstelle verlassen hat. Beide Effekte wirken sich negativ auf die Ende-zu-Ende-Verzögerung aus.

3.2.3 Asynchrones Zeitmultiplex mit fester Blocklänge

Das asynchrone Zeitmultiplex mit fester Blocklänge unterscheidet sich vom Nachrichtenmultiplex in der Verwendung von Blöcken fester Länge oder *Zellen*, in denen die Länge des Kopffelds und des Informationsfeldes für alle Blöcke gleich ist. Dadurch vereinfacht sich die Speicherverwaltung der Vermittlungsstelle erheblich. Außerdem wird die schnelle Paketvermittlung angewendet, bei der die Zellen durch reine Hardwarefunktion vermittelt werden. Verglichen mit der Paketvermittlungsstelle kann so eine sehr viel größere Zahl von Zellen pro Zeiteinheit vermittelt werden. Damit wird es möglich, bei gleicher Übertragungsrate des Mediums die Zellankunftsrate zu erhöhen und dabei die Zellgröße zu vermindern. Das wiederum vermindert die Paketierungsverzögerung und die Verzögerung in den Vermittlungsstellen durch Blockierung, und damit die Ende-zu-Ende-Verzögerung der Verbindung. Eine untere Grenze für die Zellgröße ergibt sich aus Effizienzüberlegungen, die ein vernünftiges Verhältnis zwischen der Größe der Nutzlast und des Kopffelds nahelegen.

Durch die hohen Vermittlungsraten ist die Auswahl möglicher Bedienstrategien auf einfachere Verfahren beschränkt. Mit der Vergabe von Prioritäten zwischen verschiedenen Verbindungen und innerhalb einer Verbindung können mehrere Verzögerungs- und Verlustklassen gebildet werden [Krön96].

3.3 ATM-Netzkonzept

Im folgenden sollen die Aspekte des ATM-Netzkonzepts vorgestellt werden, die sowohl öffentlichen Weitverkehrsnetzen als auch privaten Netzen gemein sind und die für die Frage der Verkehrssteuerung relevant sind. Die Standards für ATM beziehen sich auf eine Reihe von Schnittstellen:

- Die Benutzer-Netz Schnittstelle zwischen dem Endsystem, oder Terminal, des Benutzers und der Vermittlungsstelle auf der Netzseite, an die es angeschlossen ist (engl. *User-Network Interface*, UNI) [ITU-T I.413] [ATMF94].
- Die Schnittstelle zwischen Vermittlungsstellen desselben Netzes (engl. *Network-Network Interface*, NNI) [ATMF96a].
- Die Schnittstelle zwischen Vermittlungsstellen verschiedener Netze (engl. *Inter-Carrier Interface*, ICI).

Dabei wird unter einem Netz die Menge der zusammenhängenden Vermittlungsstellen verstanden, die vom selben Netzbetreiber betrieben werden.

3.3.1 Referenzmodell

Das ATM Protokoll-Referenzmodell [ITU-T I.321] ist in seiner Struktur vom Schmalband-ISDN Protokoll-Referenzmodell abgeleitet. Es besitzt den üblichen schichtweisen Aufbau von Protokollen, sowie eine vertikale Teilung in eine Steuerungsebene und eine Benutzerebene oberhalb der ATM-Schicht (siehe Bild 3.3). In der Steuerungsebene liegen die Protokolle zum Austausch von Signalisierungsnachrichten, die zur Steuerung von Verbindungen eingesetzt werden. Die Protokolle in der Benutzerebene regeln die Übermittlung der Nutzinformation. In den unteren Schichten benutzen beide Ebenen dieselben Protokolle. Dies sind die Protokolle der Bitübertragungsschicht (engl. *Physical Layer*), der ATM-Schicht (engl. *ATM Layer*) und ein Teil der ATM-Adaptierungsschicht (engl. *ATM Adaptation Layer*, AAL).

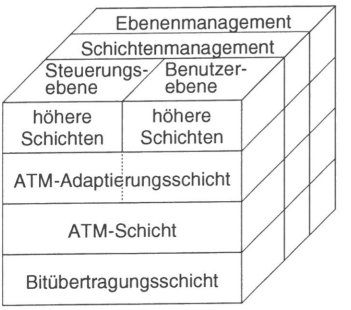


Bild 3.3: Protokoll-Referenzmodell

3.3.2 ATM-Schicht

Die ATM-Schicht [ITU-T I.361] verwendet das asynchrone Zeitmultiplexverfahren mit einer festen Blocklänge von 53 Byte. Die Blöcke (Zellen) bestehen aus einem 5 Byte langen Kopffeld und einem 48 Byte langen Feld für die Nutzinformation. Die logische Kanalnummer ist zweigeteilt mit einer Kennung für den virtuellen Pfad (engl. *Virtual Path Identifier*, VPI) und einer Kennung für den logischen Kanal (engl. *Virtual Channel Identifier*, VCI). Dies ermöglicht eine zweistufige Hierarchie logischer Verbindungen: eine Ebene der virtuellen Pfade und eine Ebene von benutzergesteuerten virtuellen Verbindungen. Beim Verbindungsaufbau tauschen die Vermittlungsstellen die Kennungen mittels Signalisiernachrichten aus.

Der Basisdienst, der von der ATM-Schicht erbracht wird, kann folgendermaßen charakterisiert werden: Die ATM-Schicht überträgt 48 Byte lange Dienstdateneinheiten über virtuelle Verbindungen. Die Reihenfolge der Dateneinheiten bleibt erhalten, jedoch können durch Fehlfunktion oder Überlast Dateneinheiten verlorengehen, bzw. durch Fehlfunktion zusätzliche Dateneinheiten eingefügt werden. Außerdem können Bitfehler die Dateneinheiten verfälschen. Die Ende-zu-Ende-Verzögerung der Dateneinheiten ist variabel. Wichtiges Kennzeichen dieses Dienstes ist die genaue Steuerung seiner Dienstgüte bezüglich Verlust und Verzögerung von Dateneinheiten durch Verkehrssteuerungsmechanismen, die in einem folgenden Abschnitt diskutiert werden.

Zur Unterstützung der ATM-Schicht gibt es mittlerweile eine große Zahl von Standards für die Protokolle der Bitübertragungsschicht, deren Kapazität von 1.5 Mbit/s bis 622 Mbit/s reicht [ITU-T I.432.1] [ATMF94]. Die Standardisierungsaktivitäten auf diesem Gebiet wer-

den vor allem davon geleitet, neben der Erschließung höherer Übertragungsraten die bestehende Infrastruktur an Übertragungsmedien, die ursprünglich für andere Protokolle entwickelt wurden, für ATM nutzbar zu machen.

3.3.3 ATM-Adaptierungsschicht

Die ATM-Adaptierungsschicht [ITU-T I.363] hat die Aufgabe, zwischen dem Basisdienst der ATM-Schicht und den speziellen Bedürfnissen der höheren Schichten zu vermitteln. Die AAL-Dienstbenutzer sind zu unterscheiden in (1) Anwendungsprotokolle der Benutzerebene und (2) das Signalisierungsprotokoll der Steuerungsebene.

Für die Benutzerebene sind, dem Ziel der Diensteintegration folgend, mehrere verschiedene AAL-Dienste vorgesehen, die jeweils von einem spezialisierten AAL-Protokoll erbracht werden. Nach dem sogenannten *core-edge* Prinzip wird diese spezialisierte Funktionalität in der Benutzerebene an den Rand des Netzes, d.h. in die Endsysteme, verlagert, während die Vermittlungsstellen mit dem Übermittlungsdienst der ATM-Schicht nur einen Basisdienst erbringen. Der Zusammenhang zwischen den verschiedenen Diensten ist in Bild 3.4 dargestellt. Ein Ziel der Trennung von AAL und ATM-Schicht ist, die ATM-Schicht, und damit die

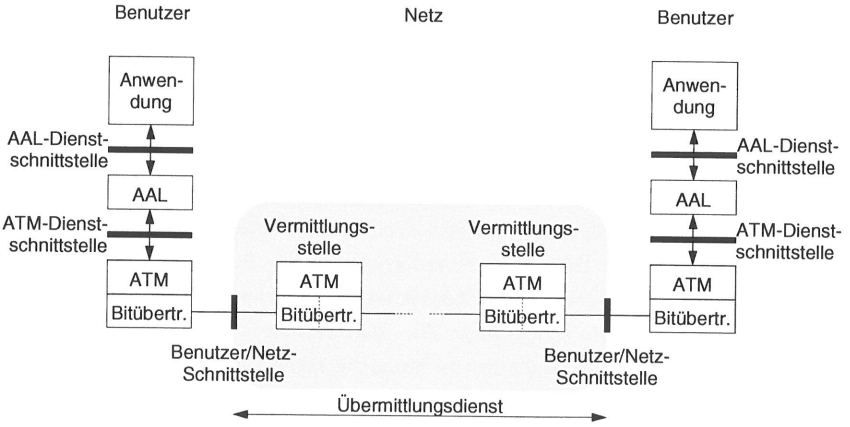


Bild 3.4: AAL-Dienst und ATM-Übermittlungsdienst in der Benutzerebene (Steuerungsebene nicht gezeigt)

Vermittlungsstellen, unabhängig von der gewählten AAL-Schicht zu machen, so daß neue AAL-Protokolle ohne Änderung der Netzinfrastruktur eingeführt werden können. Für das AAL-Protokoll der Steuerungsebene gilt das *core-edge* Prinzip nicht, da es als Transportprotokoll der Signalisiernachrichten naturgemäß in jeder Vermittlungsstelle implementiert ist.

Die Definition der Dienste und der zugehörigen AAL-Protokolle ist noch nicht abgeschlossen. Es hat sich jedoch mittlerweile eine grundlegende Unterscheidung herauskristallisiert in (Byte-) stromorientierte Dienste und paketorientierte Dienste.

Zur ersten Kategorie gehört das AAL 1-Protokoll. Es erbringt einen kanalorientierten Dienst zur Übertragung eines Bitstroms konstanter Rate mit konstanter Ende-zu-Ende-Verzögerung. Die wichtigsten Protokollfunktionen sind die Sammlung von Dienst-Datenbytes in einer Protokolldateneinheit (engl. *blocking*), die Wiederherstellung des Sendetaktes beim Empfänger und der Ausgleich von Verzögerungsschwankungen. Die Dienstgüte der verwendeten ATM-Verbindung muß eine beschränkte Verzögerung und Verzögerungsschwankung aufweisen.

Zur Kategorie der paketorientierten Protokolle gehören das AAL 3/4-Protokoll, und das AAL 5-Protokoll. Die Protokolle sind in drei Subschichten aufgeteilt, was die gemeinsame Nutzung der unteren Subschichten bei ähnlichen Diensten ermöglicht. Zunächst gibt es eine CS (*Convergence Sublayer*)-Subschicht und eine SAR (*Segmentation and Reassembly*)-Subschicht. Die SAR-Subschicht hat die Aufgabe, die um Steuerungsfelder ergänzten Pakete beim Sender in Zellen zu segmentieren und beim Empfänger wieder zusammenzusetzen. Die CS-Subschicht ist weiter unterteilt in eine CPCS (*Common Part Convergence Sublayer*)- und SSCS (*Service Specific Convergence Sublayer*)-Subschicht. In der CPCS-Subschicht sind grundlegende Protokollfunktionen zur Fehlererkennung angesiedelt. Die SSCS-Subschicht enthält spezifische Protokolle mit weitergehenden Funktionen, z.B. Fehlerkorrektur für Datendienste, wie in SSCOP (*Service Specific Connection Oriented Protocol*) [ITU-T Q.2110], das einen gesicherten Transportdienst für Signalisiernachrichten bietet, oder FR-SSCS (*Frame Relaying Service Specific Convergence Sublayer*) [ITU-T I.365.1] zur Emulation von Frame Relay über ATM. Die SSCS-Subschicht kann auch leer sein, d.h. der CPCS-Dienst wird unverändert dem Dienstbenutzer angeboten. Dies ist wichtig für echtzeitfähige Dienste, bei denen die Echtzeitbeschränkungen zeitraubende Protokollfunktionen nicht zulassen.

Der Dienst oberhalb der CPCS-Subschicht von AAL 3/4 und AAL 5 ist identisch, bis auf die Möglichkeit des Verschränkens mehrerer Pakete, die von AAL 3/4 zusätzlich geboten wird. Das Verschränken von Paketen wird beim verbindungslosen Dienst benötigt [ITU-T I.364]. Für andere Dienste, die diese Möglichkeit nicht benötigen, wird wegen seiner Einfachheit AAL 5 vorgezogen. Deshalb wird im folgenden nur auf AAL 5 eingegangen.

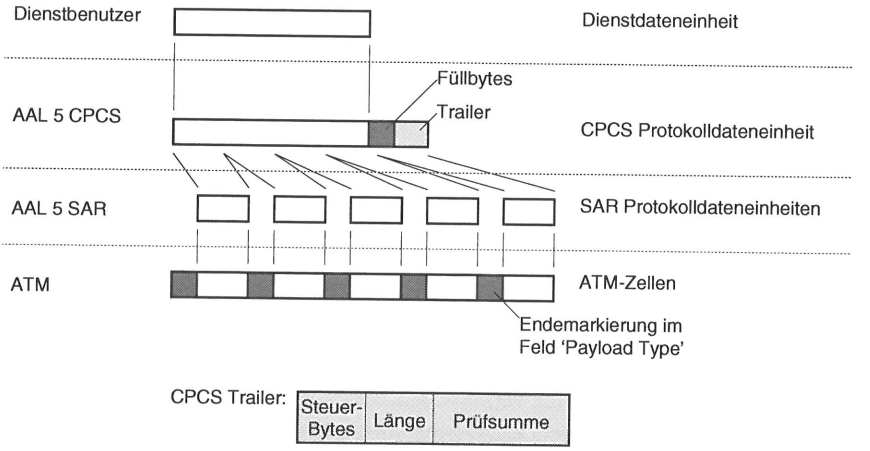


Bild 3.5: ATM-Adaptierungsschicht AAL 5

Die Blöcke, die von AAL 5 übertragen werden, können bis zu 64 KB lang sein. Die Bildung von Protokolldateneinheiten ist in Bild 3.5 dargestellt. Die AAL 5 CPCS-Protokolldateneinheit wird aus der Dienstdateneinheit der höheren Schicht gebildet, indem sie mit einem Kontrollfeld am Ende (engl. *trailer*) versehen wird, das ein Längenfeld und eine 32-bit-Prüfsumme enthält. Zwischen der Dienstdateneinheit und dem Kontrollfeld werden außerdem Füllbytes eingefügt, um die Gesamtlänge der Protokolldateneinheit auf ein Vielfaches von 48 Byte zu bringen. Damit ist sichergestellt, daß bei der anschließenden Segmentierung das Kontrollfeld immer die letzten Bytes der Nutzlast der letzten Zelle belegt. Die Prüfsumme erstreckt sich auf die gesamte Protokolldateneinheit. Mit ihr werden sowohl fehlende oder eingefügte Zellen als auch Bitfehler erkannt. Die Protokolldateneinheit wird dann von der SAR-Subschicht in ATM-Zellen segmentiert, ohne weiteren Protokolloverhead anzufügen. Das Ende der CPCS-Protokolldateneinheit wird durch eine dafür reservierte Kodierung des Feldes "Payload Type" im Kopffeld der letzten ATM-Zelle angezeigt.

3.4 Verkehrssteuerung

3.4.1 Ziele

Die Integration mehrerer unterschiedlicher Dienste in einem Netz stellt hohe Anforderungen an die Verkehrssteuerung des Netzes. Kennzeichen eines Dienstes sind auf der einen Seite die Art des Verkehrs, der in das Netz eingespeist wird, und auf der anderen Seite die Garantien bezüglich der Dienstgüte, die das Netz erbringen muß. Dabei ergibt sich die Dienstgüte, die der Benutzer erfährt, vor allem aus den Leistungsgrößen des Netzes. Einen anderen Einflußfaktor stellen die Protokolle in den Endsystemen dar. Ein diensteintegrierendes Netz bietet den Benutzern die Möglichkeit, pro Verbindung einen spezifischen Dienst zu wählen, und damit pro Verbindung eine spezifische Netzleistung einzufordern. Aufgabe der Verkehrssteuerung ist es, Überlastsituationen zu verhindern, um die geforderte, verbindungsindividuelle Netzleistung erbringen zu können. Daneben sollte die Verkehrssteuerung auch die Netzressourcen effizient nutzen.

Die ITU legt die Grundlagen der Architektur der ATM-Verkehrssteuerung in der Empfehlung I.371 [ITU-T I.371] fest. In ihrer Version vom August 1996 definiert sie vier *Transfer Capabilities*. Das ATM Forum spricht in seiner entsprechenden Spezifikation [ATMF99] von Dienstkategorien, von denen fünf definiert sind. In grossen Teilen bestehen Äquivalenzen zwischen beiden Spezifikationen, leider weicht bei inhaltlicher Äquivalenz manchmal die Namensgebung voneinander ab. Im folgenden wird auf die Festlegungen des ATM Forums eingegangen.

3.4.2 Verkehrsvertrag

Die Möglichkeit des Benutzers, pro Verbindung einen spezifischen Dienst zu wählen, läßt sich mit dem Konzept des *Verkehrsvertrags* (engl. *traffic contract*) formalisieren. Der Verkehrsvertrag wird für jede Verbindung zwischen Benutzer (Sender und Empfänger) und Netz abgeschlossen. Er regelt die Rechte und Pflichten beider Seiten während der Dauer der Verbindung. Die verschiedenen Dienste unterscheiden sich dadurch, daß ihre Verkehrsverträge unterschiedliche Rechte und Pflichten festlegen. Der Vertrag besteht aus zwei Teilen: der Festlegung einer oberen Schranke für den Verkehr, der vom Benutzer in das Netz eingespeist werden darf, und der Festlegung einer unteren Schranke für die Dienstgüte, mit der das Netz die Daten während der Dauer der Verbindung überträgt. Der Vertrag ist so abgefaßt, daß bei

einem Überschreiten der festgelegten Verkehrsschranke das Netz für den Überschußanteil nicht mehr an die vereinbarte Dienstgüte gebunden ist. Bei tarifierten Verbindungen hat der Inhalt des Verkehrsvertrags wesentlichen Einfluß auf die Entgelte.

Der Verkehrsvertrag wird beim Verbindungsaufbau mittels Signalisiernachrichten zwischen Benutzer und Netz ausgehandelt. Je nach Leistungsfähigkeit des Signalisierungsprotokolls findet eine echte Aushandlung statt (ATM Forum UNI Signalling 4.0 [ATMF96c]), oder nur eine Ja/Nein Entscheidung über die Annahme der Verbindung (ITU-T Q.2931 [ITU-T Q.2931]) (ATM Forum UNI Signalling 3.1 [ATMF94]). Dabei findet im Netz die Wegesuche der neuen Verbindung statt. Alle Vermittlungsstellen auf dem Weg sind an der Aushandlung beteiligt, denn jede Vermittlungsstelle mit ihrem ausgehenden Übertragungsabschnitt trägt einen Teil zur Ende-zu-Ende-Dienstgüte der Verbindung bei. Diese Ende-zu-Ende-Dienstgüte wird beim Verbindungsaufbau auf die einzelnen Vermittlungsstellen aufgeteilt. Durch lokale Verkehrsverträge werden diese sozusagen als "Subunternehmer" gebunden. Z.B. setzt sich die Ende-zu-Ende-Verzögerung aus der Summe der Verzögerungen in den lokalen Verkehrsverträgen zusammen.

3.4.3 Dienstgüteparameter

3.4.3.1 Verzögerungsparameter

Die *Zellübertragungsverzögerung* (engl. *Cell Transfer Delay*, CTD) zwischen Sender und Empfänger ist definiert als die Zeitspanne zwischen einem Zellabgang an der Benutzer/Netz-Schnittstelle des Senders und einer Zellankunft an der Benutzer/Netz-Schnittstelle des Empfängers. Ursachen für die Verzögerung sind Signallaufzeiten auf den Übertragungsabschnitten, feste Verzögerungen bei der Bearbeitung in Vermittlungsstellen und variable Wartezeiten in Pufferspeichern. Die Verzögerung wird als statistische Größe modelliert, die eine bestimmte Verteilung aufweist. Dabei wird das Ereignis eines Zellverlusts im Modell als unendliche Verzögerung berücksichtigt. Die Verteilung wird mit zwei Parametern beschränkt, wie in Bild 3.6 dargestellt.

Die *maximale Zellübertragungsverzögerung* (engl. *maximum Cell Transfer Delay*, max-CTD) ist definiert als das $(1-\alpha)$ -Quantil der Verteilung. Der Parameter α ist damit die Wahrscheinlichkeit für verspätete oder verlorene Zellen. Für einen Benutzer mit Echtzeitanforderungen ist das die Wahrscheinlichkeit für unbrauchbare Zellen. Der Wert von α ist durch den Parameter Zellverlustquote (s.u.) nach oben beschränkt, eine genauere Festlegung wird noch diskutiert.

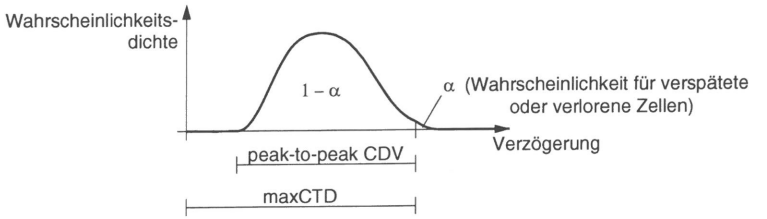


Bild 3.6: Definition der Verzögerungsparameter

Die Verzögerungsschwankung wird durch den Parameter CDV (*peak-to-peak Cell Delay Variation*) beschränkt. Er ist definiert als Differenz zwischen maxCTD und der minimal auftretenden Verzögerung, gibt also die Breite der Verteilung zwischen 0%-Quantil und $(1 - \alpha)$ -Quantil an. Dieser Parameter bestimmt bei Empfängern mit Echtzeitanforderungen, die eine konstante Verzögerung herstellen möchten, wieviel zusätzliche Verzögerung und wieviel Pufferspeicher der Empfänger zum Ausgleich der Verzögerungsschwankungen vorsehen muß.

3.4.3.2 Verläßlichkeitsparameter

Um die Verläßlichkeit einer Verbindung zu beschreiben, werden einzeln auftretende Fehlerereignisse von gehäuft auftretenden Fehlerereignissen unterschieden. Für die einzeln auftretenden Fehlerereignisse ist das Meßintervall die Dauer der Verbindung. Für die gehäuft auftretenden Fehlerereignisse betrachtet man Blöcke von N hintereinander gesendeten Zellen. Ein Block ist stark fehlerbehaftet (engl. *severely errored*), wenn mehr als M Zellen des Blocks nicht erfolgreich übertragen wurden. Die Häufigkeit eines solchen Ereignisses wird mit dem Parameter SECBR *(Severely Errored Cell Block Ratio)* wiedergegeben:

$$\text{SECBR} = \frac{\text{stark fehlerbehaftete Blöcke}}{\text{gesendete Blöcke}}$$

Gehäuft auftretende Fehlerereignisse werden aus der Statistik der einzeln auftretenden Fehlerereignisse ausgenommen.

Die einzeln auftretenden Fehlerereignisse werden weiter unterschieden in verlorene Zellen, fehlerhafte Zellen und fehlerhaft eingefügte Zellen. Eine Zelle gilt als verloren, wenn nach einer bestimmten Zeitspanne T_{max} nach dem Zellabgang an der Benutzer/Netz-Schnittstelle des Senders an der Benutzer/Netz-Schnittstelle des Empfängers noch keine Zellankunft stattgefunden hat. Ursache für verlorene Zellen sind hauptsächlich Überlastsituationen in Ver-

mittlungsstellen, wenn ein Pufferspeicher vollständig belegt ist und die Zelle nicht mehr aufnehmen kann, sowie Bitfehler im Kopffeld der Zelle, die durch die Fehlersicherung erkannt wurden und zum Verwerfen der Zelle führen. Die *Zellverlustquote* (engl. *Cell Loss Ratio*, CLR) über die gesamte Dauer einer Verbindung ist definiert als

$$\text{CLR} = \frac{\text{verlorene Zellen}}{\text{gesendete Zellen}} .$$

Eine Zelle gilt als fehlerhaft, wenn ein oder mehrere Bitfehler im Kopffeld oder in der Nutzlast aufgetreten sind (mehrfache Bitfehler im Kopffeld können dem Fehlererkennungsmechanismus entgehen). Die *Zellfehlerquote* (engl. *Cell Error Ratio*, CER) über die gesamte Dauer einer Verbindung ist definiert als

$$\text{CER} = \frac{\text{fehlerhafte Zellen}}{\text{erfolgreich übertragene Zellen + fehlerhafte Zellen}} .$$

Eine fehlerhaft eingefügte Zelle (engl. *misinserted cell*) tritt auf, wenn eine Zelle empfangen wird, für die es keine entsprechende gesendete Zelle gibt. Dies kann bei mehrfachen Bitfehlern im Kopffeld der Zelle vorkommen. Eine fehlerhaft eingefügte Zelle führt z.B. bei AAL 5 zum Verlust des gesamten Pakets. Da der Fehlermechanismus unabhängig von den gesendeten Zellen ist, ist der zugehörige Parameter CMR *(Cell Misinsertion Rate)* als Rate definiert:

$$\text{CMR} = \frac{\text{fehlerhaft eingefügte Zellen}}{\text{Zeitintervall}}$$

3.4.4 Ursachen für Verzögerungsschwankung und Verlust

Zur Untersuchung der Phänomene, die zu Verzögerungsschwankungen und Verlust führen und die durch die Verkehrssteuerung teilweise beeinflußt werden können, führt man die Betrachtung vorteilhafterweise auf verschiedenen Zeitebenen durch. Auf jeder Zeitebene werden spezifische vereinfachende Annahmen über das Verhalten der Quellen und der Netzelemente gemacht.

3.4.4.1 Verbindungsebene

Auf der Verbindungsebene wird die Gesamtzahl bestehender Verbindungen und damit die Zahl der langfristig aktiven Verkehrsquellen betrachtet. Offensichtlich führt hier eine langfristige Überlast durch zu viele Quellen zu hohen Zellverlusten. Deshalb muß die Zahl der Quel-

len begrenzt werden. Dies geschieht dadurch, daß die Quellen beim Verbindungsaufbau ihre langfristige Aktivität anmelden müssen und sie bei drohender Überlast vom Netz zurückgewiesen werden.

3.4.4.2 Burstebene

Auf der Burstebene betrachtet man die Abfolge von Phasen konstanter Aktivität einer Quelle. Es wird angenommen, daß die Summe der mittleren Raten der Verbindungen eines Multiplexers die Kapazität des Übertragungsabschnitts nicht überschreitet. Es bilden sich Warteschlangen, wenn mehrere Quellen gleichzeitig mit erhöhter Aktivität senden (engl. *burst*), und in einer momentanen Überlastsituation die Summe der momentanen Raten die Kapazität des Übertragungsabschnitts überschreitet. Grundsätzlich gibt es nun folgende Möglichkeiten:

- Das Auftreten von Bursts in einer Verbindung verhindern. Das kann dadurch geschehen, daß Übertragungskapazität aufgrund der Spitzenbitraten der Verbindungen zugeteilt wird. Nachteil ist die ineffiziente Nutzung der Kapazität.
- Eine Abschätzung der Wahrscheinlichkeit vornehmen, daß mehrere Bursts aufeinandertreffen und die Summe der momentanen Raten die Übertragungskapazität überschreitet, und sicherstellen, daß diese Wahrscheinlichkeit vernachlässigbar klein ist. Dieses *statistische Multiplexen* ist nur möglich bei einer großen Zahl von Quellen, die statistisch unabhängig sind.
- Den Pufferspeicher so groß dimensionieren, daß er die Überlast zwischenspeichern kann. Dafür müssen die Quellen die Größe der Bursts deklarieren. Dadurch wird zwar der Zellverlust klein gehalten, es ergeben sich aber Verzögerungsschwankungen.
- Eine *Quellflußregelung* einführen, die dem Netz die Steuerung der Senderaten der Quellen ermöglicht, und damit das Auftreten einer Überlast verhindert. Wegen der unvermeidlichen Verzögerungen in der Regelungsschleife muß der Pufferspeicher für die Aufnahme einer Überlast während einer Regelungsperiode dimensioniert werden. Nachteil ist das Auftreten hoher Verzögerungen. Auch haben nicht alle Quellen die Möglichkeit, ihre Senderate an äußere Vorgaben anzupassen.

3.4.4.3 Zellebene

Auf der Zellebene werden die Raten der Verbindungen als konstant angenommen. Es ergeben sich zunächst Verzögerungsschwankungen durch die Aufteilung des Bitstroms des Mediums in Zeitschlitze. Wenn die momentane Zellrate einer Verbindung nicht ein Teiler der Zellrate

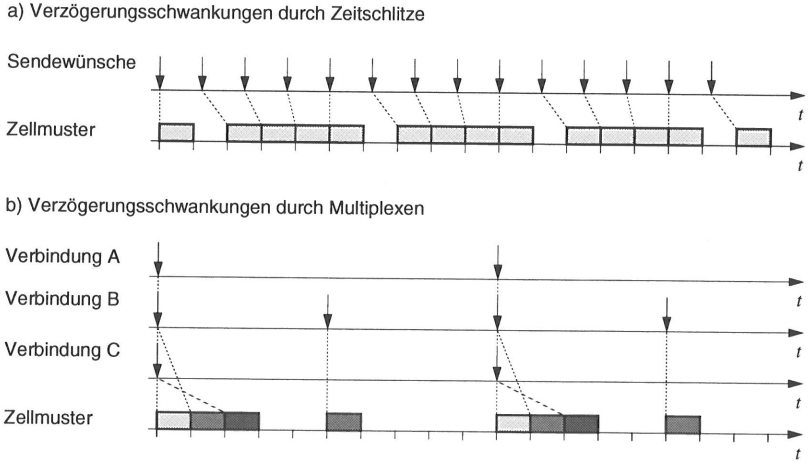


Bild 3.7: Verzögerungsschwankungen beim Sender

des Übertragungsabschnitts ist, ergeben sich Verzögerungsschwankungen bis zu einer Zellperiode. In Bild 3.7 a) ist als Beispiel das Zellmuster an der Benutzer/Netz-Schnittstelle einer konstantratigen Quelle mit einer Zellrate von 80% des Übertragungsabschnitts gezeigt.

Eine weitere unvermeidliche Quelle von Verzögerungsschwankungen auf der Zellebene ist das gleichzeitige Aufeinandertreffen einzelner Zellen beim Multiplexen mehrerer Verbindungen auf einen Übertragungsabschnitt [RoGu92]. In Bild 3.7 b) ist das Beispiel dreier konstantratiger Quellen mit Raten von jeweils 10%, 20% und 10% des Übertragungsabschnitts gezeigt. (Obwohl hier für ein Endsystem gezeigt, tritt dieses Beispiel genauso auch in Vermittlungsstellen auf.) Die Quellen haben hier eine identische Phasenlage, wodurch sich für die dritte Quelle eine Zellverzögerungsschwankung von zwei Zellperioden ergibt. Offensichtlich beträgt im allgemeinen bei Annahme konstanter Raten die Zellverzögerungsschwankung im ungünstigsten Fall bei N Quellen $N-1$ Zellperioden. Diese Abschätzung ist bei einer kleinen Zahl von Quellen brauchbar, wie sie z.B. bei der Betrachtung des Zellstroms an der Benutzer/Netz-Schnittstelle eines Senders vorliegen kann.

Für die Dimensionierung des Pufferspeichers einer Vermittlungsstelle mit einer großen Zahl von Quellen, z.B. beim Multiplexen mehrerer hundert Sprachverbindungen, ist die vorige Abschätzung zu pessimistisch. Dafür kann z.B. das $N \cdot D/D/1$ System von [RoVi91] als Modell herangezogen werden. Bei diesem Modell überlagern sich N periodische Quellen mit einer Rate $1/D$ im Verhältnis zur Rate des Übertragungsabschnitts. Die Phasenlage wird

als unabhängig und gleichverteilt angenommen. Für $N = 1000$ ergibt sich z.B. bei einem Verkehrsangebot von 0.95 und mit Wahrscheinlichkeit 10^{-10}, daß die Warteschlange länger als 84 Zellen ist. Bei einer Mischung von Quellen mit unterschiedlichen Raten zeigt sich, daß die Warteschlangenlänge stark von der Gesamtzahl der Quellen abhängt und mit ihr zunimmt.

Durch die Verzögerungsschwankungen rücken manche Zellen der Verbindung näher zusammen, andere weiter auseinander. Diese Effekte werden als *Cell Clumping* und *Cell Dispersion* bezeichnet [GuMo92]. Insgesamt erhöht sich damit die Büschelförmigkeit der Verbindung. Beim Durchlaufen mehrerer Multiplexstufen wird der Effekt des Cell Clumping verstärkt [Gole91]. Wenn der Effekt stark genug ausgeprägt ist, kann er Auswirkungen auf der Burstebene haben.

3.4.5 Komponenten der Verkehrssteuerung

Die Verkehrssteuerung besteht aus mehreren, eng miteinander verflochtenen Komponenten. Die Komponenten besitzen Mechanismen zur Überlastabwehr, die verhindern, daß Überlastsituationen entstehen, und Mechanismen zur Überlastbegrenzung, die beim Eintreten der Überlastsituation die Auswirkungen bezüglich Intensität, Ausbreitung und Dauer minimieren. Folgende Mechanismen gehören zur Überlastabwehr:

- Das *Management von Netzressourcen* (engl. *Network Resource Management*, NRM) kann dafür eingesetzt werden, Verbindungen nach ihrer Dienstcharakteristik getrennt auf separaten Wegen zu führen. Damit können z.B. Verkehrsströme konstanter Rate von büschelförmigen Verkehrsströmen getrennt werden.
- Die *Verbindungsannahmefunktion* (engl. *Connection Admission Control*, CAC) in jeder Vermittlungsstelle entscheidet darüber, ob ein Verbindungswunsch angenommen werden kann oder nicht. Beim Verbindungsaufbau findet die Wegesuche der neuen Verbindung statt, und jede Vermittlungsstelle auf dem gewählten Weg ruft ihre Verbindungssannahmefunktion auf. Falls die Verbindung von einer Vermittlungsstelle abgelehnt wird, muß ein neuer Weg gesucht werden. Falls alle Wege fehlschlagen, muß das Netz den Verbindungswunsch ablehnen. Die prinzipielle Vorgehensweise der Verbindungsannahmefunktion ist in Bild 3.8 dargestellt. Sie führt Buch über die lokalen Verkehrsverträge sämtlicher bestehender Verbindungen. Mit Hilfe eines Modells der Vermittlungsstelle kann sie eine Abschätzung der Dienstgüteparameter vornehmen, unter der Annahme, daß die neue Verbindung angenommen würde. Wenn das Ergebnis

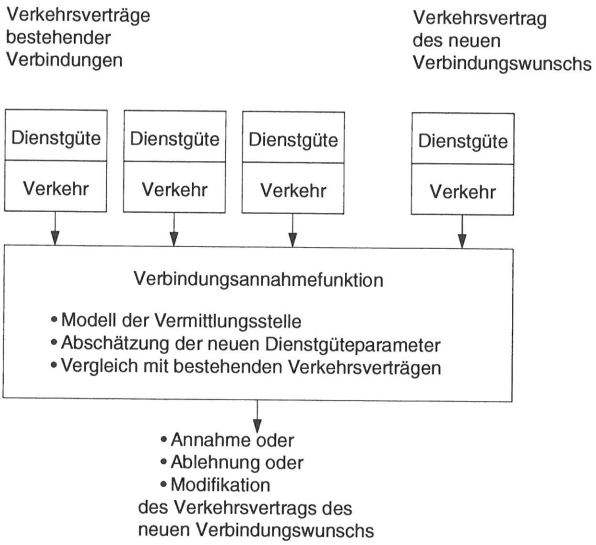


Bild 3.8: Verbindungsannahmefunktion

dieser Abschätzung keinen Verkehrsvertrag verletzt, inklusive desjenigen des neuen Verbindungswunschs, wird die Verbindung angenommen, andernfalls wird sie abgelehnt.

- Um zu überwachen, ob sich der Benutzer an seine Pflicht aus dem Verkehrsvertrag hält, führt das Netz eine *Quellflußkontrolle* (engl. *Usage Parameter Control*, UPC) möglichst nahe an der Benutzer/Netz-Schnittstelle des Senders durch. Die Quellflußkontrolle überwacht während der Dauer der Verbindung den Verkehr, der ins Netz gelangt. Falls der Benutzer die vereinbarte Schranke überschreitet, ist das Netz nicht mehr an die vereinbarte Dienstgüte gebunden und kann überschüssige Zellen sofort verwerfen oder als Überschuss markieren, der bei Überlastsituationen bevorzugt verworfen wird. In speziellen, seltenen Fällen, bei denen durch Administration darauf vertraut werden kann, daß der Benutzer die vereinbarte Schranke nie überschreitet, kann im Prinzip auf die Quellflußkontrolle verzichtet werden.
- Der Sender muß vermeiden, den im Verkehrsvertrag vereinbarten Verkehr zu überschreiten, da sonst die vereinbarte Dienstgüte nicht mehr garantiert ist. Andererseits möchte er die durch die Verbindung zur Verfügung gestellte Übertragungskapazität effi-

zient nutzen und der Schranke möglichst nahe kommen. Deshalb setzt er eine *Traffic Shaping*-Funktion ein, die den ausgehenden Verkehr in Abhängigkeit vom Verkehrsvertrag steuert.

- In den Vermittlungsstellen kommen verschiedene *Bedienstrategien* zum Einsatz. Mit Hilfe von z.B. zeitlichen Prioritäten zwischen unterschiedlichen Verbindungen können Verbindungen einer Dienstkategorie von anderen Verbindungen isoliert werden.
- Zum Entfernen des Cell Clumping kann in den Vermittlungsstellen eine *Traffic Shaping*-Funktion angewendet werden, die die Zellen jeder Verbindung individuell zwischenspeichert und mit dem richtigen Abstand wieder aussendet.
- Speziell für Datendienste, die für einen effizienten Betrieb sehr geringe Verluste fordern und gleichzeitig unempfindlich sind gegenüber Schwankungen der Übertragungsrate und der Verzögerung, wird die *ABR-Flußregelung* (engl. *ABR Flow Control*) eingesetzt. Das ist ein Regelungsmechanismus, der freie Übertragungskapazität in möglichst fairer Weise zwischen sendewilligen Endsystemen aufteilt. Dabei übt das Netz eine Flußregelung auf die Endsysteme aus, indem es regelmäßig Steuernachrichten an die Endsysteme schickt, die die aktuelle Senderate bestimmen.

Folgende Mechanismen gehören zur Überlastbegrenzung:

- Wenn in einer Überlastsituation der Pufferspeicher einer Vermittlungsstelle vollständig belegt ist, findet ein *prioritätsgesteuertes Verwerfen von Zellen* (engl. *Cell Loss Priority Control*) statt. Damit können Verbindungen mit hohen Anforderungen an die Verläßlichkeit geschützt werden.
- Speziell bei Verbindungen, bei denen eine höhere Schicht eine Fehlererkennung und eventuell Fehlerkorrektur auf Basis ganzer Pakete vornimmt (z.B. AAL 5), ist das *rahmenorientierte Verwerfen von Zellen* (engl. *Frame Discard*) vorteilhaft. Dabei erkennt die Vermittlungsstelle die Zellen, die zu einem Paket gehören und verwirft bei einer Überlastsituation alle Zellen des Pakets, statt nur einzelne Zellen zu verwerfen. Dies geschieht unter der Annahme, daß bei einem Zellverlust die höhere Schicht beim Empfänger sowieso das ganze Paket verwerfen würde. Deshalb ist es sinnvoll, die unnötigen Zellen so früh wie möglich aus dem Netz zu entfernen und die Überlast damit stärker zu reduzieren.

3.4.6 Dienstkategorien

Durch Kombination der beschriebenen Verkehrssteuerungsmechanismen können auf der ATM-Schicht verschiedene Dienstkategorien angeboten werden. Folgende fünf Dienstkategorien sind vom ATM Forum in [ATMF96b] festgelegt worden:

- *CBR (Constant Bit Rate) Service:* Dienst mit konstanter Bitrate
- *rt-VBR (Real-Time Variable Bitrate) Service:* Dienst mit variabler Bitrate und Echtzeit-Eigenschaften
- *nrt-VBR (Non-Real-Time Variable Bitrate) Service:* Dienst mit variabler Bitrate ohne Echtzeit-Eigenschaften
- *UBR (Unspecified Bitrate) Service:* Dienst ohne spezifizierte Bitrate
- *ABR (Available Bitrate) Service:* Dienst mit Zuteilung der verfügbaren Bitrate

Die in den Verkehrsverträgen der einzelnen Dienstkategorien festgelegten Attribute sind in Tabelle 3.1 zusammengefaßt.

<table>
<tr><td></td><td colspan="5">Dienstkategorie</td></tr>
<tr><td>Attribute</td><td>CBR</td><td>rt-VBR</td><td>nrt-VBR</td><td>UBR</td><td>ABR</td></tr>
<tr><td colspan="6">Verkehrsparameter</td></tr>
<tr><td>PCR, CDVT</td><td colspan="3">spezifiziert</td><td>spezifiziert[a]</td><td>spezifiziert[b]</td></tr>
<tr><td>SCR, MBS, CDVT</td><td>nicht anwendbar</td><td colspan="2">spezifiziert</td><td colspan="2">nicht anwendbar</td></tr>
<tr><td>MCR</td><td colspan="3">nicht anwendbar</td><td>nicht anwendbar</td><td>spezifiziert</td></tr>
<tr><td colspan="6">Dienstgüteparameter</td></tr>
<tr><td>peak-to-peak CDV</td><td colspan="2">spezifiziert</td><td colspan="3">nicht spezifiziert</td></tr>
<tr><td>maxCTD</td><td colspan="2">spezifiziert</td><td colspan="3">nicht spezifiziert</td></tr>
<tr><td>CLR</td><td colspan="3">spezifiziert</td><td>nicht spezifiziert</td><td>[c]</td></tr>
<tr><td colspan="6">andere Attribute</td></tr>
<tr><td>Regelungsmechanismus</td><td colspan="4">spezifiziert</td><td>nicht spezifiziert</td></tr>
</table>

Tabelle 3.1: ATM-Dienstkategorien und ihre Attribute

a. Möglicherweise nicht der Verbindungsannahmefunktion und Quellflußkontrolle unterworfen.
b. Maximal mögliche Rate, mit der die Quelle senden kann.
c. Zellverlust ist sehr gering für Quellen, die ihre Rate nach der Kontrollinformation des Regelungsmechanismus richten.

Es gibt zwei Dienstkategorien mit Echtzeit-Eigenschaften, CBR und rt-VBR. Beide sehen einen statischen Verkehrsvertrag vor, der für die Dauer der Verbindung gültig ist. (Es ist prinzipiell eine Neuverhandlung des Verkehrsvertrags während der Verbindungsdauer möglich, dies geschieht jedoch in Zeitabständen, die der Verbindungsebene zuzuordnen sind.) Er spezifiziert obere Schranken für Verzögerung, Verzögerungsschwankung und Verlust. Die Dienstkategorien unterscheiden sich in der oberen Schranke für den Verkehr. CBR beschränkt nur die Spitzenbitrate, während bei rt-VBR eine zusätzliche, langfristige Rate unterhalb der Spitzenbitrate spezifiziert wird.

Die Dienstkategorie nrt-VBR unterscheidet sich von rt-VBR nur dadurch, daß die Verzögerung und Verzögerungsschwankung nicht beschränkt ist.

Die Dienstkategorie UBR bietet eine Übertragung "nach bestem Vermögen" (engl. *best effort*) ohne spezifizierte Dienstgüte. Sie kann ohne Verbindungsannahmefunktion und Quellflußkontrolle implementiert werden. Zwar wird im Verkehrsvertrag eine Spitzenbitrate spezifiziert, jedoch kann diese vom Netz ignoriert werden. Da das Netz keine Kenntnis über den Verkehr hat, kann es auch keine Voraussagen über die Dienstgüte machen. Typischerweise werden UBR-Verbindungen für die Datenübertragung eingesetzt, solange ABR nicht verfügbar ist, und dabei im Netz das rahmenorientierte Verwerfen von Zellen angewandt [RoFl94].

Die Dienstkategorie ABR unterscheidet sich grundlegend von den anderen Dienstkategorien durch die reaktive Flußregelung, die das Netz auf den Sender ausübt. Der Verkehrsvertrag ist sozusagen in dem Teil, der die obere Schranke für den Verkehr enthält, dynamisiert. Im Verkehrsvertrag wird vereinbart eine minimale Rate (*Minimum Cell Rate*, MCR) und eine maximale Rate (*Peak Cell Rate*, PCR), zwischen denen die aktuelle Senderate schwanken kann. Diese Dienstkategorie bietet eine hohe Verläßlichkeit durch eine kleine Zellverlustquote, aber keine Schranke für die Verzögerung.

3.4.7 Verkehrsvertrag und Quellflußkontrolle für CBR und rt-VBR

3.4.7.1 Quellverkehrsdeskriptor

Die beiden Dienstkategorien CBR und rt-VBR stellen strenge Anforderungen an die vom Netz zu erbringende Verkehrsgüte. Neben dem Einsatz von Bedienstrategien, z.B. der Isolierung solcher Verbindungen durch Prioritäten, ist das wichtigste Mittel, das dem Netz zur Verfügung steht, die Beschränkung des Quellverkehrs.

Deshalb enthält der Verkehrsvertrag einen sogenannten *Quellverkehrsdeskriptor* (engl. *source traffic descriptor*). Dieser bestimmt eine obere Schranke für den Quellverkehr. Bei der CBR-Dienstkategorie ist nur eine Schranke für die maximale Zellrate, oder *Spitzenzellrate*,

definiert mit dem Parameter PCR *(Peak Cell Rate)*. Diese Dienstkategorie eignet sich damit besonders für Quellen, die mit einer konstanten Rate senden. Mit dem Quellverkehrsdeskriptor kann das Netz eine Zuteilung der Ressourcen nach der Spitzenbitrate vornehmen (engl. *peak rate allocation*).

Wenn Quellen mit variabler Rate Verbindungen der CBR-Dienstkategorie anfordern, führt dies zu einer schlechten Auslastung der Netzressourcen, weil die reservierte Übertragungskapazität nur bei Erreichen der Spitzenzellrate vollständig genutzt wird. Deshalb führt die rt-VBR-Dienstkategorie eine zusätzliche Schranke ein, die mittels eines *Leaky Bucket* formuliert wird. Der Leaky Bucket-Mechanismus mißt den Füllstand eines virtuellen Speichers, der vom Quellverkehr gefüllt wird [Turn86]. Der Speicher entleert sich mit konstanter Rate ρ, solange er nicht leer ist. Der Füllstand darf die Grenze σ nicht überschreiten, die als Kapazität des Leaky Bucket bezeichnet wird. Eine Quelle, die diese Bedingung erfüllt, nennt man konform zu einem (σ,ρ)-Leaky Bucket. Der Leaky Bucket beschränkt also die langfristige Rate der Quelle mit dem Parameter ρ. Andererseits erlaubt die Kapazität σ der Quelle, kurzfristig mit einer höheren Rate, bis hin zur deklarierten Spitzenzellrate, zu senden. Im Zusammenhang des ATM-Verkehrsvertrags wird eine spezielle Formulierung des Leaky Bucket angewandt, die weiter unten dargestellt wird.

3.4.7.2 Verbindungsverkehrsdeskriptor

Bei beiden Dienstkategorien ergibt sich ein Problem dadurch, daß der Verkehrsvertrag an der Benutzer/Netz-Schnittstelle gültig ist und der Quellverkehr bis dahin durch die in Abschnitt 3.4.4.3 beschriebenen Effekte verändert worden ist: schon im Sender tritt Cell Clumping durch Zeitschlitze und Muliplexen auf, und es können zusätzliche Verzögerungsschwankungen im Zugangsnetz auftreten. Damit ist es z.B. unmöglich, die Spitzenzellrate durch Beobachten des minimalen Zellabstands festzustellen. Deshalb wird innerhalb des Verkehrsvertrags unterschieden zwischen dem Quellverkehrsdeskriptor, der den unverfälschten Quellverkehr beschreibt, und dem *Verbindungsverkehrsdeskriptor* (engl. *connection traffic descriptor*), der an der Benutzer/Netz-Schnittstelle gültig ist und die aufgetretenen Verzögerungsschwankungen berücksichtigt.

Der Verbindungsverkehrsdeskriptor besteht aus dem Quellverkehrsdeskriptor, dem Parameter CDVT *(Cell Delay Variation Tolerance)*, der die Stärke des Cell Clumping ausdrückt, und einer Konformitätsdefinition. Die Konformitätsdefinition ist eine abstrakte Referenz für den Benutzer und das Netz, die festlegt, welcher Verkehr als konform zu gelten hat und welcher nicht. Bei der CBR-Dienstkategorie bedient sie sich eines Leaky Bucket, um die

Zellverzögerungsschwankungen bei Beobachtung der Spitzenzellrate an der Benutzer/Netz-Schnittstelle zu berücksichtigen. Bei der rt-VBR-Dienstkategorie kommt ein zweiter Leaky Bucket zur Begrenzung des langfristigen Verkehrsverhaltens hinzu.

Bevor die Konformitätsdefinition im Detail beschrieben wird, soll zunächst die Formulierung des Leaky Bucket mittels des GCR-Algorithmus dargestellt werden.

3.4.7.3 GCR-Algorithmus

Die Konformitätsdefinition verwendet eine spezielle Formulierung des Leaky Bucket-Mechanismus, den GCR *(Generic Cell Rate)*-Algorithmus, der eine eindeutige Interpretation des Mechanismus sicherstellt. Der GCR-Algorithmus bestimmt aus dem Zeitpunkt der Ankunft einer Zelle, ob die Zelle konform ist. Um von der Übertragungsrate des Mediums unabhängig zu sein, werden die Zellankünfte nicht zu diskreten Zeitpunkten betrachtet, sondern als zeitkontinuierlich angenommen.

Im folgenden soll gezeigt werden, wie aus einem wert- und zeitkontinuierlichen Leaky Bucket-Algorithmus, der den von der Zellen-Nutzlast transportierten Verkehr beschränkt, der GCR-Algorithmus abgeleitet werden kann. Betrachtet sei ein Leaky Bucket, dessen Füllstand S bei Ankunft einer Zelle sprunghaft um 48 Byte steigt und kontinuierlich mit Rate ρ abnimmt, bis er den Wert null erreicht. Die Grenze, die S nie überschreiten darf, sei σ. Offensichtlich muß gelten $\sigma \geq 48$ Byte. Bei $\sigma = 48$ Byte läßt der Leaky Bucket nur einen völlig regelmäßigen Zellstrom zu. Ein Algorithmus, der aus der Ankunftszeit einer Zelle ihre Konformität feststellt, ist in Bild 3.9 gezeigt. Der Algorithmus aktualisiert die Variable S jeweils bei einer Zellankunft und vergleicht sie dann mit der Grenze.

Durch eine Normierung der Variablen S erhält man eine Variante des GCR-Algorithmus, die *wertkontinuierlicher Leaky Bucket-Algorithmus* genannt wird (siehe Bild 3.10). Dazu substituiert man $X = S/\rho$. Die Variable X stellt die Zeitdauer dar, bis der Speicher des Leaky Bucket geleert ist. Der Parameter I *(Increment)* ersetzt den Parameter ρ mit $I = 48\ \text{Byte}/\rho$. Er stellt damit den Zwischenankunftsabstand der Zellen eines völlig regelmäßigen Zellstroms dar. Der Test auf Überschreiten der Grenze wird vorgezogen, bevor die Zelle zum Füllstand hinzugezählt wird, und der Testparameter L *(Limit)* ersetzt den Parameter σ mit $L = (\sigma - 48\ \text{Byte})/\rho$. Bei $L = 0$ läßt der Leaky Bucket nur einen völlig regelmäßigen Zellstrom zu. Ein solchermaßen parametrisierter GCR-Algorithmus wird mit GCRA(I,L) bezeichnet.

Mit einer weiteren Umformung erhält man eine zweite, äquivalente Variante von GCRA(I,L), die *Virtual Scheduling-Algorithmus* genannt wird, und die mit nur noch einer Variablen auskommt (siehe Bild 3.11). Die Variable TAT *(Theoretical Arrival Time)* ist die

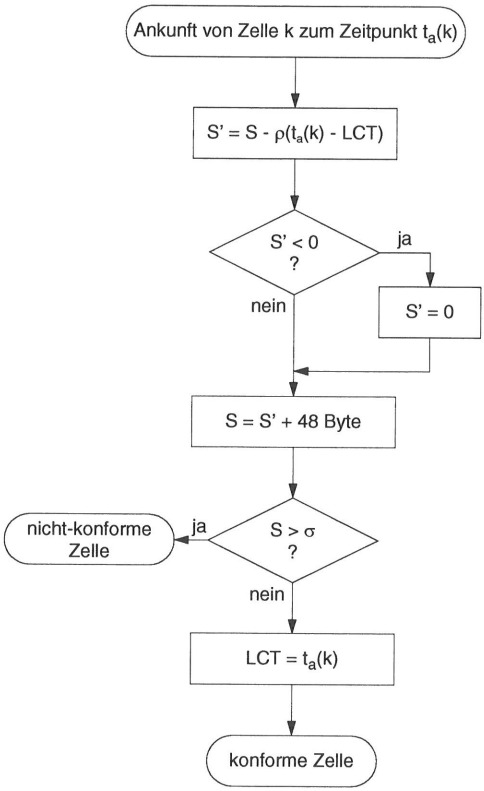


Bild 3.9: Wert- und zeitkontinuierlicher Leaky Bucket-Algorithmus

Summe aus der Ankunftszeit der letzen Zelle und der Zeitdauer, bis der Speicher des Leaky Bucket leer ist: $\mathrm{TAT} = \mathrm{LCT} + X$. TAT gibt damit den theoretischen Ankunftszeitpunkt der nächsten Zelle an, wenn der Zellstrom völlig regelmäßig wäre. Der Konformitätstest lautet jetzt $\mathrm{TAT} > t_a(k) + L$. Damit läßt sich der Parameter L neu interpretieren: er stellt die maximale Zeitdifferenz dar, um die die Zellen, verglichen mit einem völlig regelmäßigen Zellstrom, verfrüht ankommen dürfen.

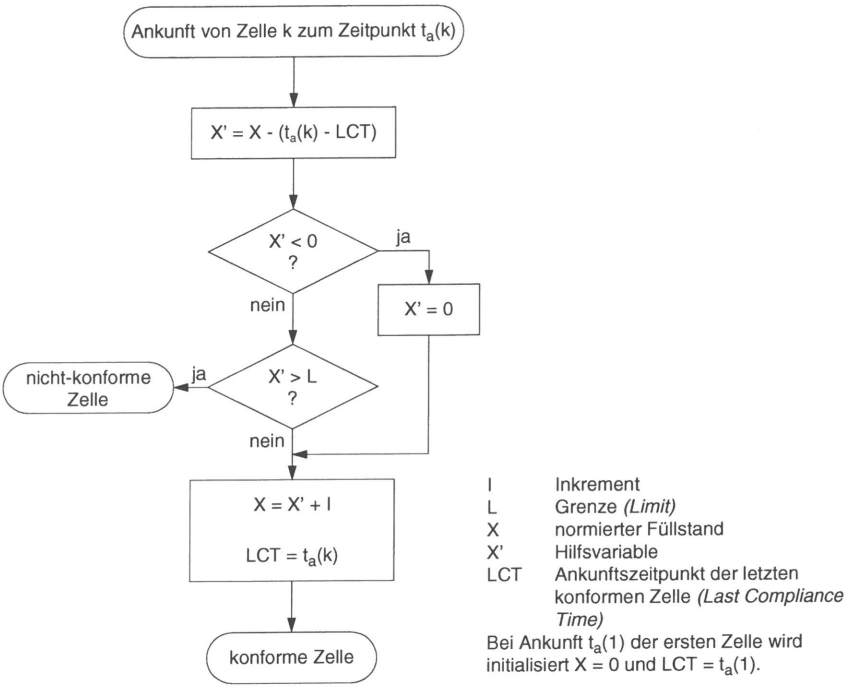


Bild 3.10: Erste Darstellung von GCRA(I,L), genannt wertkontinuierlicher Leaky Bucket-Algorithmus

3.4.7.4 Konformitätsdefinition

Für die CBR-Dienstkategorie wird eine Instanz des GCR-Algorithmus herangezogen, um die Spitzenzellrate PCR unter Berücksichtigung der Zellverzögerungsschwankung zu begrenzen. Der Parameter I ergibt sich direkt als $1/\mathrm{PCR}$. Der Parameter L stellt dann die Toleranz gegenüber Zellverzögerungsschwankungen dar. Man kann zeigen, daß ein periodischer Zellstrom, dessen Zellen eine maximal auftretende Zellverzögerungsschwankung τ aufweisen, von einem Leaky Bucket GCRA($1/\mathrm{PCR},\tau$) als konform deklariert wird. Für die CBR-Dienstkategorie wird nun die Konformität als Konformität zu GCRA($1/\mathrm{PCR}$,CDVT) festgelegt, und der Parameter CDVT so dimensioniert, daß er die bis zur Benutzer/Netz-Schnittstelle auftretenden Verzögerungsschwankungen abdeckt.

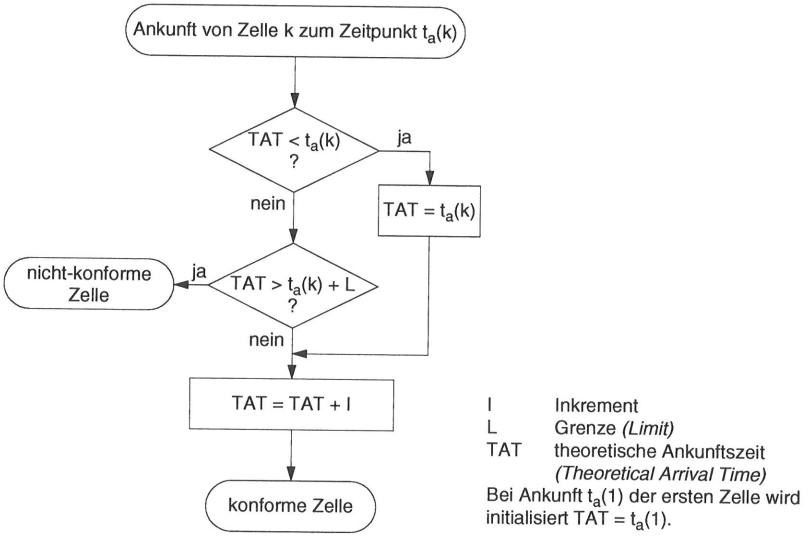


Bild 3.11: Zweite Darstellung von GCRA(I,L), genannt Virtual Scheduling-Algorithmus

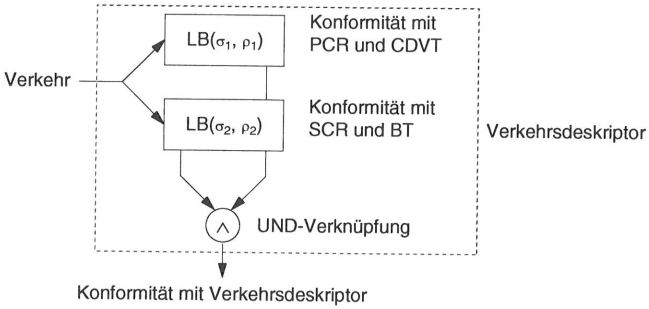


Bild 3.12: Konformitätsdefition für die rt-VBR-Dienstkategorie

Für die Konformitätsdefinition der rt-VBR-Dienstkategorie kommt zu derjenigen der CBR-Dienstkategorie eine zweite Instanz GCRA(1/SCR,BT + CDVT) hinzu, zu der der Zellstrom ebenfalls konform sein muß (siehe Bild 3.12). Der Parameter SCR *(Sustainable Cell Rate)* stellt die langfristige Rate dar (entsprechend dem Parameter ρ beim wert- und zeit-

kontinuierlichen Leaky Bucket), der Parameter BT *(Burst Tolerance)* begrenzt die Datenmenge, die mit einer Rate oberhalb SCR gesendet werden darf (entsprechend dem Parameter σ). Wie bei der CBR-Dienstkategorie berücksichtigt der Parameter CDVT die Zellverzögerungsschwankungen. Mit dem Parameter $L = \mathrm{BT} + \mathrm{CDVT}$ wird damit in einfacher Weise die Toleranz gegenüber der Büschelförmigkeit des Zellstroms an der Benutzer/Netz-Schnittstelle eingestellt. Diese hat jedoch zwei verschiedene Ursachen: die Büschelförmigkeit des Zellstroms der Quelle und die im Zugangsnetz akkumulierte Büschelförmigkeit durch Verzögerungsschwankungen.

3.5 Traffic Shaping

Wie in Abschnitt 3.4.5 schon angedeutet wurde, kann Traffic Shaping für zwei verschiedene Zwecke eingesetzt werden:

- Die Anpassung des Verkehrs einer Quelle an einen bestimmten Quellverkehrsdeskriptor. Vom Benutzer durchgeführt stellt das Traffic Shaping sicher, daß das Netz keine Zellen in der Quellflußkontrolle verwirft. Außerdem kann der Benutzer die Büschelförmigkeit des Verkehrs reduzieren, indem er einen günstigeren Verkehrsdeskriptor wählt. Dies kann ihm Vorteile bei der Tarifierung der Verbindung bringen, oder es kann, global betrachtet, zu einer besseren Netzauslastung beitragen. Diese Art von Shaping wird im folgenden *Quellshaping* genannt und in den folgenden Kapiteln genauer untersucht.
- Das Wiederherstellen der ursprünglichen Verkehrscharakteristik einer Verbindung, die beim Durchlaufen von Multiplexstufen auf der Zellebene oder Burstebene verändert wurde. Dieses wird im folgenden *Netz-Reshaping* genannt.

In allen Fällen besteht das Prinzip des Traffic Shaping darin, Bursts eines Verkehrsstroms in einem Pufferspeicher aufzufangen und über der Zeit verteilt wieder auszusenden. Traffic Shaping reduziert also die Büschelförmigkeit eines Verkehrsstroms auf Kosten einer zusätzlichen Verzögerung und eines zusätzlich benötigten Pufferspeichers.

3.5.1 Netz-Reshaping

Ein Element zum Netz-Reshaping wirkt auf Verbindungsebene. Sein prinzipieller Aufbau ist in Bild 3.13 dargestellt. Es sieht für jede Verbindung einen eigenen Shaper vor, bestehend aus einem Pufferspeicher und einer Ratensteuerung. Wenn das Element nicht schon in den Ausgang eines Koppelnetzes integriert ist, benötigt es einen Demultiplexer, der die Zellen eines Übertragungsabschnitts nach Verbindungen getrennt in die Pufferspeicher stellt. Ein Multiple-

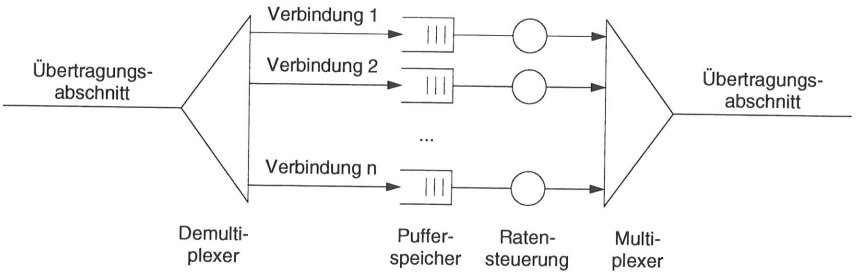


Bild 3.13: Prinzip des verbindungsindividuellen Netz-Reshaping

xer führt die Ausgänge der Ratensteuerungen wieder zusammen. In einer Implementierung wird der Pufferspeicher vorteilhafterweise von allen Verbindungen gemeinsam genutzt und es wird nur mit Verweisen auf die Speicherstellen operiert. Die Ratensteuerung und die Multiplexstufe können zusammengefaßt werden in einer *Sendeliste* aus Verweisen, die die Reihenfolge des Aussendens der Zellen festlegt. Die Sendeliste wird beim Einschreiben einer Zelle in den Zwischenspeicher jeweils aktualisiert.

Der *Spacer Controller* [BSG92] und der *Spacing Policer* [WaWo92] sind beide Elemente zum Netz-Reshaping, die anstelle einer normalen Quellflußkontrolle an der Benutzer/Netz- oder Netz/Netz-Schnittstelle eingesetzt werden können. Sie sollen wie eine Quellflußkontrolle den Verkehr überwachen und gleichzeitig das Cell Clumping ausgleichen, das durch Multiplexstufen des Zugangsnetzes oder des inneren Netzes aufgetreten ist. Beide sind für die CBR-Dienstkategorie konzipiert. Ihre Ratensteuerung reduziert die Spitzenzellrate einer Verbindung auf den im Verkehrsvertrag deklarierten Wert PCR. Wenn ein Pufferspeicher eine bestimmte maximale Länge überschreitet, werden die Zellen der Verbindung verworfen; damit wird die Quellflußkontrollfunktion realisiert. In der Multiplexstufe des Shapers treten die unvermeidlichen Verzögerungsschwankungen auf Zellebene durch Zeitschlitze und Multiplexen (Abschnitt 3.4.4) auf. Verglichen mit einer normalen Quellflußkontrolle für PCR und CDVT reduziert also das Element den Parameter CDVT auf Kosten einer zusätzlichen Verzögerung.

In [Chao92] und [KuWu96] sind Erweiterungen dieses Ansatzes für die rt-VBR-Dienstkategorie beschrieben. Dabei muß nur die Ratensteuerung abgeändert werden, so daß sie zusätzlich die Einhaltung der Parameter SCR und BT sicherstellt.

3.5.2 Quellshaping

Das Quellshaping wird vom Benutzer durchgeführt. Der Verkehr der Quelle gelangt in einen Pufferspeicher und wird von dort mit einer Ratensteuerung in das Netz gesendet. Wenn der Shaper nahe an der Quelle angeordnet ist und die Verkehrsproduktion der Quelle von außen beeinflußt werden kann, ergibt sich die Möglichkeit der Rückwirkung des Shapers auf die Quelle in einem geschlossenen Regelkreis. (Zu den Anwendungsbeispielen gehört die Dateiübertragung, aber auch steuerbare Videokodierer.) Damit kann der Shaper ein Überlaufen des Pufferspeichers verhindern. Bei Shapern mit offenem Regelkreis (Steuerung) muß ein Mechanismus zur Überlastbegrenzung Zellen verwerfen, oder der Shaper muß so dimensioniert sein, daß ein Überlaufen nicht möglich ist.

Die Auswirkung des Quellshapings mit offenem Regelkreis beim Multiplexen mehrerer Quellen mit büschelförmigem Verkehr wurde in [Nies93] und [BGLS93] untersucht. In [Nies93] reduziert der Shaper nur die Spitzenbitrate der Quelle. In [BGLS93] sendet der Shaper mit N verschiedenen, diskreten Raten; die Rate nimmt mit dem Füllstand des Pufferspeichers monoton zu. Betrachtet werden jeweils sporadische Quellen mit negativ-exponentiell-verteilter Dauer der Aktivitäts- und Pausephasen. In beiden Fällen wurde bei einer gegebenen Verlustwahrscheinlichkeit eine deutlich bessere Auslastung des Multiplexers festgestellt.

In [RVF91] wurde ein Shaper mit geschlossenem Regelkreis für eine einzelne Quelle betrachtet. Der Shaper ist zweistufig. Die erste Stufe besteht aus einer Quellflußkontrolle, die bei einer Überschreitung der zu überwachenden Parameter auf die Quelle zurückwirkt und sie stoppt. Der verwendete Mechanismus ist ein Sliding Window. Die zweite Stufe ist ein Shaper, der die Spitzenbitrate des durchgelassenen Verkehrs reduziert. Auch hier liegt der Leistungsbetrachtung eine sporadische Quelle zugrunde mit zeitdiskreten geometrisch verteilten Dauern der Aktivitäts- und Pausephasen; aus dieser werden Dimensionierungsregeln für die Shaper-Parameter abgeleitet.

3.5.3 Stochastische analytische Leistungsuntersuchung des Quellshapings

Bei einer stochastischen analytischen Leistungsuntersuchung werden stochastische Modelle herangezogen, um den Verkehr einer Quelle nachzubilden. Diese werden dann in eine mathematische Beschreibung des Shaper-Mechanismus eingesetzt. Leistungsgrößen wie Verzögerung oder Füllstand eines Zwischenspeichers sind dann abgeleitete Zufallsvariablen. Über diese Zufallsvariablen kann man statistische Aussagen machen, zum Beispiel ist es manchmal möglich, die Verteilung als geschlossenen mathematischen Ausdruck zu erhalten.

Shaper verändern die Form von Bursts, die im Quellverkehr auftreten. Eine Leistungsuntersuchung betrachtet daher die Zeitebene von einzelnen Bursts. Auf dieser Zeitebene ist der diskrete Charakter von Zellen nicht mehr sichtbar und der Verkehr kann durch seine momentane Rate beschrieben werden. Daher bieten sich Flüssigkeitsmodelle (engl. *fluid flow*) [AMS82] [Kost84] als Werkzeuge an.

In [RMV96] werden ATM-Netze einschließlich von Shapern ausführlich analytisch betrachtet. Ein Ansatz zur Untersuchung von Shapern und ATM-Multiplexern betrachtet zunächst ein G/G/1-Wartesystem mit deterministischer Bedienzeit, dessen virtuelle Wartezeit mittels der Benes-Methode bestimmt werden kann. Der Benes-Ansatz wird auf Flüssigkeitsmodelle angewendet. Es werden Lösungen unter anderem für sporadische Quellenmodelle mit unabhängig und allgemein verteilten Aktivitäts- und Pausedauern hergeleitet (siehe auch [RBC93]). Dieser Ansatz wird dann für die Analyse eines Quellshapers verwendet. Es gelingt damit, die Verteilung des virtuellen Füllstands des Shapers zu bestimmen. Daraus lässt sich die Wartezeit der letzten Zelle eines Bursts bestimmen. Diese Wartezeit kann als Maß für die Transparenz des Shapers aufgefaßt werden.

Eine weitere Analyse in [RMV96] gilt der Wirkung von Quellshaping auf den Ressourcenbedarf im Netz. Es wird gezeigt, wie für ein spezielles Quellenmodell der Bedarf an Pufferspeicher in einem Multiplexer bestimmt werden kann. Dem Quellenmodell liegt ein Diffusionsprozeß zugrunde; die im Zeitintervall $[s,t]$ in den Shaper fließende Verkehrsmenge ist $A(s,t) = mt + \sigma W(s,t)$, wobei $W(s,t)$ die von einer normalen Brown'schen Bewegung erzeugte Arbeit ist. Mittels der Theorie der großen Abweichungen angewendet auf den Benes-Ansatz gelingt es, einen näherungsweisen Ausdruck für die Füllstands-Verteilung des Multiplexers zu finden (siehe auch [BRS95]).

In [Ritt95] wird für das Quellenmodell ein Erneuerungsprozess verwendet, unter anderem eine Negativ-Binomialverteilung. Es wird ein zweistufiger Shaper untersucht mittels einer iterativen Methode. Der Shaper besteht aus einer Stufe, die eine gegebene Spitzenzellrate (PCR) sicherstellt, und einer Stufe, die eine gegebene langfristige Zellrate (SCR) mit einer Burst-Toleranz BT sicherstellt. Außerdem wird noch eine maximale Verzögerung berücksichtigt: wenn eine Zelle länger als diese Zeitdauer zwischengespeichert werden muß, um konform zu sein, wird sie verworfen. Als Leistungsgrößen werden ermittelt die Zellverwerfungswahrscheinlichkeit, die Verteilung der Zwischenabgangszeiten von Zellen und die Verteilung der Verzögerung von Zellen.

Die stochastische Analyse hat den Vorteil, daß ihre Aussagen über Leistungsgrößen mit Wahrscheinlichkeiten versehen sind. Dadurch kann man Ereignisse, die sehr selten auftreten, als irrelevant aus der Betrachtung ausschließen und sich auf die häufigen Ereignisse konzentrieren. Ohne stochastischen Ansatz muss man bei der Betrachtung von Systemen vom ungünstigsten Fall (engl. *worst case*) ausgehen, um auf der sicheren Seite zu bleiben. Dann kann es

vorkommen, daß die ermittelten Werte viel ungünstiger als in Wirklichkeit ausfallen, d.h. Systeme erscheinen schlechter als sie sind oder werden überdimensioniert. Auf der anderen Seite hat die stochastische Analyse das Problem, daß diejenigen stochastischen Modelle, die mathematisch handhabbar sind, in manchen Fällen die Eigenschaften realer Verkehrsquellen nicht vollständig nachbilden können. Komprimierte Videodaten zeigen auf verschiedenen Zeitebenen ein sehr unterschiedliches Verhalten [GaWi94] [Rose95]. Die meisten Modelle bilden nur eine bestimmte Zeitebene korrekt nach. Selbst bei der Beschränkung auf eine Zeitebene ist das Verhalten komprimierter Videodaten immer noch komplex. Zum Beispiel zeigt die Autokorrelation der Bildgrößen eine Abhängigkeit über lange Zeiträume, die Modelle mit wenig Gedächtnis nicht nachbilden können. Aus diesen Gründen wird in dieser Arbeit für die Analyse des Quellshapings ein deterministischer Ansatz gewählt, der von der Betrachtung des ungünstigsten Falls ausgeht, und "Modell-frei" ist [Cruz91a], in dem Sinne, daß die Analyse nicht nur für bestimmte Quellenmodelle durchführbar ist, sondern allgemeingültig ist. Insbesondere werden alle Zeitebenen größer als die Zellebene mit derselben Analyse abgedeckt. Ein weiterer Vorteil des deterministischen Ansatzes ist, daß er mit den standardisierten ATM-Quellflußkontrollverfahren zusammenpaßt, die ebenfalls deterministisch konstruiert sind. Deren Parameter können direkt in die Modellierung des Verkehrs einfließen.

4 Modellierung der Videoübertragung in ATM-Netzen

4.1 Charakterisierung von Verkehrsströmen durch Verkehrshüllen

Das hier verwendete Mengenmodell lehnt sich an bekannte Flüssigkeitsmodelle (engl. *fluid flow*) an [AMS82] [Kost84]. Es beschreibt einen Verkehrsstrom an einem bestimmten Punkt im Netz durch die durch den Punkt geflossene *Datenmenge* $R(t)$. Und zwar gibt $R(t)$ die im Zeitintervall $[0,t]$ geflossene Datenmenge an. Per Definition ist $R(t)$ monoton steigend. Für $t < 0$ definiert man $R(t) = 0$. Die *momentane Rate* $r(t)$ des Verkehrsstroms zum Zeitpunkt t ist die Ableitung der Datenmenge $R(t)$ nach der Zeit

$$r(t) = \frac{d}{dt}R(t). \tag{4.1}$$

In bestimmten Situationen sind Ankünfte oder Abgänge von Datenpaketen in unendlich kurzer Zeit zu modellieren. Dann ist $R(t)$ nicht stetig, sondern enthält Sprünge. Für die momentane Rate ergeben sich an den Sprungstellen Dirac-Impulse.

Des weiteren seien Netzelemente betrachtet, die den Verkehrsstrom verändern. Sie verzögern Daten oder verlieren sie. Die Datenmenge am Eingang eines Netzelements sei mit $R_0(t)$, der am Ausgang mit $R_1(t)$ bezeichnet. Die im Zeitintervall $[0,t]$ verlorengegangene Datenmenge sei mit $L(t)$ bezeichnet. Wenn keine Verluste auftreten, bleibt die gesamte Datenmenge erhalten und die im Netzelement gespeicherte Datenmenge $X(t)$ ist gleich der Datenmenge am Eingang abzüglich der Datenmenge am Ausgang

$$X(t) = R_0(t) - R_1(t). \tag{4.2}$$

Dabei wird angenommen, daß das Netzelement zum Zeitpunkt $t = 0$ leer ist.

Wiederum bei Verlustfreiheit läßt sich die *momentane Verzögerung* $d(t)$ des Netzelements definieren als minimale Zeitspanne, bis die Datenmenge am Ausgang das Niveau der Datenmenge zum Zeitpunkt t am Eingang eingeholt hat

$$d(t) = \min\{d | R_1(t+d) \geq R_0(t)\}. \tag{4.3}$$

Die Ungleichung ist an dieser Stelle statt einer Gleichung nötig, weil ein Netzelement einen Sprung am Ausgang erzeugen kann, der aus Daten besteht, die zu unterschiedlichen Zeitpunkten in das Netzelement gelangt sind (siehe das Beispiel in Bild 4.1). Bei einem Sprung am

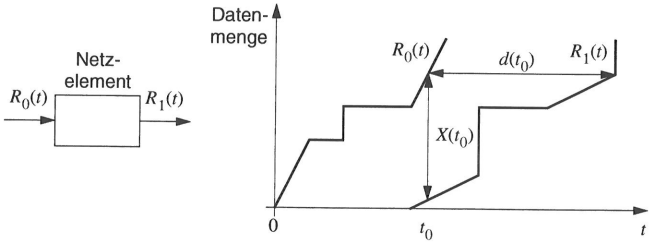


Bild 4.1: Verzögerung bei Verlustfreiheit

Eingang, d.h. einer plötzlichen Paketankunft, gibt die momentane Verzögerung die Zeitdauer an, die vergeht, bis alle Daten des Pakets das Netzelement vollständig verlassen haben. Die hier verwendete Definition beschreibt die Verzögerung aus Sicht des Eingangsverkehrs.

Im Zusammenhang mit der Verzögerung wird von einem Netzelement Kausalität verlangt. Die Bedingung dafür ist, daß es zu jedem Zeitpunkt am Ausgang nicht mehr Daten sendet als am Eingang angekommen sind

$$\forall t \geq 0: \; R_1(t) \leq R_0(t) \,. \tag{4.4}$$

Offensichtlich ist dann die gespeicherte Datenmenge immer positiv oder null. Insbesondere ist dann auch die Verzögerung immer positiv oder null, womit die Kausalität gewährleistet ist.

Die obigen Definitionen für Verzögerung und Kausalität lassen sich auf verlustbehaftete Netzelemente erweitern, wenn jeweils der Ausdruck $R_1(t)$ durch die Summe aus verlorengegangener Datenmenge $L(t)$ und übertragener Datenmenge $R_1(t)$ ersetzt wird.

4.1.1 Leaky Bucket

Ein Leaky Bucket [Turn86] ist ein allgemeiner Mechanismus, der feststellt, ob ein Verkehrsstrom eine bestimmte Schranke einhält. Der Mechanismus mißt den Füllstand eines virtuellen Speichers, der von einem Verkehrsstrom gefüllt wird. Der Speicher entleert sich mit konstanter Rate ρ, solange er nicht leer ist. Der Füllstand darf die Grenze σ nicht überschreiten. Einen Verkehrsstrom, der diese Bedingung erfüllt, nennt man konform zu einem (σ,ρ) Leaky Bucket.

Ein konformer Verkehrsstrom hat die Eigenschaft, daß seine langfristige Rate nicht größer als ρ ist, und daß die Verkehrsmenge, die kurzfristig fließen kann, durch die Grenze σ beschränkt wird. Ein Leaky Bucket beschränkt den Verkehrsstrom also auf mehreren Zeitebenen.

Der momentane Füllstand des Leaky Buckets läßt sich zum einen natürlich durch direktes Nachbilden des Speichers mit einem Zähler bestimmen: der Zähler wird bei jeder Ankunft von Daten um die Datenmenge inkrementiert und gleichzeitig mit Rate ρ dekrementiert, solange er größer null ist. Zum anderen läßt sich der Füllstand nach folgender Formel berechnen [Cruz91a], die sich für die Analyse besser eignet

$$W_\rho(R)(t) = \max_{0 \le s \le t} [R(s,t) - \rho(t-s)] \tag{4.5}$$

(dabei bezeichnet die Notation $R(s,t) = R(t) - R(s)$). Für einen konformen Verkehrsstrom gilt also

$$\forall t \ge 0 : W_\rho(R)(t) \le \sigma . \tag{4.6}$$

Ein Beispiel für einen konformen Verkehrsstrom ist in Bild 4.2 dargestellt.

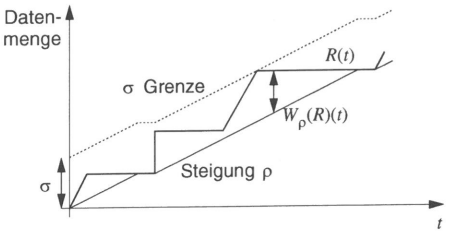


Bild 4.2: Konformität zu Leaky Bucket mit Rate ρ und Kapazität σ

Wie bereits erwähnt, ist der Leaky Bucket ein allgemeiner Mechanismus, der für die Verkehrssteuerung in verschiedenen Zusammenhängen eingesetzt wird: neben der hier gezeigten Charakterisierung von Verkehrsströmen dient er auch für die Überwachung der Verbindungsparameter und für das Traffic Shaping. Je nach dem Zusammenhang, in dem der Leaky Bucket eingesetzt wird, löst das Erreichen der Grenze unterschiedliche Aktionen aus. Bei der Überwachung der Verbindungsparameter werden Daten verworfen oder markiert, beim Traffic Shaping werden dagegen Daten zwischengespeichert.

4.1.2 Verkehrshülle

4.1.2.1 Definition

Für die Verkehrssteuerung von Kommunikationsnetzen ist eine genaue Charakterisierung der Verkehrsströme wichtig, um für die Dienstgüte der Verbindungen die geforderten Garantien geben zu können. Oft verlangen die Anwendungen vom Netz eine Dienstgüte mit deterministischen Garantien bezüglich der Verzögerung. Wie in Abschnitt 2.4 gezeigt wurde, gehört insbesondere die Übertragung von Videodaten zu dieser Klasse von Anwendungen. Um solche Garantien geben zu können, muß die Verkehrssteuerung auch deterministische Aussagen über die Verkehrsströme machen können, d.h. die Charakterisierung muß deterministisch sein. Eine solche Charakterisierung, die zudem zeitunabhängig ist, ist die *Verkehrshülle* (engl. *traffic envelope*) [Cruz91a], [KWLZ95] eines Verkehrsstroms. Sie ist definiert als eine obere Schranke für die in einem Zeitintervall der Dauer τ geflossene Verkehrsmenge. Sie ist eine Funktion von τ für alle $\tau \geq 0$.

Definition: Eine monoton steigende Funktion $b(\tau)$, $\tau \geq 0$, ist eine Verkehrshülle des durch $R(t)$ beschriebenen Verkehrsstroms genau dann, wenn

$$\forall s,t, 0 \leq s \leq t : R(s,t) \leq b(t-s) \; . \tag{4.7}$$

Wie ein Leaky Bucket beschränkt auch eine Verkehrshülle den Verkehrsstrom auf mehreren Zeitebenen, nämlich für jeden Wert von τ.

4.1.2.2 Anwendungsbeispiel

Eine Menge von Verkehrsströmen mit Datenmengen $R_i(t)$ werde über einen Multiplexer mit Pufferspeicher übertragen. Der Multiplexer besitze am Ausgang eine Übertragungskapazität c. Die Bedienstrategie sei *Abfertigung nach Ankunftsreihenfolge* (engl. *FCFS, first come, first served*). Von jedem Verkehrsstrom sei eine Hülle $b_i(\tau)$ bekannt. Damit kann nun eine obere Schranke für die Verzögerung im Pufferspeicher wie folgt bestimmt werden. Offensichtlich lassen sich Verkehrshüllen addieren, d.h. die Summe der Datenmengen $R_S(t) = \sum R_i(t)$ wird beschränkt durch die Summe der Hüllen $b_S(\tau) = \sum b_i(\tau)$. Betrachtet man den gesamten Füllstand des Pufferspeichers für die Summe der Datenmengen, so verhält sich das System identisch zu einem Leaky Bucket mit Rate $\rho = c$, der von $R_S(t)$ gespeist wird. Also läßt sich (4.5) zur Bestimmung des Füllstands $X(t)$ anwenden

$$X(t) = W_c(R_S)(t) = \max_{0 \leq s \leq t} [R_S(s,t) - c(t-s)] \; . \tag{4.8}$$

Nun gibt (4.7) mit der Hülle $b_S(t-s)$ eine obere Schranke für die Datenmenge $R(s,t)$ an, aus der sich eine obere Schranke $\hat{X}$ für den Füllstand ableiten läßt

$$X(t) \le \hat{X} = \max_{0 \le s \le t} [b_S(t-s) - c(t-s)]$$

$$= \max_{\tau \ge 0} [b_S(\tau) - c\tau] \,. \tag{4.9}$$

Gleichung (4.9) läßt sich geometrisch interpretieren als y-Achsenabschnitt der tiefsten Geraden mit Steigung c, die oberhalb der Funktion $b_S(\tau)$ liegt (siehe Bild 4.3). Aus der oberen

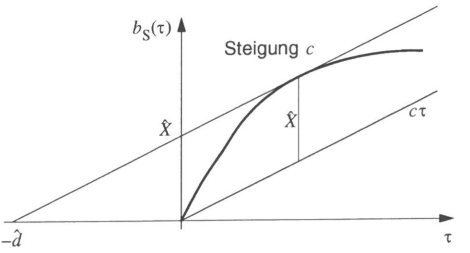


Bild 4.3: Bestimmung oberer Schranken für Füllstand und Verzögerung

Schranke für den Füllstand ergibt sich unmittelbar eine obere Schranke für die Verzögerung. Aus der angenommenen arbeitserhaltenden Bedienstrategie folgt nämlich, daß ein Beobachter im Pufferspeicher nicht länger warten muß, als die Bedienung der Datenmenge dauert, die bei seiner Ankunft im Pufferspeicher vorlag. Eine obere Schranke für die Verzögerung ist also $\hat{d} = \hat{X}/c$. Geometrisch entspricht dies dem Schnittpunkt der Geraden mit der x-Achse.

4.1.2.3 Empirische Hülle

Für jeden durch $R(t)$ beschriebenen Verkehrsstrom existiert eine minimale Verkehrshülle $\varepsilon(R)(\tau)$, die *empirische Hülle* genannt wird [KWLZ95]. Sie stellt die bestmögliche Beschränkung des Verkehrsstroms mit einer Hülle dar. Es gilt

$$\varepsilon(R)(\tau) = \max_{t \ge 0} R(t, t+\tau) \,. \tag{4.10}$$

Die empirische Hülle ist offensichtlich monoton steigend. Darüberhinaus besitzt sie die Eigenschaft der *Subadditivität* [KWLZ95], d.h.

$$\forall a > 0, b > 0: \; \varepsilon(R)(a+b) \le \varepsilon(R)(a) + \varepsilon(R)(b) \,, \tag{4.11}$$

die sich unmittelbar aus der Definition ergibt:

$$\begin{aligned}\varepsilon(R)(a+b) &= \max_{t \geq 0} R(t,t+a+b) \\ &= \max_{t \geq 0} [R(t,t+a) + R(t+a,t+a+b)] \\ &\leq \max_{t \geq 0} R(t,t+a) + \max_{t \geq 0} R(t+a,t+a+b) \\ &= \varepsilon(R)(a) + \varepsilon(R)(b)\,.\end{aligned}$$

Die empirische Hülle stellt eine Verfeinerung bekannter Charakterisierungen von Verkehrsströmen dar. Der Ausdruck $\varepsilon(R)(\tau)/\tau$ gibt die maximale Rate des Verkehrsstroms bezüglich eines Zeitintervalls der Dauer τ an. Wenn der Grenzwert für $\tau \to 0$ existiert, ist er offensichtlich gleich der Ableitung der empirischen Hülle an der Stelle $\tau = 0$. Dieser Wert ist gleich der maximalen momentanen Rate $r_{\max}$:

$$\begin{aligned}\frac{d}{d\tau}\varepsilon(R)(\tau)\bigg|_{\tau=0} &= \lim_{\tau \to 0} \frac{\varepsilon(R)(\tau)}{\tau} \\ &= \lim_{\tau \to 0} \frac{\max_{t \geq 0} R(t,t+\tau)}{\tau} \\ &= \max_{t \geq 0} \lim_{\tau \to 0} \frac{R(t,t+\tau)}{\tau} \\ &= \max_{t \geq 0} r(t) = r_{\max}\,. \end{aligned} \tag{4.12}$$

(Hier wurde angenommen, daß $R(t)$ bestimmte Bedingungen erfüllt, die die Vertauschung des Maximum mit dem Limes erlauben.)

Wenn der Grenzwert für $\tau \to \infty$ existiert, ist er bei unendlich andauernden Verkehrsströmen gleich der mittleren Rate $\bar{r}$ des Verkehrsstroms

$$\bar{r} = \lim_{\tau \to \infty} \frac{\varepsilon(R)(\tau)}{\tau}\,. \tag{4.13}$$

(Auch hier muß $R(t)$ bestimmte Bedingungen erfüllen, siehe [Chan94].) Bei einem endlichen Verkehrsstrom der Dauer t_0 bezieht sich die mittlere Rate auf t_0, sie ist dann gleich

$$\frac{\varepsilon(R)(t_0)}{t_0}\,.$$

Damit umfaßt die Charakterisierung eines Verkehrsstroms durch seine empirische Hülle auch den Burstiness-Faktor f, der als das Verhältnis von maximaler Rate zu mittlerer Rate definiert ist: $f = r_{\max}/\bar{r}$. Die empirische Hülle ist eine detaillierte Beschreibung der Büschel-

förmigkeit eines Verkehrsstroms auf mehreren Zeitebenen, nämlich für jeden Wert von τ. Nebenbei sei angemerkt, daß man für den ideal glatten Verkehrsstrom mit konstanter Rate ρ, $R(t) = \rho t$, eine Ursprungsgerade mit Steigung ρ erhält: $\varepsilon(R)(\tau) = \rho\tau$.

Wenn die Mengenfunktion des Verkehrsstroms Sprünge enthält, also unendliche momentane Raten aufweist, existiert der betrachtete Grenzwert für $\tau \to 0$ nicht. Die empirische Hülle besitzt dann einen Sprung an der Stelle $\tau = 0$, dessen Höhe der maximalen Höhe aller Sprünge der Mengenfunktion entspricht. Dieser Sprung ist gewissermaßen das Analogon zur maximalen momentanen Rate.

4.1.2.4 Anwendung in der Zuweisung von Netzressourcen

Die Aussagekraft der empirischen Hülle soll mit einer erneuten Betrachtung des Beispiels von Abschnitt 4.1.2.2 deutlich gemacht weren. Angenommen, die empirische Hülle des Summen-Verkehrsstroms $\varepsilon(R_S)(\tau)$ sei bekannt, dann kann aus (4.8) der maximal auftretende Füllstand bestimmt werden

$$X_{\max} = \max_{t \geq 0} X(t) = \max_{t \geq 0} \max_{0 \leq s \leq t} [R_S(s,t) - c(t-s)]$$

$$= \max_{\tau \geq 0} [\varepsilon(R_S)(\tau) - c\tau] \,. \tag{4.14}$$

Die maximal auftretende Verzögerung ist $d_{\max} = \max_{t \geq 0} d(t) = X_{\max}/c$. Im Gegensatz zu vorhin sind diese Größen jetzt tatsächliche, im ungünstigsten Fall auftretende (engl. *worst case*) Werte, oder minimale obere Schranken. Das System wird damit also optimal dimensioniert.

Leider ist üblicherweise die empirische Hülle des Summen-Verkehrsstroms nicht bekannt. Dazu wäre die Kenntnis über den zeitlichen Verlauf aller Verkehrsströme nötig. Da typischerweise für die Verkehrsströme individuelle, voneinander unabhängige Verkehrsverträge abgeschlossen werden, können höchstens die empirischen Hüllen der einzelnen Verkehrsströme bekannt sein. Ihre Summe ist eine (nicht-minimale) Hülle für den Summen-Verkehrsstrom, die zur Zuweisung von Netzressourcen verwendet werden kann. Wenn die individuellen Hüllen eine günstige Form aufweisen, kann bei der Überlagerung mehrerer Verkehrsströme ein Multplexgewinn realisiert werden, wie weiter unten gezeigt wird.

Es sei zunächst angenommen, der Eingangsverkehr in den Multiplexer bestehe aus einem einzelnen Verkehrsstrom. Aus dem Beispiel von Bild 4.3 wird deutlich, daß die Übertragungskapazität des Multiplexers c kleiner sein kann als die Steigung der empirischen Hülle an der Stelle $\Delta t = 0$, und damit kleiner als die maximale Rate des Verkehrsstroms. Damit kann die Verkehrssteuerung eine Zuteilung der Übertragungskapazität vornehmen, die geringer ist als die Zuteilung der maximalen Rate (engl. *peak rate allocation*). Der Preis dafür ist eine Verzögerung, die jedoch durch eine deterministische obere Schranke begrenzt ist. Wenn

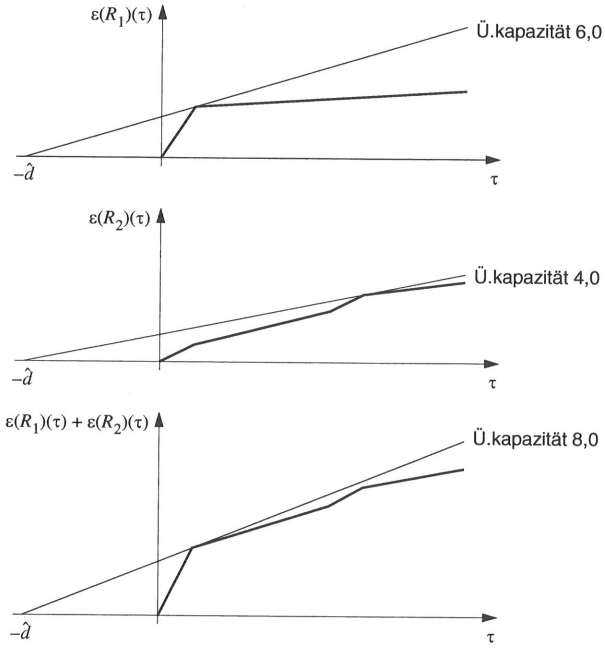


Bild 4.4: Multiplexgewinn mit deterministischer Dienstgüte

die Verkehrssteuerung keine Verzögerung tolerieren kann, muß sie die maximale Rate zuteilen. Es sei noch angemerkt, daß der Multiplexer bei $c < r_{max}$ als Traffic Shaper arbeitet, dies wird später genauer untersucht.

Jetzt seien mehrere sich überlagernde Verkehrsströme betrachtet. In diesem Fall kann zusätzlich ein Multiplexgewinn realisiert werden. Zum Beispiel werde ein Verkehrsstrom mit Mengenfunktion $R_1(t)$, der hohe Büschelförmigkeit bei kurzen Zeitintervallen aufweist, mit einem Verkehrsstrom mit Mengenfunktion $R_2(t)$, der hohe Büschelförmigkeit bei langen Zeitintervallen aufweist, gemultiplext (siehe Bild 4.4). Bei gleicher Verzögerung benötigt die Summe der Verkehrsströme nur die (normierte) Übertragungskapazität 8,0, während die Verkehrsströme einzeln die Kapazität 6,0 und 4,0 benötigen, also ein Gewinn von 20%.

Dieser Multiplexgewinn wird ohne Verluste erzielt und stellt sich auch bei wenigen Verkehrsströmen ein. Voraussetzung ist, daß die empirischen Hüllen der Verkehrsströme unterschiedliche Verläufe aufweisen, also hohe Büschelförmigkeit bei verschiedenen Zeitintervallen auftritt.

Hier wurde die Zuteilung von Netzressourcen am Beispiel der empirischen Hülle gezeigt, um die theoretisch mögliche Leistungsfähigkeit von Mengenmodellen zu demonstrieren. Wie in Abschnitt 4.1.3 ausgeführt werden wird, ist in der Praxis der Verbindungsannahmefunktion im allgemeinen nur eine nicht-minimale Hülle bekannt. Ziel ist es natürlich, die Differenz zwischen der bekannten Hülle und der tatsächlichen empirischen Hülle so klein wie möglich zu machen, um gute Schranken zu erhalten.

4.1.2.5 Spezialfälle Leaky Bucket und Moving Window

Die Beschränkung durch einen Leaky Bucket ist ein Spezialfall einer Verkehrshülle. Für einen Verkehr, der konform zu einem (σ,ρ) Leaky Bucket ist, ergibt sich aus (4.5) und (4.6)

$$\forall s,t, 0 \le s \le t : R(s,t) \le \sigma + \rho(t-s) \; . \tag{4.15}$$

Die Beschränkung durch einen Leaky Bucket ist daher äquivalent zur Beschränkung durch die lineare Verkehrshülle

$$b(\tau) = \sigma + \rho\tau \tag{4.16}$$

(siehe Bild 4.5 a), in dem der ausgeschlossene Bereich schattiert dargestellt ist). Im nächsten

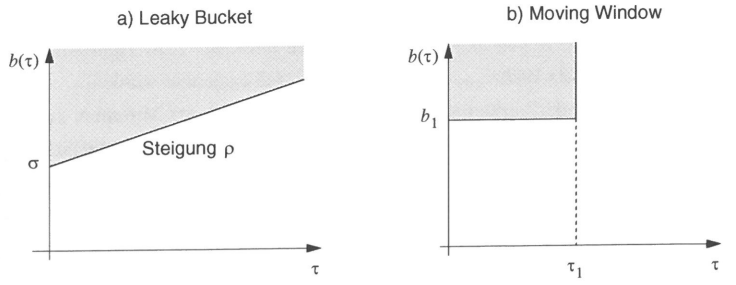


Bild 4.5: a) Leaky Bucket und b) Moving Window als Spezialfälle einer Verkehrshülle

Abschnitt wird gezeigt, wie durch den Einsatz mehrerer Leaky Buckets allgemeine Verkehrshüllen approximiert werden können.

Ein zweiter Spezialfall einer Verkehrshülle ist die Beschränkung eines Verkehrsstroms durch ein Moving Window [Rath91]. Bei diesem Mechanismus wird die in jedem Zeitintervall der Dauer τ_1 geflossene Verkehrsmenge auf den Wert b_1 beschränkt. Dazu wird ein Zähler mit $R(t)$ inkrementiert und mit $R(t-\tau_1)$ dekrementiert. Der Zähler darf den Wert b_1 nicht überschreiten. Dies ist äquivalent zu einer Verkehrshülle

$$b(\tau) = \begin{cases} b_1 \text{ für } \tau \leq \tau_1 \\ \infty \text{ für } \tau > \tau_1 \end{cases}$$

(siehe Bild 4.5 b)).

4.1.3 Verkehrsdeskriptor

Für den Einsatz im Betrieb von Kommunikationsnetzen sind allgemeine Verkehrshüllen unpraktisch, weil sie zu detailliert sind. Zum Beispiel wären Quellflußkontrollmechanismen für allgemeine Verkehrshüllen gar nicht oder nur mit sehr hohem Aufwand zu realisieren. Deshalb versucht man, Verkehrshüllen durch die im vorigen Abschnitt erwähnten Sonderfälle zu approximieren, für die einfache Quellflußkontrollmechanismen bekannt sind. Dabei reduziert sich die Charakterisierung des Verkehrs auf wenige Parameter, was auch die Aushandlung des Verkehrsvertrags in den Signalisiernachrichten erleichtert. Die Approximation einer Verkehrshülle kann hier als *Verkehrsdeskriptor* [ATMF94] bezeichnet werden.

Aus der Definition der Verkehrshülle ist offensichtlich, daß das Minimum zweier Funktionen, die beide Verkehrshüllen für einen Verkehrsstrom darstellen, ebenfalls eine Hülle des Verkehrsstroms ist. Angewendet auf die Quellflußkontrolle bedeutet dies, daß mehrere parallel betriebene Instanzen eines Mechanismus wie Leaky Bucket oder Moving Window eine Verkehrshülle überwachen können, die das Minimum der Verkehrshüllen der einzelnen Instanzen darstellt. Damit wird eine Approximation möglich.

Wählt man Leaky Buckets als zugrundeliegenden Mechanismus, so läßt sich mit n parallel betriebenen Instanzen mit den Parametern (σ_i,ρ_i) die folgende allgemeine Verkehrshülle überwachen:

$$b(\tau) = \min_{i=1..n}\{\sigma_i + \rho_i\tau\} \; . \tag{4.17}$$

In der Standardisierung wurde dieser Ansatz mit $n = 2$ für den Verkehrsvertrag beim ATM Übermittlungsdienst mit variabler Bitrate festgelegt [ATMF94]. Ein Beispiel ist in Bild 4.6 a) zusammen mit den den Parametern zugeordneten englischen Begriffen dargestellt.

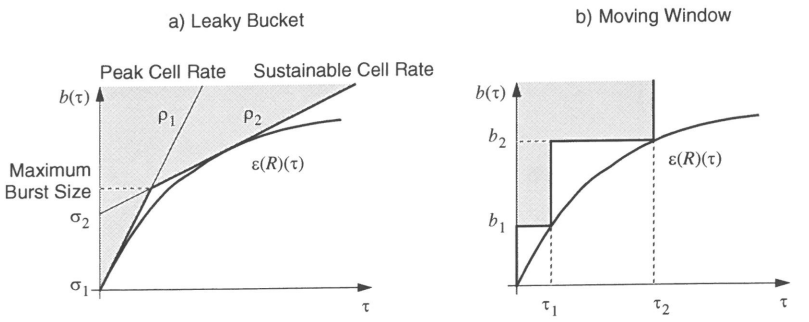


Bild 4.6: Mehrere Instanzen von a) Leaky Buckets, b) Moving Windows zur Überwachung einer Verkehrshülle

n Instanzen eines Moving Windows mit Parametern (τ_i, b_i), $\tau_i < \tau_{i+1}$, $b_i < b_{i+1}$ können die folgende Hülle überwachen

$$b(\tau) = \begin{cases} b_1 \text{ für } 0 \le \tau \le \tau_1 \\ b_2 \text{ für } \tau_1 < \tau \le \tau_2 \\ \quad \dots \\ \infty \text{ für } \tau > \tau_n \end{cases} . \tag{4.18}$$

Ein Beispiel für $n = 2$ ist in Bild 4.6 b) dargestellt.

Was die Güte der Approximation anbelangt, so sind die von Leaky Buckets überwachbaren Hüllen stückweise linear und konkav. Sie können also selbst bei beliebig vielen Instanzen den Verkehr nicht so eng wie die empirische Hülle beschränken, da letztere im allgemeinen nicht konkav, sondern nur subadditiv ist[1]. Das Moving Window hat diesen Nachteil nicht, es kann offensichtlich mit genügend vielen Instanzen die empirische Hülle beliebig genau approximieren. Allerdings zeigt das Beispiel, daß im allgemeinen bei gleicher Zahl von Instanzen die auf Leaky Buckets basierende Quellflußkontrolle eine engere Hülle überwachen kann. Das Moving Window benötigt mit seiner stufenförmigen Hülle für dieselbe Güte sehr viele Instanzen, da eine Instanz nur ein einzelnes Zeitintervall überwacht. Es sind natürlich Verkehrsströme denkbar, deren Hülle eine ausgeprägte Stufenform aufweist, und für die deshalb Moving Windows besser geeignet sind. Bei den in Abschnitt 6 untersuchten Videokodie-

1. Man kann zeigen, daß Konkavität hinreichend ist für Subadditivität.

rern konnte das jedoch nicht festgestellt werden. Dabei weisen beide Mechanismen eine vergleichbare Komplexität auf, die eher zum Nachteil für das Moving Window ausfällt. Deshalb werden in der Praxis Leaky Buckets für die Quellflußkontrolle vorgezogen.

4.2 Modellierung des Empfängers

4.2.1 Pufferspeicher

In diesem Abschnitt wird ein Video-Empfänger mit dem im vorigen Abschnitt eingeführten Mengenmodell modelliert. Der Empfänger enthält einen Pufferspeicher, der die vom Netz kommenden Daten zwischenspeichert, und einen Video-Dekodierer, der die Daten aus dem Pufferspeicher entnimmt und dekodiert (siehe Bild 4.7). Das Entnehmen der Daten folgt

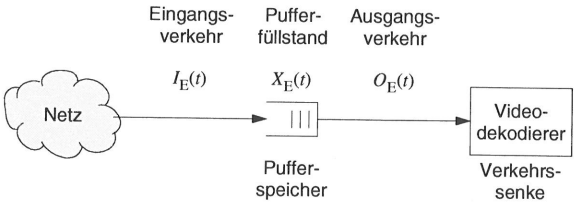


Bild 4.7: Empfängermodell

einem bestimmten Zeitplan. Unter der Annahme eines starren Empfängers, wie er in Abschnitt 2.4 beschrieben wurde, ist der relative zeitliche Ablauf fest vorgegeben. Der Dekodierer kann den Zeitplan nur nach vorne oder nach hinten verschieben, indem er den Beginn des Dekodierens entsprechend verzögert. Diese Verzögerung ist die *Ausspielverzögerung*. Der unveränderliche *relative Zeitplan* des Dekodierers wird durch die Funktion $C(t)$ beschrieben. $C(t)$ stellt die Datenmenge dar, die im Zeitintervall $[0,t]$ ausgelesen wird, wenn die ersten Daten zum Zeitpunkt $t = 0$ ausgelesen werden.

Im folgenden wird ein Beispiel für $C(t)$ angegeben, das die wesentlichen Aspekte der Kodierverfahren mit variabler Bitrate für eine quantitative Analyse modelliert. Eine genauere Modellierung, die sich auch für die Implementierung eignet, wird in Abschnitt 4.2.2 beschrieben. Als Beispiel werde das Motion-JPEG Verfahren [ISO 10918-1] von Abschnitt 2.2 betrachtet. Die kodierten Videodaten bestehen aus einer Folge von Bildern F_i mit Größe s_i, die zu äquidistanten Zeitpunkten ausgelesen werden. Die Ausleseperiode sei gleich der Bild-

periode T. Das Auslesen eines Bildes geschehe plötzlich, so daß alle Daten des Bildes zum selben Zeitpunkt dem Pufferspeicher entnommen werden. Das bedeutet, daß $C(t)$ Sprünge enthält. Mit $s(t)$ sei die Sprungfunktion[2] bezeichnet. Dann ist

$$C(t) = \sum_{i=0}^{n-1} s_i s(t - iT) . \tag{4.19}$$

Zusätzliches Datenaufkommen, das durch die Steuerfelder von Protokollen zwischen der Video-Anwendung und der ATM-Schicht verursacht wird (z.B. Protokolle zur Kapselung der Bilder und Protokolle der ATM-Adaptierungsschicht), kann dadurch berücksichtigt werden, daß es den Größen s_i zugeschlagen wird.

Die Ausspielverzögerung D_E legt die absolute zeitliche Position des Empfängerzeitplans fest. Damit ergibt sich für den Ausgangsverkehr aus dem Pufferspeicher $O_E(t) = C(t - D_E)$ (siehe Bild 4.8). Mit $I_E(t)$ werde die Mengenfunktion des Ankunftspro-

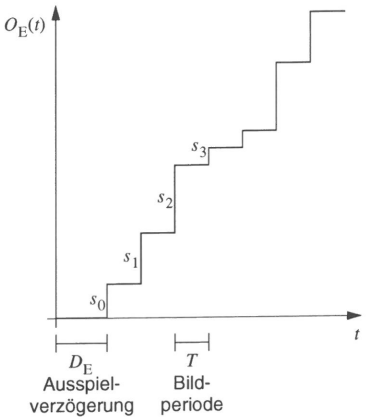


Bild 4.8: Zeitplan des Empfängers

zesses vom Netz in den Pufferspeicher bezeichnet. Für den Füllstand des Pufferspeichers $X_E(t)$ ergibt sich dann nach der Erhaltungsgleichung (4.2) (siehe auch Bild 4.7)

$$X_E(t) = I_E(t) - O_E(t) . \tag{4.20}$$

2. Die Sprungfunktion ist definiert als $s(t) = \begin{cases} 0 \text{ für } t < 0 \\ 1 \text{ für } t \geq 0 \end{cases}$

Der Pufferspeicher hat die Aufgabe, den Leseprozeß des Dekodierers vom Ankunftsprozeß der Daten aus dem Netz zu entkoppeln, indem er die Echtzeitanforderungen an den Ankunftsprozeß lockert. Im folgenden wird gezeigt, wie man diese Anforderungen ausschließlich durch Anforderungen an den Füllstand des Pufferspeichers ausdrücken kann (siehe auch [ReHa92]). Zum einen dürfen die Daten nicht zu spät ankommen. Wenn der Dekodierer ein kodiertes Bild aus dem Pufferspeicher entnimmt, aber noch nicht alle Daten des Bildes im Pufferspeicher angekommen sind, wird im Modell (4.20) der Füllstand negativ, es liegt ein "Unterlaufen" des Pufferspeichers vor (das Gegenteil des Überlaufens). Zu spät ankommende Daten werden vermieden, wenn gilt

$$\forall t \geq 0\colon\ X_{\mathrm{E}}(t) \geq 0\,. \tag{4.21}$$

Zum anderen dürfen die Daten nicht zu früh ankommen. Ein Beobachter, der zusammen mit zu früh ankommenden Daten in den Pufferspeicher gelangt, sieht einen Füllstand, der größer ist als die Kapazität $X_{\mathrm{K,E}}$, also ein Überlaufen. Zu früh ankommende Daten werden vermieden, wenn gilt

$$\forall t \geq 0\colon\ X_{\mathrm{E}}(t) \leq X_{\mathrm{K,E}}\,. \tag{4.22}$$

Das Verhalten des Empfängers wird, neben dem relativen Dekodier-Zeitplan $C(t)$, durch die Ausspielverzögerung D_E und die Kapazität des Pufferspeichers $X_{\mathrm{K,E}}$ bestimmt. Wenn diese Größen gegeben sind, bestimmen die Gleichungen (4.21) und (4.22) die Grenzen einer *Region erlaubter Mengenfunktionen* $I_{\mathrm{E}}(t)$ für den Ankunftsprozeß. Diese Region ist eine Punktmenge $(t,I_{\mathrm{E}}(t))$ definiert durch

$$\forall t \geq 0\colon\ O_{\mathrm{E}}(t) \leq I_{\mathrm{E}}(t) \leq O_{\mathrm{E}}(t) + X_{\mathrm{K,E}}\,. \tag{4.23}$$

Ein Beispiel für eine solche Region ist in Bild 4.9 dargestellt. Ein Verkehrsstrom, dessen Mengenfunktion innerhalb der Region bleibt, verhindert ein Über- oder Unterlaufen des Pufferspeichers. Dabei bestimmt die Ausspielverzögerung die horizontale Lage der unteren und oberen Schranke und die Speicherkapazität die vertikale Lage der oberen Schranke. Die Form beider Schranken ist durch den relativen Zeitplan des Dekodierers bestimmt.

Während (4.23) die erlaubten Mengenfunktionen für gegebene Größen D_{E} und $X_{\mathrm{K,E}}$ bestimmt, ist es manchmal auch nötig, die Fragestellung umzukehren, nämlich für eine gegebene Mengenfunktion $I_{\mathrm{E}}(t)$ minimale Werte für D_{E} und $X_{\mathrm{K,E}}$ zu bestimmen, die die Bedingungen (4.21) und (4.22) erfüllen. Im folgenden wird angenommen, daß der Ankunftsprozeß dadurch entstanden ist, daß eine Quelle, deren Ausgangsverkehr $R(t)$ dem relativen Dekodier-

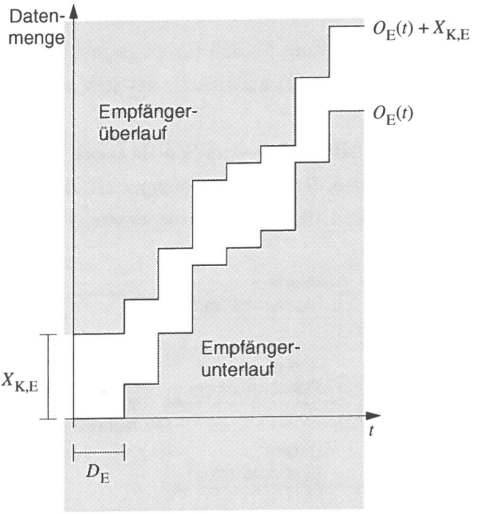


Bild 4.9: Region erlaubter Mengenfunktionen für den Ankunftsprozeß

Zeitplan $C(t)$ entspricht, eine variable Verzögerung $d(t)$ erfahren hat. Zunächst werde die minimale Ausspielverzögerung D_E bestimmt. Aus der Bedingung zur Vermeidung des Unterlaufens (4.21) erhält man

$$\forall t \geq 0: \; X_E(t) = I_E(t) - C(t - D_E) \geq 0$$

$$\Leftrightarrow \forall t' \geq -D_E: \; I_E(t' + D_E) \geq C(t') \; . \tag{4.24}$$

Aus der Definition der Verzögerung (4.3) ergibt sich, daß $d(t)$ das kleinste d ist, das die Bedingung $I_E(t + d) \geq C(t)$ erfüllt. Das kleinste D_E, das (4.24) erfüllt, ist damit

$$D_E = \max_{t \geq 0} d(t) \; . \tag{4.25}$$

Zur Verhinderung des Unterlaufens muß also die Ausspielverzögerung gleich der maximalen Verzögerung gesetzt werden, was sich durch direkte Überlegung bestätigen läßt.

Wenn D_E bestimmt ist, ergibt sich die minimal benötigte Kapazität des Pufferspeichers einfach aus dem maximal auftretenden Füllstand gemäß (4.20)

$$X_{K,E} = \max_{t \geq 0} [I_E(t) - C(t - D_E)] \; . \tag{4.26}$$

4.2.2 Dekodier-Zeitplan von MPEG-2 kodierten Strömen

In diesem Abschnitt wird ein genaues Modell für einen multimedialen Verkehrsstrom nach dem MPEG-2 Standard [ISO 13818-1] entwickelt, das sich auch für eine Implementierung eignet.

Ein kodierter Strom der MPEG-2 System-Schicht (engl. *system layer*) ist aus mehreren Elementarströmen (Audio, Video, Daten) zusammengesetzt, die ineinander verschränkt sind (engl. *interleaved*) (siehe Bild 4.10). Jeder Elementarstrom besteht aus einer Folge von

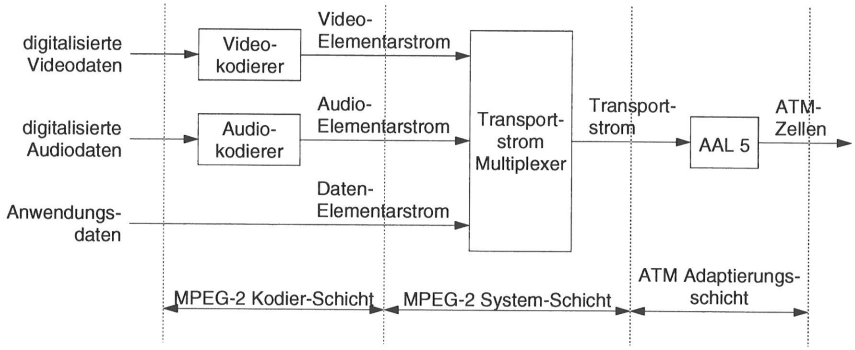


Bild 4.10: Überblick MPEG-2 über ATM

Zugriffseinheiten (engl. *access unit*). Diese können z.B. im Falle von Video ein Bild oder im Falle von Audio ein Block von zeitlich aufeinanderfolgenden Abtastwerten umfassen. Der Standard definiert mit dem *Dekodierzeitpunkt* einer Zugriffseinheit den Zeitpunkt der Entnahme aus dem Pufferspeicher (relativ zum Start des Dekodiervorgangs). Die Dekodierzeitpunkte eines Elementarstroms sollen näherungsweise äquidistant sein. Die Abtasttaktgeneratoren der verschiedenen Elementarströme können jedoch implementierungsbedingt Schwankungen aufweisen und müssen auch nicht untereinander synchronisiert sein. Bei Schwankungen des Abtasttaktes wird der Dekodierzeitpunkt im Verkehrsstrom explizit in der Form eines Zeitstempels kodiert. Andernfalls kann der Zeitstempel entfallen und der Dekodierer extrapoliert den Dekodierzeitpunkt aus vorhergehenden Zeitstempeln desselben Elementarstroms. Im Modell besteht ein Elementarstrom ES_j aus einer Folge von Zugriffseinheiten mit Größe $s_{j,i}$ und Dekodier-Zeitstempel $t_{j,i}$. Die Größe $s_{j,i}$ beinhaltet auch alle Steuerfelder der bis zur ATM-Schicht auftretenden Protokolle (MPEG-Systemschicht und AAL 5). Im Unterschied zum Modell in Abschnitt 4.2.1 werden also (1) mehrere Elementarströme verschränkt und (2) sind die Dekodierzeitpunkte nicht strikt äquidistant.

Eine weitere Verfeinerung des Modells berücksichtigt das Datenvolumen der Steuerfelder der System-Schicht, die nicht den Elementarströmen zugeordnet werden können (dazu zählen z.B. Programmtabellen). Dazu wird ein zusätzlicher Elementarstrom ES_0 mit Zugriffseinheiten $s_{0,i}$ eingeführt. Um nicht jedes Steuerfeld mit einer separaten Zugriffseinheit zu modellieren, werden die Steuerfelder, die während eines Zeitintervalls konstanter Dauer Δt_0 auftreten (z.B. eine Sekunde), in einer Zugriffseinheit mit Dekodier-Zeitstempel $t_{0,i} = i\Delta t_0$ zusammengefaßt.

Damit erhält man für den Dekodier-Zeitplan bei m "echten" Elementarströmen ES_1 bis ES_m und einem Pseudo-Elementarstrom ES_0 für die System-Schicht

$$C(t) = \sum_{j=0}^{m} \sum_{i=0}^{n-1} s_{j,i} s(t - t_{j,i}) \,. \tag{4.27}$$

4.3 Modellierung des Senders

Beim Sender können zwei Fälle nach der Art der Anwendung unterschieden werden: Im ersten Fall komprimiert die Anwendung die Videodaten in Echtzeit und sendet sie dann sofort. Im zweiten Fall werden die Daten nach der Kompression zuerst auf einem Massenspeicher abgelegt. Die Anwendung sendet dann die bereits komprimierten Daten zu einem späteren Zeitpunkt.

4.3.1 Echtzeit-Kodierung

Bei der Echtzeit-Kodierung enthält der Sender einen Kodierer und einen Pufferspeicher, der die kodierten Daten zwischenspeichert, bevor sie ins Netz gesendet werden (siehe Bild 4.11 a)). Traffic Shaping wird durch das Zwischenspeichern der Daten, kombiniert mit einer Ratensteuerung, erreicht.

Wie der Dekodierer in Abschnitt 4.2.1 generiert auch der Kodierer die Daten nach einem bestimmten Zeitplan $I_S(t)$. Es sei im folgenden angenommen, daß der Produktionsprozeß der Daten genau dem relativen Zeitplan des Dekodierers aus Abschnitt 4.2.1 entspricht, d.h.

$$I_S(t) = C(t) \,. \tag{4.28}$$

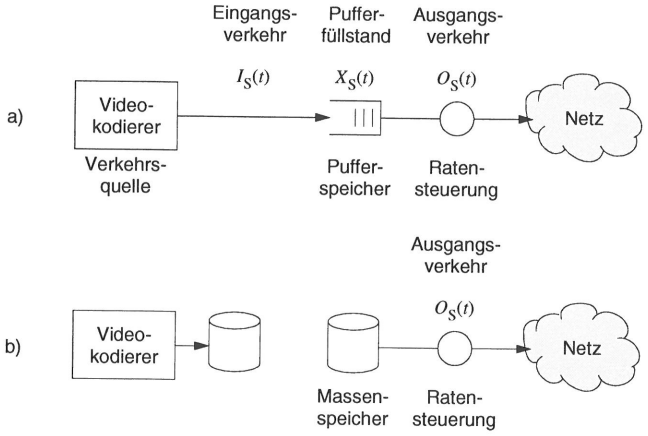


Bild 4.11: Sendermodell für a) Echtzeit-kodierte, b) gespeicherte Videodaten

Insbesondere wird auch hier angenommen, daß der Kodierer alle Daten eines Bildes oder einer Zugriffseinheit zum selben Zeitpunkt generiert und im Pufferspeicher ablegt, wodurch sich in $C(t)$ Sprünge ergeben. Mit $O_S(t)$ werde die Mengenfunktion des von der Ratensteuerung erzeugten Verkehrsstroms bezeichnet, mit dem der Pufferspeicher ausgelesen wird. Damit ergibt sich für den Füllstand des Pufferspeichers

$$X_S(t) = I_S(t) - O_S(t) . \tag{4.29}$$

Wie in Abschnitt 4.2.1 muß auch hier ein Unterlauf oder Überlauf verhindert werden. Mit einer Kapazität des Pufferspeichers $X_{K,S}$ lauten die entsprechenden Bedingungen

$$\forall t \geq 0: \; X_S(t) \geq 0 , \tag{4.30}$$

$$\forall t \geq 0: \; X_S(t) \leq X_{K,S} . \tag{4.31}$$

Der Pufferspeicher entkoppelt die Ratensteuerung vom Produktionsprozeß des Kodierers und gibt ihr dadurch Raum, die Rate als Funktion über die Zeit zu glätten. Wenn der Zeitplan des Kodierers $I_S(t)$ gegeben ist, bestimmen die Gleichungen (4.30) und (4.31) eine Region erlaubter Mengenfunktionen für den Sendeprozeß $O_S(t)$. Sie ist definiert durch

$$\forall t \geq 0: \; I_S(t) - X_{K,S} \leq O_S(t) \leq I_S(t) . \tag{4.32}$$

Ein Beispiel für eine solche Region ist in Bild 4.12 dargestellt.

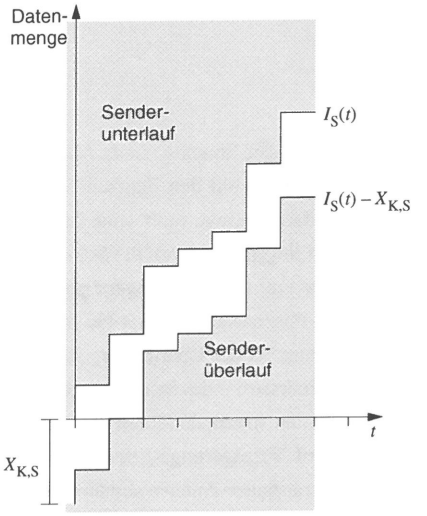


Bild 4.12: Region erlaubter Mengenfunktionen für den Sendeprozeß

4.3.2 Gespeicherte Videodaten

Anwendungen, die die bereits kodierten Videodaten von einem Massenspeicher lesen, haben die im vorigen Abschnitt diskutierten Einschränkungen nicht. Wie in Bild 4.11 b) dargestellt, ist die Ratensteuerung direkt mit dem Massenspeicher gekoppelt (eventuell wird in einer Implementierung ein Pufferspeicher benötigt). Die Daten können mit jeder beliebigen Rate gelesen und gesendet werden (solange sie im Rahmen der durch die Implementierung gegebenen Einschränkungen bleibt). Die Bilder können im Vergleich zum Fall der Echtzeit-Kodierung früher als im Kodier-Zeitplan vorgesehen gesendet werden. Daraus folgt, daß für den Sendeprozeß jeder Verkehrsstrom erlaubt ist. D.h. die Region der erlaubten Mengenfunktionen umfaßt den ganzen ersten Quadranten.

4.4 Verzögerungsschwankungen bei der Übertragung

4.4.1 Problemstellung

Die Aufgabe der Mechanismen zum Traffic Shaping, die in Abschnitt 5 vorgestellt werden, ist es, Verkehrsströme zu generieren, die sowohl den Echtzeit-Anforderungen des Senders als auch des Empfängers genügen. Genauer gesagt, muß seine Mengenfunktion innerhalb einer Region erlaubter Mengenfunktionen liegen, die durch die Schnittmenge der Regionen erlaubter Mengenfunktionen des Senders und des Empfängers gebildet wird. (Wie im vorigen Abschnitt gezeigt wurde, kann die Region des Senders bei gespeichertem Video der ganze erste Quadrant sein.) Das Netz erzeugt bei der Übertragung jedoch eine stochastische Verzögerung, die den ursprünglichen Verkehrsstrom des Senders verändert. Die Verzögerung besitzt neben einem konstanten einen variablen Anteil, der durch variable Warteschlangenlängen hervorgerufen wird. Dieser Anteil wird Verzögerungs-Jitter genannt. Der Verkehrsstrom wird nicht nur in der Zeit verschoben (konstanter Anteil), sondern durch den Verzögerungs-Jitter auch zeitlich verzerrt.

Um diesen Effekt der stochastischen Verzögerung zu berücksichtigen, sind zwei Ansätze möglich. Im ersten Ansatz berücksichtigt bereits der Sender die Verzögerung und erzeugt einen entsprechend verändertern Verkehrsstrom (Behandlung ex ante). Im zweiten Ansatz ist es der Empfänger, der die Verzögerung berücksichtigt, indem er seine Ausspielverzögerung und seinen Pufferspeicher anpaßt (Behandlung ex post).

Die Behandlung im Sender ist natürlich dann nicht möglich, wenn die Größe des Jitters dem Sender nicht bekannt ist. Daneben gibt es auch den Fall von Verteildiensten, bei denen das Netz den Verkehrsstrom in Multicast-fähigen Verkehrsknoten dupliziert und gleichzeitig mehreren Empfängern zustellt. Wegen unterschiedlicher Wege durch das Netz weist der Jitter dann bei den Empfängern eine unterschiedliche Größe auf. In diesen Fällen ist dann nur eine Behandlung im Empfänger möglich.

Im folgenden sei angenommen, daß die Verkehrssteuerung des Netzes deterministische oder quasi-deterministische untere und obere Schranken für die Verzögerung angeben kann (unter quasi-deterministisch seien sehr geringe, vernachlässigbare Übertretungswahrscheinlichkeiten verstanden). Die untere und obere Schranke seien mit $\underline{d}$ und $\hat{d}$ bezeichnet. Die Differenz $\delta = \hat{d} - \underline{d}$ ist der Verzögerungs-Jitter.

Solche Schranken ergeben sich zum Beispiel aus den in Abschnitt 3.4.3.1 vorgestellten, vom ATM Forum festgelegten Verzögerungsparametern. Wenn man $\hat{d}$ = maxCTD und δ = CDV setzt, dann liegt ein Anteil von $(1 - \alpha)$ Zellen innerhalb dieser Schranken.

4.4.2 Behandlung im Sender (vor der Übertragung)

Zur Behandlung der stochastischen Verzögerung im Sender wird die Schnittmenge der Regionen beim Sender gebildet. Dafür werden die Schranken des Empfängers (4.23) auf den Sender abgebildet, indem auf der Zeitachse eine stochastische Verzögerung abgezogen wird (siehe Bild 4.13). Dabei wird das Schaubild der Schranken durch den Verzögerungs-Jitter über der

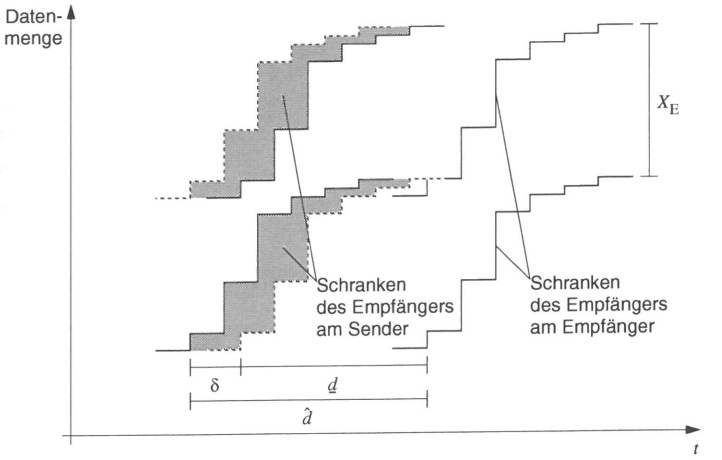


Bild 4.13: Abbildung der Schranken des Empfängers auf den Sender

Zeitachse "verschmiert". Daraus lassen sich engere Schranken ableiten, die einen Unterlauf oder Überlauf des Pufferspeichers auch bei Jitter verhindern. Die Schranke zur Verhinderung eines Unterlaufs ergibt sich durch eine Verschiebung um $\hat{d}$ nach links, die Schranke zur Verhinderung eines Überlaufs durch Verschiebung um $\underline{d}$ nach links. Der Verzögerungs-Jitter bewirkt also eine Verengung der Schranken um den Wert δ. Die Region erlaubter Mengenfunktionen des Empfängers am Sender ist gegeben durch

$$\forall t \geq 0: \; O_E(t - \hat{d}) \leq I_E(t) \leq O_E(t - \underline{d}) + X_E . \tag{4.33}$$

4.4.3 Behandlung im Empfänger (nach der Übertragung)

Bei der Behandlung der stochastischen Verzögerung beim Empfänger wird zunächst für die Erzeugung des Verkehrsstroms im Sender eine Verzögerung null angenommen. Der erzeugte Verkehrsstrom mit Mengenfunktion $R(t)$ genüge den Schranken (4.23) für eine Ausspielver-

zögerung D_E und eine Pufferspeicherkapazität X_E. Am Empfänger hat die stochastische Verzögerung das Schaubild der Mengenfunktion über der Zeitachse verschmiert. Es ergibt sich eine Mengen-Region $\bar{R}$ (siehe Bild 4.14).

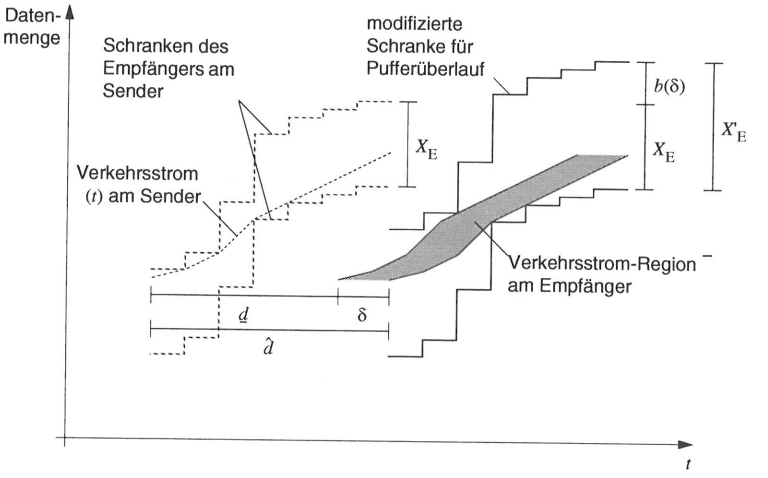


Bild 4.14: Abbildung des Verkehrsstroms des Senders auf den Empfänger

Der Empfänger vergrößert jetzt seine Ausspielverzögerung und seine Pufferspeicherkapazität so, daß die gesamte Region $\bar{R}$ innerhalb der Schranken liegt. Offensichtlich ist die dafür nötige Ausspielverzögerung

$$D'_E = D_E + \hat{d}. \tag{4.34}$$

Die Kapazität des Pufferspeichers muß erhöht werden, weil nun Daten, die während eines Intervalls der Dauer δ gesendet wurden, gleichzeitig zu Beginn des Intervalls beim Empfänger ankommen können. Die zusätzlich benötigte Kapazität ergibt sich aus der Datenmenge, die im ungünstigsten Fall während eines Intervalls der Dauer δ gesendet werden kann. Wenn eine Hülle $b(t)$ des Verkehrsstroms bekannt ist, kann diese Menge mit $b(\delta)$ abgeschätzt werden. Somit ergibt sich

$$X'_E = X_E + b(\delta). \tag{4.35}$$

Wenn keine Hülle bekannt ist, kann z.B. die maximale Rate des Verkehrsstroms oder die Übertragungskapazität des letzten Übertragungsabschnitts zur Abschätzung herangezogen werden. Diese Größe sei mit c bezeichnet. Dann gilt

$$X'_E = X_E + c\delta . \tag{4.36}$$

5 Traffic Shaping-Verfahren für komprimierte Videodaten

Mit Hilfe der Modellierung aus dem vorigen Kapitel werden in diesem Kapitel verschiedene Verfahren zum Traffic Shaping vorgestellt. Dabei gibt es zwei große Klassen von Verfahren. Bei den direkten Verfahren arbeitet der Shaping-Algorithmus direkt mit der Mengenfunktion des Videomaterials (d.h. hauptsächlich den einzelnen Datenmengen der kodierten Videobilder). Bei den Leaky Bucket-basierten Verfahren dagegen fließt die Mengenfunktion nur indirekt über ihre Hülle ein. Zuerst werden die indirekten, dann die direkten Verfahren vorgestellt.

Eine differenziertere Klassifizierung der Verfahren ergibt sich, wenn man betrachtet, welche Kenntnis über die kodierten Videodaten währen des Shaping-Vorgangs benutzt wird:

- Bei gespeicherten Videodaten ist die vollständige Mengenfunktion der kodierten Daten im voraus bekannt. Dann kann ein direktes Verfahren angewendet werden. Dieses kann eine globale Optimierung der Mengenfunktion des Shapers vornehmen. (Abschnitt 5.3)
- Wenn nur eine Hülle der Mengenfunktion der kodierten Daten im voraus bekannt ist, kann ein Leaky Bucket-basiertes Verfahren eingesetzt werden. Bei gespeicherten Videodaten kann die Hülle kompakter dargestellt werden als die Mengenfunktion. Bei Echtzeit-kodierten Daten kann eine Hülle geschätzt werden mittels einer repräsentativen Stichprobe. Im letzten Fall muß aber das Videomaterial und der Kodierer statistische Eigenschaften besitzen, die eine gute Schätzung erlauben. (Abschnitt 5.2)
- Wenn die Mengenfunktion des Videomaterials nicht im voraus bekannt ist, können stattdessen bestimmte Eigenschaften des Kodierers, z.B. das GOP-Muster, bekannt sein. Dann kann in Echtzeit die unmittelbaren Zukunft der Mengenfunktion geschätzt werden aus ihrem Verlauf in der Vergangenheit. Das ermöglicht wieder die Anwendung eines direkten Verfahrens, das jetzt mit einer Schätzung arbeitet. Im Gegensatz zum ersten Fall optimiert es die Mengenfunktion des Shapers nur lokal, soweit die Schätzung des zukünftigen Verlaufs einigermaßen zuverlässig ist. (Abschnitt 5.3.2)

5.1 Einfache Ansätze

Zur Einführung in dieses Kapitel sollen die schon länger bekannten einfachen Ansätze für das Traffic Shaping zusammengefaßt werden (siehe z.B. [Rath93]), die sich später als Spezialfälle der in dieser Arbeit behandelten Verfahren herausstellen werden. Hier sei der Fall Echtzeit-kodierten Videos angenommen. In Bild 5.1 sind die Ansätze in Beispielen anhand des Verlaufs der momentanen Rate $r(t)$ dargestellt.

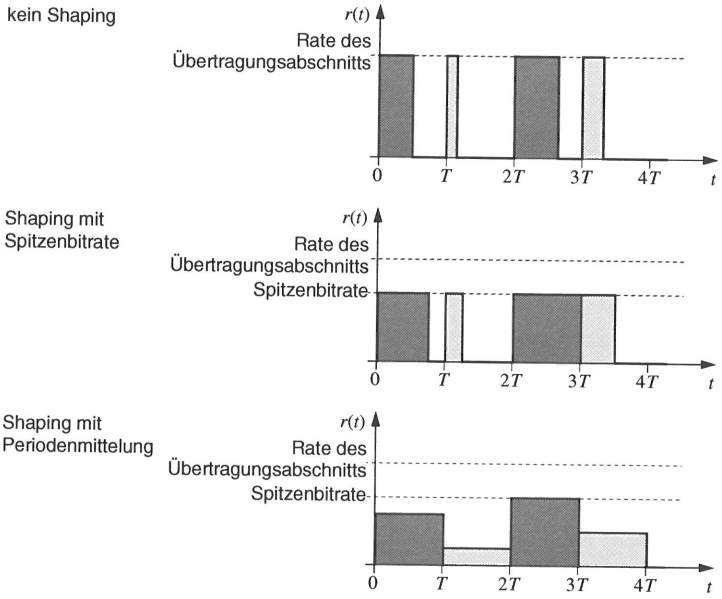


Bild 5.1: Einfache Traffic Shaping-Verfahren

- kein Shaping: Die Bilder werden mit der vollen Rate des Übertragungsabschnitts am Sender übertragen.
- Shaping mit Spitzenbitrate: Der Sender ermittelt die Größe des größten Bildes und dividiert diese durch die Bildperiode, das ergibt die sogenannte Spitzenbitrate des Videostroms. Die Bilder werden einheitlich mit dieser Rate gesendet.
- Shaping mit Periodenmittelung: Für jedes Bild ermittelt der Sender die Rate, die benötigt wird, um das Bild während genau einer Bildperiode zu senden. Diese Rate muß für jedes Bild neu eingestellt werden.

Offensichtlich garantieren die beiden letzten Fälle, daß jedes Bild innerhalb der Bildperiode T gesendet wird, daher ist T die maximale Verzögerung. Der Empfänger kann also seine Ausspielverzögerung auf T einstellen. Sowohl Sender als auch Empfänger müssen höchstens ein Bild zwischenspeichern, und die benötigte Speicherkapazität X_S und X_E entspricht der Größe des größten Bildes.

5.2 Leaky Bucket-basierte Verfahren

In diesem Abschnitt wird ein Traffic Shaper analysiert, der aus mehreren Instanzen eines Leaky Bucket-Shapers aufgebaut ist. Untersuchungsgegenstand sind die Verzögerung [Cruz91a], [Cruz91b], die benötigte Speicherkapazität und die Hülle des Ausgangsverkehrs. Zuerst wird ein einzelner Leaky Bucket-Shaper betrachtet, dann die Zusammenschaltung mehrerer Leaky Bucket-Shaper in Serie.

5.2.1 Einfacher Leaky Bucket

Betrachtet sei ein Netzelement mit Verkehrsströmen am Eingang und Ausgang, deren Mengenfunktionen jeweils mit $R_0(t)$ und $R_1(t)$ bezeichnet seien. Das Netzelement soll den Verkehrsstrom glätten, so daß seine Hülle bestimmte Eigenschaften aufweist. Ein (σ,ρ)-Shaper ist definiert als ein Netzelement, dessen Ausgangsverkehrsstrom konform zu einem (σ,ρ)-Leaky Bucket ist [Cruz91a]. Nun gibt es mehrere Strategien, den Verkehr zwischenzuspeichern und zu verzögern, um dieses Ziel zu erreichen. Ein pathologischer Shaper könnte z.B. den gesamten Verkehrsstrom auf unbestimmte Zeit absorbieren. Deshalb wird in der Definition zusätzlich gefordert, daß die momentane Verzögerung minimal ist. Daraus läßt sich dann in eindeutiger Weise ein Mechanismus ableiten. Hier soll stattdessen ein Shaper aus der Forderung nach minimaler Zwischenspeicherung abgeleitet werden. Wie später gezeigt werden wird, führt dies zum selben Ergebnis.

Im folgenden werden die Bezeichnungen "(σ,ρ)-Shaper" und "Leaky Bucket-Shaper" synonym verwendet. Wie sich herausstellen wird, liegt dem (σ,ρ)-Shaper ein Leaky Bucket-Mechanismus zugrunde.

5.2.1.1 Speicher

Definition: Ein (σ,ρ)-Shaper ist ein Netzelement, das die Hülle seines Ausgangsverkehrs $R_1(t)$ beschränkt gemäß der Schranke

$$\forall t > 0: W_\rho(R_1)(t) \leq \sigma \tag{5.1}$$

und dabei die momentan zwischengespeicherte Datenmenge $X(t)$ minimiert.

Aus der Schranke (5.1) läßt sich unmittelbar folgende untere Schranke für die Datenmenge $X(t)$ ableiten

$$X(t) \geq (W_\rho(R_0)(t) - \sigma)^+ \tag{5.2}$$

(hier wird die abkürzende Notation $(a)^+ = \max\{a,0\}$ verwendet). Gleichung (5.2) wird in Anhang 9.1 bewiesen. Diese Schranke gibt eine minimale Zwischenspeicherung an, die für jeden Shaper gilt, der (5.1) garantiert. Konstruierte man einen Shaper, der seinen Füllstand genau auf die untere Schranke einstellte, d.h.

$$X(t) = (W_\rho(R_0)(t) - \sigma)^+ , \tag{5.3}$$

so wäre er minimal, er erfüllte den zweiten Teil der Definition. Tatsächlich erfüllt (5.3) auch den ersten Teil der Definition (5.1), dies wird in Anhang 9.2 gezeigt. Daraus folgt, daß ein Shaper nach (5.3) insgesamt die Definition eines (σ,ρ) -Shapers erfüllt.

Aus (5.3) läßt sich die Funktionsweise des Shapers ersehen. Der Ausdruck $W_\rho(R_0)(t) - \sigma$ entspricht dem Anteil des Füllstandes eines Leaky Bucket, der die Grenze σ überschreitet. Solange der Verkehrsstrom am Eingang die Beschränkung der Hülle einhält, d.h. $W_\rho(R_0)(t) \le \sigma$, ist der Ausdruck kleiner gleich null und damit der Zwischenspeicher leer. Das bedeutet, daß sich der Shaper für den Verkehr transparent verhält. Im Fall daß die Hülle überschritten wird, d.h. $W_\rho(R_0)(t) > \sigma$, wird die überschüssige Datenmenge zwischengespeichert. Der Shaper "tritt in Aktion". In diesem Zustand erlaubt die Beschränkung der Hülle dem Shaper, den Speicher mit Rate ρ zu leeren. Ein Beispiel für die Funktionsweise wird in Bild 5.2 gezeigt.

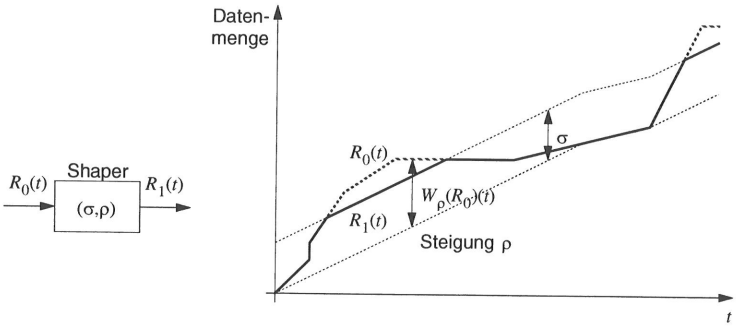


Bild 5.2: Funktionsweise eines Leaky Bucket Shapers

Für die Dimensionierung der Speicherkapazität des Shapers muß der maximal auftretende Füllstand bekannt sein. Die Speicherkapazität $X_{\sigma,\rho}(R_0)$, die zum Glätten des Verkehrsstroms mit Mengenfunktion $R_0(t)$ nötig ist, ist gegeben durch

$$X_{\sigma,\rho}(R_0) = \max_{t \ge 0} X(t) = (W_{\rho,\max}(R_0) - \sigma)^+ , \tag{5.4}$$

wobei die Bezeichnung $W_{\rho,\max}(R_0) = \max_{t \geq 0} W_\rho(R)(t)$ für den maximal auftretenden Füllstand des Leaky Bucket verwendet wird.

5.2.1.2 Verzögerung

Die durch den Shaper verursachte momentane Verzögerung richtet sich nach dem momentanen Füllstand. Offensichtlich ist die Verzögerung null, wenn der Füllstand null und der Shaper transparent ist. Im anderen Fall gelangen die Daten in den Speicher, der mit Rate ρ geleert wird. Ein Beobachter, der zum Zeitpunkt t in den Speicher gelangt, muß warten, bis die vor ihm im Speicher befindlichen Daten, das ist die Menge $X(t)$, den Speicher mit Rate ρ verlassen haben. Diese Überlegung legt für die momentane Verzögerung die folgende Beziehung nahe

$$d(t) = \frac{1}{\rho} X(t) . \tag{5.5}$$

Ein formaler Beweis für (5.5) ist in Anhang 9.3 angegeben. Dieses Ergebnis ist identisch mit dem Shaper, den man erhält, wenn eine minimale momentane Verzögerung gefordert wird [Cruz91a]. Aus der Beschränkung der Hülle (5.1) folgt nämlich

$$d(t) \geq \frac{1}{\rho} (W_\rho(R_0)(t) - \sigma)^+ . \tag{5.6}$$

Ein (σ,ρ)-Shaper minimiert also sowohl die momentane Zwischenspeicherung, als auch die momentane Verzögerung.

Die vom Shaper eingeführte variable Verzögerung muß am Empfänger wieder ausgeglichen werden. Für die Dimensionierung der Ausspielverzögerung muß nach (4.25) die maximal auftretende Verzögerung bekannt sein. Die bei einem Verkehrsstrom mit Mengenfunktion $R_0(t)$ maximal auftretende Verzögerung $d_{\sigma,\rho}(R_0)$ ist

$$d_{\sigma,\rho}(R_0) = \max_{t \geq 0} d(t) = \frac{1}{\rho} X_{\sigma,\rho}(R_0) = \frac{1}{\rho} (W_{\rho,\max}(R_0) - \sigma)^+ . \tag{5.7}$$

5.2.2 Mehrere Leaky Buckets

Die Verknüpfung mehrerer (σ,ρ) Shaper in Serie wird im folgenden Mehrfach-Leaky Bucket (MLB)-Shaper genannt. Der MLB-Shaper bestehe aus n (σ,ρ)-Shapern mit den Parametern (σ_m,ρ_m), $m = 1..n$. Die Shaper seien so verknüpft, daß der Verkehrsstrom am Ausgang von

Shaper m mit Mengenfunktion $R_m(t)$ den Verkehrsstrom am Eingang von Shaper $m+1$ darstellt. Der Verkehrsstrom am Eingang des MLB-Shapers habe die Mengenfunktion $R_0(t)$, der am Ausgang $R_n(t)$ (siehe Bild 5.3).

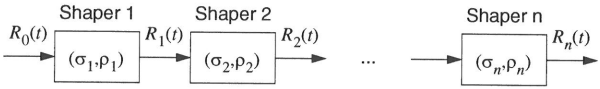


Bild 5.3: MLB-Shaper

Bei dieser Anordnung stellt sich die Frage, wie der durch einen Shaper geglättete Verkehrsstrom durch die nachfolgenden Shaper verändert wird. Insbesondere stellt sich die Frage nach der Gültigkeit der Schranke (5.1). Zur Beantwortung wird zunächst die im Gesamtsystem zwischengespeicherte Datenmenge untersucht.

5.2.2.1 Speicher

Für die Bestimmung der momentan im gesamten MLB-Shaper zwischengespeicherten Datenmenge wird folgendes Lemma für einen (σ,ρ) Shaper benötigt:

Lemma 5.1: ((σ,ρ) Shaper) Gegeben sei ein $\bar{\rho} \geq 0$. Es gilt für jedes $t \geq 0$

$$W_{\bar{\rho}}(R_1)(t) = (W_{\bar{\rho}}(R_0)(t) - X(t))^+ . \tag{5.8}$$

Ein Beweis ist in Anhang 9.4 angegeben. Das Lemma ermöglicht, den Füllstand eines dem (σ,ρ)-Shaper folgenden Leaky Bucket mit Rate $\bar{\rho}$ zu bestimmen. Mit $X_k(t)$ werde die Summe der Füllstände der k ersten Shaper bezeichnet.

Satz 5.1: (n (σ,ρ) Shaper in Serie) Für alle $k = 1..n$ gilt

$$X_k(t) = \max_{m=1..k} (W_{\rho_m}(R_0)(t) - \sigma_m)^+ . \tag{5.9}$$

Ein Beweis ist in Anhang 9.5 angegeben. Mit anderen Worten, die gesamte zwischengespeicherte Datenmenge entspricht dem Maximum (über alle Shaper) der Datenmenge, die von einem einzelnen Shaper zwischengespeichert würde, wenn dieser vom Verkehrsstrom gespeist würde.

Die für einen Verkehrsstrom mit Mengenfunktion $R_0(t)$ maximal gespeicherte Datenmenge $X_{\eth,\eth}(R_0)$ ($\eth$ und $\eth$ sollen für die Parameter des MLB-Shapers stehen) ergibt sich unmittelbar aus (5.9) und (5.4)

$$X_{\eth,\eth}(R_0) = \max_{t \geq 0} X_n(t) = \max_{m=1..n} X_{\sigma_m,\rho_m}(R_0)$$

$$= \max_{m=1..n} (W_{\rho_m,\max}(R_0) - \sigma_m)^+ \,. \tag{5.10}$$

5.2.2.2 Paralleles Ersatzschaltbild

Für die Mengenfunktion des Verkehrsstroms am Ausgang des MLB-Shapers gilt mit (5.9)

$$\begin{aligned} R_n(t) &= R_0(t) - X_n(t) \\ &= R_0(t) - \max_{m=1..n} (W_{\rho_m}(R_0)(t) - \sigma_m)^+ \\ &= \min_{m=1..n} [R_0(t) - (W_{\rho_m}(R_0)(t) - \sigma_m)^+] \,. \end{aligned} \tag{5.11}$$

Der Ausdruck in eckigen Klammern ist der Ausgangsverkehr eines (σ_m,ρ_m)-Shapers, der von $R_0(t)$ gespeist wird. Damit ergibt sich ein Ersatzschaltbild für den MLB-Shaper, das aus n parallel geschalteten (σ_m,ρ_m)-Shapern besteht. Die Ausgänge der Shaper werden dabei mit einem Netzelement zusammengefaßt, das am Ausgang die minimale Verkehrsmenge seiner Eingänge durchläßt (siehe Bild 5.4).

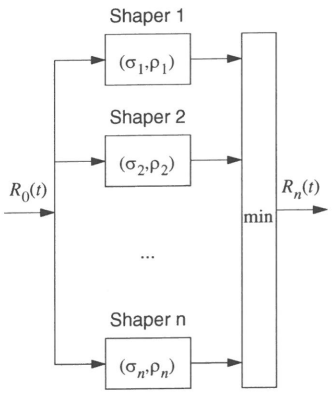


Bild 5.4: Ersatzschaltbild für den MLB-Shaper

Damit ist es jetzt möglich, die Frage nach der Veränderung des Verkehrsstroms durch nachfolgende Shaper zu beantworten. Das Ersatzschaltbild macht deutlich, daß sich die Effekte der Shaper gegenseitig nicht aufheben. Genauer gesagt, die von jedem Shaper gemäß

Definition (5.1) garantierte Hülle $b_m(\Delta t) = \sigma_m + \rho_m \Delta t$ gilt auch für den Ausgang des MLB-Shapers. Insgesamt ist der Verkehrsstrom am Ausgang des MLB-Shapers also durch folgende Hülle beschränkt

$$b(\Delta t) = \min_{m = 1..n} \{\sigma_m + \rho_m \Delta t\} . \tag{5.12}$$

Ein anderer Beweis dieser Aussage ist in [Cruz91b] zu finden.

5.2.2.3 Verzögerung

Die gesamte Verzögerung eines MLB-Shapers ist in [Cruz91a] und [Cruz91b] untersucht worden. Hier sollen die Ergebnisse zusammengefaßt werden. Das folgende Lemma ist das Analogon zu Lemma 5.1:

Lemma 5.2: ((σ,ρ)-Shaper) Gegeben sei ein $\bar{\rho} \geq 0$. Es gilt für jedes $t \geq 0$

$$W_{\bar{\rho}}(R_1)(t + d(t)) = (W_{\bar{\rho}}(R_0) - \bar{\rho} d(t))^+ . \tag{5.13}$$

Mit $d_k(t)$ werde die Summe der Verzögerungen in den k ersten Shapern bezeichnet. Das Analogon zu Satz 5.1 lautet:

Satz 5.2: Für alle $k = 1..n$ gilt

$$d_k(t) = \max_{m = 1..k} \left[\frac{1}{\rho_m} (W_{\rho_m}(R_0)(t) - \sigma_m)^+ \right] . \tag{5.14}$$

Mit anderen Worten, die gesamte Verzögerung entspricht dem Maximum (über alle Shaper) der Verzögerung eines einzelnen Shapers, wenn dieser vom Verkehrsstrom gespeist würde.

Die bei einem Verkehrsstrom mit Mengenfunktion $R_0(t)$ maximal auftretende Verzögerung $d_{\vec{\sigma},\vec{\rho}}(R_0)$ ist

$$d_{\vec{\sigma},\vec{\rho}}(R_0) = \max_{t \geq 0} d_n(t) = \max_{m = 1..n} d_{\sigma_m,\rho_m}(R_0)$$

$$= \max_{m = 1..n} \frac{1}{\rho_m} (W_{\rho_m,\max}(R_0) - \sigma_m)^+ . \tag{5.15}$$

5.2.3 MLB-Shaper für Echtzeit-kodierte Videodaten

In diesem Abschnitt wird der MLB-Shaper im Zusammenhang mit dem Modell für die Übertragung Echtzeit-kodierter Videodaten von Abschnitt 4.3.1 behandelt. Dabei geht es um die Abwägungen, die zu treffen sind, wenn ein MLB-Shaper für einen gegebenen Verkehrsstrom parametrisiert wird. Ziel ist es, die Hülle des Verkehrsstroms am Ausgang des Shapers zu minimieren, und damit eine sparsame Zuweisung von Netzressourcen zu ermöglichen. Der dabei verfolgte Ansatz beginnt mit einem einzelnen (σ,ρ)-Shaper und fügt dann weitere Shaper schrittweise zu.

5.2.3.1 Leistungsabwägung beim Ein/Aus-Shaper

5.2.3.1.1 Abhängigkeiten zwischen Leistungsgrößen

Betrachtet werde das Sendermodell von Abschnitt 4.3.1. Der Sender sei mit einem einzelnen (σ,ρ)-Shaper ausgestattet. Der Verkehrsstrom am Eingang des Shapers wird vom Kodierer erzeugt, es gilt $R_0(t) = C(t)$. Bei der Modellierung des Kodierers wurde die plötzliche Generierung der Daten eines Bildes oder einer Zugriffseinheit angenommen, wodurch $C(t)$ Sprünge enthält, welche eine unendlich hohe Rate darstellen. Um eine endliche Rate am Ausgang des Shapers zu erhalten, darf der Shaper deshalb nie transparent sein. Das bedeutet $\sigma = 0$. Solch ein Shaper wird im folgenden *Ein/Aus-Shaper* genannt, weil er entweder Daten mit Rate ρ (Zustand "ein") oder gar nicht (Zustand "aus") sendet (siehe Bild 5.5). Er garan-

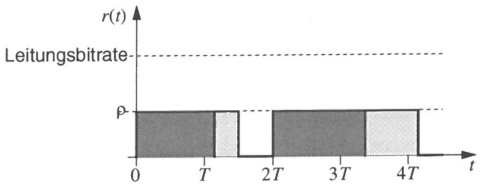


Bild 5.5: Verkehrsstrom des Ein/Aus-Shapers

tiert damit am Ausgang eine Hülle $b(\Delta t) = \rho\Delta t$. Für einen Ein/Aus-Shaper vereinfacht sich der Ausdruck für seinen maximalen Füllstand (5.4) zu

$$X_{0,\rho}(R_0) = W_{\rho,\max}(R_0), \tag{5.16}$$

und der Ausdruck für seine maximale Verzögerung (5.7) zu

$$d_{0,\rho}(R_0) = \frac{1}{\rho} W_{\rho,\max}(R_0) . \tag{5.17}$$

Die Größe $W_{\rho,\max}(R_0)$ spielt also eine zentrale Rolle. Sie kann aus der empirischen Hülle $\varepsilon(R_0)(\tau)$ des Verkehrsstroms bestimmt werden analog zu dem Vorgehen in Abschnitt 4.1.2.4

$$\begin{aligned} W_{\rho,\max}(R_0) &= \max_{t \geq 0} \max_{0 \leq s \leq t} [R(s,t) - \rho(t-s)] \\ &= \max_{\tau \geq 0} [\varepsilon(R_0)(\tau) - \rho\tau] . \end{aligned} \tag{5.18}$$

Geometrisch läßt sich $W_{\rho,\max}(R_0)$ interpretieren als y-Achsenabschnitt der tiefsten Geraden mit Steigung ρ, die oberhalb der Funktion $\varepsilon(R_0)(\tau)$ liegt.

Betrachtet man $W_{\rho,\max}(R_0)$ als Funktion von ρ, so stellt diese Funktion die Abwägung dar zwischen einer Verminderung der Ausgangsrate des Shapers, und damit der belegten Netzressourcen, auf der einen Seite und einer Erhöhung der Verzögerung und des benötigten Speichers im Shaper auf der anderen Seite. Für die Diskussion ihres prinzipiellen Verlaufs sei als Beispiel das Kodierermodell (4.19) herangezogen. Es gelte $R_0(t) = C(t)$. Die empirische Hülle ist dann

$$\varepsilon(R_0)(\tau) = \max_{i \geq 0} \sum_{k=i}^{i + \lfloor \tau/T \rfloor} s_k . \tag{5.19}$$

Dies ist eine nicht-stetige, stufenförmige Funktion mit einem Sprung bei jeder Bildperiode T (siehe Bild 5.6).

Wenn ρ groß genug ist, berührt die Gerade die Hülle offensichtlich genau einmal am Punkt $\tau = 0$ und für den y-Achsenabschnitt gilt $W_{\rho,\max}(R_0) = \varepsilon(R_0)(0)$. Wenn ρ verkleinert wird, bleibt der y-Achsenabschnitt konstant, bis die Gerade die Hülle an einem zweiten Punkt berührt. Wenn ρ weiter verkleinert wird, rotiert die Gerade um den zweiten Berührungspunkt und der y-Achsenabschnitt beginnt zu wachsen, bis ein neuer Berührungspunkt erreicht wird, usw. Für kleiner werdende ρ liegen die Berührungspunkte immer weiter rechts. Dadurch verlängert sich der Hebel und die Ableitung $\frac{d}{d\rho} W_{\rho,\max}(R_0)$ nimmt zu. Es ergibt sich eine stückweise lineare Funktion, deren prinzipieller Verlauf in Bild 5.7 dargestellt ist.

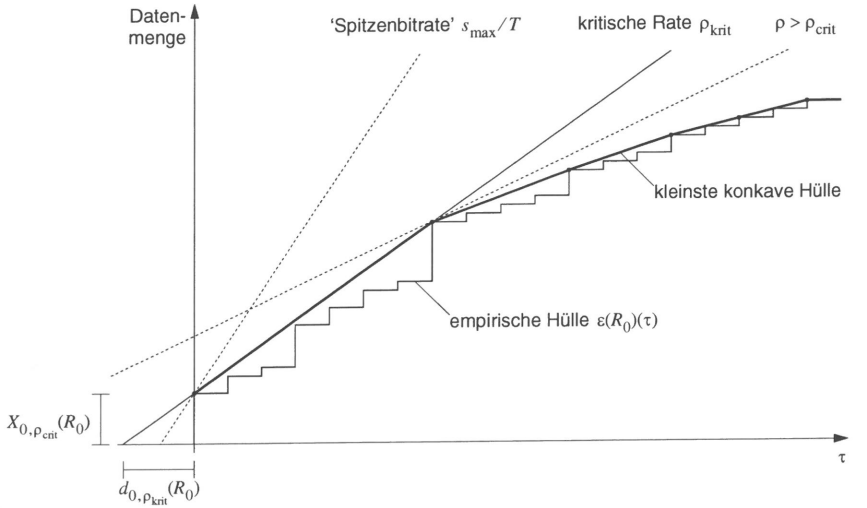


Bild 5.6: Bestimmung von Ein/Aus-Shaper-Parametern aus der empirischen Hülle

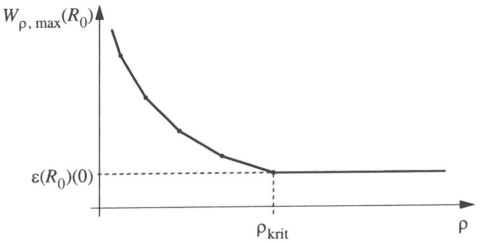


Bild 5.7: Prinzipieller Verlauf von $W_{\rho,\,max}(R_0) = f(\rho)$

5.2.3.1.2 Kritische Rate

Bei einer Funktion, die eine Abwägung zweier Kosten darstellt (hier Ausgangsrate und Speicher/Verzögerung), sind Knicke im Verlauf von besonderer Bedeutung, da sich an diesen Stellen die Steigung sprunghaft ändert und damit das Verhältnis der marginalen Kosten. Für die Einstellung eines Arbeitspunktes sind diese Stellen vorteilhaft, weil sie oft lokale Minima der

Gesamtkosten darstellen. Ein Knick liegt bei der vorigen Betrachtung immer dann vor, wenn sich der Berührungspunkt ändert. Ein ausgeprägter Knick, der auch unabhängig vom Verlauf der empirischen Hülle ist, ergibt sich an der Stelle, bei der sich der Berührungspunkt zum ersten Mal ändert. Die Rate an dieser Stelle wird im folgenden *kritische Rate* ρ_{krit} genannt. Für die Region $\rho \geq \rho_{krit}$ gilt

$$X_{0,\rho}(R_0) = \varepsilon(R_0)(0), \tag{5.20}$$

$$d_{0,\rho}(R_0) = \frac{1}{\rho}\varepsilon(R_0)(0), \tag{5.21}$$

also ein von der Rate unabhängiger, konstanter Speicherbedarf und eine umgekehrt proportionale Verzögerung. Ein Arbeitspunkt in dieser Region ist damit besonders vorteilhaft, weil keine zusätzlichen Speicher-Ressourcen benötigt werden. Die kritische Rate ist eine Eigenschaft des Verkehrsstroms. Aus der geometrischen Konstruktion ergibt sich

$$\rho_{crit}(R_0) = \max_{\tau \geq 0} \frac{\varepsilon(R_0)(\tau) - \varepsilon(R_0)(0)}{\tau}. \tag{5.22}$$

Für den Fall eines nach dem Kodierermodell (4.19) erzeugten Verkehrsstroms gilt, daß die empirische Hülle an der Stelle $\tau = 0$ der Größe des größten Bildes entspricht (nach (5.19) ist $\varepsilon(R_0)(0) = \max_{k \geq 0} s_k = s_{max}$). Da die Größe jeden Bildes s_i nach oben durch die maximale Bildgröße beschränkt werden kann, ergibt sich $\varepsilon(R_0)(iT) \leq (i+1)s_{max}$. Mit (5.22) erhält man daraus eine obere Schranke für die kritische Rate

$$\rho_{krit} \leq \frac{s_{max}}{T}. \tag{5.23}$$

Üblicherweise wird s_{max}/T die "Spitzenbitrate" einer Folge von Bildern genannt. Für $\rho_1 = s_{max}/T$ erhält man den Fall des Shapings mit Spitzenbitrate von Seite 83. Bei dieser Rate entspricht die Verzögerung genau der Bildperiode: $d_{0,\rho}(R_0) = T$.

5.2.3.1.3 Kleinste konkave Hülle

Die Funktion $W_{\rho, max}(R_0) = f(\rho)$ stellt eine Art Abtastung der empirischen Hülle dar. Wie deutlich wurde, spielen dabei nur die Berührungspunkte mit der Hülle eine Rolle. Der Verlauf der Hülle zwischen den Berührungspunkten ist nicht relevant. Verbindet man die Berührungs-

punkte, erhält man eine stückweise lineare Funktion, die die *kleinste konkave*[1] *Hülle* des Verkehrsstroms darstellt. Sie ist allein für die Größe von $W_{\rho,\max}(R_0)$ ausschlaggebend. Ihre allgemeine Formel ist

$$b(\Delta t) = \min_{i=1..n} \{\sigma_i + \rho_i \Delta t\} .$$

Sie hat damit dieselbe Form wie der Verkehrsdeskriptor aus Leaky Buckets aus Abschnitt 4.1.3.

5.2.3.2 Dimensionierung des MLB-Shapers

Nach der Betrachtung eines einzelnen Ein/Aus-Shapers folgt jetzt die Dimensionierung eines MLB-Shapers, der aus einem Ein/Aus-Shaper und zusätzlichen Shapern mit niedrigerer Rate und $\sigma > 0$ besteht. Die Shaper mit niedrigerer Rate verringern nach (5.12) zusätzlich die Hülle am Ausgang. Ein zusätzlicher Shaper kann ein zusätzliches Geradensegment zur Hülle beitragen. Es gibt also eine Abwägung zwischen der Zahl der Shaper und der Güte der Hülle.

5.2.3.2.1 Gegebene Verzögerung

Zuerst sei der Fall betrachtet, daß die maximale Verzögerung durch den Shaper gegeben ist und die Speicherkapazität daran angepaßt wird. Mit der Bedingung $d_{\sigma,\rho}(R_0) \le D$ erhält man aus (5.15) für die Shaper Parameter σ_m und ρ_m

$$\sigma_m \ge W_{\rho_m,\max} - \rho_m D . \tag{5.24}$$

Es sei angenommen, daß der durch D gegebene Freiraum vollständig genutzt wird und damit für jeden Shaper gilt

$$\sigma_m = W_{\rho_m,\max} - \rho_m D . \tag{5.25}$$

Der erste Shaper sei der Ein/Aus-Shaper mit $\sigma_1 = 0$. Gleichung (5.25) soll nach ρ_1 als Funktion von D aufgelöst werden. Dies ist die Umkehrfunktion zu $d_{0,\rho}(R_0)$ in (5.17). Mit den geometrischen Argumenten von Abschnitt 5.2.3.1.1 wird deutlich, daß $d_{0,\rho}(R_0)$ streng monoton fallend und daher umkehrbar ist. Offensichtlich ergibt sich aus der Geometrie

$$\rho_1(D) = \max_{\tau > 0} \frac{\varepsilon(R_0)(\tau)}{\tau + D} \text{ für } D \ge 0 \tag{5.26}$$

---

1. Eine stetige Funktion $f(x)$ ist konkav, wenn $\forall x,y, 0 \le \alpha \le 1$: $f(\alpha x + (1-\alpha)y) \ge \alpha f(x) + (1-\alpha) f(y)$.

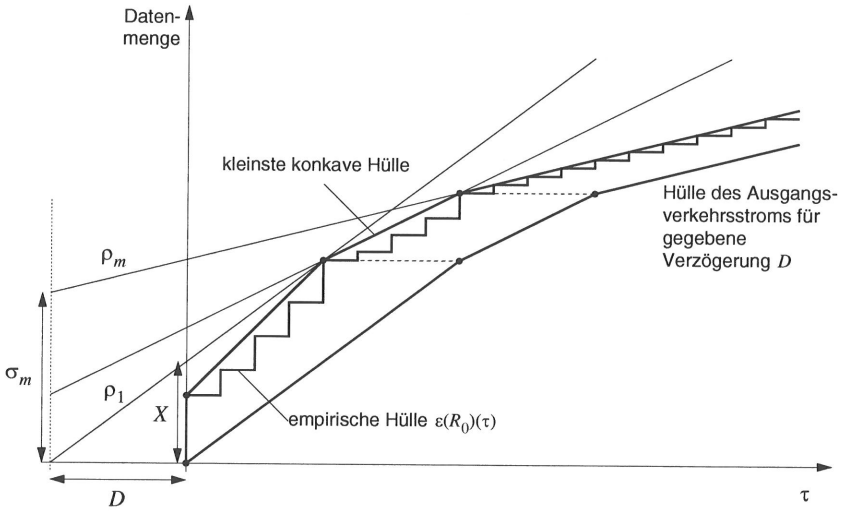


Bild 5.8: Dimensionierung für gegebene Verzögerung

(siehe Bild 5.8).

Für die übrigen Shaper mit $\sigma_m > 0$ läßt (5.25) einen Freiheitsgrad. Ihre Rate kann frei gewählt werden. Außerdem ist die Zahl der Shaper noch offen. Zunächst werde die Auswirkung der Rate ρ_m auf die Hülle des Ausgangsverkehrs untersucht, um dann Schlüsse auf die Zahl der Shaper zu ziehen. Geometrisch interpretiert ist σ_m der Schnittpunkt der tiefsten Gerade mit Steigung ρ_m, die über der Funktion $\varepsilon(R_0)(\tau)$ liegt (siehe Bild 5.8). Shaper m trägt zur Hülle (5.12) die Gerade $b(\tau) = \sigma_m + \rho_m\tau$ bei, die eine um D Einheiten nach rechts verschobene Kopie der ersten Geraden darstellt. Wie gezeigt wurde, hängt $W_{\rho,\max}(R_0)$ nur von der kleinsten konkaven Hülle von R_0 ab. Hätte man unbegrenzt viele Shaper zur Verfügung, könnte man jeden Wert ρ_m im Intervall $[0,\rho_1)$ abdecken. Man erhielte für die Hülle am Ausgang eine Geradenschar. Das Minimum der Geradenschar ist eine stückweise lineare Funktion. Ihre Segmente, das erste ausgenommen, entsprechen den Segmenten mit Rate kleiner ρ_1 der kleinsten konkaven Hülle von R_0 wenn sie um D Einheiten nach rechts geschoben werden.

Mit l werde die Zahl der Segmente der kleinsten konkaven Hülle bezeichnet, deren Rate kleiner ist als ρ_1. Tatsächlich braucht man nur l Shaper (zusätzlich zum Ein/Aus-Shaper) um dieselbe Hülle wie bei unbegrenzt vielen Shapern zu erreichen, nämlich für jedes Segment ein Shaper. Dabei entspricht die Rate ρ_m der Steigung des Segments. In [GGPS95]

wurde gezeigt, daß solch ein MLB-Shaper optimal ist in dem Sinne, daß er die kleinste Hülle aller Shaper mit konkaver Hülle besitzt. In einer Implementierung ist jedoch die Zahl der verfügbaren Shaper begrenzt. Ein optimaler MLB-Shaper ist daher im allgemeinen nicht realisierbar. Wenn die Implementierung nur $n < l$ Shaper zuläßt, besteht ein sinnvoller Ansatz darin, die Raten der Shaper auf die ersten $n-1$ Segmente einzustellen. (In spezifischen Kontexten können eventuell andere Ansätze besser geeignet sein.) Es gibt eine Abwägung zwischen der Zahl der Shaper und der Hülle, die für den Verkehrsstrom am Ausgang garantiert werden kann. Für kleine n leidet die Güte der Schranke von (5.12). In einem kommenden Abschnitt wird anhand zusätzlicher Analyse ein Satz vorgestellt, der eine engere Schranke aufstellt.

Als nächstes werde die nötige Speicherkapazität eines nach (5.25) dimensionierten Shapers ermittelt. Sie ist nach (5.10)

$$X_{\vartheta,\hat{\rho}}(R_0) = \max_{m=1..n} \rho_m D = \rho_1 D . \tag{5.27}$$

Als wichtiges Ergebnis ergibt sich, daß der Speicherbedarf nur von den Parametern des Ein/Aus-Shapers abhängt, und nicht von den Parametern der Shaper mit niedrigerer Rate. Daraus folgt auch, daß das Hinzufügen weiterer Shaper den Speicherbedarf nicht erhöht.

5.2.3.2.2 Gegebene Speicherkapazität

Nun sei der Fall betrachtet, daß der MLB-Shaper für eine gegebene Speicherkapazität X dimensioniert wird, und die Verzögerung nachträglich ermittelt wird. Mit der Bedingung $X_{\vartheta,\hat{\rho}}(R_0) \le X$ erhält man aus (5.10)

$$\sigma_m \ge W_{\rho_m,\max}(R_0) - X , \tag{5.28}$$

und es gelte wie vorhin bei der Verzögerung die volle Ausnutzung des Freiraums

$$\sigma_m = W_{\rho_m,\max}(R_0) - X . \tag{5.29}$$

Um die Rate des Ein/Aus-Shapers mit $\sigma_1 = 0$ zu bestimmen, ist (5.29) nach ρ_1 als Funktion von X aufzulösen. Diese Funktion ist die Umkehrfunktion von $X_{0,\rho}(R_0)$ in (5.16). Aus der geometrischen Konstruktion ergibt sich, daß $X_{0,\rho}(R_0)$ streng monoton fallend ist, und daher umkehrbar ist, wenn $\rho > \rho_{krit}$ oder $X > \varepsilon(R_0)(0)$. Die Umkehrfunktion ist

$$\rho_1(X) = \max_{\tau > 0} \frac{\varepsilon(R_0)(\tau) - X}{\tau} \text{ für } X \ge \varepsilon(R_0)(0) , \tag{5.30}$$

wobei die Definition von $\rho_1(X)$ für $X = \varepsilon(R_0)(0)$ erweitert wurde durch

$$\rho_1(\varepsilon(R_0)(0)) = \max_{\tau > 0} \frac{\varepsilon(R_0)(\tau) - \varepsilon(R_0)(0)}{\tau} = \rho_{krit}.$$

Die Dimensionierung der Shaper und die Konstruktion der Hülle am Ausgang geschieht analog zum vorigen Fall durch Betrachten der Geradenschar bei unbegrenzt vielen Shapern. Diesmal entspricht das Minimum der Geradenschar einer um X Einheiten nach unten verschobenen Kopie der Segmente mit Rate kleiner ρ_1 der kleinsten konkaven Hülle (siehe Bild 5.9). Wieder werden die Shaper diesen Segmenten zugeordnet. Im allgemeinen ist die

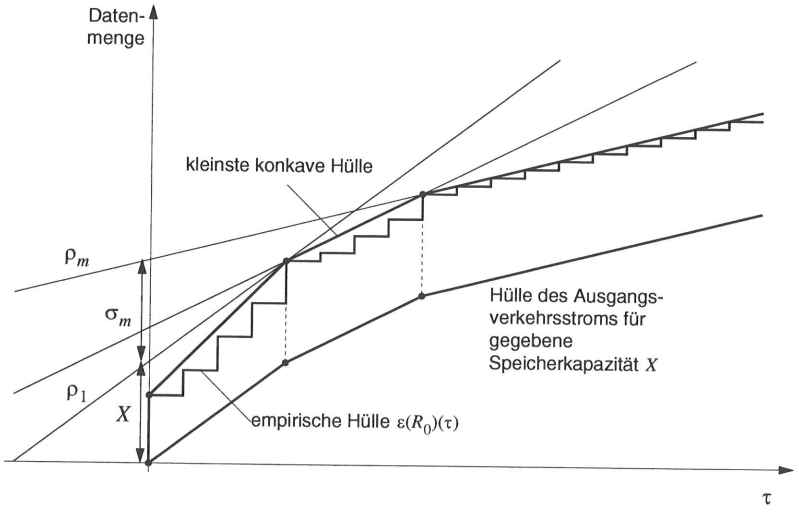


Bild 5.9: Dimensionierung für gegebene Speicherkapazität

gesamte Hülle bei gleicher Speicherkapazität kleiner als im vorigen Fall, wie in den Beispielen von Bild 5.8 und Bild 5.9 gezeigt. Auf der anderen Seite erhält man für die Abschätzung der Verzögerung, die bei (5.29) auftritt, nach (5.15)

$$d_{\vec{\sigma},\vec{\rho}}(R_0) = \max_{m=1..n\rho_m} \frac{X}{} \quad . \tag{5.31}$$

Dieser Wert kann beliebig groß werden, in Abhängigkeit von der kleinsten Rate aller Shaper.

5.2.3.3 Optimalität der maximalen Rate des MLB-Shapers

Hier soll gezeigt werden, daß die maximale Rate eines MLB-Shapers, d.h. die Rate seines Ein/Aus-Shapers, optimal ist. Genauer formuliert:

Satz 5.3: (1) Jeder Shaper, der eine maximale Verzögerung $D \geq 0$ garantiert, erzeugt eine maximale Rate, die größer oder gleich der Rate in (5.26) ist. (2) Jeder Shaper, der einen maximalen Füllstand seines Zwischenspeichers $X \geq \varepsilon(R_0)(0)$ garantiert, erzeugt eine maximale Rate, die größer oder gleich der Rate in (5.30) ist.

Beweis: Es sei angenommen, der Shaper garantiere eine maximale Verzögerung $D \geq 0$ und produziere einen Verkehrsstrom mit maximaler Verzögerung $\rho' < \rho_1(D)$. Dann garantiert er eine Hülle $b(\tau) = \rho'\tau$ und seine Verzögerung im ungünstigsten Fall ist mit (5.6) nach unten beschränkt durch

$$D \geq \frac{1}{\rho'} W_{\rho', \max}(R_0) = \rho_1^{-1}(\rho') .$$

Da $\rho_1^{-1}(\rho')$ streng monoton fallend ist, ergibt sich $D > D$, also ein Widerspruch.

Die gleiche Argumentation kann für die Speicherkapazität herangezogen werden. ♦

5.2.3.4 Speicherkapazität des Empfängers

Bis jetzt wurde nur der Pufferspeicher am Sender betrachtet. Der Empfänger besitzt einen zweiten Pufferspeicher, um den aus dem Netz kommenden Verkehrsstrom vom Zeitplan des Dekodierers abzukoppeln. Dieser Zeitplan wurde gleich dem Zeitplan des Kodierers angenommen. Gewissermaßen sitzt im Empfänger ein zweiter Shaper, der den ursprünglichen Zeitplan des Kodierers wieder herstellt. Zusätzlich zu der für diesen Zweck benötigten Speicherkapazität braucht der Empfänger auch noch Platz, um die durch das Netz verursachten Verzögerungsschwankungen auszugleichen, wie in Abschnitt 4.4.3 dargestellt wurde. Dies wird hier nicht berücksichtigt.

Es sei also angenommen, daß der Pufferspeicher direkt vom Ausgang des MLB-Shapers gefüllt und vom Dekodierer geleert wird. Der MLB-Shaper sei für eine Verzögerung D dimensioniert worden. Die vom Dekodierer ausgelesene Datenmenge ist $R_0(t - D_E)$, wobei D_E die Ausspielverzögerung ist. Um ein Unterlaufen des Speichers zu verhindern, muß nach (4.25) $D_E \geq D$ sein, andererseits soll die Speicherkapazität des Empfängers minimiert werden, die mit D_E zunimmt. Deshalb sei $D_E = D$ gewählt, wie auch in (4.26) angenommen. Der momentane Füllstand des Empfängers ist dann

$$X_r(t) = R_n(t) - R_0(t - D)$$

$$= R_0(t) - \max_{m=1..n} (W_{\rho_m}(R_0)(t) - \sigma_m)^+ - R_0(t-D)$$

$$\leq R_0(t-D,t) - W_{\rho_1}(R_0)(t). \tag{5.32}$$

Nun kann $W_{\rho_1}(R_0)(t) = \max_{0 \leq s \leq t} [R_0(s,t) - \rho_1(t-s)]$ für $s = t-D$ nach unten beschränkt werden

$$W_\rho(R_0)(t) \geq R_0(t-D,t) - \rho D. \tag{5.33}$$

Daraus folgt

$$X_r(t) \leq R_0(t-D,t) - R_0(t-D,t) + \rho_1 D$$

$$= \rho_1 D = X. \tag{5.34}$$

Mit anderen Worten ist die Speicherkapazität des Empfängers nach oben durch die Speicherkapazität des Senders begrenzt.

5.2.4 Hülle für den Verkehrsstrom am Ausgang eines MLB-Shapers

Die von (5.12) vorhergesagte Hülle für den Verkehrsstrom am Ausgang eines MLB-Shapers kann ziemlich weit sein. Zum Beispiel wenn eine Implementierung nur wenige (σ,ρ)-Shaper zuläßt. Oder wenn der MLB-Shaper für einen bestimmten Verkehrsstrom, genauer gesagt für eine Hülle dimensioniert wurde, aber dann für einen anderen Verkehrsstrom mit kleinerer Hülle eingesetzt wird. Hier werden jetzt engere Schranken abgeleitet, unter der Annahme, daß die Parameter σ und ρ des Shapers und eine Hülle des Verkehrsstroms am Eingang bekannt ist. Zunächst für einen einzelnen (σ,ρ)-Shaper, dann für einen MLB-Shaper.

Satz 5.4: ((σ,ρ)-Shaper) Wenn der Verkehrsstrom am Eingang durch eine Hülle $b_0(\tau)$ beschränkt ist, dann ist der Verkehrsstrom am Ausgang durch folgende Hülle beschränkt:

$$b_1(\tau) = b_0(\tau) - (W_\rho(b_0)(\tau) - \sigma)^+. \tag{5.35}$$

Ein Beweis ist in Anhang 9.6 angegeben. Es gibt eine Übereinstimmung in der Form zwischen (5.35) und dem Ausdruck für die Mengenfunktion des Verkehrsstrom am Ausgang des Shapers

$$R_1(t) = R_0(t) - X(t) = R_0(t) - (W_\rho(R_0)(t) - \sigma)^+.$$

Mit anderen Worten: Wenn eine Hülle des Eingangsverkehrs als hypothetischer Eingangsverkehr aufgefaßt und in den Shaper gespeist wird, dann ist der hypothetische Ausgangsverkehr eine Hülle für den tatsächlichen Ausgangsverkehr.

Der folgende Satz erweitert diese Eigenschaft auf k in Serie geschaltete (σ,ρ)-Shaper.

Satz 5.5: (k (σ,ρ)-Shaper in Serie) Wenn der Verkehrsstrom am Eingang durch eine Hülle $b_0(\tau)$ beschränkt ist, dann ist der Verkehrsstrom am Ausgang durch folgende Hülle beschränkt:

$$b_k(\tau) = b_0(\tau) - \max_{m=1..k}(W_{\rho_m}(b_0)(\tau) - \sigma_m)^+ . \tag{5.36}$$

Beweis: Es wird gezeigt, daß (5.36) für das parallele Ersatzschaltbild aus Abschnitt 5.2.2.2 gilt, und damit auch für den seriellen MLB-Shaper. Der Ausgang des parallelen Shapers wird durch das Minimum der Ausgänge der einzelnen Shaper beschränkt. Da nach Satz 5.4 der Ausgang jedes einzelnen (σ,ρ)-Shapers durch (5.35) beschränkt ist, ist der Ausgang des parallelen Shapers durch (5.36) beschränkt. ♦

Die Verbesserung der Schranken von Satz 5.5 gegenüber den Schranken aus der Definition des Shapers (5.12) ist in Bild 5.10 anhand eines Beispiels mit einem Ein/Aus-Shaper dargestellt. Aus der vorhergesagten Hülle ist ein Verkehrsdeskriptor mit zwei Leaky Buckets

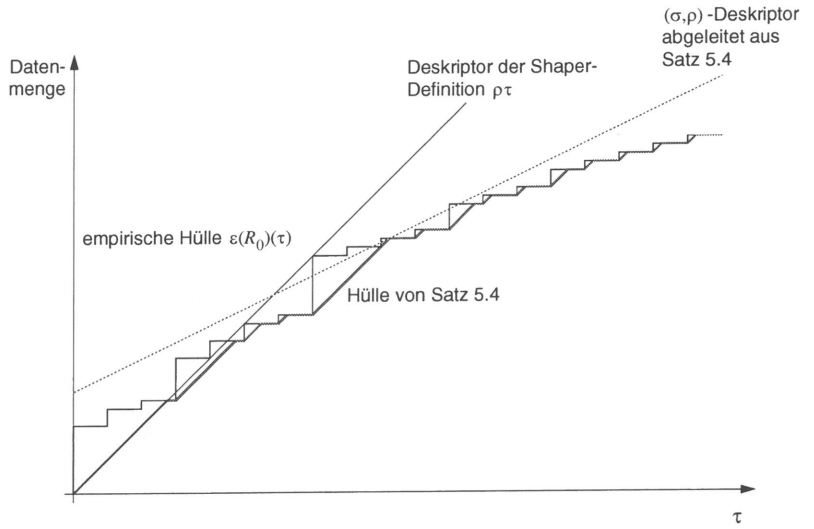


Bild 5.10: Hülle des Ausgangsverkehrsstroms (Ein/Aus-Shaper)

gewonnen worden. Der erste Leaky Bucket fällt mit der Definition des Shapers zusammen, während der zweite Leaky Bucket mit niedrigerer Rate eine Schranke darstellt, die auch die Hülle des Eingangsverkehrs unterschreitet.

5.2.5 MLB-Shaper für gespeicherte Videodaten

Der Mechanismus des MLB-Shapers wird in diesem Abschnitt modifiziert, um ihn für gespeicherte Videodaten geeignet zu machen. Es sei daran erinnert, daß in diesem Fall der Sender keine Schranken und der Empfänger Schranken für ein Unter- und Überlaufen des Pufferspeichers aufstellt. Das Problem besteht darin, einen Verkehrsstrom mit einer Mengenfunktion zu finden, der in die Empfängerschranken paßt, welche um den Wert der Speicherkapazität X_E voneinander entfernt sind. Eine wichtige Beobachtung in diesem Zusammenhang ist, daß im Falle von Echtzeit-kodierten Videodaten die Schranken des Senders genau diese Form aufweisen, wenn $X_S = X_E$ gesetzt wird; nur die Position der Schranken unterscheidet sich (man vergleiche Bild 4.9 mit Bild 4.12). Für letzteren Fall ist ein Verfahren bekannt, einen passenden Verkehrsstrom zu erzeugen: es kann ein MLB-Shaper eingesetzt werden, der nach Abschnitt 5.2.3.2 für eine Speicherkapazität X_E dimensioniert wurde.

Um den MLB-Shaper an den Fall gespeicherter Videodaten anzupassen, wird die Steuerungsfunktion des Shapers von seiner Datenübertragungsfunktion getrennt. Der MLB-Shaper steuert nach wie vor die Rate, mit der kodierte Daten ins Netz gesendet werden, aber der Verkehrsstrom am Eingang ist jetzt virtuell und wird von einem sogenannten *Zeitplan-Dekodierer* aus den kodierten Videodaten erzeugt (siehe Bild 5.11). Die Mengenfunktion des generierten

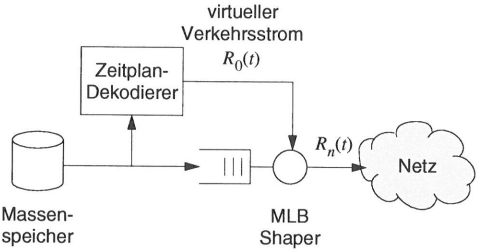


Bild 5.11: MLB-Shaper für gespeicherte Videodaten

virtuellen Verkehrsstroms entspricht dabei dem Ausdruck für die Schranke zur Verhinderung eines Überlaufs des Pufferspeichers ((4.22) mit (4.20))

$$R_0(t) = C(t - D) + X_E s(t) \tag{5.37}$$

($s(t)$ ist die Sprungfunktion). Sie unterscheidet sich vom Fall Echtzeit-kodierter Videodaten, bei dem gilt $R_0(t) = C(t)$, durch einen Initialisierungsschritt $X_E s(t)$ und eine Verschiebung um D Zeiteinheiten. Der Grund für die Verzögerung D wird weiter unten erklärt. Der Zeit-

plan-Dekodierer erhält die Funktion $C(t)$ durch ein teilweises Dekodieren der Videodaten. Die Parameter D und X_E müssen bei der Initialisierung bekannt sein. Ein Puffer entkoppelt das Dekodieren vom Sendeprozeß der Daten.

Ein Beispiel für einen solcherart erzeugten Verkehrsstrom wird in Bild 5.12 gezeigt. Die durchgezogene Linie zeigt die Mengenfunktion für Echtzeit-kodierte Videodaten. Die Mengenfunktion für gespeicherte Videodaten weicht am Anfang und Ende davon ab. Die Abweichung am Ende liegt einfach daran, daß das Ende der Videodaten früher erreicht wird (die Schranken für Überlauf und Unterlauf sind ja vertauscht). Am Anfang muß der Speicher des Empfängers gefüllt werden, was durch den zusätzlichen Ausdruck $X_E s(t)$ in der Mengenfunktion des Eingangsverkehrs erreicht wird. Deshalb ist die Verzögerung nötig, bevor der Empfänger mit dem Dekodieren beginnt.

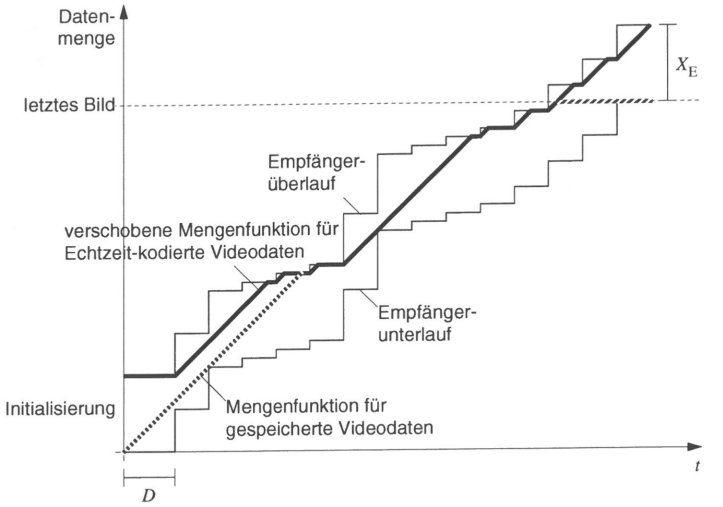


Bild 5.12: MLB-Shaper für gespeicherte Videodaten (Beispiel Ein/Aus-Shaper)

Im folgenden wird angenommen, daß für eine Anwendung eine kleine Hülle, die den durch X_E gegebenen Freiraum vollständig ausnutzt, wichtiger ist, als eine kleine Verzögerung D. Also wird der MLB-Shaper gemäß Abschnitt 5.2.3.2 für einen Eingangsverkehr $R_0(t) = C(t)$ dimensioniert, ohne die Initialisierung zu berücksichtigen, und D wird anschließend angepaßt. Eine konservative untere Schranke für D ist die Zeit, die der Sprung $X_E \sigma(t)$ braucht, um aus dem "Gedächtnis" des MLB-Shapers $W_{\rho_m}(R_0)(t)$ gelöscht zu werden, d.h. $W_{\rho_m}(X_E s(t))(D) = 0$. Diese Schranke ist

$$D \geq \frac{X_E}{\min_{m=1..n} \rho_m} . \tag{5.38}$$

Wenn der MLB-Shaper auf diese Weise dimensioniert wird, ist seine maximale Rate am Ausgang optimal. Diese Eigenschaft erbt er vom Fall Echtzeit-kodierter Videodaten. Das heißt jeder Shaper, der mit einer Empfänger-Speicherkapazität X arbeitet, hat eine maximale Rate, die größer oder gleich der durch (5.30) gegebenen Rate ist.

5.2.6 Implementierungsaspekte

5.2.6.1 Mechanismus des MLB-Shapers

Ein MLB-Shaper ist einfach zu implementieren, weil er aus einfachen Grundbausteinen, den (σ,ρ)-Shapern, aufgebaut ist. In einer ATM-Übertragungseinheit befindet sich der Shaper typischerweise an der Stelle, an der die Pakete einer Verbindung in einzelne Zellen segmentiert werden. Wie in Abschnitt 5.2.2.2 gezeigt wurde, kann ein MLB-Shaper mit mehreren parallel geschalteten Instanzen von (σ,ρ)-Shapern realisiert werden. Eine Instanz benötigt zum Bestimmen von $W_{\rho_m}(R)(t)$ einen Zähler, der von der Verkehrsmenge am Eingang $R(t)$ inkrementiert wird und mit der Rate des Shapers ρ dekrementiert wird, solange er größer null ist. Die Aktualisierung der Zähler kann zeitlich diskret zu jeder Periode des Zelltakts des Übertragungsabschnitts erfolgen. Zu jedem Zelltakt wird entschieden, ob eine Zelle erzeugt werden kann. Die Erzeugung erfolgt dann, wenn *alle* Leaky Buckets die Zelle als konform deklarieren, das bedeutet für den einzelnen Leaky Bucket, daß $W_{\rho_m}(R)(t)$ kleiner gleich σ_m ist.

Der Zähler eines (σ,ρ)-Shapers, das Dekrement und der Vergleichswert sollten annähernd kontinuierliche Werte annehmen können (etwa mit einer Fixkommadarstellung), damit auch Raten realisiert werden können, die keine ganzen Teiler der Zellrate des Übertragungsabschnitts sind. Die Darstellung einer beliebigen reellen Zahl ist jedoch nicht möglich. Eine andere Stelle, bei der die Darstellung beliebiger Raten nicht praktikabel ist, sind die Signalisierungsnachrichten mit denen der Benutzer dem Netz den Verkehrsdeskriptor mitteilt. Es muß also für die Raten eine minimale Granularität definiert werden, die bei der Dimensionierung des Hardware-Shapers berücksichtigt werden muß. Beginnend mit den Bedingungen (5.24) und (5.28) lassen sich leicht Formeln für modifizierte ρ_m und σ_m angeben. Weil mit einer geeigneten Wahl der Granularität die Auswirkungen beliebig klein gehalten werden können, wird dieser Aspekt hier nicht weiter berücksichtigt.

Bevor die Zellen einer Verbindung auf den Übertragungsabschnitt gelangen, werden sie noch mit Zellen anderer Verbindungen gemultiplext, wofür ein Pufferspeicher gebraucht wird, um eventuelle Konflikte bei gleichzeitig auftretenden Segmentierentscheidungen mehrerer Verbindungen aufzulösen. Dieses Multiplexen erzeugt einen Verzögerungsjitter, der bei der anschließenden Quellflußkontrolle am Netzeingang berücksichtigt werden muß.

5.2.6.2 Einsatz bei gespeicherten Videodaten

Bevor die Übertragung von gespeicherten Videodaten beginnen kann, muß der Shaper dimensioniert werden, d.h. seine Parameter σ und ρ müssen bestimmt werden, und für den Verkehrsvertrag muß der Quellverkehrsdeskriptor des Verkehrsstroms am Ausgang des Shapers bestimmt werden.

Bei gespeicherten Videodaten liegt es nahe, soviele Berechnungen wie möglich im voraus vorzunehmen, und die Berechnungen zum Zeitpunkt des Abspielens auf ein Minimum zu reduzieren. Für die Dimensionierung des MLB-Shapers kann die empirische Hülle des Verkehrsstroms im voraus bestimmt und mit den Videodaten abgespeichert werden. Beim Abspielen wird damit der MLB-Shaper für eine gegebene Empfängerkapazität nach Abschnitt 5.2.3.2.2 dimensioniert und eine Hülle für die Reservierung der Netzressourcen nach Satz 5.5 bestimmt. Anstatt der empirischen Hülle kann es aber auch die kleinste konkave Hülle, oder eine weiter vereinfachte Hülle sein. Dies spart Speicherplatz und Rechenzeit, führt aber dazu daß die vorhergesagten Schranken nicht mehr so eng sind und möglicherweise mehr Netzressourcen belegt werden. Hier kann Aufwand gegen Leistung abgewogen werden. Insgesamt ist mit diesem Ansatz der Rechenaufwand zum Initialisieren eines Abspielvorgangs gering und kann auch von weniger leistungsfähigen Prozessorsystemen durchgeführt werden.

5.2.6.3 Einsatz bei Echtzeit-kodierten Videodaten

Bei Echtzeit-kodierten Videodaten ist die kleinstmögliche Schranke, wie sie die empirische Hülle des Verkehrsstroms darstellt, normalerweise nicht bekannt, da ja der zu kodierende Szeneninhalt einen wesentlichen Einfluß auf die empirische Hülle hat. Falls der Szeneninhalt vorhersehbar ist, kann man versuchen, eine statistisch repräsentative Hülle zu bestimmen, die nur mit einer sehr geringen Wahrscheinlichkeit überschritten wird. Wenn der Kodierer diese Hülle überschreitet, müssen die überschüssigen Daten bereits am Sender verworfen werden, ansonsten bildet sich ein Rückstau, der die Daten über die vereinbarte Verzögerung hinaus verzögert, und der Empfänger gerät außer Tritt. Der Sender muß hier also zusätzlich eine Quellflußkontrollfunktion wahrnehmen.

Ein anderer Ansatz setzt den MLB-Shaper als Teil der Ratenregelung eines rückgekoppelten Kodierers ein. Damit verhindert man ein Überschreiten der Hülle von vorneherein. Wenn man den Kodierer mit konstanter Rate aus Abschnitt 2.3.2 betrachtet, so erkennt man, daß seine Ratenregelung aus einem virtuellen Ein/Aus-Shaper mit Rate ρ besteht, dessen Füllstand die Quantisierung steuert. Dieser Ansatz läßt sich unmittelbar auf einen virtuellen MLB-Shaper erweitern (siehe Bild 5.13). Diese Anordnung garantiert für den Ausgangsverkehr die Hülle, für die der MLB-Shaper dimensioniert worden ist. Ein nachgeschalteter realer MLB-Shaper mit den gleichen Parametern läuft also nie über. (Natürlich liegt es nahe, in der Implementierung den virtuellen und realen MLB-Shaper zu integrieren.) Die Hülle kann durch Messungen von für die Anwendung typischem Szenenmaterial bei offener Regelschleife bestimmt werden. Im normalen Einsatz wird diese Hülle nicht überschritten, und die Qualität bleibt annähernd konstant. Andernfalls regelt ein zu hoher Füllstand die Quantisierung herunter, bis die Hülle wieder erreicht wird. Der Vorteil gegenüber der Ratenregelung mit konstanter Rate liegt darin, daß die Hülle besser an die "natürliche" Charakteristik des Szenenmaterials und des Kodierers angepaßt werden kann, und dadurch das Herunterregeln der Quantisierung zu einem sehr seltenen Ereignis wird.

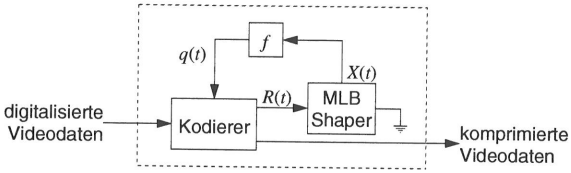


Bild 5.13: MLB-Shaper als Ratenregelung für einen Kodierer mit konstanter Hülle

5.3 Direkte Verfahren

5.3.1 Verfahren des kürzesten Wegs

5.3.1.1 Mechanismus

Das in diesem Abschnitt vorgestellte Verfahren arbeitet direkt mit den Regionen erlaubter Mengenfunktionen aus Abschnitt 4.2.1 und Abschnitt 4.3. Für eine gegebene Region kann das Problem des Traffic Shapings folgendermaßen formuliert werden: Ein Shaper hat eine möglichst glatte Mengenfunktion, geometrisch interpretiert einen Weg, zwischen einem Start- und

Zielpunkt zu finden, der vollständig innerhalb der Region liegt [ReBe95], [SZKT96]. Eine intuitiv plausible Lösung dieses Problems ist der kürzeste Weg zwischen den beiden Punkten (siehe Bild 5.14).

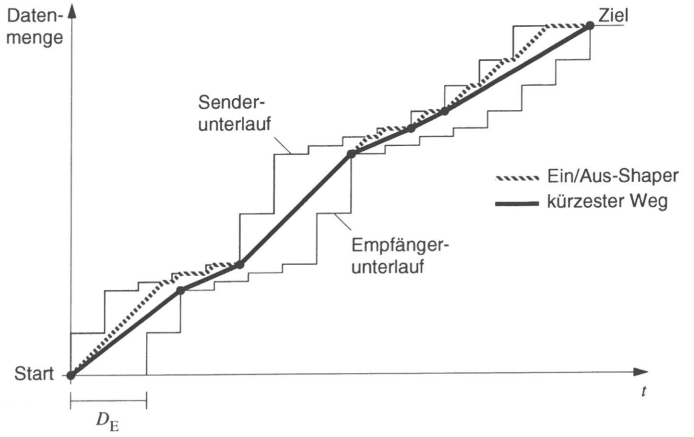


Bild 5.14: Verfahren des kürzesten Wegs im Vergleich mit Ein/Aus-Shaper

Der kürzeste Weg weist bestimmte Optimalitätseigenschaften auf. In [SZKT96] wurde gezeigt, daß er von allen Wegen innerhalb der Region das Minimum einer Halbordnung aus der Majorisierungstheorie darstellt. Daraus konnten Aussagen über den Maximalwert und die Varianz der Rate abgeleitet werden. Der Weg sei zeitlich diskretisiert und bestehe aus einer Folge von Segmenten mit Rate r_i, die alle die gleiche Länge T auf der Zeitachse haben. Die (reflexive) Halbordnung ist folgendermaßen definiert:

Gegeben seien zwei n-dimensionale Vektoren $\vec{u} = (u_1,u_2,\ldots,u_n)$ und $\vec{v} = (v_1,v_2,\ldots,v_n)$. Eine Permutation der Elemente von $\vec{u}$ (bzw. $\vec{v}$), die die Elemente nach monoton fallenden Werten sortiert, sei mit $u_{[1]},u_{[2]},\ldots,u_{[n]}$ (bzw. $v_{[1]},v_{[2]},\ldots,v_{[n]}$) bezeichnet. $\vec{u}$ wird von $\vec{v}$ majorisiert, oder $\vec{u} \leq_M \vec{v}$, genau dann, wenn

$$\text{für } j = 1,\ldots,n-1: \sum_{i=1}^{j} u_{[i]} \leq \sum_{i=1}^{j} v_{[i]}$$

$$\text{und } \sum_{i=1}^{n} u_{[i]} = \sum_{i=1}^{n} v_{[i]} .$$

Aus der Minimierung dieser Halbordnung folgt, daß der kürzeste Weg die kleinste maximale Rate $r_{max} = \max_{i=1..n} r_i$ aller Wege aufweist und ebenfalls die kleinste Varianz

$$var(\hat{r}) = \sum_{i=1}^{n} (r_i - \bar{r})^2 / n \ .$$

Es sei hier die Vermutung aufgestellt, daß der kürzeste Weg auch die kleinste empirische Hülle aller Wege aufweist, jedoch fehlt bis jetzt dafür der Beweis.

Ein Shaper, dessen Mengenfunktion dem kürzesten Weg entspricht, sei im folgenden KW-Shaper (für "kürzester Weg") genannt. Gegenüber dem MLB-Shaper ist der KW-Shaper flexibler, da die Grenzen der Region erlaubter Zeitpläne beliebig festgelegt werden können. Insbesondere kann durch das Netz verursachter Verzögerungsjitter schon im Sender berücksichtigt werden, wie in Abschnitt 4.4.2 diskutiert. Die Dimensionierung des MLB-Shapers nach Verzögerung oder Speicherkapazität setzt implizit voraus, daß die Region mindestens eine Teilregion enthält, die durch ein Paar korrespondierender Grenzen gebildet wird, die identische, horizontal (gegebene Verzögerung) oder vertikal (gegebene Speicherkapazität) verschobene Kopien sind. Für diese Fälle kann der MLB- mit dem KW-Shaper verglichen werden und die maximale Rate des KW-Shapers bestimmt werden: sie ist durch (5.26) und (5.30) gegeben, da diese Werte nach Satz 5.3 minimale Werte für alle Shaper darstellen und die maximale Rate des KW-Shapers minimal ist. Daraus folgt, daß die maximale Rate eines MLB- und KW-Shapers identisch ist. Im Falle, daß Netzressourcen basierend auf der maximalen Rate zugeteilt werden, zeigen also der MLB- und der KW-Shaper identische Leistung.

Das Verfahren des kürzesten Wegs eignet sich in dieser Form nicht für eine Implementierung bei Echtzeit-kodierten Videodaten, da es für die Bestimmung des kürzesten Wegs die Kenntnis der gesamten Mengenfunktion des Videomaterials benötigt. Eine in einem folgenden Abschnitt vorgestellte Variante des Verfahrens arbeitet deshalb mit einer lokalen Schätzung des zukünftigen Verlaufs der Mengenfunktion.

5.3.1.2 Implementierungsaspekte

Verglichen mit dem MLB-Shaper ist der KW-Shaper wesentlich aufwendiger zu implementieren. Jeder Sendevorgang benötigt die Bestimmung des kürzesten Weges, und dies erfordert einen wahlfreien Zugriff auf die gesamte Mengenfunktion des zu kodierenden Videomaterials. Zwar zeigt der Zugriff auf die Mengenfunktion eine gewisse Lokalität, diese ist aber schwierig im vorhinein zu bestimmen (je weiter z.B. die Schranken zeitlich auseinanderliegen desto mehr Kanten müssen geprüft werden um das kürzeste Geradensegment zu bestimmen).

Der kürzeste Weg muß jedesmal aufs neue bestimmt werden, weil jede Änderung eines Parameters (Speicherkapazität oder Verzögerung) einen neuen kürzesten Weg zur Folge hat. Es gibt im Gegensatz zum MLB-Shaper keine Berechnungsschritte, die unabhängig von Parametern im voraus ausgeführt werden können.

Ohne besondere Vorkehrungen haben die Geradensegmente, die den kürzesten Weg ausmachen, beliebige Steigungen, d.h. vom Shaper würde verlangt, mit beliebigen Raten zu arbeiten. Wie schon beim MLB-Shaper erwähnt wurde, muß jedoch eine minimale Granularität der Rate definiert werden. Der Algorithmus muß also modifiziert werden. Er liefert dann eine Annäherung des kürzesten Weges bei der die Steigungen der Geradensegmente die geforderte Granularität aufweisen. Weil die Auswirkungen beliebig klein gehalten werden können wird dies hier nicht weiter betrachtet.

5.3.2 Shaping für Echtzeit-kodierte Videodaten

Bei Echtzeit-kodierten Videodaten erhält der Shaper erst während der Übertragung Kenntnis vom Verlauf der Mengenfunktion des Videomaterials, nämlich in dem Maße wie der Kodierer die kodierten Videodaten an seinem Ausgang zur Verfügung stellt. Die Auswirkung dieses Problems und seine Lösung sind an einem Beispiel in Bild 5.15 dargestellt. Hier seien vereinfachend nur zwei der vier Schranken betrachtet, nämlich Senderunterlauf und Empfängerunterlauf, das Verfahren kann aber die anderen Schranken genauso berücksichtigen. Bekannt ist dem Shaper die Vergangenheit, d.h. der Verlauf der Mengenfunktion bis zum zuletzt kodierten Bild (durchgezogene Linien). Unbekannt ist der zukünftige Verlauf der Mengenfunktion. Weil sie aber, unter anderem durch den Einfluß des GOP-Musters, kurzfristig stark korreliert ist, kann die Mengenfunktion für die unmittelbare Zukunft mit guter Treffsicherheit aus dem vergangenen Verlauf vorhergesagt werden (Prädiktion, gestrichelte Linien). Zwei einfache Heuristiken sind zum Beispiel:

- Die Größe von Bild i wird mit der Größe von Bild $i-n$ vorhergesagt (n sei die Länge des GOP-Musters),
- Die Größe eines Bildes wird mit der Größe des letzten Bildes desselben Typs (I, P oder B) vorhergesagt.

Natürlich bieten sich auch statistische Vorhersageverfahren an, die hier aber nicht weiter untersucht werden..

Der Shaper kann jetzt die vorhergesagte Mengenfunktion als Hypothese seinen Berechnungen zugrundelegen und damit wieder den kürzesten Weg zwischen dem gegenwärtigen Punkt und dem vorhergesagten Zielpunkt bestimmen. Wie das Bild zeigt, kann der Shaper

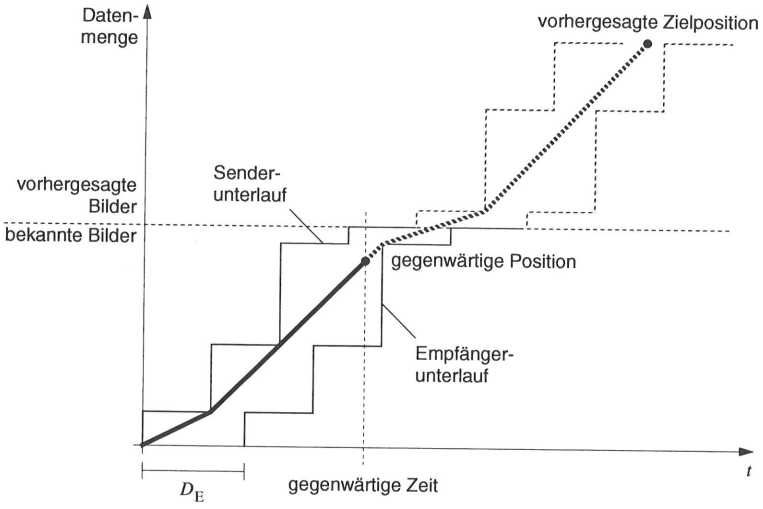


Bild 5.15: Verfahren des kürzesten Wegs für Echtzeit-kodierte Videodaten

einen Überlauf oder Unterlauf sicher verhindern, weil er die Schranken des Senders bis zum gegenwärtigen Bild kennt, und von den Schranken des Empfängers – wegen der mit der Ausspielverzögerung gegebenen Verzögerung – sogar ein Stück weit den zukünftigen Verlauf kennt. Die Größe, die jetzt im Vergleich zum KW-Shaper unsicher geworden ist, ist die Rate des Shapers: Wenn sich die Hypothese des Shapers als falsch herausstellt, hat er nicht den tatsächlich kürzesten Weg gewählt. D.h. er hat zeitweilig eine zu hohe Rate oder eine zu niedrige Rate gewählt. In letzterem Fall muß er diese durch eine höhere Rate ausgleichen. In beiden Fällen hat er also unter Umständen mehr Bursts erzeugt.

Weil es über den Verkehr des Shapers jetzt keine Sicherheit mehr gibt, ist die Reservierung im Netz erschwert. Gut geeignet sind in diesem Fall Netzdienste, die eine Neuverhandlung des Verkehrsdeskriptors während der Verbindungsdauer vorsehen. Zum Beispiel wurde mit Renegotiated Constant Bitrate (RCBR) [GKT97] ein Dienst mit konstanter Bitrate vorgeschlagen, bei dem die Quelle die Rate von Zeit zu Zeit neu bestimmen kann.

Die häufige Ratenänderung kann ein Problem darstellen. Der Shaper wird jedesmal, wenn der Kodierer ein neues Bild zur Verfügung stellt, eine neue Vorhersage aufstellen und einen neuen Plan für die Ratenfunktion berechnen. Weil die Vorhersage die tatsächliche Mengenfunktion fast nie völlig genau trifft, hat dies zur Folge, daß der Shaper seine Rate nach einer Bildperiode, wenn auch meist geringfügig, ändert. Bei der Verwendung von RCBR ist es

jedoch sinnvoll, die Zahl der Neuverhandlungen mit dem Netz zu reduzieren. In diesem Fall kann der Shaper in einer anderen Betriebsart arbeiten, in der die gegenwärtige Rate solange beibehalten wird, bis die Mengenfunktion eine Schranke zu verletzen droht.

6 Leistungsuntersuchung

In Kapitel 5 wurde über analytische Methoden die Leistung verschiedener Verfahren des Traffic Shaping miteinander verglichen und es konnten dabei allgemeingültige Aussagen wie z.B. die Optimalitätseigenschaft des Ein/Aus-Shapers gemacht werden. Diese Aussagen konnten jedoch nur qualitativ sein, nämlich eine bessere, schlechtere oder gleichgute Leistung feststellen, da sie den Einfluß des Videomaterials auf die Leistung der Shaper nicht berücksichtigten. In diesem Kapitel sollen Simulationen mit repräsentativem Videomaterial quantitative Aussagen ermöglichen und Anhaltspunkte für den praktischen Einsatz geben, so z.B. die Abwägung zwischen Implementierungsaufwand und erzieltem Nutzen.

6.1 Simulationen

6.1.1 Methodik

Das im theoretischen Teil verwendete zeit- und wertkontinuierliche Modell für die Verkehrsströme und Netzelemente läßt sich auch gut für eine Simulation im Rechner verwenden. Dieser Ansatz hat z.B. gegenüber einer zeitdiskreten Simulation auf Zellebene den Vorteil einer höheren Geschwindigkeit. Zum Beispiel kann man im kontinuierlichen Modell einen Verkehrsfluß mit konstanter Rate durch ein einziges Geradensegment darstellen, wohingegen im diskreten Modell eine Reihe von Zellankünften nachgebildet werden muß. Durch die Reduktion der Zahl der zu simulierenden Ereignisse sind auch lange Simulationen mit Echtzeitdauern in Kinofilmlänge praktikabel.

Das Ziel der Simulation ist die Mengenfunktion $R(t)$ des Verkehrsstroms am Ausgang eines Shapers zu berechnen und diese dann zu analysieren. Die Mengenfunktion ist bei allen Verfahren abschnittsweise linear. Es genügt daher, die Anfangs- und Endpunkte der Abschnitte zu berechnen. Zur Abbildung kontinuierlicher Zeit- oder Mengenwerte scheint sich zunächst die Fließkommadarstellung anzubieten. Die Zahlendarstellung muß eine hohe Präzision aufweisen: Die Rate kann sich innerhalb von Bruchteilen einer Bildperiode ändern, und gleichzeitig soll der simulierte Zeitraum für eine hohe Aussagekraft bis zu mehreren hunderttausend Bildperioden umfassen. Weil bei einer Fließkommadarstellung die absolute Präzision mit steigendem Absolutwert abnehmen würde, verwendet die Simulation für Zeitvariablen 64 bit-Festkommazahlen, die geeignet skaliert werden. Aus demselben Grund wird diese Darstellung auch für die Datenmenge verwendet.

Die berechnete Mengenfunktion kann dann verschiedenen Analysen zugeführt werden. Für die Bestimmung von Verkehrsdeskriptoren und die Zuteilung von Ressourcen ist die Empirische Hülle relevant. Mit einem relativ einfachen Ansatz kann man sie an äquidistanten Werten von τ abtasten und den Funktionswert direkt aus der Definition der empirischen Hülle bestimmen. Dazu braucht nur ein Fenster der Dauer τ über den abschnittsweise linearen Verkehrsstrom geschoben werden und die maximale Differenz an den Enden bestimmt werden. Um den Fehler durch die Abtastung klein zu halten, muß an sehr vielen Punkten abgetastet werden, was den Aufwand in die Höhe treibt. Deshalb wird die Empirische Hülle in einem anderen Ansatz in einer geschlossenen Form berechnet. Dieser stützt sich auf die Beobachtung, daß die Empirische Hülle ebenfalls abschnittsweise linear ist und nur die Endpunkte der Geradensegmente berechnet werden müssen. Für ein bestimmtes τ werden der Mengenfunktion alle Abschnitte bestehend aus Geradensegmenten entnommen, die länger als τ sind. Die Abschnitte werden mit einem gemeinsamen Ursprung übereinandergelegt, und Schnittpunkte zwischen den Geradensegmenten bestimmt. Dann wird der nächste Endpunkt der empirischen Hülle bestimmt, der entweder ein Schnittpunkt zweier Geradensegmente oder ein Endpunkt eines Geradensegments sein kann. Das ganze Verfahren wird mit dem τ des gefundenen Punkts wiederholt und man erhält sukzessive die Punkte der empirischen Hülle.

6.1.2 Nachgebildete Szenarien

Den Simulationen liegen das Szenario Echtzeit-kodierter Videodaten von Abschnitt 4.3.1 und das Szenario gespeicherter Videodaten von Abschnitt 4.3.2 zugrunde. Die Simulationen sollen ein leichtgewichtiges Protokoll nachbilden, wie z.B. die ATM Forum-Vereinbarung zur Übertragung von MPEG-komprimiertem Video über AAL 5 [ATMF97]. Der Traffic Shaper ist in der SAR-Schicht angesiedelt.

Weil die MPEG-Audioschicht typischerweise mit konstanter Ausgangsrate arbeitet, wurde ihr Verkehrsaufkommen nicht berücksichtigt. Das Verkehrsaufkommen der MPEG-Systemschicht ist schwierig nachzubilden und hängt stark von der jeweiligen Implementierung ab. Typischerweise ist es aber sehr viel kleiner als das der MPEG-Videoschicht und wurde deshalb vernachlässigt. Ebenfalls vernachlässigt wurde der sehr geringe Protokolloverhead des Kapselungsprotokolls von [ATMF97] und von AAL 5.

Bei der Echtzeit-Kodierung stellt der Kodierer zu Beginn jeder Bildperiode ein neues Bild in den Zwischenspeicher des Traffic Shapers. Bei gespeicherten Videodaten fordert der Traffic Shaper die Daten vom Massenspeicher an. Es wird angenommen, daß er bei der Wahl der momentanen Rate keinen Einschränkungen unterliegt.

6.2 Simulationsdaten

6.2.1 Datensätze

Die Simulationen verwenden eine Reihe von Datensätzen, die aus längeren Segmenten komprimierten Videos gewonnen wurden (siehe Tabelle 6.1). Das Videomaterial umfaßt verschiedene Szeneninhalte und Kompressionsverfahren. Bei allen Datensätzen bis auf einen ist das MPEG-1-Verfahren [ISO 11172-2] angewendet worden. Beim Datensatz *(cnn_j)* wurde jedes Bild einzeln mit JPEG [ISO 10918-1] komprimiert, was oft als Motion-JPEG bezeichnet wird. Wegen der Ähnlichkeit des Verfahrens dürfte Motion-JPEG ähnliche Eigenschaften wie eine MPEG-Kodierung mit ausschließlich I-Bildern aufweisen. Die Datensätze enthalten für jedes Bild der Videoschicht die Datenmenge und bei MPEG-kodiertem Material außerdem noch den Bildtyp (I, P oder B).

Datensätze	**Auflösung**	**Kompression, GOP-Muster**	**Dauer**
asterix, atp, dino, lambs, mrbean, mtv, news, race, simpsons, soccer, talk1, talk2, terminator [Rose95]	25 Bilder/s, 384x288 Pixel, Format 4:2:0	MPEG-1, IBBPBBPBBPBB	40000 Bilder
starwars [GaWi94]	24 Bilder/s, 352x240 Pixel, Format 4:2:0	MPEG-1, IBBPBBPBBPBB	174138 Bilder
cnn-j [Eger94]	25 Bilder/s, 640x480 Pixel, Format 4:2:0	Motion-JPEG, I	36664 Bilder

Tabelle 6.1: Eigenschaften des Videomaterials

6.2.2 Empirische Hüllen

Die Empirische Hülle $\varepsilon(\tau)$ des Verkehrs am Shaper-Eingang hat sich in den vorigen Kapiteln als bestimmend für die Leistung aller Shaping-Verfahren erwiesen. Weil sie für jede Intervalldauer τ die maximal geflossene Datenmenge angibt, kann man aus ihr das Ausmaß von Phasen erhöhter Aktivität (Bursts) auf verschiedenen Zeitebenen ablesen. Die Zusammenhänge seien an den drei Beispielen von Bild 6.1 erklärt:

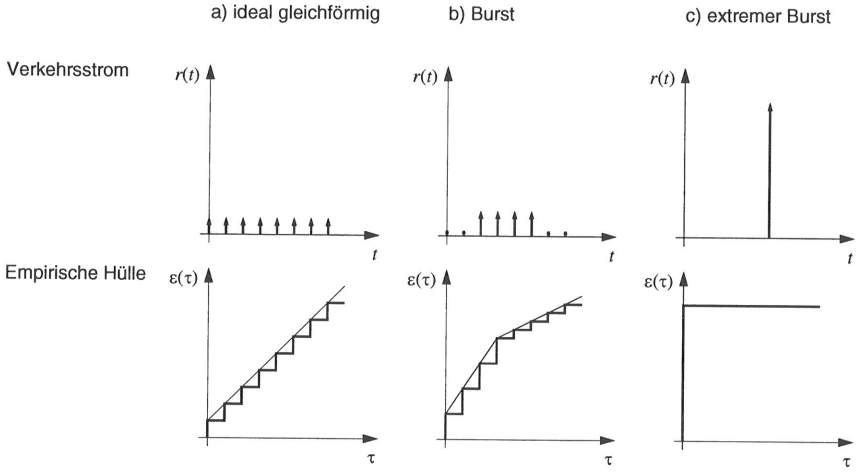


Bild 6.1: Drei exemplarische Videosequenzen mit gleicher mittlerer Rate und unterschiedlicher Büschelförmigkeit

- Eine ideal gleichförmige Videosequenz mit gleichgroßen Bildern hat eine Empirische Hülle, die einer Ursprungsgeraden folgt (Beispiel a)).
- Ein Videosequenz mit einem einzelnen Burst erzeugt in der Empirischen Hülle einen Knick (Beispiel b)). Der Quotient $\varepsilon(\tau)/\tau$ an der Stelle des Knicks entspricht der Rate des Bursts. Der Wert von τ entspricht der Dauer des Bursts. In wirklichen Videosequenzen überlagern sich mehrere Bursts unterschiedlicher Dauer, so daß es unter Umständen keinen ausgeprägten Knick mehr gibt, sondern einen kontinuierlichen Verlauf.
- Der theoretisch ungünstigste Fall einer Videosequenz wäre der, bei dem die ganze Datenmenge in einem einzigen Bild konzentriert wäre (Beispiel c)). Hier weicht die Empirischen Hülle am meisten von einer Ursprungsgeraden ab..

Einige Empirische Hüllen ausgewählter Datensätze sind in Bild 6.2 dargestellt. Der Sprung bei $\tau = 0$ wird durch das größte Bild der Videosequenz verursacht. Bei MPEG-Kodierung bildet sich auf der Zeitebene weniger Bildperioden die GOP-Struktur in einem periodischen Muster ab: auf einen großen Sprung, verursacht durch ein I- oder P-Bild, folgen zwei kleinere Sprünge durch B-Bilder. Manchmal ist die volle GOP-Periode sichtbar, manchmal aber auch

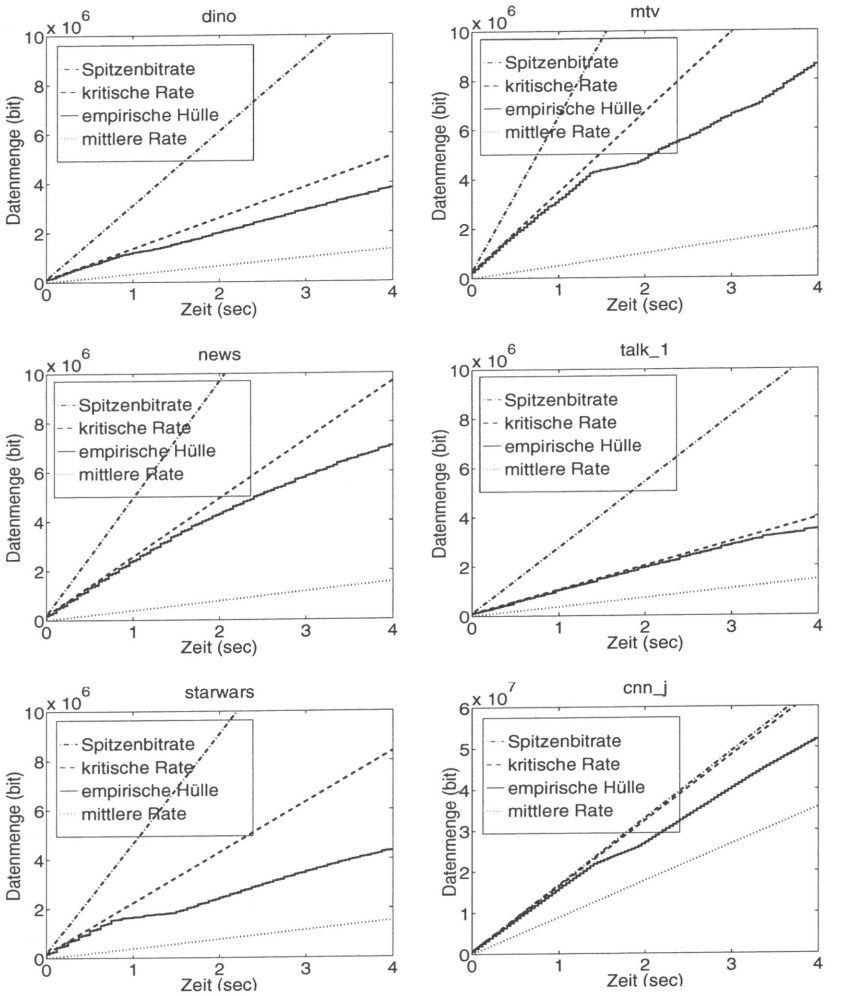


Bild 6.2: Empirische Hüllen ausgewählter Datensätze

nur eine Periode von drei Bildern. Letzteres läßt sich darauf zurückführen, daß die größten P-Bilder annähernd so groß wie die größten I-Bilder werden und dadurch nicht mehr unterscheidbar sind. Das Muster kann auch bei anderen Intervalldauern τ gefunden werden.

Bei einigen Videosequenzen zeigt sich auf der Zeitebene weniger Sekunden ein mehr oder weniger ausgeprägter Knick, also das Anzeichen eines Bursts. In einem späteren Abschnitt soll untersucht werden, welche Zeitebenen für die Leistung bei der Übertragung von Video relevant sind.

6.3 Ein/Aus-Shaper

6.3.1 Leistung

Wie in Abschnitt 5.2.3.3 gezeigt wurde, stellt der Ein/Aus-Shaper Schranken für die Leistung aller Shaper auf. Seine Rate ist die kleinstmögliche Spitzenbitrate aller Shaper, und das sowohl für das Szenario Echtzeit-kodierter und gespeicherter Videodaten. Im folgenden wird die Abwägung zwischen den Shaper-Größen *Rate*, *Verzögerung* und *Speicherkapazität* bei einem idealen Netz (keine Verzögerung) untersucht. Bei Echtzeit-kodierten Videodaten ist normalerweise die Shaper-Verzögerung die kritische Größe, da sie gleich der Verzögerung zwischen Aufnahme und Wiedergabe ist, welche direkt den Gesetzen der menschlichen Wahrnehmung unterliegt. Die Shaper-Speicherkapazität bestimmt die Größe des Pufferspeichers bei Sender und Empfänger, wie in Abschnitt 5.2.3.4 gezeigt. Bei gespeicherten Videodaten ist dagegen die Shaper-Speicherkapazität die kritische Größe, da sie die Größe des Pufferspeichers des Empfängers bestimmt.

Die Einträge im linken Teil von Tabelle 6.2 geben die Rate ρ des Shapers an, normiert auf die mittlere Rate der Videosequenz. Weil der Shaper keinen Einfluß auf die mittlere Rate eines Verkehrsstroms hat, können die Tabelleneinträge gleichzeitig als Burstiness-Faktor f des Verkehrsstroms am Ausgang des Shapers gelesen werden. Die einzelnen Spalten definieren verschiedene Betriebspunkte des Shapers.

Der Betrieb mit einer Verzögerung von 40 ms kann als Bezugspunkt herangezogen werden. Dann entspricht nämlich die Verzögerung der Periodendauer der Bilder (das gilt für alle Sequenzen außer *starwars*) und es liegt der Spezialfall des Shaping mit Spitzenbitrate (siehe Abschnitt 5.1) vor. Der Burstiness-Faktor entspricht hier dem Quotienten des größten Bilds dividiert durch die mittlere Bildgröße, diese Größe wird oft als Burstiness-Faktor des "rohen" Videostroms (vor dem Shaping) bezeichnet.

Die Tabelle 6.2 zeigt, daß bei MPEG-Kodierung der Übergang von 40 ms zu 200 ms Verzögerung die Rate um Werte zwischen 50% und 75% reduziert. Eine weitere Erhöhung der Verzögerung auf eine oder vier Sekunden verringert die Rate nicht annähernd so stark. Wie in Abschnitt 5.2.3.1.2 gezeigt wurde, fällt die Rate für steigende Verzögerung am schnellsten oberhalb der kritischen Rate, nämlich in einem umgekehrt proportionalen Verhältnis. Es zeigt

	Normierte Shaper-Rate für gegebene Verzögerung				gegebene Speicherkapazität	Kritische Rate	
	40 ms	200 ms	1 s	4 s	8 Mbit	normierte Rate	Verzögerung
asterix	6,59	3,41	2,18	1,74	1,47	3,99	66,19 ms
atp	8,72	2,73	2,32	1,75	1,13	3,28	106,2 ms
dino	9,15	3,19	2,45	1,90	1,22	3,80	96,21 ms
lambs	18,36	4,57	3,48	2,86	2,02	4,94	148,6 ms
mrbean	12,98	3,70	2,96	2,35	1,66	3,83	135,5 ms
mtv	12,71	5,50	3,76	3,08	2,96	6,53	77,80 ms
news	12,36	5,35	3,82	2,37	1,35	6,18	80,06 ms
race	6,58	3,25	2,21	1,63	1,40	5,38	48,94 ms
simpsons	12,94	3,07	2,24	1,48	1,29	5,07	102,2 ms
soccer	7,58	3,41	2,66	1,76	1,33	3,84	79,04 ms
talk1	7,35	2,48	2,05	1,78	1,21	2,65	110,9 ms
talk2	7,41	2,85	2,44	1,93	1,15	3,06	96,72 ms
terminator	7,30	2,72	1,71	1,25	1,03	3,32	87,89 ms
starwars	12,37	4,31	2,35	1,69	1,48	5,47	90,50 ms
cnn-j	1,82	1,54	1,21	1,10	1,25	1,79	40,76 ms

Tabelle 6.2: Leistung des Ein/Aus-Shapers

sich, daß bei MPEG-Kodierung die kritische Rate signifikant unter der Spitzenbitrate liegt, nämlich zwischen 39% und 70% darunter. Beim Datensatz mit M-JPEG-Kodierung dagegen liegt die kritische Rate nur 1.6% unter der Spitzenbitrate. Deshalb reduziert sich die Rate beim Übergang von 40 ms zu 200 ms auch nur um 15%. Die GOP-Struktur mit der Inter-Kodierung ist also für eine günstige, d.h. niedrig liegende kritische Rate verantwortlich.

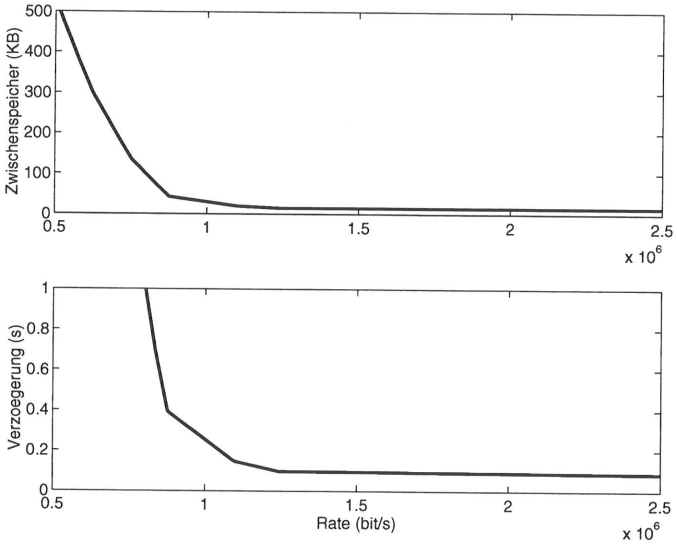


Bild 6.3: Leistungsabwägung beim Ein/Aus-Shaper *(dino)*

Ein kompletter Verlauf der Verzögerung und der Speicherkapazität als Funktion der Rate ist in Bild 6.3 am Beispiel des Datensatzes *dino* gezeigt.

Die Spalte '8 Mbit Speicherkapazität' soll einen Anhaltspunkt für den Einsatz in einem Videoabrufdienst geben, bei dem der Dekodierer einen Pufferspeicher von rund 1 MByte zur Verfügung hat. Bis auf drei Ausnahmen weisen die Datensätze eine sehr kleine Rest-Büschelförmigkeit mit einem Burstiness-Faktor kleiner als 1,5 auf.

6.3.2 Relevante Zeitebenen

Aus der geometrischen Konstruktion der Shaper-Leistungsgrößen durch Anlegen einer Geraden an die Empirische Hülle ergibt sich ein Punkt, an dem die Gerade die emprisiche Hülle berührt (in Sonderfällen können es auch mehrere Punkte sein). Der Wert von τ an diesem Punkt gibt die Dauer des Bursts an, der für die Leistung des Shapers verantwortlich ist. In Tabelle 6.3 sind die Berührungspunkte für die Betriebspunkte von Tabelle 6.2 angegeben.

	τ am Berührungspunkt					
	40 ms Verz.	**200 ms Verz.**	**1 s Verz.**	**4 s Verz.**	**8 Mbit Speicherk.**	**kritische Rate**
asterix	0,000 s	960,0 ms	1,320 s	19,92 s	30,36 s	480,0 ms
atp	0,000 s	360,0 ms	7,680 s	8,880 s	58,08 s	360,0 ms
dino	0,000 s	840,0 ms	6,240 s	14,52 s	229,0 s	280,0 ms
lambs	0,000 s	480,0 ms	6,480 s	15,00 s	57,24 s	480,0 ms
mrbean	0,000 s	2,040 s	3,600 s	12,12 s	13,08 s	1,440 s
mtv	0,000 s	1,360 s	7,080 s	32,40 s	32,40 s	160,0 ms
news	0,000 s	1,440 s	2,400 s	4,920 s	83,88 s	120,0 ms
race	0,000 s	1,440 s	1,600 s	16,64 s	16,64 s	40,00 ms
simpsons	0,000 s	600,0 ms	2,240 s	15,00 s	133,8 s	40,00 ms
soccer	0,000 s	2,160 s	3,360 s	7,680 s	33,60 s	120,0 ms
talk1	0,000 s	2,880 s	6,720 s	25,92 s	26,40 s	480,0 ms
talk2	0,000 s	3,840 s	9,600 s	11,16 s	107,0 s	960,0 ms
terminator	0,000 s	480,0 ms	1,520 s	20,20 s	326,9 s	360,0 ms
starwars	0,000 s	625,0 ms	3,375 s	9,500 s	211,0 s	125,0 ms
cnn-j	0,000 s	1,400 s	18,40 s	206,4 s	4,000 s	200,0 ms

Tabelle 6.3: Lage des Berührungspunkts der Shaper-Geraden mit der Empirischen Hülle

Je größer der Wert von τ ist, desto länger ist der Hebel der Berührungsgeraden und desto stärker steigen Verzögerung und Speicherkapazität mit fallender Rate. Ein hoher Wert von τ bedeutet also, daß die Möglichkeiten des Shaping schon ausgeschöpft sind.

Für alle Raten größer als die kritische Rate ist $\tau = 0$, das ist bei allen Datensätzen bei einer Verzögerung von 40 ms der Fall. Insgesamt zeigen sich die Datensätze sehr uneinheitlich und es ist schwierig, allgemeine Trends auszumachen. Es läßt sich jedoch feststellen, daß die Zeit, die zum Abbau eines Bursts benötigt wird (das ist die Verzögerung) in allen Fällen kleiner ist als die Dauer des Bursts.

6.4 MLB-Shaper

6.4.1 Bewertung der Verkehrshülle

Der Vorteil des allgemeinen MLB-Shapers gegenüber dem Spezialfall Ein/Aus-Shaper liegt in der kleineren garantierten Verkehrshülle bei gleicher Spitzenbitrate. Diese Ausgangshülle soll jetzt genauer untersucht werden. In den meisten Anwendungsszenarien kommt es auf eine im voraus bekannte, vorhergesagte Hülle an und nicht auf die sich tatsächlich ergebende minimale Empirische Hülle. Denn wenn die Reservierung der Netzressourcen auf einem während der Verbindungsdauer festen Verkehrsvertrag basiert (siehe Abschnitt 3.4.2), wird die Hülle zum Zeitpunkt des Verbindungsaufbaus benötigt, um daraus den Verkehrsdeskriptor im Verkehrsvertrag abzuleiten. Im folgdenden werden daher verschiedene Vorhersagen für die Ausgangshülle verglichen.

Die Bewertung eines mehrdimensionalen Verkehrsdeskriptors oder gar einer Verkehrshülle ist schwieriger als die Bewertung einer Spitzenbitrate. In der Praxis kommt es auf die gesamte Zahl der Verbindungen an, die von einem Netz übermittelt werden können. Diese Zahl hängt vom Algorithmus der Verbindungsannahmeentscheidung ab, der nicht normiert ist, und dem spezifischen Mix der geforderten Verkehrsverträge mit ihren Verkehrsdeskriptoren und Dienstgütegarantien. Als Grundlage eines Vergleichs verschiedener Shaping-Verfahren wird deshalb beispielhaft ein einzelner Übertragungsabschnitt herausgegriffen und untersucht, wieviele Verkehrsströme er tragen könnte, wenn alle mit demselben zu untersuchenden Verkehrsdeskriptor beschrieben wären. In einem ersten Schritt sollen die praktischen Beschränkungen eines Verkehrsvertrags vernachlässigt werden und der Verkehrsdeskriptor soll eine allgemeine Verkehrshülle sein.

Im einfachsten Fall eines Übertragungsabschnitts mit der Bedienstrategie *Abfertigung nach Ankunftsreihenfolge* (FCFS) sieht die Verbindungsannahmeentscheidung folgendermaßen aus: Der Übertragungsabschnitt verursacht einen Verzögerungsjitter, der maximal den Wert δ betragen soll. Eine Verbindung mit einem gegebenen Verkehrsdeskriptor (hier: Verkehrshülle $b_i(\tau)$) wird genau dann angenommen, wenn die neue Abschätzung der maximalen Verzögerung kleiner gleich δ ist. Die Abschätzung wird aus der Summe der Verkehrsdeskriptoren der Verbindungen gebildet. Anknüpfend an Abschnitt 4.1.2.4 ergibt sich als Annahmebedingung (mit c wird die Kapazität des Übertragungsabschnitts bezeichnet)

$$\frac{1}{c} \max_{\tau \geq 0} \left[\sum_i b_i(\tau) - c\tau \right] \leq \delta \,. \tag{6.1}$$

Daraus läßt sich die (maximal mögliche) Zahl der Verbindungen bestimmen.

Eine Abschätzung der Hülle erhält man, wenn man die Zahl der Verbindungen als Funktion von δ aufträgt. Bei $\delta = 0$ teilt der Übertragungsabschnitt die Ressourcen nach der Spitzenbitrate zu (Steigung der Hülle bei $\tau = 0$). Bei Werten größer Null spielt dann der Verlauf der Hülle für größere Zeitintervalle τ eine Rolle. In den folgenden Simulationen wird jeweils von einem Übertragungsabschnitt der Kapazität 125 Mbit/s ausgegangen und diese Funktion für einen Wertebereich zwischen 0 und 400 ms graphisch dargestellt.

6.4.2 Einfache Vorhersage

In der Simulation sind MLB-Shaper mit verschiedenen Zahlen m von (σ,ρ)-Shapern für eine gegebene Shaper-Verzögerung dimensioniert worden. Dabei wurden die einzelnen (σ,ρ)-Shaper den Segmenten der kleinsten konkaven Hülle zugeordnet und für steigende Intervalldauern τ dimensioniert bis die Zahl m erreicht war.

Eine einfache Vorhersage der Hülle des MLB-Shapers erhält man aus (5.12) als Minimum der Hüllen der einzelnen (σ,ρ)-Shaper.

Das Ergebnis für den Datensatz *dino* ist in Bild 6.4 dargestellt. Das Bild zeigt, daß zusätzliche (σ,ρ)-Shaper eine starke Leistungsverbesserung bringen, solange bis die kleinste konkave Hülle von den (σ,ρ)-Shapern vollständig abgedeckt wird. Im dargestellten Datensatz ist bei einer Shaper-Verzögerung von 80 ms der Idealfall bei 47 (σ,ρ)-Shapern erreicht (für 200 ms sind es 42). Bei wenigen (σ,ρ)-Shapern bleibt die Zahl der Verbindungen ab einem bestimmten δ konstant, weil die Abschätzung (6.1) das letzte Segment der Hülle erreicht hat.

6.4.3 Vorhersage mit Kenntnis der Eingangshülle und Satz 5.4

Wenn vom Verkehrsstrom am Eingang des Shapers eine Hülle bekannt ist, läßt sich die Vorhersage der Hülle verbessern, indem Satz 5.4 angewendet wird. Die Simulation des vorigen Abschnitts wurde wiederholt und die Hülle diesmal unter Zugrundelegung der empirischen Hülle des Datensatzes vorhergesagt.

Das Ergebnis für den Datensatz *dino* ist in Bild 6.5 dargestellt. Die Leistung ist für kleine Werte von m deutlich verbessert. Der Idealfall $m = 47$ (bzw. $m = 42$) bleibt unverändert. Die Zahl der Verbindungen ist für kleine m jetzt nicht mehr konstant, sondern steigt über den gesamten Wertebereich. Insgesamt kommen die Kurven dem Idealfall näher, es gibt nicht mehr so starke Unterschiede für unterschiedliche m. Schon mit nur 3 (σ,ρ)-Shapern erreicht man eine Leistung, die nur unwesentlich unter dem Idealfall liegt.

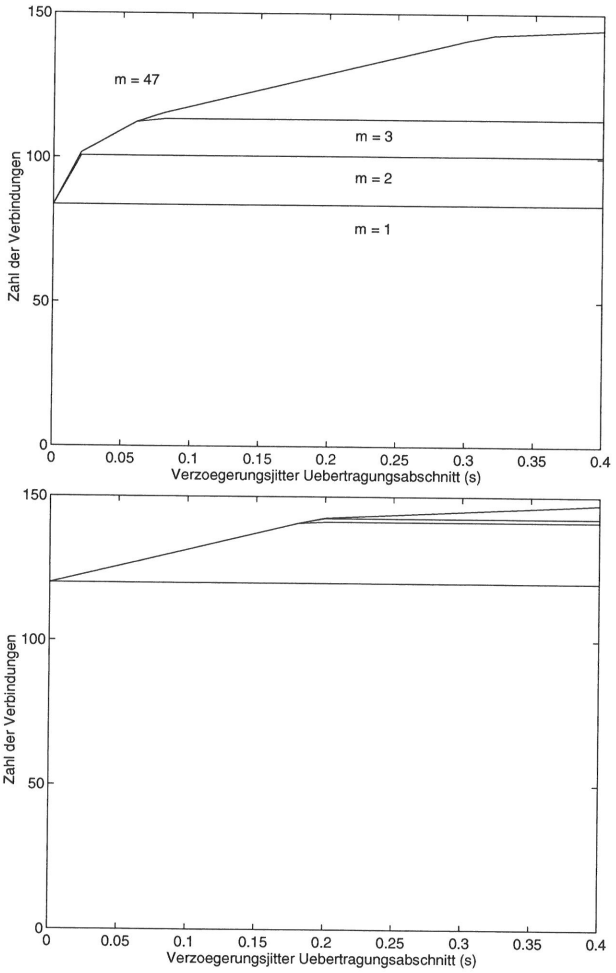


Bild 6.4: MLB-Shaper dimensioniert für Verzögerung 80 ms (oben) und 200 ms (unten), einfache Vorhersage der Hülle

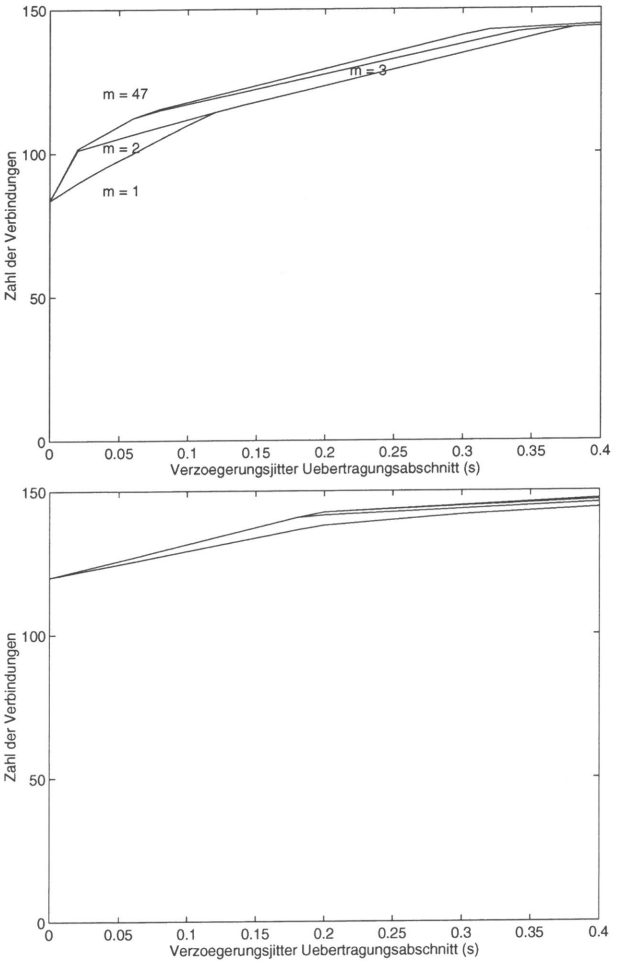


Bild 6.5: MLB-Shaper dimensioniert für Verzögerung 80 ms (oben) und 200 ms (unten), Vorhersage der Hülle mit Satz 5.4

Der Implementierungsaufwand für eine fast-optimale Leistung ist für mehrere Datensätze in Tabelle 6.4 a) zusammengefaßt. Als Leistungsziel wurde definiert, daß der MLB-Shaper über den bekannten Wertebereich von δ (0 bis 400 ms) mindestens 95% der Zahl der Verbindungen eines idealen MLB-Shapers erzielt. Es sind tatsächlich bei allen Datensätzen nur einige wenige (σ,ρ) -Shaper nötig.

	Zahl *m* der (σ,ρ) -Shaper (Idealer Deskriptor, Referenz: idealer MLB-Shaper)		**Zahl *l* der (σ,ρ) -Deskriptoren (Ein/Aus-Shaper, Referenz: idealer Deskriptor)**	
	80 ms Verz.	**200 ms Verz.**	**80 ms Verz.**	**200 ms Verz.**
asterix	1	1	5	2
atp	2	1	4	3
dino	3	1	6	2
lambs	2	3	7	2
mrbean	1	1	7	2
mtv	1	1	5	2
news	1	1	8	2
race	2	1	4	2
simpsons	3	1	6	2
soccer	1	1	6	3
talk1	2	1	8	2
talk2	2	1	8	2
terminator	2	2	4	2
cnn-j	1	1	2	5

Tabelle 6.4: Aufwand um 95% Leistung des Idealfalls zu erzielen:
a) Shaper (links), b) Verkehrsdeskriptor (rechts)

Zusammenfassend kann man feststellen, daß eine verbesserte Vorhersage durch Kenntnis einer Hülle des Videomaterials den Implementierungsaufwand signifikant vermindern hilft.

6.4.4 Einfluß des Verkehrsdeskriptors

Bis jetzt wurde in den Simulationen des MLB-Shapers ein Idealfall betrachtet, weil der Verkehrsdeskriptor gleich der vorhergesagten Hülle gesetzt wurde, was eine bestmögliche Zuteilung von Ressourcen ermöglicht. In der Praxis ist der Verkehrsdeskriptor jedoch normalerweise parametrisiert, z.B. wie die Quellverkehrsdeskriptoren in Abschnitt 3.4.7.1, um die Menge der zu übertragenden Information zu reduzieren. Deswegen wurde in weiteren Simulationen der Umfang des Verkehrsdeskriptors auf die ersten l Segmente der vorhergesagten Hülle aus dem vorigen Abschnitt beschränkt.

Die Auswirkung auf die Ressourcenzuteilung beim Ein/Aus-Shaper (80 ms und 200 ms Verzögerung) ist für den Datensatz *dino* in Bild 6.6 dargestellt. Auch hier wirkt sich eine Verkehrshülle, die in der Zahl ihrer Segmente reduziert ist, dadurch aus, daß die Zahl von Verbindungen ab einem bestimmten Wert von δ konstant bleibt. Die Leistungseinbuße für kleine l ist signifikant.

Die Anforderung an die Güte des Verkehrsdeskriptors für eine fast-optimale Leistung ist für mehrere Datensätze in Tabelle 6.4 zusammengefaßt. Ähnlich wie im vorigen Abschnitt wurde als Leistungsziel definiert, daß die Ressourcenzuteilung über den bekannten Wertebereich von δ (0 bis 400 ms) mindestens 95% der Zahl der Verbindungen eines idealen Verkehrsdeskriptors erzielt. Bei den MPEG-kodierten Datensätzen benötigt die Ausgangshülle bei kleiner Verzögerung deutlich mehr (σ,ρ)-Deskriptoren als die Definition der Standards zur Zeit erlaubt (nämlich höchstens zwei). Für größere Verzögerung reduziert sich die Zahl auf einige wenige (σ,ρ)-Deskriptoren.

Die Ergebnisse dieses Abschnitts über den MLB-Shaper zeigen, daß es für das untersuchte Videomaterial nur wenige (σ,ρ)-Shaper braucht, d.h. wenig Implementierungsaufwand, um dem idealen MLB-Shaper sehr nahe zu kommen. Die vollständige Beschreibung der Hülle für die Ressourcenzuteilung ist jedoch kritisch für die Leistung. Die zusätzliche Information, die aus Satz 5.4 gewonnen werden kann, kann durch eingeschränkte Verkehrsdeskriptoren wieder verloren gehen und die Leistungsverbesserung zunichte gemacht werden.

6.5 KW-Shaper

Weil er den kürzesten Weg berechnet, benötigt der KW-Shaper die Kenntnis der Bildgrößen des gesamten Videomaterials. Er eignet sich deshalb nur für das Szenario gespeicherter Videodaten.

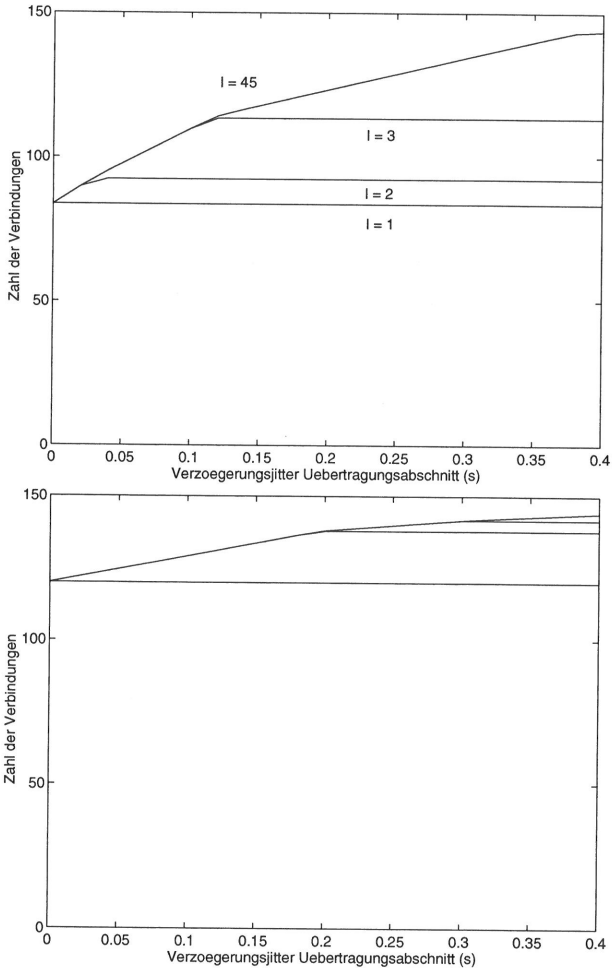


Bild 6.6: Auswirkung eines reduzierten Verkehrsdeskriptors, Verzögerung 80 ms (oben) und 200 ms (unten)

Der KW-Shaper arbeitet mit Regionen erlaubter Mengenfunktionen die durch vier Schranken gebildet werden, so daß die Speicherkapazität des Senders, die Speicherkapazität des Empfängers und die Ausspielverzögerung des Empfängers unabhängig voneinander eingestellt werden können. Damit wird es möglich, den Verzögerungsjitter des Netzes schon im voraus zu berücksichtigen.

Im folgenden bildet das Szenario gespeicherter Videodaten den Rahmen eines Vergleiches zwischen KW- und MLB-Shaper. Dabei sollen verschiedene Speicherkapazitäten des Empfängers betrachtet werden. Wie in den vorigen Abschnitten wird die Leistung anhand der Zahl der Verbindungen eines Übertragungsabschnitts bewertet. Zusätzlich wird berücksichtigt, daß der Übertragungsabschnitt einen Verzögerungsjitter verursacht. Beim MLB-Shaper muß dieser nachträglich im Empfänger ausgeglichen werden gemäß dem Vorgehen in Abschnitt 4.4.3. Der dafür nötige Pufferspeicher geht vom Gesamtspeicher ab und steht dem Shaping nicht mehr zur Verfügung. Der KW-Shaper dagegen kann den Verzögerungsjitter bereits bei der Berechnung seiner Mengenfunktion berücksichtigen gemäß Abschnitt 4.4.2.

Damit die Shaper unter gleichen Rahmenbedingungen arbeiten, wurden die Shaping-Parameter folgendermaßen bestimmt. (1) Festlegung der Speicherkapazität des Empfängers. Diese geht direkt in eine Schranke der Mengenfunktion des KW-Shapers ein. (2) Festlegung des Verzögerungjitters des Übertragungsabschnitts. Dieser geht direkt in eine Schranke des KW-Shapers ein. (3) Bestimmung der Speicherkapazität, die für den MLB-Shaper zur Verfügung steht. Diese wird durch ein iteratives Verfahren bestimmt. Die Summe aus diesem und dem Speicher zum Ausgleich des Verzögerungsjitters muß am Ende gleich der Speicherkapazität des Empfängers sein. Dabei wird der Speicher zum Ausgleich des Verzögerungsjitters mit einer Abschätzung der Hülle des Shaper-Ausgangsverkehrs nach Satz 5.4 bestimmt. In den Simulationen wurde ein idealer Verkehrsdeskriptor angenommen.

In der Simulation wurde die Speicherkapazität des Empfängers zunächst so gesetzt, daß sie zum Ausgleich eines Verzögerungsjitters von 400 ms ausreichend ist (beim Datensatz *starwars* sind das 126 KB). Die Ergebnisse für den Datensatz *starwars* sind in Bild 6.7 dargestellt. Wie erwartet zeigt der KW-Shaper die beste Leistung über den gesamten Wertebereich von δ. Der MLB-Shaper zeigt annähernd gleiche Leistung bei kleinen Werten des Verzögerungsjitters des Übertragungsabschnitts. Die Leistung läßt für steigende Werte von δ nach. Das liegt daran, daß ein immer größerer Teil des Speichers zur Entfernung des Jitters gebraucht wird, und weniger Speicher für das Shaping verbleibt. Mit einer zusätzlichen Speicherkapazität von 256 KB gibt es jedoch für einen einzelnen (σ,ρ)-Shaper fast keinen Leistungsabfall über den gesamten Wertebereich. Im Fall von nur zwei (σ,ρ)-Shapern ist der

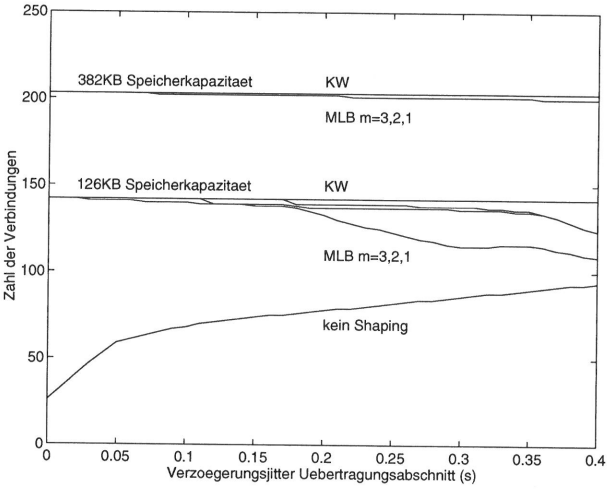


Bild 6.7: Vergleich KW- und MLB-Shaper *(starwars)*

MLB-Shaper vom KW-Shaper nicht mehr unterscheidbar. Bei allen Datensätzen beträgt die maximale Differenz zwischen MLB- und KW-Shaper bis auf eine Ausnahme eine Verbindung.

7 Zusammenfassung und Ausblick

Die vorliegende Arbeit untersuchte die Übertragung zeitabhängiger Medien in einem dienstintegrierenden Netz. Betrachtet wurden paketvermittelnde Netze basierend auf dem Asynchronen Transfermodus, deren Mechanismen zur Verkehrssteuerung vorgestellt wurden. Auf das Konzept der verbindungsindividuellen Ressourcenreservierung wurde näher eingegangen. Dieser Ansatz hat den Vorteil, daß er den Anwendungen eine bestimmte Dienstgüte garantieren kann, die beim Verbindungsaufbau ausgehandelt wird. Er erfordert gleichzeitig eine genaue Charakterisierung des Verkehrs durch die Anwendung.

Das hier betrachtete Medium Video weist Eigenschaften auf, die es exemplarisch für andere Medien machen. Die Kompressionsverfahren für Video, die redundante Information aus den abgetasteten Rohdaten entfernen, erzeugen eine variable Datenmenge für die kodierten Bilder. Gleichzeitig stellen verschiedene Anwendungsszenarien Echtzeit-Anforderungen an die Übertragung der Daten. Aus diesen Faktoren ergibt sich eine Verkehrscharakteristik, die schwierig zu handhaben ist: Der Verkehr ist büschelförmig auf mehreren Zeitebenen, und die Zeitebenen erstrecken sich von der Ebene einzelner Bilder im Bereich zweistelliger Millisekunden bis zur Ebene mehrerer Minuten.

Ein Traffic Shaper ist ein Netzelement, das die Charakteristik eines Verkehrsstroms verbessert. Eine Ratensteuerung reguliert die abgehende Verkehrsmenge und ein Pufferspeicher fängt ankommende Verkehrsspitzen auf. Die verbesserte Charakteristik wird immer erkauft mit einer zusätzlichen Verzögerung und dem Aufwand für die Bereitstellung eines genügend großen Pufferspeichers. Näher untersucht wurden Traffic Shaper, die auf dem Weg des Verkehrsstroms möglichst früh zum Einsatz kommen, nämlich beim Sender. Dieser möchte einen günstigen Verkehrsdeskriptor erzielen, der ihm beim Abschluß des Verkehrsvertrags mit dem Netz zugute kommt. Traffic Shaper unterscheiden sich im Algorithmus, der für die Ratensteuerung zum Einsatz kommt. Um sie in ihrer Leistung vergleichen zu können, wurde ein Modell für den Übertragungsvorgang und seine beteiligten Elemente entwickelt, bei dem die Verkehrsströme mit zeit- und wertkontinuierlichen Mengenfunktionen dargestellt werden (Kapitel 4). Mit Hilfe des Modells konnte die Wahl einer geeigneten Rate der Ratensteuerung als Optimierung eines Weges durch eine Region erlaubter Mengenfunktionen in der Daten-Zeit-Ebene dargestellt werden. Das Problem der stochastischen Verzögerungen im Netz wurde systematisch behandelt und zwei Lösungsansätze, die Behandlung vor oder nach dem Shaping, wurden vorgestellt. Ausserdem wurde für die Charakterisierung von Verkehrsströmen die Methode der *Verkehrshüllen* verwendet. Eine Verkehrshülle ist eine deterministische und zeitunabhängige obere Schranke für die Datenmenge und eine Verallgemeinerung anderer bekannter Verkehrsdeskriptoren, wie der Leaky Bucket oder das Moving Window. Die Ver-

kehrshülle hat den großen Vorteil, daß sie die Büschelförmigkeit eines Verkehrsstroms auf allen Zeitebenen beschreiben kann. Bei der verbindungsindividuellen Reservierung von Ressourcen ist die Kenntnis einer Verkehrshülle für jede Verbindung Voraussetzung dafür, daß die Vermittlungsstellen deterministische Dienstgüteparameter vorhersagen können.

Für die Konstruktion eines Traffic Shapers wurden zwei sehr unterschiedliche Ansätze verfolgt (Kapitel 5): der *Mehrfach-Leaky Bucket* (MLB)-Shaper, den man erhält durch Hintereinanderschalten mehrerer einfacher (σ, ρ) -Shaper, die auf dem Leaky Bucket-Mechanismus beruhen, und die direkten Verfahren, die die Rate als Weg in der Region erlaubter Mengenfunktionen bestimmen. Zu letzteren gehört der *Kürzeste Weg* (KW)-Shaper, der bezüglich der Glätte des Verkehrsstroms den optimalen Shaper darstellt. Es wurde gezeigt, wie das Prinzip des MLB-Shapers, der unmittelbar nur für Echtzeit-kodierte Daten einzusetzen ist, durch eine Uminterpretation des Shaper-Mechanismus auch bei gespeicherten Videodaten angewendet werden kann. Für den MLB-Shaper wurden zu den Formeln für die Verzögerung von Cruz duale Formeln für die Speicherkapazität hergeleitet. Daraus konnten allgemeine qualitative Aussagen über die Leistung der Shaper abgeleitet werden. So wurde festgestellt, daß schon der einfachste Ein/Aus-Shaper (ein einstufiger MLB-Shaper mit $\sigma = 0$) die günstigste Verkehrscharakteristik bezüglich der Spitzenbitrate erzeugt, d.h. andere Shaper nur im weiteren Verlauf der Verkehrshülle für $\tau > 0$ Verbesserungen erwirken können. Auch konnte gezeigt werden, daß die Abwägung zwischen einer Senkung der maximalen Rate und der sich daraus ergebenden Erhöhung der Speicherkapazität und Verzögerung für hohe Raten am günstigsten ist und bei sinkender Rate immer ungünstiger wird; eine einfache graphische Konstruktion an der empirischen Hülle des Eingangsverkehrs vermittelt diesen Zusammenhang. Bei der Annahme von stufenförmigen Mengenfunktionen bei Erzeugung und Verbrauch der kodierten Daten konnte die Existenz einer kritischen Rate nachgewiesen werden, oberhalb derer die Speicherkapazität konstant bleibt und die Verzögerung umgekehrt proportional zur maximalen Rate mit dem kleinsten Proportionalitätsfaktor steigt. Schließlich wurde eine verbesserte Vorhersage für eine Verkehrshülle am Ausgang eines MLB-Shapers entwickelt, die bei Kenntnis einer Verkehrshülle für den Eingang eine sehr viel engere Schranke aufstellt als die Definition des Shapers.

Die Leistung der Shaper im realen Einsatz wurde mit Simulationen mit kodierten Videodaten untersucht. Die Datensätze variieren sowohl den Inhalt des Szenenmaterials als auch die Kodierung, genauer den Einsatz von räumlicher und zeitlicher Redundanzminderung (ein Datensatz mit reiner Intra-Kodierung mit M-JPEG und vierzehn Datensätze mit abwechselnder Intra-, uni- und bidirektionale Inter-Kodierung mit MPEG). Simulationen mit dem einfachsten Shaper, dem Ein/Aus-Shaper, zeigen die Unterschiede. Bei den MPEG-kodierten Daten kann die maximale Rate ohne große Nachteile sehr weit gesenkt werden (die kritische

Rate ist niedrig), beim M-JPEG-kodierten Datensatz ist das nicht der Fall. Der Grund dafür ist, daß bei MPEG-Kodierung die großen Bilder im Regelfall von kleinen Bildern gefolgt werden, so daß für kleine τ die empirische Hülle nur langsam ansteigt.

Die größte Wirkung zeigt das Traffic Shaping beim Szenario gespeicherter Videodaten, weil hier keine strengen Bedingungen an die Verzögerung gestellt werden und die Speicherkapazität des Shapers großzügig bemessen werden kann. Im Fall von rund 8 MByte Speicherkapazität verbleibt bei 12 von 15 Datensätzen eine geringe Rest-Büschelförmigkeit (maximale Rate geteilt durch mittlere Rate) kleiner als 1,5; dies gilt auch für den M-JPEG-Datensatz. Das bedeutet, daß auch stark büschelförmiges Material mittels Traffic Shaping effizient übertragen werden kann, selbst wenn das Netz nur eine Zuteilung der Spitzenbitrate (engl. *peak rate allocation*) unterstützt.

Wenn die Ressourcenreservierung des Netzes auch andere Teile der Verkehrshülle berücksichtigt (mittels der Verkehrsdeskriptoren), kommt es darauf an, welche Verkehrshülle man für eine Kombination eines bestimmten Datensatzes und eines bestimmten Shapers vorhersagen kann. Beim MLB-Shaper kommt es außerdem darauf an, wieviele Stufen gebraucht werden, um eine gute Verkehrshülle zu erzeugen. Das Ergebnis der Simulationen ist, daß bei allen Datensätzen ein dreistufiger MLB-Shaper genügt, um 95% Leistung des idealen MLB-Shapers zu erreichen (gemessen in der Zahl der Verbindungen, die von einem Übertragungsabschnitt angenommen werden können). Voraussetzung für eine effiziente Nutzung des Netzes ist außerdem, daß der Verkehrsdeskriptor die Verkehrshülle genügend genau beschreiben kann. Wenn der Verkehrsdeskriptor, wie in den Standards vorgesehen, auf zwei Stufen begrenzt ist, ist die Formung des hinteren Verlaufs der Verkehrshülle (für größere τ) vergebens und die Effizienz erleidet Einbußen. Die Simulationen zeigen, daß schwach geformte Videoströme (80 ms Verzögerung) durch zwei Stufen schlecht beschrieben werden, dagegen stärker geformte Videoströme (200 ms Verzögerung) bei MPEG-Kodierung in 11 von 13 Datensätzen mit zwei Stufen 95% Leistung des idealen Verkehrsdeskriptors erreichen. Dieses Ergebnis leuchtet ein, da ein stark formender Traffic Shaper die Bursts der kürzeren Zeitebenen vollständig entfernt und deshalb die Verkehrshülle zu Beginn (kleine τ) lange linear verläuft.

Der KW-Shaper hat gegenüber dem MLB-Shaper zwei Vorteile: er erzeugt den glattesten Ausgangsverkehr und er kann Verzögerungsschwankungen im Netz schon beim Shaping berücksichtigen, anstatt erst nachträglich, was der Leistung zusätzlich zugute kommen könnte. Er scheint aber sehr viel aufwendiger in der Implementierung. Ein Simulationsszenario, in dem signifikante Verzögerungsschwankungen eingearbeitet wurden, verglich beide Shaper und variierte die Zahl der Stufen des MLB-Shapers. Im Ergebnis liegen die Shaper in

der Leistung nahe beieinander, und zwar je näher je stärker der Verkehr geformt wird (großer Zwischenspeicher). Bei einer Speicherkapazität von 256 KB sind ein MLB-Shaper mit zwei Stufen und ein KW-Shaper für praktische Belange nicht mehr unterscheidbar.

Die künftige Entwicklung von Abrufdiensten mit Multimedia-Inhalten wird den Bedarf nach Traffic Shaping erhöhen. Der Wunsch, dem Benutzer mehr Möglichkeiten zu geben, in den Ablauf einer Präsentation einzugreifen, die Notwendigkeit heterogene Empfänger mit unterschiedlichen Fähigkeiten zu unterstützen, und der Wunsch der Anwendungsentwickler nach mehr Flexibilität beim Entwickeln von Präsentationen führt dazu, daß die einzelnen Medienelemente, die eine Präsentation ausmachen, separat verarbeitet, gespeichert und übertragen werden. Das Zusammensetzen der Präsentation geschieht erst zum spätmöglichsten Zeitpunkt, nämlich beim Empfänger. Die Trennung der Medienelemente kann sehr weit gehen, z.B. könnten die Untertitel eines Films in einem eigenen Elementarstrom enthalten sein. Solch ein Ansatz wird zum Beispiel von der Arbeitsgruppe für die MPEG-4-Spezifikationen verfolgt [ISO 14496-1]. Dabei kann ein Elementarstrom gleichzeitig in mehreren Versionen vorliegen, z.B. in unterschiedlichen Sprachen, oder in unterschiedlichen Kodierungs-Qualitäten und -Datenmengen. Dies bedeutet, daß die genaue Zusammensetzung einer Präsentation aus Elementarströmen erst zum Zeitpunkt des Abspielens bekannt ist. Es ist daher nicht sinnvoll, zum Zeitpunkt der Kodierung die Elementarströme auf eine vorgegebene Verkehrscharakteristik hin zu kodieren unter Einbußen in der Qualität. Stattdessen kann der Sender beim Abspielen einen Traffic Shaper einsetzen, der dem Gesamtverkehr der Elementarströme die gewünschte Verkehrscharakteristik aufprägt. Die Mengenfunktion, die der Dimensionierung des Shapers zugrundeliegt, ist jetzt von der Zusammensetzung der Elementarströme abhängig. Eine weitere Änderung der Mengenfunktion ergibt sich, wenn der Anwender den zeitlichen Ablauf der Präsentation manipulieren kann, z.B. durch schnellen Vor- und Rücklauf oder wahlfreien Zugriff auf einen beliebigen Zeitpunkt.

8 Literaturverzeichnis

[AMS82] D. Anick, D. Mitra, M.M. Sondhi, "Stochastic theory of a data-handling system with multiple sources", *Bell Systems Technical Journal*, Vol. 61, 1982

[ATMF94] The ATM Forum, *ATM User-Network Interface Specification Version 3.1*, Prentice Hall, New Jersey, September 1994

[ATMF96a] The ATM Forum, *Private Network-Network Interface Specification Version 1.0*, März 1996

[ATMF96b] The ATM Forum, *Traffic Management Specification Version 4.0*, April 1996

[ATMF96c] The ATM Forum, *User-Network Interface (UNI) Signalling Specification Version 4.0*, Juli 1996

[ATMF97] The ATM Forum, *Audio/Visual Multimedia Services: Video on Demand Version 1.1*, März 1997

[ATMF99] The ATM Forum, *Traffic Management Specification Version 4.1*, März 1999

[BCS93] E. Biagioni, E. Cooper, R. Sansom, "Designing a Practical ATM LAN", *IEEE Network Magazine*, März 1993

[BGLS93] F. Bernabei, L. Gratta, M. Listanti, A. Sarghini, "Analysis of On-Off Source Shaping for ATM Multiplexing", *Proc. IEEE INFOCOM'93*, 1993

[BGRS94] B. Bensaou, B. Guibert, J. Roberts, A. Simonian, "Performance of an ATM multiplexer in the fluid approximation using the Benes approach", *Annals of Operations Research*, Vol. 49, pp.137-160, 1994

[BRS95] F. Brichet, J. W. Roberts, A. Simonian, "Performance of an ATM multiplexer fed by leaky bucet controlled sources", *Proc. IFIP Workshop on ATM Traffic Management*, Paris, Dezember 1995

[BSG92] P. Boyer, M. Servel, F. Guillemin, "The Spacer-Controller: An Efficient UPC/NPC for ATM Networks", *Proc. International Switching Symposium'92*, Yokohama, Japan, November 1992

[Chan94] C.-S. Chang, "Stability, Queue Length, and Delay of Deterministic and Stochastic Queuing Networks", *IEEE Transactions on Automatic Control*, Vol. 39, No. 5, Mai 1994

[Chao92] H. J. Chao, "Architecture Design for Regulating and Scheduling User's Traffic in ATM Networks", *Proc. ACM SIGCOMM'92*, August 1992

[Cruz91a] R. L. Cruz, "A Calculus for Network Delay, Part I: Network Elements in Isolation", *IEEE Transactions on Information Theory*, Vol. 37, No. 1, Januar 1991

[Cruz91b] R. L. Cruz, "A Calculus for Network Delay, Part II: Network Analysis", *IEEE Transactions on Information Theory*, Vol. 37, No. 1, Januar 1991

[Cruz95] R. L. Cruz, "Quality of Service Guarantees in Virtual Circuit Switched Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 13, 1995

[DaTo95] I. Dalgic, F. A. Tobagi, "Constant Quality Video Encoding", *Proc. IEEE ICC'95*, Juni 1995

[Eger94] M. Egerter, "Untersuchung und Bewertung von Videodaten unter Verwendung unterschiedlicher Kompressionsverfahren", Diplomarbeit am Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, Juni 1994

[GaWi94] M. Garrett, W. Willinger, "Analysis, modelling and generation of self-similar VBR video traffic", *Proc. ACM SIGCOMM*, London, England, UK, August 1994

[GBR92] F. Guillemin, P. Boyer, L. Romoeuf, "Spacer-Controller: Architecture and First Assessments", *Proc. IFIP Workshop on Broadband Communications*, Estoril, Portugal, Januar 1992

[GGPS95] L. Georgiadis, R. Guérin, V. Peris, K.N. Sivarajan, "Efficient Network QoS Provisioning Based on per Node Traffic Shaping", *IEEE/ACM Transactions on Networking*, Vol. 4, No. 4, August 1996

[GKT97] M. Grossglauser, S. Keshav , D. Tse, "RCBR: A Simple and Efficient Service for Multiple Time-Scale Traffic", *IEEE/ACM Transactions on Networking*, Vol. 5, No. 6, Dezember 1997

[Gole91] S. J. Golestani, "Congestion-free communication in high-speed packet networks", *IEEE Transactions on Communications*, Vol. 39, No. 2, Dezember 1991

[GuMo92] F. Guillemin, W. Monin, "Management of Cell Delay Variation in ATM Networks", *Proc. IEEE GLOBECOM'92*, Dezember 1992

[HHS94] R. Händel, M. N. Huber, S. Schröder, *ATM Networks – Concepts, Protocols, Applications*, Addison-Wesley, Reading, 1994

[IEEE 1180] IEEE Standard Specifications for the Implementations of 8 by 8 Inverse Discrete Cosine Transform, IEEE Std 1180-1990, 6. Dezember 1990

[ISO 10918-1] ISO/IEC 10918-1:1994 / ITU-T Recommendation T.81 (JPEG), "Information Technology – Digital Compression and Coding of Continuous-Tone Still Images: Requirements and Guidelines"

[ISO 11172-1] ISO/IEC 11172-1:1993 (MPEG-1), "Information Technology – Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s – Part 1: Systems"

[ISO 11172-2] ISO/IEC 11172-2:1993 (MPEG-1), "Information Technology – Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s – Part 2: Video"

[ISO 11172-3] ISO/IEC 11172-3:1993 (MPEG-1), "Information Technology – Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s – Part 3: Audio"

[ISO 13818-1] ISO/IEC 13818-1 / ITU-T Recommendation H.222.0 (MPEG-2), "Information Technology – Generic Coding of Moving Pictures and Associated Audio – Part 1: Systems"

[ISO 13818-2] ISO/IEC 13818-2 / ITU-T Recommendation H.262 (MPEG-2), "Information Technology – Generic Coding of Moving Pictures and Associated Audio – Part 2: Video"

[ISO 13818-3] ISO/IEC 13818-3:1995 (MPEG-2), "Information Technology – Generic Coding of Moving Pictures and Associated Audio – Part 3: Audio"

[ISO 14496-1] ISO/IEC 14496-1:1999 (MPEG-4), "Information Technology – Coding of audio-visual objects – Part 1: Systems"

[ITU-T G.114] ITU-T Recommendation G.114, "Transmission Systems And Media, General Characteristics of International Telephone Connections And International Telephone Circuits – One-Way Transmission Time", Genf, Februar 1996

[ITU-T G.783] ITU-T Recommendation G.783, "Characteristics of synchronous digital hierarchy (SDH) equipment functional blocks", Genf, April 1997

[ITU-T H.261] ITU-T Recommendation H.261, "Video Codec for Audiovisual Services at p x 64 kbit/s", Genf, März 1993

[ITU-T I.121] ITU-T Recommendation I.121, "Broadband Aspects of ISDN", Genf, April 1991

[ITU-T I.150] ITU-T Recommendation I.150, "B-ISDN Asynchronous Transfer Mode Functional Characteristics", Genf, Februar 1999

[ITU-T I.233] ITU-T Recommendation I.233, "Frame Mode Bearer Services", Genf, Oktober 1991

[ITU-T I.311] ITU-T Recommendation I.311, "B-ISDN General Network Aspects", Genf, August 1996

[ITU-T I.321] ITU-T Recommendation I.321, "B-ISDN Protocol Reference Model and its Applications", Genf, April 1991

[ITU-T I.356] ITU-T Recommendation I.356, "B-ISDN ATM Layer Cell Transfer Performance", Genf, März 1993

[ITU-T I.361] ITU-T Recommendation I.361, "B-ISDN ATM Layer Specification", Genf, Februar 1999

[ITU-T I.363] ITU-T Recommendation I.363, "B-ISDN ATM Adaptation Layer (AAL) Specification", Genf, März 1993

[ITU-T I.364] ITU-T Recommendation I.364, "Support of the Broadband Connectionless Data Service by the B-ISDN", Genf, Februar 1999

[ITU-T I.365.1] ITU-T Recommendation I.365.1, "B-ISDN ATM Adaptation Layer Sublayers: Frame Relaying Service Specific Convergence Sublayer (FR-SSCS)", Genf, November 1993

[ITU-T I.371] ITU-T Recommendation I.371, "Traffic Control and Congestion Control in B-ISDN", Genf, März 1993

[ITU-T I.413] ITU-T Recommendation I.413, "B-ISDN User-Network Interface", Genf, März 1993

[ITU-T I.432.1] ITU-T Recommendation I.432.1, "B-ISDN User-Network Interface – Physical Layer Specification: General characteristics", Genf, Februar 1999

[ITU-T Q.2110] ITU-T Recommendation Q.2110, "B-ISDN ATM Adaptation Layer – Service Specific Connection Oriented Protocol (SSCOP)", Genf, Juli 1994

[ITU-T Q.2931] ITU-T Recommendation Q.2931, "Broadband Integrated Services Digital Network (B-ISDN) – Digital Subscriber Signalling System No. 2 (DSS 2) – User-Network Interface (UNI) Layer 3 Specification for Basic Call/ Connection Control", Genf, Februar 1995

[ITU-T X.25] ITU-T Recommendation X.25, "Interface Between Data Terminal Equipment (DTE) and Data Circuit-Terminating Equipment (DCE) for Terminals Operating in the Packet Mode and Connected to Public Data Networks by Dedicated Circuit", Genf, Oktober 1996

[JSS93] K. Jeffay, D.L. Stone, F.D. Smith, "Transport and Display Mechanisms For Multimedia Conferencing Across Packet-Switched Networks", *Computer Networks and ISDN Systems*, Vol. 26, No. 10, Juli 1994

[Kost84] L. Kosten, "Stochastic Theory of Data Handling Systems with Groups of Multiple Sources", in *Performance of Computer-Communication Systems*, Hrsg. H. Rudin und W. Bux, Elsevier, 1984

[Krön96] H. Kröner, "Verkehrssteuerung in ATM-Netzen: Verfahren und verkehrstheoretische Analysen zur Zellpriorisierung und Verbindungssannahme", 62. Bericht über verkehrstheoretische Arbeiten (Dissertationsschrift) Universität Stuttgart, Institut für Nachrichtenvermittlung und Datenverarbeitung, 1995

[Kuro93] J. Kurose, "Open Issues and Challenges in Providing Quality of Service Guarantees in High-Speed Networks", *ACM Computer Communication Review*, 1993

[KuWu96] F.-J. Kuo, J.-S. Wu, "Design and Evaluation of GCRA Shaper and Enforcer Architecture in ATM Networks", *Intl. Journal of Communication Systems*, Vol. 9, 1996

[KWLZ95] E. Knightly, D. Wrege, J. Liebeherr, H. Zhang, "Fundamental Limits and Tradeoffs of Providing Deterministic Guarantees to VBR Video Traffic", *Proc. ACM SIGMETRICS'95*, Ottawa, Mai 1995

[LCY94] S. S. Lam, S. Chow, D. K. Y. Yau, "An Algorithm for Lossless Smoothing of MPEG Video", *Proc. ACM SIGCOMM'94*, August 1994

[LeGa91] D. LeGall, "MPEG-A Video Compression Standard for Multimedia Applications", Communications of the ACM, Vol. 34, No. 4, April 1991

[LMT93] I. M. Leslie, D. R. McAuley, D. L. Tennenhouse, "ATM Everywhere?", *IEEE Network*, März 1993

[LoVa91] S. Low, P. Varaiya, "A Simple Theory of Traffic and Resource Allocation in ATM", *Proc. IEEE GLOBECOM'91*, p. 1633-1637, 1991

[LPT93] J.-Y. Le Boudec, E. Port, H.-L. Truong, "Flight of the Falcon", *IEEE Communications Magazine*, Februar 1993

[NaKu92] R. Nagarajan, J. Kurose, "On Defining, Computing, and Guaranteeing Quality-of-Service in High-Speed Networks", *Proc. IEEE INFOCOM'92*, 1992

[Nies93] G. Niestegge, "The Impact of Traffic Shaping on Broadband Network Performance", *Archiv für Elektronik und Übertragungstechnik*, Vol. 47, No. 5/6, 1993

[PeMi93] W. B. Pennebaker, J. L. Mitchell, *JPEG Still Image Data Compression Standard, Van Nostrand Reinhold*, New York, 1993

[Rath91] E. P. Rathgeb, "Verkehrsflüsse in ATM-Netzen – Modellierung und Analyse von Verkehrsquellen und Quellflußkontrollverfahren", 51. Bericht über verkehrstheoretische Arbeiten (Dissertationsschrift) Universität Stuttgart, Institut für Nachrichtenvermittlung und Datenverarbeitung, 1991

[Rath93] E. P. Rathgeb, "Policing of Realistic VBR Video Traffic in an ATM Network", *International Journal of Digital and Analog Communication Systems*, Vol. 6, 1993

[RBC93] J. W. Roberts, B. Bensaou, Y. Canetti, "A traffic control framework for high speed data transmission", *IFIP Transactions C-15: Modelling and Performance Evaluation of ATM Technology*, Hrsg. H. Perros, G. Pujolle, Y. Takahashi, North Holland-Elsevier Science Publishers, Januar 1993

[ReBe92] A. R. Reibman, A. W. Berger, "On VBR Video Teleconferencing Over ATM Networks", *Proc. IEEE GLOBECOM'92*, Dezember 1992

[ReBe95] A. R. Reibman, A. W. Berger, "Traffic Descriptors for VBR Video Teleconferencing Over ATM Networks", *IEEE/ACM Transactions on Networking*, Vol. 3, No. 3, Juni 1995

[ReHa92] A. R. Reibman, B. G. Haskell, "Constraints on Variable Bit-Rate Video for ATM Networks", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 2, No. 4, p. 361-372, Dezember 1992

[Ritt95] M. Ritter, "Performance analysis of the dual cell spacer in ATM systems", *Proc. IFIP 6th International Conference on High Performance Networking*, September 1995

[RMV96] *Broadband Network Teletraffic – Performance Evaluation and Design of Broadband Multiservice Networks – Final Report of Action COST 242*, Hrsg. J. Roberts, U. Mocci, J. Virtamo, Springer-Verlag, Heidelberg, 1996

[Rose95] O. Rose, "Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems", *Universität Würzburg, Institut für Informatik, Forschungsbericht Nr. 101*, Februar 1995

[RFC 791] J. Postel, "Internet protocol, protocol specification", RFC 791, September 1981

[RFC 1661] W. Simpson, "The Point-to-Point Protocol (PPP)", RFC 1661, Juli 1994

[RFC 1819] L. Delgrossi, L. Berger, "Internet Stream Protocol Version 2 (ST2) Protocol Specification - Version ST2+", RFC 1819, August 1995

[RFC 1883] S. Deering, R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 1883, Januar 1996

[RKTS94] R. Ramjee, J. Kurose, D. Towsley, H. Schulzrinne, "Adaptive playout mechanisms for packetized audio applications in wide-area networks", Proc. IEEE INFOCOM, Toronto, Canada, June 1994

[Robe94] J. W. Roberts, "Virtual Spacing for Flexible Traffic Control", *International Journal of Communications Systems*, Vol. 7, 1994

[RoFl94] A. Romanow, S. Floyd, "Dynamics of TCP Traffic over ATM Networks", *Proc. ACM SIGCOMM'94*, London, England UK, August 1994

[RoGu92] J. Roberts, F. Guillemin, "Jitter in ATM networks and its impact on peak rate enforcement", *Performance Evaluation Journal*, Special Issue on Modelling of High Speed Telecommunications Systems, September 1992

[RoVi91] J. W. Roberts, J. T. Virtamo, "The Superposition of Periodic Cell Arrival Streams in an ATM Multiplexer", *IEEE Transactions on Communications*, Vol. 39, No. 2, Februar 1991

[RVF91] G. Rigolio, L. Verri, L. Fratta, "Source Control and Shaping in ATM Networks", *Proc. IEEE GLOBECOM'91*, Dezember 1991

[Stei93] R. Steinmetz, *Multimedia-Technologie – Einführung und Grundlagen*, Springer-Verlag, Heidelberg, 1993

[SZKT96] J. Salehi, Zhi-Li Zhang, J. Kurose, D. Towsley, "Supporting Stored Video: Reducing Rate Variability and End-to-End Resource Requirements through Optimal Smoothing", *Proc. ACM SIGMETRICS 1996*, Philadelphia, PA, Mai 1996; sowie *University of Massachusetts Department of Computer Science Technical Report 95-98*, November 1995

[Turn86] J. S. Turner, "New directions in communications (or which way to the information age?)", *IEEE Communications Magazine*, Vol. 24, No. 10, Oktober 1986

[WaWo92] E. Wallmeier, T. Worster, "The Spacing Policer, an Algorithm for Efficient Peak Bit Rate Control in ATM Networks", *Proc. International Switching Symposium '92*, Yokohama, Japan, Oktober 1992

9 Anhänge

9.1 Beweis von (5.2)

Aus der Erhaltungsgleichung bei Verlustfreiheit (4.2) und der Kausalitätsbedingung (4.4) ergibt sich folgende untere Schranke für $X(t)$:

$$\begin{aligned} R_1(s,t) &= R_1(t) - R_1(s) \\ &= R_0(t) - X(t) - R_1(s) \\ &\geq R_0(s,t) - X(t) \\ &\Leftrightarrow X(t) \geq R_0(s,t) - R_1(s,t) \end{aligned} \tag{9.1}$$

Aus der Definition von $W_\rho(R)(t)$ (4.5) folgt, daß ein $s < t$ existiert, so daß

$$W_\rho(R_0)(t) = R_0(s,t) - \rho(t-s) .$$

Eingesetzt in (9.1) erhält man

$$\begin{aligned} X(t) &\geq W_\rho(R_0)(t) - [R_1(s,t) - \rho(t-s)] \\ &\geq W_\rho(R_0)(t) - \sigma , \end{aligned}$$

wobei die letzte Ungleichung aus der Definition (5.1) folgt. Weil $X(t) \geq 0$ (Kausalität), folgt daraus (5.2). ♦

9.2 Beweis, daß (5.3) hinreichend ist für (5.1)

Statt (5.1) wird im folgenden gezeigt, daß aus (5.3) eine noch stärkere Bedingung folgt, die hinreichend ist für (5.1), nämlich

$$W_\rho(R_1)(t) = \max\{W_\rho(R_0)(t), \sigma\} . \tag{9.2}$$

Die Erhaltungsgleichung (4.2) eingesetzt in die Definition von $W_\rho(R)(t)$ (4.5) ergibt für $R_1(t)$

$$\begin{aligned} W_\rho(R_1)(t) &= \max_{0 \leq s \leq t}[R_0(s,t) - X(t) + X(s) - \rho(t-s)] \\ &= \max_{0 \leq s \leq t}[R_0(s,t) - \rho(t-s) + X(s)] - X(t) . \end{aligned} \tag{9.3}$$

Mit $M(t)$ werde der Ausdruck links des Minuszeichens bezeichnet

$$M(t) = \max_{0 \leq s \leq t}[R_0(s,t) - \rho(t-s) + X(s)] . \tag{9.4}$$

Weiter unten wird gezeigt werden, daß $M(t) = W_\rho(R_0)(t)$. Mit diesem Ergebnis kann (9.3) weiter entwickelt werden

$$\begin{aligned} W_\rho(R_1)(t) &= W_\rho(R_0)(t) - X(t) \\ &= W_\rho(R_0)(t) - (W_\rho(R_0)(t) - \sigma)^+ \\ &= \max\{W_\rho(R_0)(t), \sigma\} . \end{aligned}$$

Damit ist (9.2) gezeigt. Jetzt wird wie angekündigt $M(t)$ untersucht. Zunächst wird eine untere Schranke bestimmt, die später benötigt wird. Aus der Kausalitätsbedingung $X(s) \geq 0$ folgt

$$M(t) \geq W_\rho(R_0)(t) . \tag{9.5}$$

Aus (9.4) folgt, daß ein s mit $0 \leq s \leq t$ existiert, so daß

$$M(t) = R_0(s,t) - \rho(t-s) + X(s) .$$

Fall (A): $X(s) = 0$. Daraus folgt

$$\begin{aligned} M(t) &= R_0(s,t) - \rho(t-s) \\ &= W_\rho(R_0)(t) . \end{aligned}$$

Fall (B): $X(s) > 0$. Daraus folgt

$$M(t) = R_0(s,t) - \rho(t-s) + W_\rho(R_0)(s) - \sigma .$$

Aus der Definition von $W_\rho(R)(t)$ (4.5) folgt, daß ein u mit $0 \leq u \leq s$ existiert, so daß $W_\rho(R_0)(s) = R_0(u,s) - \rho(s-u)$. Daraus folgt

$$\begin{aligned} M(t) &= R_0(s,t) + R_0(u,s) - \rho(t-s) - \rho(s-u) - \sigma \\ &= R_0(u,t) - \rho(t-u) - \sigma \\ &\leq W_\rho(R_0)(t) - \sigma . \end{aligned}$$

Fall (B.1): $\sigma = 0$. Mit (9.5) folgt

$$M(t) = W_\rho(R_0)(t) .$$

Fall (B.2): $\sigma > 0$. Mit (9.5) ergibt sich ein Widerspruch, deshalb kann dieser Fall nicht vorkommen. ◆

9.3 Beweis von (5.5)

Gemäß der Definition der momentanen Verzögerung (4.3) wird das kleinste d gesucht, für das gilt

$$\begin{aligned} & R_1(t+d) \geq R_0(t) \\ & \Leftrightarrow X(t+d) \leq R_0(t,t+d) \end{aligned} \tag{9.6}$$

wobei (9.6) aus der Erhaltungsgleichung (4.2) folgt.

Fall (A): $X(t) = 0$. Offensichtlich ist das kleinste $d = 0$ und $d(t) = \frac{1}{\rho}X(t)$ ist bestätigt.

Fall (B): $X(t) > 0$. Mit der Gleichung für den Füllstand (5.3) ergibt sich aus der Fallannahme $W_\rho(R_0)(t) > \sigma$. Außerdem sei angenommen, daß gilt

$$\forall t' \in [t,t+d]: W_\rho(R_0)(t') \geq \sigma . \tag{9.7}$$

Die Gültigkeit dieser Annahme wird weiter unten gezeigt werden. Aus (9.6) und (5.3) erhält man damit

$$\begin{aligned} & W_\rho(R_0)(t+d) - \sigma \leq R_0(t,t+d) \\ & \Leftrightarrow \max_{0 \leq s \leq t+d}[R_0(s,t+d) - \rho(t+d-s)] - \sigma \leq R_0(t,t+d) \\ & \Leftrightarrow \max_{0 \leq s \leq t+d}[R_0(s,t) - \rho(t-s)] - \sigma \leq \rho d \\ & \Leftrightarrow \max\{\max_{0 \leq s \leq t}[R_0(s,t) - \rho(t-s)], \\ & \qquad \max_{t \leq s \leq t+d}[R_0(s,t) - \rho(t-s)]\} - \sigma \leq \rho d \\ & \Leftrightarrow W_\rho(R_0)(t) - \sigma \leq \rho d \wedge M(t,d) - \sigma \leq \rho d \end{aligned} \tag{9.8}$$

mit $M(t,d) = \max_{t \leq s \leq t+d}[R_0(s,t) - \rho(t-s)]$. Da $R_0(s,t) \leq 0$ für $s \geq t$, ergibt sich $M(t,d) \leq \rho d$, daher ist die zweite Ungleichung immer erfüllt. Aus der ersten Ungleichung ergibt sich die Bedingung

$$d \geq \frac{1}{\rho}(W_\rho(R_0)(t) - \sigma) \geq \frac{1}{\rho}(W_\rho(R_0)(t) - \sigma)^+ = \frac{1}{\rho}X(t) \quad . \tag{9.9}$$

Das kleinste d, das (9.9) erfüllt, ist

$$d(t) = \frac{1}{\rho}X(t) .$$

Damit ist (5.5) gezeigt.

Es bleibt zu zeigen, daß (9.7) gilt. Mit der Definition von $W_\rho(R)(t)$ (4.5) ergibt sich aus (9.7)

$$\begin{aligned} W_\rho(R_0)(t+a) &= \max_{0\le s\le t+d}[R_0(s,t+a)-\rho(t+a-s)] \\ &= \max_{0\le s\le t+d}[R_0(s,t)-\rho(t-s)]+R_0(t,t+a)-\rho a \\ &\ge \max_{0\le s\le t}[R_0(s,t)-\rho(t-s)]-\rho a \\ &= W_\rho(R_0)(t)-\rho a\,. \end{aligned} \tag{9.10}$$

Aus (9.10) folgt, daß gilt $W_\rho(R_0)(t+a)\ge\sigma$, solange

$$a\le\frac{1}{\rho}(W_\rho(R_0)(t)-\sigma)\ .$$

Dies ist wegen (9.9) der Fall. Damit ist (9.7) gezeigt. ♦

9.4 Beweis von Lemma 5.1

Die Definition von $W_\rho(R)(t)$ (4.5) angewandt auf $\bar\rho$ und $R_1(t)$ ergibt mit der Erhaltungsgleichung (4.2)

$$\begin{aligned} W_{\bar\rho}(R_1)(t) &= \max_{0\le s\le t}[R_0(s,t)-X(t)+X(s)-\bar\rho(t-s)] \\ &= \max_{0\le s\le t}[R_0(s,t)-\bar\rho(t-s)+X(s)]-X(t)\ . \end{aligned} \tag{9.11}$$

Mit $M(t)$ werde der Ausdruck in eckigen Klammern bezeichnet

$$M(t) = \max_{0\le s\le t}[R_0(s,t)-\bar\rho(t-s)+X(s)]\ . \tag{9.12}$$

Für $M(t)$ werden zwei untere Schranken bestimmt, die später benötigt werden: Aus dem Fall $s = t$ folgt

$$M(t)\ge X(t)\,. \tag{9.13}$$

Da $X(s)\ge 0$ (Kausalitätsbedingung) folgt außerdem

$$M(t)\ge W_{\bar\rho}(R_0)(t)\,. \tag{9.14}$$

Aus (9.12) folgt, daß ein s mit $0\le s\le t$ existiert, so daß

$$M(t) = R_0(s,t)-\bar\rho(t-s)+X(s)\ . \tag{9.15}$$

Fall (A): $X(s) = 0$. Daraus folgt

$$M(t) = R_0(s,t) - \bar{\rho}(t-s) = W_{\bar{\rho}}(R_0)(t) \,, \tag{9.16}$$

und daher wird aus (9.11)

$$W_{\bar{\rho}}(R_1)(t) = W_{\bar{\rho}}(R_0)(t) - X(t) \,. \tag{9.17}$$

Aus (9.16) und (9.13) folgt $W_{\bar{\rho}}(R_0)(t) \geq X(t)$, deshalb ist (9.17) äquivalent zu (5.8).

Fall (B): $X(s) > 0$. In diesem Fall ist nach (5.3)

$$X(s) = W_{\rho}(R_0)(s) - \sigma = \max_{0 \leq u \leq s}[R_0(u,s) - \rho(s-u)] - \sigma \,.$$

Daraus folgt, daß ein u mit $0 \leq u \leq s$ existiert, so daß

$$X(s) = R_0(u,s) - \rho(s-u) - \sigma \,.$$

Also wird aus (9.15)

$$M(t) = R_0(s,t) - \bar{\rho}(t-s) + R_0(u,s) - \rho(s-u) - \sigma$$

$$= R_0(u,t) - \bar{\rho}(t-s) - \rho(s-u) - \sigma \,.$$

Im folgenden wird $M(t)$ von oben beschränkt.

Fall (B.1): $\rho \geq \bar{\rho}$. In diesem Fall

$$M(t) \leq R_0(u,t) - \bar{\rho}(t-u) - \sigma \leq W_{\bar{\rho}}(R_0)(t) - \sigma \,.$$

Fall (B.1.a): $\sigma = 0$. Mit (9.14) folgt

$$M(t) = W_{\bar{\rho}}(R_0)(t) \,, \tag{9.18}$$

und daher wird aus (9.11)

$$W_{\bar{\rho}}(R_1)(t) = W_{\bar{\rho}}(R_0)(t) - X(t) \,. \tag{9.19}$$

Aus (9.18) und (9.13) folgt $W_{\bar{\rho}}(R_0)(t) \geq X(t)$, deshalb ist (9.19) äquivalent zu (5.8).

Fall (B.1.b): $\sigma > 0$. Mit (9.14) ergibt sich ein Widerspruch, daher kann dieser Fall nicht vorkommen.

Fall (B.2): $\rho < \bar{\rho}$. In diesem Fall

$$M(t) \leq R_0(u,t) - \rho(t-u) - \sigma \leq W_{\rho}(R_0)(t) - \sigma = X(t) \,.$$

Mit (9.13) folgt

$$M(t) = X(t) \,, \tag{9.20}$$

und daher wird aus (9.11)

$$W_{\bar{\rho}}(R_1)(t) = 0. \tag{9.21}$$

Aus (9.20) und (9.14) folgt $W_{\bar{\rho}}(R_0)(t) \le X(t)$, deshalb ist (9.21) äquivalent zu (5.8). ♦

9.5 Beweis von Satz 5.1

Durch m-maliges Anwenden von (5.8) erhält man

$$W_{\bar{\rho}}(R_m)(t) = (W_{\bar{\rho}}(R_0)(t) - X_m(t))^+. \tag{9.22}$$

Gleichung (5.9) soll durch vollständige Induktion gezeigt werden. Die Gültigkeit für den Fall $k = 1$ wird durch (5.3) bestätigt. Als nächstes wird gezeigt, daß wenn (5.9) für k gilt, es auch für $k+1$ gilt. Damit gilt (5.9) für alle k. Aus (5.3), angewandt auf den k-ten Shaper, erhält man

$$\begin{aligned} X_{k+1}(t) - X_k(t) &= (W_{\rho_{k+1}}(R_k)(t) - \sigma_{k+1})^+ \\ &= [(W_{\rho_{k+1}}(R_0)(t) - X_k(t))^+ - \sigma_{k+1}]^+ , \end{aligned} \tag{9.23}$$

wobei man (9.23) durch Anwendung von (9.22) erhält.

Fall (A): $X_{k+1}(t) - X_k(t) = 0$. Es folgt

$$(W_{\rho_{k+1}}(R_0)(t) - X_k(t))^+ \le \sigma_{k+1} .$$

Fall (A.1): $W_{\rho_{k+1}}(R_0)(t) - X_k(t) < 0$. Dann $X_k(t) > W_{\rho_{k+1}}(R_0)(t)$.
Fall (A.2): $W_{\rho_{k+1}}(R_0)(t) - X_k(t) \ge 0$. Dann

$$X_k(t) \ge W_{\rho_{k+1}}(R_0)(t) - \sigma_{k+1} .$$

Durch Kombination der Fälle (A.1) und (A.2) und mit $X_k(t) \ge 0$ (Kausalitätsbedingung) erhält man

$$X_k(t) \ge (W_{\rho_{k+1}}(R_0)(t) - \sigma_{k+1})^+ .$$

Fall (B): $X_{k+1}(t) - X_k(t) > 0$. Es ergibt sich

$$X_{k+1}(t) = W_{\rho_{k+1}}(R_0)(t) - \sigma_{k+1} .$$

Aus der Fallannahme folgt $X_{k+1}(t) > X_k(t) \ge 0$, also

$$X_{k+1}(t) = (W_{\rho_{k+1}}(R_0)(t) - \sigma_{k+1})^+ .$$

Durch Kombination der Fälle (A) und (B) erhält man

$$X_{k+1}(t) = \max\{X_k(t), (W_{\rho_{k+1}}(R_0)(t) - \sigma_{k+1})^+\} ,$$

und damit die Gleichung (5.9). ♦

9.6 Beweis von Satz 5.4

Für den Beweis von Satz 5.4 wird zunächst eine Reihe von Lemmata für einfache Shaper und einfache Hüllen am Ausgang abgeleitet. Das folgende Lemma hilft, den Ausgangsverkehr eines Ein-Aus Shapers nach oben zu begrenzen.

Lemma 9.1: ((σ,ρ) Shaper mit $\sigma = 0$) Für jedes $t_0 \geq 0$ und $\tau \geq 0$ gibt es ein $t_1 \leq t_0$, so daß

$$X(t_1) = 0 \text{ und } R_1(t_1,t_1 + \Delta t) \geq R_1(t_0,t_0 + \tau) .$$

Mit anderen Worten, wenn für die am Ausgang während eines Zeitintervalls der Länge τ geflossene Datenmenge eine obere Schranke gesucht ist, kann immer ein leerer Speicher zu Beginn des Intervalls angenommen werden.

Beweis: Wenn $X(t_0) = 0$, gilt das Lemma trivialwerweise für $t_1 = t_0$. Ansonsten sei $t_1 = \max\{t | t \leq t_0 \wedge X(t) > 0\}$. Dann sendet der Shaper dauernd mit Rate ρ während des Intervalls $[t_1,t_0]$, daraus folgt $R_1(t_1,t_0) = \rho(t_0 - t_1)$. Auf der anderen Seite gilt $R_1(t_1 + \tau,t_0 + \tau) \leq \rho(t_0 - t_1)$ per Definition des Shapers (5.1). Daraus folgt

$$R_1(t_1,t_1 + \tau) - R_1(t_0,t_0 + \tau) = R_1(t_1,t_0) - R_1(t_1 + \tau,t_0 + \tau) \geq 0 .$$ ♦

Das nächste Lemma trifft dieselbe Aussage wie Satz 5.4 für den Spezialfall eines Ein-Aus Shapers.

Lemma 9.2: ((σ,ρ) Shaper mit $\sigma = 0$) Wenn der Verkehrsstrom am Eingang $R_0(t)$ durch eine Hülle $b_0(\tau)$ beschränkt ist, dann ist der Verkehrsstrom am Ausgang $R_1(t)$ durch die Hülle $b_1(\tau) = b_0(\tau) - W_\rho(b)(\tau)$ beschränkt.

Beweis: Zunächst werde ein kompakter Ausdruck für den Verkehrsstrom am Ausgang gebildet:

$$\begin{aligned} R_1(t) &= R_0(t) - X(t) = R_0(t) - W_\rho(R_0)(t) \\ &= R_0(t) - \max_{0 \leq s \leq t}[R_0(s,t) - \rho(t-s)] \\ &= -\max_{0 \leq s \leq t}[-R_0(s) - \rho(t-s)] \\ &= \min_{0 \leq s \leq t}[R_0(s) + \rho(t-s)] . \end{aligned} \tag{9.24}$$

Um eine obere Schranke für die während des Zeitraums $[t_0,t_0 + \tau]$ geflossene Datenmenge zu finden, wird Lemma 9.1 angewandt. Danach gibt es ein t_1, so daß

$$R_1(t_0,t_0+\tau) \le R_1(t_1,t_1+\tau) \text{ und} \tag{9.25}$$

$$X(t_1) = 0 \Leftrightarrow R_1(t_1) = R_0(t_1). \tag{9.26}$$

Mit Hilfe von (9.24) und (9.26) ergibt sich aus (9.25)

$$\begin{aligned} R_1(t_0,t_0+\tau) &\le R_1(t_1+\tau) - R_0(t_0) \\ &= \min_{0 \le s \le t_1+\Delta t}[R_0(s) + \rho(t_1+\tau-s)] - R_0(t_1) \\ &= \min_{0 \le s \le t_1+\Delta t}[R_0(s) + \rho(t_1-s)] + \rho\tau - R_0(t_1). \end{aligned} \tag{9.27}$$

Als nächstes wird eine engere obere Schranke für den Wertebereich des Minimumoperators bestimmt. Gleichung (9.24) eingesetzt in (9.26) ergibt

$$\min_{0 \le s \le t_1}[R_0(s) + \rho(t_1-s)] = R_0(t_1) .$$

Daraus folgt, daß

$$\min_{0 \le s \le t_1+\tau}[R_0(s) + \rho(t_1-s)] = \min_{t_1 \le s \le t_1+\tau}[R_0(s) + \rho(t_1-s)].$$

Die weitere Entwicklung von Gleichung (9.27) mit diesem Ergebnis ergibt

$$\begin{aligned} R_1(t_0,t_0+\tau) &\le \min_{t_1 \le s \le t_1+\tau}[R_0(t_1,s) + \rho(t_1+\tau-s)] \\ &= \min_{0 \le u \le \tau}[R_0(t_1,t_1+u) + \rho(\tau-u)] && (9.28) \\ &\le \min_{0 \le u \le \tau}[b_0(u) + \rho(\tau-u)] && (9.29) \\ &= b_0(\tau) - W_\rho(b_0)(\tau). && (9.30) \end{aligned}$$

Für (9.28) wurde die Substitution $u = s - t_1$ angewandt, für (9.29) die Annahme, daß $R_0(t)$ durch die Hülle $b_0(\tau)$ beschränkt wird, und für (9.30) wurde (9.24) herangezogen. ♦

Für einen allgemeinen (σ,ρ) Shaper beschränkt das nächste Lemma den Verkehrsstrom am Ausgang mit jeder Hülle, die auch den Eingang beschränkt.

Lemma 9.3: ((σ,ρ) Shaper) Wenn der Verkehrsstrom am Eingang $R_0(t)$ durch eine Hülle $b_0(\tau)$ beschränkt ist, dann ist auch der Verkehrsstrom am Ausgang $R_1(t)$ durch dieselbe Hülle beschränkt.

Beweis: Die in der Zeitspanne $[t_0,t_0+\tau]$ geflossene Datenmenge am Ausgang ist

$$\begin{aligned} &R_1(t_0,t_0+\tau) = \\ &\quad R_0(t_0+\tau) - (W_\rho(R_0)(t_0+\tau) - \sigma)^+ - R_0(t_0) + (W_\rho(R_0)(t_0) - \sigma)^+. \end{aligned} \tag{9.31}$$

Fallunterscheidung:

	$W_\rho(t_0) \le \sigma$	$W_\rho(t_0) > \sigma$
$W_\rho(t_0 + \tau) \le \sigma$	Fall (A)	Fall (B)
$W_\rho(t_0 + \tau) > \sigma$	Fall (C)	Fall (D)

Fälle (A) und (D): Mit den Fallannahmen wird aus (9.31) jeweils

$$\begin{aligned} R_1(t_0,t_0+\tau) &= R_0(t_0,t_0+\tau) \\ &\le b_0(\tau)\,. \end{aligned}$$

Fall (B): Aus der Fallannahme wird aus (9.31)

$$\begin{aligned} R_1(t_0,t_0+\tau) &= R_0(t_0,t_0+\tau) + W_\rho(R_0)(t_0) - \sigma \\ &= [R_0(t_0+\tau) - W_\rho(R_0)(t_0+\tau) - (R_0(t_0) - W_\rho(R_0)(t_0))] \\ &\quad + (W_\rho(R_0)(t_0+\tau) - \sigma)\,. \end{aligned}$$

Der Ausdruck in eckigen Klammern entspricht dem Verkehrsstrom am Ausgang eines (σ,ρ) Shapers mit $\sigma = 0$, für den mit Lemma 9.2 eine obere Schranke bekannt ist: $[\ldots] \le b_0(\tau) - W_\rho(b_0)(\tau)$. Aus der Fallannahme folgt $W_\rho(R_0)(t_0+\tau) \le \sigma$. Insgesamt also

$$\begin{aligned} R_1(t_0,t_0+\tau) &\le b_0(\tau) - W_\rho(b_0)(\tau) \\ &\le b_0(\tau)\,. \end{aligned}$$

Fall (C): Aus der Fallannahme $W_\rho(R_0)(t_0+\tau) > \sigma$ wird aus (9.31)

$$\begin{aligned} R_1(t_0,t_0+\tau) &= R_0(t_0,t_0+\tau) - (W_\rho(R_0)(t_0+\tau) - \sigma) \\ &\le b_0(\tau)\,. \end{aligned}$$

♦

Mit diesen Vorarbeiten ist es jetzt möglich, Satz 5.4 zu zeigen.

Beweis von Satz 5.4: Zuerst werde der Ausdruck für den Verkehrsstrom am Ausgang umgeschrieben

$$\begin{aligned} R_1(t) &= R_0(t) - X(t) \\ &= R_0(t) - (W_\rho(R_0)(t) - \sigma)^+ \\ &= R_0(t) - \max\{W_\rho(R_0)(t) - \sigma, 0\} \\ &= \min\{R_0(t) - W_\rho(R_0)(t) + \sigma, R_0(t)\}\,. \end{aligned} \tag{9.32}$$

Um für die im Zeitraum $[t_0,t_0+\tau]$ geflossene Datenmenge eine obere Schranke zu bestimmen, wird aus (9.32) eine untere Schranke für $t = t_0$ und eine obere Schranke für $t = t_0+\tau$ abgeleitet:

$$R_1(t_0) \geq R_0(t_0) - W_\rho(R_0)(t_0)\,,$$

$$R_1(t_0+\tau) \leq R_0(t_0+\tau) - W_\rho(R_0)(t_0+\tau) + \sigma\,.$$

Daraus folgt, daß die Datenmenge beschränkt ist durch

$$R_1(t_0,t_0+\tau) \leq [R_0(t_0+\tau) - W_\rho(R_0)(t_0+\tau) - (R_0(t_0) - W_\rho(R_0)(t_0))] + \sigma\,.$$

Der Ausdruck in eckigen Klammern entspricht dem Verkehrsstrom am Ausgang eines (σ,ρ) Shapers mit $\sigma = 0$, für den mit Lemma 9.2 eine obere Schranke bekannt ist: $[\ldots] \leq b_0(\tau) - W_\rho(b_0)(\tau)$. Daraus folgt

$$\leq b_0(\tau) - (W_\rho(b_0)(\tau) - \sigma)\,.$$

Aus diesem Ergebnis folgt zusammen mit Lemma 9.3

$$R_1(t_0,t_0+\tau) \leq b_0(\tau) - (W_\rho(b_0)(\tau) - \sigma)^+\,. \qquad \blacklozenge$$