Technische Universität München

Josef Glasmann

Ressourcenmanagement für Echtzeitverkehre in Intranets

Lehrstuhl für Kommunikationsnetze

Lehrstuhl für Kommunikationsnetze

Ressourcenmanagement für Echtzeitverkehre in Intranets

Josef Glasmann

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender:	Univ.-Prof. Dr.-Ing. Klaus Diepold
Prüfer der Dissertation:	1. Univ.-Prof. Dr.-Ing. Jörg Eberspächer 2. Univ.-Prof. Dr. rer. nat. habil. Martina Zitterbart, Universität Karlsruhe (TH)

Die Dissertation wurde am 23. April 2003 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 14. Nov. 2003 angenommen.

Vorwort

Diese Arbeit ist im Laufe meiner Tätigkeit als wissenschaftlicher Assistent am Lehrstuhl für Kommunikationsnetze der Technischen Universität München entstanden. In dieser Zeit konnte ich bei meinen vielfältigen Aufgaben in Forschung und Lehre wertvolle Erfahrungen sowohl im Umgang mit Studenten und Diplomanden als auch bei der Planung und Durchführung von Industrieprojekten sammeln.

Mein besonderer Dank gilt meinem Doktorvater Prof. Dr.-Ing. Jörg Eberspächer, der mich durch zahlreiche Diskussionen, konstruktive Anmerkungen und seine freundliche Unterstützung in allen Phasen begleitet und maßgeblich zum Gelingen der Arbeit beigetragen hat.

Frau Prof. Dr. Martina Zitterbart von der Technischen Hochschule Karlsruhe möchte ich meinen herzlichen Dank für die Übernahme des Zweitgutachtens aussprechen.

Danke an alle Kolleginnen und Kollegen für die freundschaftliche, offene und kreative Atmosphäre am Lehrstuhl. Insbesondere möchte ich die Mitglieder der Arbeitsgruppe NAT (*Network Architecture Team)* herausstellen, die aufgrund des straffen Führungsstils von Stefan Butenweg, des Weitblickes von Anton Riedl und der unerschöpflichen Kreativität von Andrea Bör eine kontinuierliche Plattform für kritische, ermutigende und manchmal auch heitere Diskussionen bot.

Ebenso möchte ich Herrn Dr. Harald Müller und Herrn Jürgen Totzke von der Siemens AG für die konstruktiven Diskussionen und die gute Zusammenarbeit innerhalb des gemeinsamen Forschungsprojektes danken.

An dieser Stelle möchte ich auch alle Diplomanden nicht vergessen, die an zahlreichen Verkehrsmessungen sowie der Implementierung des Prototyps beteiligt waren.

Ein besonders herzliches Dankeschön gilt meiner Freundin Melanie, die durch ihr Verständnis und ihre tatkräftige Unterstützung, vor allem im Kampf gegen Rechtschreibfehler, einen erheblichen Beitrag geleistet hat. Abschließend möchte ich mich bei meinen Eltern Elfriede und Josef Glasmann bedanken, sie haben den Grundstein gelegt.

München, im April 2003 *Josef Glasmann*

Kurzfassung

Um die Integration von Daten- und Kommunikationsdiensten in IP-basierten Intranets voranzutreiben, wurde eine skalierbare Ressourcenmanagement-Architektur entwickelt, die in der vorliegenden Arbeit beschrieben ist.

Mit Hilfe dieser Architektur ist es möglich, in Firmennetzen interaktive Kommunikationsdienste wie Telefonie, Videotelefonie oder Videokonferenzen mit einer Güte (*Quality of Service* QoS abzuwickeln, wie sie die Anwender von PSTN-Netzen gewohnt sind. Aufbauend auf der DiffServ-Technologie ermöglicht sie die Realisierung verschiedener, vordefinierter Dienstklassen mit harten und weichen Dienstgütegarantien.

Für die Realisierung der Ressourcenmanagement-Architektur wurde ein zentralisierter Ansatz gewählt. Er basiert auf der funktionalen Trennung von Ressourcenreservierung und paketverarbeitenden Prozessen. Dadurch können in den Netzknoten verbindungsbezogene Zustandsinformationen und Verarbeitungsschritte vermieden werden.

Während die Paketverarbeitung in den Netzknoten stattfindet, wird die Verarbeitung der Reservierungsnachrichten in einen zentralen Server ausgelagert. Dazu wird ein Ressourcen-Manager RM definiert, der Reservierungsanfragen entgegen nimmt, eine Zugangskontrolle durchführt und die Zustandsinformationen des Netzes verwaltet. Für große Intranets sieht der Ansatz die Aufteilung in mehrere Netz-Domänen vor. Jede Domäne wird zentral von einem RM verwaltet, der mit anderen benachbarten RM über ein Signalisierungsprotokoll verbunden ist.

Für die Bestimmung geeigneter Zugangskontrollverfahren wurden Verkehrsmessungen an realen Quellen durchgeführt, Methoden zur Charakterisierung von Quellenverkehren entwickelt und für den Systemkontext in Frage kommende Zugangskontrollverfahren untersucht. Die analytischen Ergebnisse wurden durch Simulationen validiert.

Um darüber hinaus eine leichte Einführbarkeit sicherzustellen, wurde beim Systemdesign darauf geachtet, die Anzahl der neu einzuführenden Schnittstellen zu minimieren und, soweit möglich, standardisierte Protokolle zu verwenden. Daher wurde die RM-Architektur an bereits vorhandene Dienststeuerungseinheiten im Netz (z.B. H.323-GK, SIP-Proxy) angebunden. Über die Dienstsignalisierung wird eine Zugangskontrolle angestoßen, welche dann Ende-zu-Ende, d.h. für jeden Link des Nutzdatenpfades, eine Ressourcenreservierung durchführt.

Mit der RM-Architektur wird ein Reservierungsverfahren bestehend aus einem Protokoll und einer Zugangskontrollfunktion eingeführt. Dieser Ansatz skaliert auch für größere IP-Netze und zeichnet sich durch seine gute Integrationsfähigkeit in eine bestehende Systemumgebung aus.

Um eine bestmögliche Entkopplung der Architektur von der darunter liegenden Netztechnologie zu erreichen, wurde unabhängig vom Ressourcenmanagement ein Topologieerkennungsdienst definiert. Für dessen Realisierung wurde ebenfalls ein zentralisierter Ansatz gewählt und ein Topologie-Manager spezifiziert, der die Netztopologie ermittelt und diese in einem einheitlichen Topologiemodell speichert. Über eine offene Schnittstelle bietet er eine abstrakte Sicht auf die Topologie und Konfiguration des Netzes, auf der das Ressourcenmanagement aufsetzen kann.

Das Ressourcenmanagement-System und der Topologieerkennungsdienst wurden prototypisch implementiert und in realen Intranet-Umgebungen getestet. Dadurch konnten die Realisierbarkeit des Ansatzes nachgewiesen und die Funktionsweise der Protokolle und Verfahren gezeigt werden.

Abkürzungsverzeichnis

ACF	Admission Confirm
AF	Assured Forwarding
API	Application Programming Interface
AR	Auto-Regressive
ARMA	Auto Regressive Moving Average
ARQ	Admission Request
AS	Assured Service
ASN.1	Abstract Syntax Number One
ATM	Asynchronous Transfer Mode
BB	Bandwidth Broker
BGP	Border Gateway Protocol
BGRP	Border Gateway Reservation Protocol
BHCA	Busy Hour Call Attempts
BR	Border Router
CAC	Call Admission Control
CBR	Constant Bit Rate
CCITT	Comité Consultatif International Télégraphique et Téléphonique
CLI	Command Line Interface
CLS	Controlled Load Service
COPS	Common Open Policy Service
CoS	Class of Service
CSMA-CD	Carrier Sense Multiple Access - Collision Detect
CTB	Complex Token Bucket
DB	Delay-Based
DiffServ	Differentiated Services Architecture
DPS	Dynamic Packet State
DRQ	Disconnect Request
DS	Dienststeuereinheit
DSCP	DiffServ CodePoint
DSP	Digital Signaling Processor
EDF	Earliest Deadline First
EF	Expedited Forwarding
EPD	Early Packet Discard

FCFS	First Come First Serve
FIFO	First In First Out
FP	Fixed Priority
FTP	File Transfer Protocol
fr-ARIMA	fractional Auto-Regressive Integrated Moving Average
fr-BM	fractional Brownian Motion
GARA	Global Architecture for Reservation and Allocation
GCRA	Generic Cell Rate Algorithm
GK	Gatekeeper
GPS	Generalized Processor Sharing
GS	Guaranteed Service
GSM	Global System for Mobile Telecommunications
HDTV	High Density Television
H-FSC	Hierarchical Fair Service Curve
ICMP	Internet Control Management Protocol
IETF	Internet Engineering Task Force
IN	Intelligentes Netz
IntServ	Integrated Services Architecture
IP	Internet Protocol
IPP	Interrupted Poisson Process
IRQ	Information Request
IRR	Information Response
ISDN	Integrated Services Digital Network
IST	Information Society Technologies
ITU	International Telecommunication Union
ITU-T	International Telecommunication Union Telecommunication Sector
LAN	Local Area Network
LL	LiveLan
LSP	Label Switched Path
MA	Moving Average
MCU	Multipoint Control Unit
MGC	Media Gateway Controller
MEGACO	Media Gateway Control
MIB	Management Information Base
MIP	Mixed-Integer Programm
MLE	Maximum Likelyhood Estimation

MMFP	Markov Modulated Fluid Process
MMPP	Markov Modulated Poisson Process
MO	Managed Objects
MPLS	Multiple Protocol Lable Switching
MTU	Maximum Transmission Unit
NM	NetMeeting
NS	Network Simulator
NSIS	Next Steps In Signaling
NTP	Network Time Protocol
OSI	Open Systems Interconnection
PDB	Per Domain Behavior
PHB	Per Hop Behavior
PIB	Policy Information Base
PL	Paketlänge
PP	Poisson Process
PS	Premium Service
PSTN	Public Switched Telephone Network
PZA	PaketZwischenAnkunftszeit
QoS	Quality of Service
QoS-RM	Quality of Service Resource Manager
R/S	Rescaled Adjusted Range Statistic
RAP	Resource Allocation Protocol
RB	Rate-Based
RCA	Resource Control Agent
RED	Random Early Discard
RM	Ressourcen-Manager / Ressourcenmanagement
RRA	Resource Reservation Agents
RSVP	Resource Reservation Protocol
RTCP	Real Time Transport Control Protocol
RTP	Real Time Transport Protocol
SDP	Session Description Protocol
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SLS	Service Level Specification
SMI	Structure Management Information
SNMP	Simple Network Management Protocol

SPPI	Structure of Policy Provisioning Information
STB	Simple Token Bucket
TB	Token Bucket
TCA	Traffic Conditioning Agreement
TCP	Transmission Control Protocol
TDM	Time Division Multiplex
TE	Terminal Equipment
TES	Transpond Expand Sample
TM	Topolgie-Manager / Topologiemanagement
TOS	Type of Service Field
TSPEC	Traffic Specification
UDP	User Datagram Protocol
VBR	Variable Bit Rate
VoIP	Voice over IP
VPN	Virtual Private Network
WFQ	Weighted Fair Queuing
WWW	World Wide Web

Inhaltsverzeichnis

1. Einführung

Das Internet hat in den vergangenen Jahren eine rasante Entwicklung vollzogen. Seit Erfindung des WWW-Browsers stieg die Anzahl der Internetnutzer und damit das Verkehrsaufkommen exponentiell an. Der kostengünstige und anwenderfreundliche Zugang zu einer schnell wachsenden Menge weltweit verfügbarer Informationen hat ebenso zu dieser Entwicklung beigetragen wie die Einführung immer neuer Applikationen.

Mittlerweile ist das Internet nicht mehr nur ein Netz für reine Datenapplikationen wie z.B. Email, FTP oder WWW. Das Internet lebt vielmehr von der Vielfalt seiner Anwendungen und den offenen Programmierschnittstellen. Im Laufe der letzten Jahre haben sich neue Anwendungen wie IP-Telefonie, *Streaming* Audio und Video, *Broadcasting* (Radio, Fernsehen) oder Netzspiele mit höheren Anforderungen an die Netze verbreitet. Auf Seiten der Netze konnten durch den Einsatzes modernster optischer Transporttechnologien die notwendigen Voraussetzungen für diese Anwendungen geschaffen werden.

Deshalb entstand die Vision, mit dem Internet eine einheitliche, anwenderfreundliche und weltumspannende Informations- und Kommunikationsinfrastruktur mit nahezu grenzenlosen Möglichkeiten für völlig neue Anwendungen zu schaffen.

Bis heute wird das Internet diesem Anspruch jedoch nicht gerecht. Die Anwendungsmöglichkeiten, die es bietet, übersteigen immer noch seine Fähigkeiten. So wurden zwar komplexere *E-Commerce*-, *Collaboration*- und *Tele-Education*-Systeme entwickelt, die sich häufig aus Daten- und Echtzeitdiensten zusammensetzen. Jedoch werden in den meisten Realisierungen die echtzeitkritischen Dienste wie Telefonie, Videotelefonie oder Multimedia-Konferenzen nach wie vor über die herkömmlichen PSTN-Netze abgewickelt. Das Internet bietet trotz großer Bandbreiten im Backbone momentan noch nicht die technischen Voraussetzungen, um echtzeitkritischen Verkehr dienstgerecht zu transportieren. Zudem sind die Datennetze auf der letzten Meile noch nicht überall so ausgebaut, dass alle Nutzer entsprechend angebunden werden könnten.

Ähnlich verhält es sich im Bereich privater Firmennetze (Intranets). Die Anbindung von Zweigstellen an größere Standorte erfolgt häufig über teure Mietleitungen öffentlicher Netze mit niedriger Kapazität. Dabei werden in den meisten Fällen Daten- und Telefonverkehre auf separaten Leitungen übertragen. Innerhalb eines Firmenstandortes werden nach wie vor zwei getrennte Netze für Telefonie- und Datenanwendungen betrieben. Die Datennetze (LANs) sind meist so gut dimensioniert, dass sie prinzipiell beide Verkehrsarten transportieren könnten. Dennoch zögern viele Firmen mit der Einführung rein IP-basierter Telefonsysteme oder gar Multimediaapplikationen, da sie an deren Zuverlässigkeit zweifeln.

Dies trägt dazu bei, dass die Einführung echtzeitkritischer Anwendungen sowohl im Intranet als auch im Internet momentan auf geringe Kundenakzeptanz stößt und daher nur sehr schleppend vorangeht. Folglich wurden bis heute die Möglichkeiten der Integration von Informations- und Kommunikationsdiensten noch nicht annähernd ausgenutzt.

1.1 Anforderungen von Echtzeitverkehren an die Datennetze

Damit die Vision einer einheitlichen, anwenderfreundlichen und weltumspannenden Informations- und Kommunikationsinfrastruktur Wirklichkeit werden kann, sind zuerst die notwendigen Voraussetzungen auf Seiten der Netzinfrastruktur zu schaffen. In [Ebe01] wurden unter anderem folgende Defizite des Internets festgestellt: „zu geringe Übertragungsraten (’Geschwindigkeit’), unzureichende Dienstqualität (’*Quality of Service*’), unzureichende Zuverlässigkeit und unzureichende Rechnerleistung der Informationsquellen (Server)“. Interaktive Kommunikationsdienste über IP-Netze (Intranet, Internet) werden nur dann Kundenakzeptanz erlangen, wenn sie mindestens mit vergleichbarer Güte (QoS) angeboten werden können, wie es Teilnehmer von PSTN-Netzen (*Public Switched Telephone Networks*) gewohnt sind. Ein PSTN-Nutzer akzeptiert eher, vom Netz blockiert zu werden, als Schwankungen der Verbindungsqualität oder gar einen Abbruch während einer laufenden Verbindung in Kauf nehmen zu müssen.

Um echtzeitkritische Verkehre über Paketnetze übertragen und dem Nutzer eine gewisse Dienstgüte garantieren zu können, müssen in den Netzen Verzögerungen (*Delay*), Verzögerungsschwankungen (*Jitter*) und Verluste (*Loss*) von Paketen begrenzbar sein. Insbesondere bei der Übertragung von Sprachdaten reagiert der Anwender sehr empfindlich auf kurzzeitige Schwankungen der Verbindungsqualität. Hoher Jitter und jegliche Art von Paketverlusten wirken sich beim Empfänger unmittelbar auf die Verständlichkeit der Sprachinformation aus. Ebenso führt eine zu große Gesamtverzögerung der Sprachübertragung zu einer Verschlechterung des Kommunikationsverhaltens.

Diese unerwünschten Effekte treten bei der Aggregation von Verkehren im Netz auf, wenn es zur Überlastung einzelner Links kommt. Auch wenn die Netze gut dimensioniert sind, kann es dennoch kurzfristig zu Stausituationen kommen. Diese sind jedoch ohne die Einführung weitergehender Maßnahmen kaum zu vermeiden, da das Nutzerverhalten, die Verkehrsbeziehungen und die statistischen Eigenschaften der Verkehre nie genau vorhersagbar sind. Um Netze so zu planen, dass ohne weitere Maßnahmen auch kurzfristige Stausituationen weitgehend vermieden und QoS garantiert werden können, ist eine enorme Überdimensionierung erforderlich. Zudem stehen in den Zugangsnetzen die notwendigen Kapazitäten oft nicht zur Verfügung oder sind dort nach wie vor sehr teuer. Im Kernnetz sind die reinen Bandbreitenkosten zwar geringer, dennoch ist es für einen Netzbetreiber auch im Kernnetz ein Wettbewerbsvorteil, den tatsächlichen Ressourcenbedarf möglichst gut abschätzen und damit gering halten zu können.
Um nun eine bestimmte Dienstgüte für Echtzeitverkehre zu garantieren, müssen Stausituationen im Netz kontrolliert werden. Dazu müssen neue Mechanismen in IP-Netzen eingeführt werden, welche dynamisch die vorhandenen Netzressourcen verwalten. Dies kann beispielsweise dadurch geschehen, dass man den Verkehrszufluss in die Netze regelt (Netzzugangskontrolle). Ein positiver Nebeneffekt einer Zugangskontrollfunktion ist, dass man solche Netze niedriger dimensionieren und damit effektiver betreiben kann.

1.2 Technisches Umfeld

Um Stausituationen im Netz kontrollieren zu können, sind über die Netzplanung hinaus weitergehende Maßnahmen wie die Einführung eines dynamischen Netzmanagements notwendig.

Ein Netzmanagement besteht aus operativer Sicht aus einem statischen und einem dynamischen Teil. Der statische Teil wird im Folgenden als Konfigurationsmanagement, der dynamische Teil als QoS-Management bezeichnet.

Das Konfigurationsmanagement setzt die von der Netzplanung vorgegebenen Dimensionierungswerte in eine Netzkonfiguration um. Das QoS-Management hingegen ist für den laufenden Netzbetrieb zuständig. Es stellt einem Netzbetreiber einen Satz von Techniken zur Verfügung, mit dem die vorhandenen Ressourcen dynamisch verwaltet werden können. Sie dienen der Kontrolle von kurz- und mittelfristig auftretenden Stausituationen im Netz. Dabei werden zwei Ziele verfolgt:

- Optimale Ausnutzung der vorhandenen Netzressourcen.
- Gewährung von Dienstgütegarantien.

Um die beiden Ziele zu erreichen, wird das QoS-Management in zwei Teilprozesse aufgespalten: Das Ressourcenmanagement und das Traffic-Engineering. Während das **Ressourcenmanagement** den Zufluss des Verkehrs in das Netz regelt (*Admission Control*) und die zugelassenen Verkehre den Netzressourcen zuweist (Klassifizierung), versucht das **Traffic-Engineering** durch gezielte Eingriffe in das Routing (*Adaptive-, QoS-based Routing*) und/oder das Einrichten von MPLS-Pfaden für eine gleichmäßige Auslastung der Links im Netz zu sorgen. Die Voraussetzung für Dienstgütegarantien ist jedoch eine ausreichende Dimensionierung des Netzes.

Um für die weitere Arbeit eine einheitliche Sicht auf die an der Bereitstellung von QoS in IP-Netzen beteiligten Prozesse zu schaffen, wird in Abbildung 1-1 exemplarisch das Zusammenspiel dargestellt.

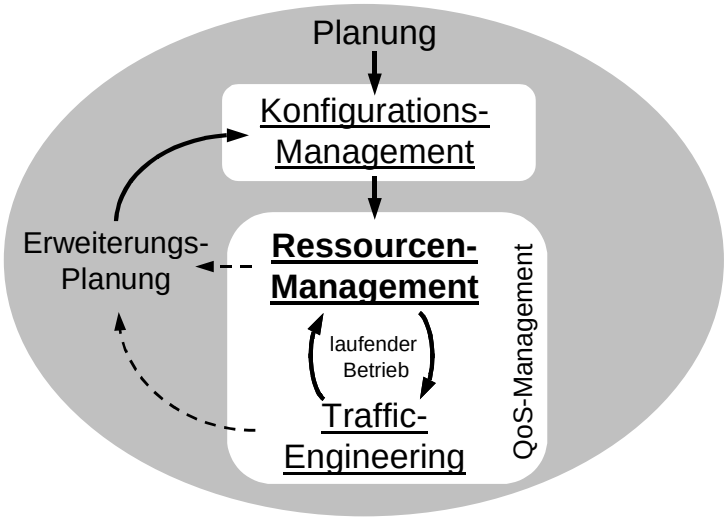


Abbildung 1-1: Netzplanung & optimierter Netzbetrieb - Prozessablaufdiagramm

Alle hier betrachteten Vorgänge gehören zum Aufbau, effektiven Betreiben und inkrementellen Erweitern der Netze. Im Gegensatz zu den Planungsprozessen sind die Prozesse des QoS-Managements (Ressourcenmanagement, Traffic-Engineering) weitgehend automatisierbar und laufen synchron zum Netzbetrieb ab.

Der *Netzplanungsprozess* liefert geeignete Standorte von Netzknoten, eine günstige Struktur der Vermaschung sowie eine Dimensionierungsvorschrift der Links. Darüber hinaus gibt er die Startkonfiguration des Netzes hinsichtlich der Routing-Metriken und der Dienstklassendimensionierung vor, die dann mit Hilfe eines *Konfigurations-Managements* auf die Netzknoten heruntergeladen wird.

Das *Ressourcenmanagement* wird als ein Teil des operativen QoS-Managements benötigt, um echtzeitkritische Verkehre vor kurz- und mittelfristig auftretenden Überlastsituationen (Millisekunde bis Sekunde) im Netz zu schützen. Es verwaltet die Netzressourcen dynamisch, regelt den Teilnehmerzugang und überwacht die Nutzung der Netzressourcen. Steigt das

Verkehrsaufkommen, wächst die Blockierungswahrscheinlichkeit zunächst punktuell auf einzelnen Links stark an.

Das *Traffic-Engineering* bietet die Möglichkeit, die Verkehrsflüsse im Netz den geänderten Verhältnissen anzupassen. Durch das Optimieren der Routen während des laufenden Netzbetriebs können mittelfristig (Stunde bis Woche) Blockierungswahrscheinlichkeiten gesenkt und der Zeitpunkt einer Kapazitätserweiterung verzögert werden. Das Traffic-Engineering kann insbesondere in vermaschten Netzen effektiv eingesetzt werden, in denen kleine Teile des Netzes nahezu den gesamten Verkehr tragen. Eine gleichmäßigere Auslastung wird mit Hilfe von Modifikationen der verwendeten Routen/Pfade im Netz (*QoS-Routing, Load Sharing, MPLS, ect.*) erzielt, die sich dynamisch auf veränderte Lastverhältnisse im Netz anpassen. Die Methoden des Traffic-Engineerings sollten schon in den Planungsprozess einbezogen werden.

Das Traffic-Engineering kann vom Ressourcen-Management getriggert werden, wenn z.B. die Blockierungsrate von Verbindungen einen bestimmten Grenzwert übersteigt. Bei stetig wachsendem Verkehrsaufkommen im Netz können zu einem bestimmten Zeitpunkt keine neuen Pfade mehr gefunden werden und die Lastgrenze des Netzes ist erreicht. Die notwendige Re-Dimensionierung oder Erweiterungsplanung des bestehenden Netzes ist in diesem Fall unumgänglich.

Die zeitliche Dimension, in der die jeweiligen Prozesse arbeiten, ist in der Abbildung 1-2 noch einmal zusammenfassend dargestellt.

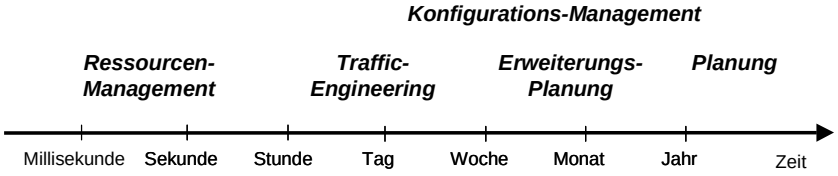


Abbildung 1-2: Zeitliche Granularität der Prozesse

1.3 Der Ende-zu-Ende Dienstgütebegriff

In diesem Abschnitt wird der Ende-zu-Ende Dienstgütebegriff aus Anwendersicht genauer betrachtet. Dazu wird ein einfaches Systemmodell entworfen, das aus einem Kommunikationssystem und aus Teilnehmern besteht. Das Kommunikationssystem selbst wiederum setzt sich analog zu [SZ95] aus Terminals und einem Transportnetz zusammen.

Abbildung 1-3 zeigt ein Beispiel für eine Videoapplikation, bei der echtzeitkritische Daten von der Quelle (Kamera, Terminal A) zur Senke (Bildschirm, Terminal B) übertragen werden. Die Qualität des Dienstes wird vom Empfänger bei der Darstellung auf dem Bildschirm wahrgenommen und nach subjektiven Kriterien bewertet. Der Dienstgütebegriff gilt somit aus der Sicht der Teilnehmer auf Applikationsebene.

Bei Echtzeitanwendungen erwartet der Teilnehmer, dass ein Sprach- oder Videosignal vom Kommunikationssystem so transportiert wird, dass es dem Empfänger unverändert zur Verfügung gestellt wird. Die Verarbeitung des Ursprungssignals in den Endgeräten ist für den Teilnehmer ebenso transparent wie die Übertragung der Daten über das Transportnetz. Für die Bewertung eines Systems hat die ITU-T (*International Telecommunication Union*) in ihrer P-Standardserie zur Bestimmung der Ende-zu-Ende Übertragungsqualität von Multimediaverkehren objektive Mess- und subjektive Testmethoden definiert.

Das Kommunikationssystem ist für die entsprechende Aufbereitung, Übertragung und Darstellung eines Ursprungssignals zuständig. Jeder dabei notwendige Verarbeitungsschritt

des Ursprungssignals hat einen Einfluss auf die vom Teilnehmer wahrgenommene Dienstqualität. Im Folgenden werden die einzelnen Verarbeitungsschritte am Beispiel des Videosignals von der Quelle bis zur Senke genauer betrachtet (siehe Abbildung 1-3).

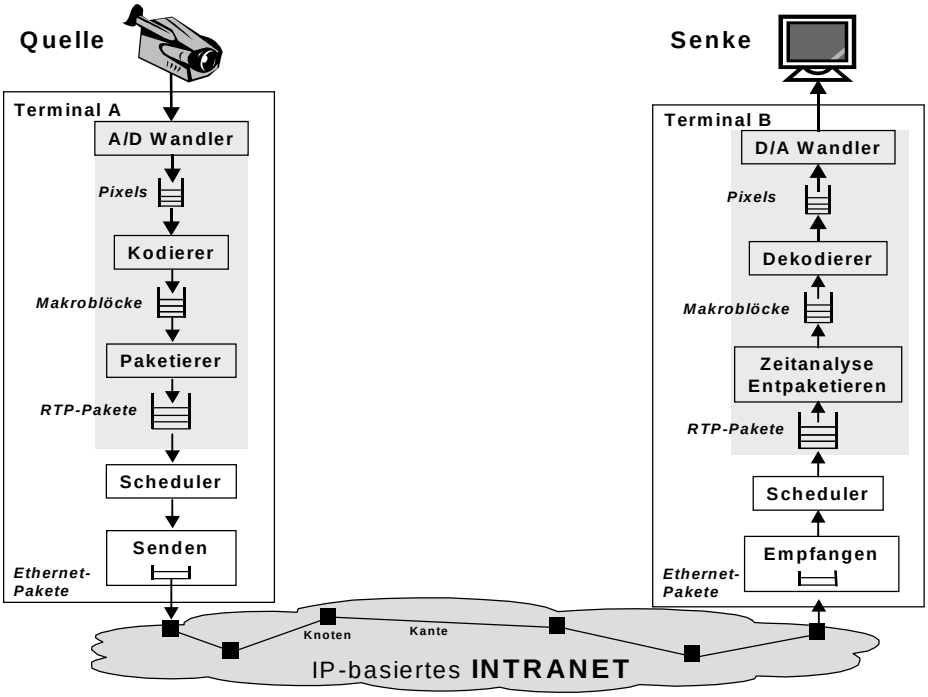


Abbildung 1-3: Verarbeitungsschritte eines Signals auf dem Weg von der Quelle zur Senke

Der Signalpfad kann aus den Verarbeitungsprozessen in den Terminals sowie in den Netzknoten modelliert werden. Das Terminal führt in der Regel mehrere Prozesse gleichzeitig aus. Es koordiniert alle aktiven Prozesse und weist die Systemressourcen den einzelnen Prozessen gemäß ihrer Priorität zu. Eine Echtzeitanwendung stellt einen Anwendungsprozess dar. Er besteht aus einem Analog/Digital-Wandler und der Implementierung eines Kodierverfahrens, mit dem die abgetasteten Analogsignale komprimiert werden. Die kodierten Nutzdatenframes werden anschließend nach einem bestimmten Paketierungsschema zu einzelnen Übertragungseinheiten (RTP-Pakete) zusammengefasst. Sowohl das Kodierungsverfahren als auch die Paketierung verursachen eine Verzögerung der originalen Sprach- oder Videosignale. Die RTP-Pakete werden anschließend für die Übertragung der Nutzdaten aufbereitet und auf dem Übertragungsmedium ausgesendet.

Das Terminal ist über eine Zugangsleitung an das Kommunikationsnetz angebunden, welches durch Knoten und Kanten (Links) repräsentiert wird. Ein Knoten besitzt mehrere Ein-/ Ausgänge und vermittelt die Pakete auf ihrem Weg zum Ziel. Aus Sicht der Dienstgüte stellt jeder Knoten eine Stelle dar, an der die Pakete je nach Auslastung des Knotens mehr oder weniger stark verzögert werden. Die Links hingegen stellen eine konstante Verzögerungskomponente dar, die durch die Signallaufzeit vorgegeben ist.

Der Dienstgütebegriff aus der Sicht der Teilnehmer schließt somit alle signalverarbeitenden Prozesse in den Terminals und im Netz mit ein. Um Ende-zu-Ende eine gewisse Dienstqualität zu garantieren, wird eine Ressourcenreservierung in den Endgeräten und im Netz benötigt, die über eine Signalisierung koordiniert werden muss.

1.4 Zielsetzung der Arbeit

Das Ziel der hier vorgestellten Arbeit ist es, eine Ressourcenmanagement-Architektur für Echtzeitverkehre in Intranets zu entwickeln. Das Ressourcen-Management (RM) soll ermöglichen, echtzeitkritische Dienste über IP-basierte Intranets mit verschiedenen garantierbaren Qualitätsstufen abzuwickeln. Bei diesen Diensten handelt es sich vorwiegend um Telefonie-, Videotelefonie- und Videokonferenz-Anwendungen.

Der Dienstgütebegriff umfasst Ende-zu-Ende Verzögerungen und Verluste von IP-Paketen entlang des Nutzdatenpfades zwischen den Terminals und schließt darüber hinaus die Verbindungsaufbauzeiten der Dienststeuerung ein. Für die Bereitstellung der Übertragungsressourcen im Netz ist das Ressourcen-Management zuständig, für die Bereitstellung der Ressourcen im Terminal die Applikation. Die Koordination soll ein neues Reservierungsverfahren übernehmen.

Bei dem Design der RM-Architektur soll auf eine gute Skalierbarkeit geachtet werden. Das RM-Konzept soll nicht nur für Intranets geeignet, sondern auch auf größere IP-Netze übertragbar sein. Darüber hinaus sollen Aspekte der Migration, d.h. der leichten Einführbarkeit des RM-Systems in eine bestehende Infrastruktur von Anfang an berücksichtigt werden. Das Ressourcen-Management ist dabei so in eine Systemumgebung einzubinden, dass der Installations- und Managementaufwand möglichst gering gehalten werden kann.

1.5 Gliederung

Kapitel 2 gibt zunächst einen kurzen Überblick über die Forschungsgebiete im Bereich QoS in IP-Netzen. Danach werden einige aus der Literatur bekannte QoS-Architekturen beschrieben, anhand ihrer Merkmale diskutiert und bewertet. Das Kapitel schließt mit einem Architekturenvergleich.

Kapitel 3 gibt einen Überblick über die Konzepte der RM-Architektur. Es werden Ziele und Merkmale des Systems vorgestellt. Anschließend werden verschiedene Betreiberszenarien für die RM-Architektur aufgezeigt und die prinzipielle Funktionsweise des Reservierungsverfahrens anhand eines einfachen Beispiels erläutert. Das Kapitel schließt mit einer Abgrenzung zu den standardisierten IETF-Architekturen und der Einordnung der RM-Architektur in den Kontext der in Kapitel 2 vorgestellten Arbeiten.

Kapitel 4 liefert die Grundlagen für die Auswahl geeigneter Zugangskontrollverfahren für die RM-Architektur. Es werden Messungen von realen Echtzeitverkehrsquellen in IP-Netzen durchgeführt und die Ergebnisse anhand verschiedener statistischer Methoden analysiert.

In Kapitel 5 wird zunächst ein Verfahren zur Charakterisierung der in Kapitel 4 gemessenen Verkehre ausgewählt. Danach wird eine Methodik zum Parametrisieren von Messreihen entwickelt und auf die Messungen aus Kapitel 4 angewendet. Ferner werden für das RM-System in Frage kommende Zugangskontrollverfahren ausgewählt und anhand der parametrisierten Messreihen untersucht. Die Ergebnisse werden mit Hilfe von Simulationen validiert. Alle untersuchten Verfahren werden abschließend bewertet und zwei davon für das RM-System ausgewählt.

In Kapitel 6 wird die RM-Architektur im Detail vorgestellt. Dazu werden die einzelnen Komponenten und Protokolle der RM-Architektur spezifiziert. Ferner werden verschiedene Anwendungs-Szenarien der RM-Architektur betrachtet. Abschließend wird ein Einblick in die prototypische Implementierung gegeben.

Kapitel 7 schließlich fasst die wesentlichen Neuerungen und Ergebnisse dieser Arbeit zusammen.

2. Vergleich und Bewertung von QoS-Architekturen

In den vergangenen Jahren wurde viel auf dem Gebiet der verteilten Multimediasysteme geforscht. Die Steuerung von komplexen multimedialen Diensten sowie die Verarbeitung der Sprach- und Videosignale in Echtzeit stellen große Anforderungen an die Endgeräte und das Netz. Um diesen Anforderungen gerecht zu werden, sind für die Steuerung der Dienste mehrere Dienstarchitekturen z.B. H.323/SIP, ICEBERG, TINA und für die Steuerung der Netze mehrere Netzarchitekturen z.B. ATM, Internet (IntServ, DiffServ) entwickelt worden. Vereinfachend wird von folgender Aufgabenverteilung zwischen der Dienstarchitektur und der Netzarchitektur ausgegangen:

- **Dienstarchitekturen:** Sie dienen dem Aushandeln der Dienstmerkmale zwischen den Terminals. Durch sie werden die Anzahl und Art der Verbindungen (Medium, Kodierverfahren) und die Mindestanforderung an die Übertragungsqualität (QoS-Parameter) aus Teilnehmersicht festgelegt.
- **Netzarchitekturen:** Sie verwalten die Netzressourcen und sorgen dafür, dass die von den Teilnehmern geforderte Übertragungsqualität für die Dauer der Verbindung gewährleistet wird. Sie werden daher im Folgenden auch als QoS-Architekturen bezeichnet.

Ein Teilnehmer nimmt die Qualität eines Kommunikationsdienstes „Ende-zu-Ende“ wahr. Das schließt alle an der Verarbeitung des Ursprungssignals beteiligten Prozesse mit ein (OSI-Schichtenmodell*: Data-Plane* [Tan02]). Für die Einhaltung der QoS-Parameter ist daher eine Vielzahl von QoS-Mechanismen in Netzknoten und Endgeräten notwendig. Sie werden über eine Ressourcensteuerung koordiniert. Netzarchitekturen besitzen eine Ressourcensteuerung (*OSI: Control-Plane*) und sehen in der Regel eine Überwachung der tatsächlich erzielten Übertragungsqualität vor (*OSI: Management-Plane*).

In diesem Kapitel wird zunächst ein kurzer Überblick über die verschiedenen Forschungsgebiete im Bereich von “QoS in IP-Netzen“ gegeben. Alle hier aufgeführten Forschungsfelder beschäftigen sich mit Teilaspekten des Ende-zu-Ende QoS-Problems. Die Arbeiten auf den jeweiligen Gebieten liefern konkrete Lösungsvorschläge, welche im Folgenden als **QoS-Mechanismen** bezeichnet werden. Diese laufen in Endgeräten oder Netzknoten. Ein einzelner QoS-Mechanismus kann isoliert oder in Verbindung mit anderen eingesetzt werden und stellt einen Baustein einer QoS-Architektur dar. Bei einer **QoS-Architektur** werden mehrere QoS-Mechanismen in einen einheitlichen, übergeordneten Systemkontext gebettet und von einer Systemsteuerung koordiniert. In jüngster Vergangenheit wurde eine Vielzahl unterschiedlicher QoS-Architekturen entworfen, um die Qualität gegenüber klassischen IP-Netzen (*Best Effort*) zu verbessern. Dabei lassen sich zwei Klassen von QoS-Architekturen bilden, die sich in ihrer Zielsetzung unterscheiden:

- Klasse-A: Diese Ansätze versuchen primär einen Übertragungsdienst zu realisieren, der analog zu einem verbindungsorientierten Netz (ATM, TDM) harte QoS-Garantien ermöglicht, wie z.B. die Architekturen IntServ (Abschnitt 2.2.1), RRA (Abschnitt 2.3.1.2), SCORE (Abschnitt 2.3.2.1), GLOBUS (Abschnitt 2.4.1), DARWIN (Abschnitt 2.4.2).
- Klasse-B: Viele andere Ansätze versuchen primär der Philosophie des Internet zu folgen und auf der Grundlage einer einfachen und robusten Architektur einen bestmöglichen Übertragungsdienst zu realisieren. Beispiele sind die Architekturen DiffServ (Abschnitt 2.2.2), Bandwidth Broker (Abschnitt 2.3.1.1), AQUILA (Abschnitt 2.3.2.2), TEQUILA (Abschnitt 2.3.3), Egress-AC (Abschnitt 2.3.2.3), EP-AC (Abschnitt 2.3.2.4).

Erstere basieren auf verteilten Mechanismen und qualitativen Dienstgütespezifikationen, letztere auf einer dynamischen Dienstgütevereinbarung zwischen Teilnehmern und Netz, einer quantitativen Dienstgütespezifikation sowie einer Zugangskontrolle auf Verbindungsebene (pro *IP-Flow*).

Zum besseren Verständnis der QoS-Architekturen werden zunächst wichtige Forschungsfelder im Bereich der Endgeräte und Netze identifiziert und dazu jeweils einige QoS-Mechanismen beispielhaft aufgeführt. Im Anschluss daran werden übergreifende Systemlösungen (QoS-Architekturen) für das Internet vorgestellt. Das Kapitel endet mit einem Architekturenvergleich.

2.1 QoS-Mechanismen und Methoden

2.1.1 Endgeräte

In diesem Abschnitt werden QoS-Mechanismen vorgestellt, welche ausschließlich in den Endgeräten zum Einsatz kommen. Sie umfassen die Bereiche Nutzdatenverarbeitung (OSI: *User Plane*) und Steuerung (OSI: *Control Plane*). Die Forschung im Rahmen von Endgeräten konzentriert sich auf folgende Themenschwerpunkte:

- **Kodierung**
- **Task Scheduling**
- **Synchronisation**
- **Transportmechanismen**
- **Teilnehmersignalisierung**

Bei den **Kodierverfahren** geht es um Techniken zur Datenreduktion (*silence suppression, object-based video coding*), die mehrere Qualitätsstufen zulassen (*layered coding*), flexibel an die Lastsituation des Netzes angepasst werden können und unanfällig gegen Übertragungsfehler sind (*adaptive delay and error control*) [REH99], [BB01], [ACZ95], [TZ99].

Ferner werden **Scheduling**-Verfahren betrachtet. Sie sind ein Teil des Betriebssystems und dafür zuständig, die zentralen Ressourcen (z.B. Prozessorzeit) eines Multimediaterminals auf mehrere Anwendungsprozesse mit unterschiedlichen Performance-Anforderungen aufzuteilen [Aud91], [RS94], [Kan96]. Eng verbunden mit der Kodierung von Medienströmen einerseits und echtzeitfähigen Betriebssystemen andererseits ist das Feld der Synchronisation von echtzeitkritischen Medienströmen.

Die **Synchronisation** in einem verteilten System ist vom Empfänger eines oder mehrerer Medienströme durchzuführen. Sie sorgt für den Erhalt bzw. die Wiederherstellung des zeitlichen Bezugs von Paketen innerhalb eines Medienstromes sowie zwischen Paketen mehrerer unabhängiger Medienströme. Dazu wurden sowohl System- und Speichermodelle

[Ste90], [LG90], Protokolle [EDP92], [RH95], [Sch96] als auch ganze Software-Architekturen [LL96] entwickelt.

Als **Transportmechanismus** für die Übertragung von Echtzeitverkehren in IP-Netzen hat sich das RTP-Protokoll der IETF [Sch96] durchgesetzt. Neben den Nutzdaten enthält Informationen über den Nutzdateninhalt (z.B. Medientyp, Kodierverfahren), Informationen zur Transportkontrolle (Paketsequenznummer, Zeitstempel) und einen *Feedback*-Mechanismus (RTCP).

Die **Teilnehmersignalisierung** in IP-Netzen setzt sich aus zwei voneinander entkoppelten Prozessen zusammen, der Dienstsignalisierung (z.B. ITU-Standard H.323 [H.323], IETF-Standard SIP [HSS99]) und der Ressourcenreservierung (z.B. IETF-Standard RSVP). Die Dienstsignalisierung handelt die Medienströme und deren QoS-Anforderungen zwischen den Endgeräten aus. Danach wird die Ressourcenreservierung dazu verwendet, die für die Übertragung der Medienströme notwendigen Ressourcen im Netz zu belegen. Das Reservierungsprotokoll RSVP (*Ressource Reservation Protocol*) [Bra97] beispielsweise läuft sowohl auf der Teilnehmer-Netz Schnittstelle als auch innerhalb des Netzes zwischen den Netzknoten.

2.1.2 Netze

In diesem Abschnitt werden einige Forschungsgebiete im Umfeld der IP-Netze vorgestellt. Sie umfassen auf der einen Seite Methoden der Verkehrsmodellierung, Verkehrslenkung und Verkehrskontrolle, auf der anderen Seite Mechanismen der Netz-internen Signalisierung und des Managements. Die entwickelten Verfahren laufen in den Netzknoten und betreffen die Bereiche Paketverarbeitung (OSI: *Data-Plane*), Steuerung (OSI: *Control-Plane*) und Netzmanagement (OSI: *Management-Plane*). Die Themenschwerpunkte sind im Einzelnen:

- **Verkehrsmodellierung**
- **Verkehrslenkung**
- **Verkehrskontrolle**
  - *Traffic Conditioning (Classifying, Policing, Marking, Shaping)*
  - *Scheduling (service disciplines)*
  - *Puffermanagement*
- **Zugangskontrolle**
- **Reservierungsprotokolle**
- **Konfigurationsmanagement**

Bei der **Verkehrsmodellierung** wurde das Verhalten von Sprachquellen (ON-/OFF [Bra65], [JS00], MMPP [HL86]), Videoquellen (MMPP [HTL91], ARMA [Gru91], TES [Lee91]), Datenquellen ([LN91], [GW94]) sowie das langzeitabhängige Verhalten von aggregierten Ethernet-Verkehren ([Lel94], [Wil97]) untersucht. In [Ada97], [RK96] wird ein Überblick über die am häufigsten verwendeten Verkehrsmodelle gegeben. Die Modelle werden zur Ermittlung des Pufferverhaltens und des Bandbreitenbedarfes der Verkehre verwendet und liefern die Grundlage für die Entwicklung von Scheduling-, Zugangskontroll- und lastabhängigen Verkehrslenkungsverfahren (*Adaptive-Routing*) sowie von Methoden zur Planung und Dimensionierung der Netze.

Im Bereich des Traffic-Engineering wurden neue **Verkehrslenkungsverfahren** entwickelt, um den laufenden Netzbetrieb zu optimieren. Damit kann zum einen eine höhere Auslastung der Netze, zum anderen eine bessere Kontrolle von Stausituationen in den Netzen erreicht werden. Basierend auf einer Auslastungsüberwachung wird das Routing der Daten durch das Netz dynamisch angepasst. Die zugrundeliegenden Mechanismen beruhen beispielsweise auf einer Wegewahl, die nicht nur Ziel-basiert sondern z.B. Quell- und Ziel-basiert ist, auf einer

Optimierung der Routen unter Berücksichtigung von Nebenbedingungen (*Constraint-based Routing*) [WC96], [CN98] und auf dem Einstellen der Routen über Metrik-Anpassung [Rie02] oder Routen-Fixierung (*Route-Pinning*: z.B. MPLS). Diese Verfahren werden hier nicht näher betrachtet, da sie zwar die Lastsituation und damit die QoS verbessern, jedoch ungeeignet sind, den Teilnehmern eine QoS-Garantie zu gewähren.

Unter den Verfahren der **Verkehrskontrolle** werden im Folgenden alle QoS-Mechanismen verstanden, die ein IP-Paket auf seinem Weg durch einen Netzknoten durchlaufen kann (*Forwarding Process*). Sie werden normalerweise an den Ausgängen eines Netzknotens implementiert und beinhalten Methoden des *Traffic Conditionings, Schedulings* und *Puffermanagements*. Im Rahmen der IETF Standardisierung der QoS-Architekturen IntServ [Wr97] und DiffServ [Bla98] wurde der Begriff des „***Traffic Conditionings***“ geprägt. Er umfasst die Prozesse der Klassifizierung (*Classifying*) und Kennzeichnung (*Marking*) eines Paketes als zu einem bestimmten IP-Flow zugehörig, der Überwachung (*Policing*) eines IP-Flows hinsichtlich eines vorkonfigurierten Verkehrsprofils sowie des Formens (*Shaping*) eines IP-Flows auf ein bestimmtes Verkehrsprofil. Diese Mechanismen sind aus der Standardisierung des ATM bekannt und werden nun auch für IP-Netze verwendet. Eine Definition der Begriffe ist in [Bla98] zu finden.

Scheduling-Verfahren wurden bereits Anfang der 90iger Jahre für Breitbandnetze (ATM) entwickelt. Sie bestimmen, wie die Übertragungsbandbreite einer Ausgangsleitung eines Netzknotens auf die dort gleichzeitig ankommenden Pakete aufgeteilt wird. Sie legen die Reihenfolge fest, mit der die gepufferten Pakete bedient werden und bilden damit die Basis jeder QoS-Architektur. Scheduling-Verfahren verwalten in der Regel mehrere (meist virtuelle) Puffer und ermöglichen pro Puffer eine anteilsmäßige Zuweisung der zur Verfügung stehenden Speicherkapazität und Bedienrate. Sie unterscheiden sich in der Methode, wie sie die Bedienzeit des ausgehenden Links unter Berücksichtigung der Reservierungen auf die einzelnen Puffer aufteilen. Sie passen dazu entweder die Bedienrate einer Verkehrsklasse dynamisch an die aktuelle Lastsituation an (*rate based*) oder sie vergeben zeitabhängige Prioritäten (*delay based*). Darüber hinaus zeichnet sich das dynamische Verhalten eines Schedulers durch seinen Umgang mit ungenutzten Ressourcen aus. Entweder werden sie anteilig auf die anderen Verkehrsklassen aufgeteilt (*work conserving*) oder sie bleiben ungenutzt (*non-work conserving*). Letztere haben zwar Nachteile hinsichtlich der erzielbaren Netzauslastung, jedoch Vorteile bei der Übertragung Jitter-sensitiver Echtzeitverkehre [Zh95]. Ein guter Überblick über die bekanntesten Verfahren wird in [Zha95] gegeben.

Puffermanagementverfahren wurden speziell für TCP-Verkehre entwickelt, um die Pufferfüllstände in den Netzknoten zu steuern. Sie sorgen in Stausituationen dafür, dass frühzeitig, d.h. noch bevor die Puffer überlaufen, Pakete verworfen werden. Eine Steuerung der Pufferfüllstände ist nur dann möglich, wenn die betroffenen Applikationen in den Endgeräten die Verluste feststellen und ihre Senderate reduzieren. Transportprotokolle für Echtzeitverkehre (RTP) und für Datenverkehre (TCP) unterstützen einen solchen Mechanismus. Beispiele für solche Verfahren sind: *Early Packet Discard* EPD [RF95], *Random Early Discard* RED [FJ93] und *Fair Random Early Discard* FRED [LM97].

Bei den **Zugangskontrollverfahren** wird zwischen mess- und parameterbasierten Verfahren unterschieden. Ein Überblick wird in den Artikeln [Jam95], [PE96], [KS99] gegeben. Auf die Zugangskontrollverfahren wird detailliert in Kapitel 5 eingegangen.

Um harte QoS-Garantien zu ermöglichen, ist eine explizite **Reservierung** von Netzressourcen in den Netzknoten notwendig. Dafür werden entsprechende Protokolle und Zugangskontroll-verfahren benötigt. Für das Internet wurde ein Empfänger-orientiertes Reservierungserfahren (RSVP) standardisiert [Bra97]. Eine Aggregation von einzelnen Reservierungsnachrichten

wird im RFC 3175 [Bak01] vorgeschlagen. Daneben gibt es zahlreiche andere Ansätze, die den Signalisierungsverkehr reduzieren und dadurch besser skalierbar sind [ABF98], [PS98], [Feh99]. Um in einer Multi-Domänen Umgebung die Zustandsinformationen zu reduzieren, wird in [PHS00] basierend auf DiffServ-Netzen und BGP (*Border Gateway Protocol*) ein Reservierungsverfahren BGRP entwickelt, das Reservierungszustände hinsichtlich gemeinsamer Zieldomänen aggregiert (*Sink-Tree-Based*). Bei der IETF werden gerade in der Arbeitsgruppe NSIS (*Next Steps in Signaling*) Anstrengungen unternommen, um für verschiedene Reservierungsverfahren einen einheitlichen Transportmechanismus zu schaffen [Han03]. Dazu werden die Funktionen „Transport“ und „Nachrichtenverarbeitung“ getrennt (*Signaling Layer, Transport Layer*).

Mit der Einführung von QoS-Mechanismen in IP-Netze entstand unter anderem der Bedarf, die **Konfiguration** der Netzelemente dynamisch zu steuern. Dies bedeutet, dass das die Belegung von Ressourcen (z.B. Einrichten der Verkehrsklassen) nicht ein einmaliger Vorgang ist, sondern dynamisch an die aktuellen Bedürfnisse der Teilnehmer angepasst werden kann. Ferner muss der Teilnehmerzugang zu diesen Ressourcen durch das Konfigurieren von Klassifizierungs- und Filter-Einheiten gesteuert und überwacht werden. Darüber hinaus macht die immer größer werdende Vielfalt von verschiedenen Gerätetypen, welche unterschiedliche QoS-Mechanismen unterstützen, einen neuen Management-Ansatz erforderlich.
Die IETF sieht zwei Ansätze vor, die dieser Komplexität und dem damit verbundenen hohen Managementaufwand durch ein möglichst einfaches Konzept begegnen. Ein Entwurf wurde in der Arbeitsgruppe SNMPCONF erarbeitet und basiert auf dem SNMP-Konzept [Fad03]. Der Zweite wurde in der Arbeitsgruppe RAP (*Resource Allocation Protocol*) erarbeitet und basiert auf dem COPS-Protokoll (*Common Open Policy Service*) [Cha01]. Beiden liegt das Modell eines zentralen Servers und mehrerer Clients zugrunde. Die Konfiguration des Netzes wird von Seiten der Administration in einer einheitlichen Form beschrieben (*Policy = Set of Rules*) [Str01] und dem Server übergeben. Für die Spezifikation solcher *Policies* wurde in Anlehnung an SMI (*Structure Management Information*) [CPS99] eine entsprechende Syntax SPPI (*Structure of Policy Provisioning Information*) [Clo03] und entsprechend der MIB (*Management Information Base*) eine Datenstruktur PIB (*Policy Information Base*) definiert. Neben einer Technologie-unabhängigen PIB [WSH01] sind mehrere Technologie-abhängige PIBs vorgesehen [FM02]. Der Server verteilt die netzweit gültigen PIB-Objekte an die Clients. Die Clients bilden sie auf ihre lokal zur Verfügung stehenden QoS-Mechanismen ab und kümmern sich um die Umsetzung der *Policy-Rules* in den Netzknoten. Die Steuerung der Netzknoten kann dabei entweder direkt über die PIB-Objekte oder, nach Umsetzung auf entsprechende MIB-Objekte, über diese erfolgen.

2.2 QoS-Architekturen: IETF-Standards

In diesem Abschnitt werden die beiden IETF-Standards kurz vorgestellt und bewertet. Die beiden Ansätze dienen bei den Bewertungen in den Abschnitten 2.5 bzw. 3.5 als Referenzarchitekturen.

2.2.1 Integrated Services Architecture (IntServ)

IntServ (Klasse-A) wurde 1994 entwickelt, um IP-Netze für die Anforderungen echtzeitkritischer Anwendungen zu rüsten und den Teilnehmern Qualitätsgarantien geben zu können. In [BCS94] werden das Architekturkonzept und die notwenigen Mechanismen zur Verkehrskontrolle vorgestellt. Darüber hinaus wurden zwei Dienstklassen definiert: der *Guaranteed Service GS* [SPG97] und der *Controlled Load Service CLS* [Wro97].
IntServ ist ein zustandsbasierter Ansatz mit einem Reservierungsverfahren. Es handelt sich dabei um einen verbindungsorientierten Ansatz, der für jede Kommunikationsbeziehung (IP-

Datenstrom) die Reservierung von Netzressourcen Ende-zu-Ende ermöglicht. Die Reservierung erfordert die Implementierung von Protokollzustandsautomaten in den Routern und Terminals. Von der IETF wird als Reservierungsprotokoll das RSVP (*Ressource Reservation Protocol*) [Bra97] vorgeschlagen, das auf Sitzungen mit vielen Teilnehmern ausgelegt ist und auf *Soft-States* basiert. Letzt genannte Eigenschaft erfordert das Aktualisieren des Reservierungszustandes in den Routern durch das periodische Senden von sogenannten *Refresh*-Nachrichten.
In den Routern müssen nicht nur die ankommenden Nutzdatenpakete, sondern auch Signalisierungsnachrichten verarbeitet werden. Dabei sind für jeden Reservierungsvorgang die erforderlichen Ressourcen zu berechnen und bereitzustellen. Im Vergleich zu herkömmlichen Routern sind bei der Paketverarbeitung zusätzliche Mechanismen wie Klassifizierung (pro Paket), *Policing* (pro Datenstrom) und *Shaping* (pro Datenstrom) zu durchlaufen. Auch das Puffermanagement und der Scheduling-Prozess sind wesentlich komplexer, da entsprechend der Reservierung pro Datenstrom ein eigener Puffer verwaltet und bedient werden muss. Die Vielzahl der Verarbeitungsschritte pro Paket und die Tatsache, dass insbesondere in größeren Backbone-Netzen über einen Router gleichzeitig mehr als eine Million IP-Datenströme laufen können, stellen hohe Anforderungen an die Router-Performance. Neben der Komplexität des Puffermanagements führt bei Verwendung von RSVP auch die Menge der Signalisierungsnachrichten zu Skalierungsproblemen. Die Nachrichtenflut kommt dadurch zustande, dass zum einen für jeden einzelnen Medienstrom einer Multimediasitzung ein eigener RSVP-Signalisierungskanal benötigt wird, und zum anderen, dass periodisch *Refresh*-Nachrichten gesendet werden. In [Ber01] wurde ein Mechanismus zur Bündelung mehrerer RSVP-Nachrichten und zur Vereinfachung des *Refresh*-Mechanismus standardisiert. Darüber hinaus wurde in [Bak01] ein Mechanismus zur Aggregation von Reservierungszuständen (*State Reduction*) an Domänen-Grenzen standardisiert.

Bewertung
Für einen Netzbetreiber liegen die Stärken der IntServ-Architektur in der großen Flexibilität und der zeitlichen wie räumlichen Granularität des QoS-Angebotes. Die Teilnehmer können entsprechend ihrer Bedürfnisse und den Anforderungen individuell Ressourcen reservieren und damit die Dienstqualität selbst bestimmen. Die Schwächen des Ansatzes liegen in der schlechten Skalierbarkeit, dem großen Managementaufwand und der schwierigen Einführbarkeit.

Probleme bei der Migration ergeben sich aus der Tatsache, dass die IntServ-Technologie Ende-zu-Ende, d.h. sowohl auf allen Endgeräten als auch auf allen Netzknoten, eingeführt werden muss. Probleme bei der Skalierbarkeit sind sowohl durch einen komplexen Paketverarbeitungsprozess (Puffermanagement, Scheduling) als auch durch die Verarbeitung von Signalisierungsnachrichten und die Verwaltung von Zustandsinformationen begründet. Die Komplexität steigt dabei linear mit der Anzahl der gleichzeitig aktiven Reservierungen an.

Vorteile	Nachteile
harte und weiche QoS-Garantien	großer Aufwand hinsichtlich des Netzmanagements
hohe Granularität des QoS-Angebotes (Verkehrsaggregation)	schlechte Skalierbarkeit
	Probleme bei der Einführung (Migration)

Tabelle 2-1: Bewertung der IntServ-Architektur

Insbesondere die hohe Komplexität der IntServ-Architektur hat innerhalb der IETF dazu geführt, einen zweiten Ansatz zu entwickeln: DiffServ.

2.2.2 Differentiated Services Architecture (DiffServ)

DiffServ wurde 1998 entwickelt, um eine robuste, einfache und skalierbare QoS-Architektur für IP-Backbones zu schaffen. Die Architektur basiert auf dem Konzept der Dienstklassen und einem relativem Dienstgüteverständnis, im Gegensatz zu absoluten Gütegarantien. DiffServ stellt einen Satz an Konzepten und QoS-Mechanismen zur Verfügung, der die Realisierung verschiedener Übertragungsdienste ermöglicht. Der Standard enthält nur eine Beschreibung des Verhaltens eines einzelnen Knotens (PHB: *Per Hop Behavior*) und keine Spezifikation der Ende-zu-Ende erzielbaren Dienstgüte (PDB: *Per Domain Behavior*). In [Bla98] wurden das Architekturkonzept sowie die Mechanismen zur Verkehrskontrolle vorgestellt.

Die Architektur basiert auf einem Domänen-Konzept, welches an den Domänengrenzen eine statische Verkehrs- und Zugangskontrolle vorsieht und innerhalb dieser völlig zustandslos arbeitet. Auf allen Netzknoten einer Domäne sind für die Realisierung der Dienstklassen folgende Mechanismen vorgesehen:

- **Klassifizierung** (*Classifying*): Unterscheidung verschiedener Anwendungsdatenströme nach Klassen.
- **Scheduling**: Zuweisung von Ressourcen zu einer Klasse. Es bestimmt maßgeblich das PHB.

In den Grenzknoten (*Border-Nodes*) können darüber hinaus weitere Mechanismen eingesetzt werden:

- **Markierung** (*Marking*): Kennzeichnung von Paketen als zu einer Klasse zugehörig (*DiffServ Codepoints*: *DSCP*).
- **Überwachung** hereinkommender (*Policing*) und abgehender (*Shaping*) Verkehre entsprechend der zuvor z.B. mit eine Nachbar-Domäne getroffenen Vereinbarung (*Traffic Conditioning Agreement: TCA*).

Vor dem Einrichten einer Dienstklasse werden Regeln zum Klassifizieren der Datenströme, eine eindeutige Markierung (DSCP), ein bestimmtes Verhalten (PHB) der Netzknoten bei der Paketverarbeitung (*Forwarding*) und die zu verwendenden QoS-Mechanismen an den Domänen-Grenzen definiert.

Zur Identifikation eines Datenstromes können mehrere Felder des MAC-, IP-, TCP-/UDP-Headers verwendet werden (*Multi Field Classification*). Die Markierung eines Datenstromes erfolgt durch das Setzen bestimmter Bits im Paketkopf (*DS-Field*). Welche Bits im IPv4 und IPv6 dafür vorgesehen sind, ist in [Nic02] definiert.

Während des Netzbetriebes klassifizieren die Knoten ein Paket anhand des DSCP, bilden diesen auf ein PHB ab und weisen das Paket den belegten Ressourcen der Dienstklasse zu.

Für die DiffServ-Architektur wurden beispielhaft zwei PHB standardisiert: *Expedited Forwarding* [Dav02] und *Assured Forwarding (AF)* [Hei99]. Das *Expedited Forwarding (EF) PHB* ist für echtzeitkritische Anwendungen vorgesehen, definiert einen DSCP und beschreibt qualitativ das Verhalten eines Netzknotens bei der Paketverarbeitung hinsichtlich der zu erwartenden Verzögerungen und Verluste. Es wird maßgeblich durch das Scheduling und die Auslastung im Netz bestimmt. Aus dem PHB lassen sich notwendige Anforderungen an die technische Realisierung ableiten. Das EF-PHB ist beispielsweise wie folgt definiert: „die Bedienrate eines Ausgangs muss die Ankunftsrate für kurze und lange Zeitintervalle übersteigen, sodass nur geringe Verzögerungen, Jitter und Verluste auftreten“. Wie diese Anforderung sicherzustellen ist, wird nicht definiert.

Das *AF-PHB* definiert im Gegensatz zum EF-PHB mehrere DSCP und beschreibt, welche QoS-Mechanismen (Verkehrskontrolle, Puffermanagement) an welcher Stelle im Netz zur

Realisierung dieser Dienstklassen vorgesehen sind. Wie schon bei dem EF-PHB, werden auch hier keine absoluten QoS-Parameterwerte definiert, sondern nur qualitative Angaben gemacht. Zwischen verschiedenen AF-Klassen enthält der Standard nur relative Aussagen.

Ein Netzbetreiber kann innerhalb seines Netzes die im Rahmen des Standards vorgesehenen Mechanismen prinzipiell frei auswählen und beliebige Dienste realisieren. Schwierigkeiten können dann entstehen, wenn über mehrere Netzbetreiber hinweg eine für den Teilnehmer vorhersagbare Dienstqualität erreicht werden soll. Um diese erzielen zu können, muss ein Netzbetreiber das Ende-zu-Ende Verhalten seines Netzes kennen (*PDB: Per Domain Behavior*), kann dann einen entsprechenden Dienst definieren (*SLS*: *Service Level Specification*) und an den Netzgrenzen Vereinbarungen mit anderen Netzbetreibern treffen (*SLA*: *Service Level Agreement*). Ein *SLA* stellt einen Vertrag zwischen zwei Netzbetreibern dar. Er beinhaltet unter anderem Informationen über Verkehrstyp, Verkehrsvolumen (*TCA*) und Dienstgüte (*SLS*). Der ursprüngliche DiffServ-Ansatz basiert auf einer Vereinbarung, die statischer Natur ist und von der Netzadministration manuell eingerichtet wird.

Bewertung

Die Stärken der DiffServ-Architektur liegen aufgrund der geringen Komplexität sicherlich in der guten Skalierbarkeit, dem geringen Managementaufwand und der leichten Einführbarkeit. Harte QoS-Garantien für Verkehre mit Echtzeitanforderungen können in DiffServ-Netzen allerdings nicht gegeben werden, da keine dynamische Ressourcenreservierung vorgesehen ist. Eine Teilnehmersignalisierung und ein Zugangskontrollverfahren fehlen. Die erzielbare Dienstqualität ist umso eher vorhersagbar, je besser das Verkehrsaufkommen pro Dienstklasse an den Domänengrenzen und die Verkehrsbeziehungen innerhalb der Domänen bekannt sind. Die für Echtzeitverkehre erzielbare QoS wurde bereits mehrfach untersucht. In [TZ01] wurden anhand von Simulationen die Auswirkungen verschiedener Scheduling und Puffermanagement-Mechanismen auf das Ende-zu-Ende Verhalten von Sprachverkehren über mehrere DiffServ-Domänen hinweg analysiert. Ebenso gibt es bereits erste Erfahrungen mit realen Netzen, wie z.B. Messungen in einem Testnetz [Fer00], [FPR00] oder in dem pan-europäischen Backbone GÉANT zeigen [Cam03]. Die Ergebnisse machen deutlich, dass DiffServ-Mechansimen eine erhebliche Verbesserung im Vergleich zum *Best Effort Service* des heutigen Internets darstellen, jedoch harte QoS-Garantien ohne Zugangskontrolle nicht möglich sind.

In den Fällen, in denen für ein Verkehrsaggregat an einem Eingangsnetzknoten viele potentielle Zielnetzknoten in Frage kommen (große räumliche Granularität) und das Verkehrsaufkommen (Anzahl der Verbindungen) über der Zeit betrachtet stark schwankt (große zeitliche Granularität), ist ein effektiver Netzbetrieb kaum möglich. Garantiert ein Netzbetreiber Echtzeitverkehren beispielsweise eine minimale Verzögerung, muss er sein Netz so stark überdimensionieren, dass ein wirtschaftlicher Betrieb schwer erreichbar ist.

Vorteile	Nachteile
leichte Einführbarkeit	geringe Granularität
einfaches Management	Qualitative QoS, keine QoS-Garantie
hohe Freiheiten bei der Dienstklassenspezifikation	kein Ende-zu-Ende QoS-Konzept
gute Skalierbarkeit	Netze schwer zu dimensionieren, geringe erzielbare Netzauslastung

Tabelle 2-2: Bewertung der DiffServ-Architektur

2.3 DiffServ-Erweiterungen

Im Folgenden werden mehrere Ansätze zur Erweiterung der in Abschnitt 2.2.2 beschriebenen DiffServ-Architektur diskutiert. Alle hier vorgestellten Konzepte führen eine dynamische Ressourcenmanagementfunktion mit dem Ziel ein, die Vorhersagbarkeit der Verbindungsqualität für echtzeitkritische Verkehre zu verbessern. Sie basieren auf den Grundsätzen der DiffServ-Philosophie und versuchen dabei die Zustandsinformationen innerhalb des Netzes möglichst gering zu halten.
Zunächst werden zentrale Ressourcenmanagement-Ansätze vorgestellt. Danach wird eine verteilte Architektur beschrieben, welche dieses Ziel mit Hilfe eines integrierten Ansatzes von Routenoptimierung, Netzdimensionierung und einem verteilten Ressourcenmanagement zu erreichen versucht. Darüber hinaus werden weitere verteilte Zugangskontrollverfahren erläutert, die auf einer schlanken Signalisierung und einer messbasierten Überwachungsfunktion der Lastsituation im Netz beruhen.

2.3.1 Ansätze mit zentraler Zugangskontrolle

In diesem Abschnitt werden zwei Architekturen diskutiert, die eine dynamische Ressourcenmanagementfunktion in Form eines zentralen Servers im Netz einführen.

2.3.1.1 Bandwidth Broker

Die IETF hat ein Implementierungsbeispiel für die DiffServ-Architektur *"A Two Differentiated Services Architecture for the Internet"* standardisiert [NJZ99] (Klasse-B). In der Architektur sind zwei Dienstklassen vorgesehen: ein *Premium* (*PS*) und ein *Assured* Service (*AS*). Sie werden durch einen einfachen Klassifizierer (2 Bit: *P, A*) und zwei Puffer realisiert, die von einem Priority-Scheduler bedient werden. Als Verkehrsprofil für *PS*-Verkehre wird eine Spitzenbitrate und für *AS*-Verkehre eine mittlere Rate sowie eine maximale Burstgröße vereinbart. Die Verkehre werden am Netzzugang durch Filter überwacht. Im Fall des *PS* besitzt der Filter eine maximale Burstlänge von nur einem Paket (*peak rate allocation*).

Das Besondere an diesem Beispiel ist, dass es eine Zugangskontrolle auf Verbindungsebene und eine dynamische Belegung von Ressourcen vorsieht. Dazu werden Ressourcenmanagement-Agenten, sogenannten *„Bandwidth Broker"* BB und eine Signalisierung pro IP-Flow zwischen Teilnehmer und BB sowie zwischen benachbarten BB eingeführt. Aufgabe eines BB ist, den Teilnehmerzugang zu den Ressourcen einer Domäne dynamisch zu steuern. Dazu führt er folgende Operationen durch:

- Bearbeitung der Reservierungsnachrichten,
- Durchführung der Zugangskontrolle und
- Konfiguration der Zugangsknoten (*Classifier, Policer, Marker, Shaper*).

Eine Reservierungsnachricht enthält Angaben zur gewünschten Dienstklasse (PS oder AF), Übertragungsrate, maximalen Burstgröße, Zeitpunkt und Dauer. Bei der Zugangskontrolle werden zunächst die Identität und das Teilnehmerprofil des Senders überprüft. Danach wird kontrolliert, ob die verfügbare Bandbreite (SLS) auf dem Link zur Nachbar-Domäne ausreicht. Falls ja, werden zum einen die verfügbare Bandbreite um den geforderten Betrag reduziert und zum anderen die Verbindungs-Daten (Flow-Spec) gespeichert. Falls nein, kann er bei einem BB in der Nachbar-Domäne zusätzliche Ressourcen (Kapazitäten) anfordern. Nach erfolgter Reservierung konfiguriert der BB den Zugangsknoten (*Classifier, Policer, Marker*) mit den Verbindungsparametern der zugelassenen Verbindung.

Das hier vorgestellte BB-Konzept basiert auf der Annahme, dass der Anteil des PS-Verkehrs am gesamten Netzverkehr sehr gering (< 10%) ist und dass die Netze innerhalb einer

DiffServ-Domäne ausreichend dimensioniert sind. Es wird davon ausgegangen, dass die Kosten für Leitungskapazitäten innerhalb einer Domäne niedrig, für Leitungskapazitäten zu Nachbar-Domänen hingegen hoch sind. Daher ist eine Zugangskontrolle nur an Domänengrenzen vorgesehen.

Internet 2: QBone

QBone steht für die praktische Umsetzung des BB-Konzeptes. Der IETF BB-Standard sieht zwar einen zentralen Server vor, spezifiziert ihn jedoch nicht näher. Um ein gemeinsames Inter-Domain QoS-Konzept zu erarbeiten, dieses prototypisch zu realisieren und in einem Feldversuch zu testen wurde das Internet2-Projekt Ende 1998 gegründet. Es handelt sich dabei um eine Partnerschaft von ca. 130 amerikanischen Universitäten und ca. 40 Firmen. Es wurde ein Testbett über mehrere Campus-, nationale Behörden- und internationale Forschungs-Netze hinweg aufgebaut, um praktische Erfahrung beim Aufbau und Betrieb eines solchen Netzes zu gewinnen.

Das Inter-Domain Konzept basiert auf einem Realisierungsentwurf von *Bandwidth-Brokern BB* [QB99]. Jeder BB regelt den Verkehrsfluss in einer DiffServ-Domäne. Die QBone Architektur definiert in Anlehnung an den „*Virtual Leased Line Service*“ von Van Jacobson [NJZ99] einen „*QBone Premium Service*“, der eine Spitzenbitratenreservierung über mehrere Domänen hinweg vorsieht. Die Dienstklassenspezifikation sieht möglichst geringe Verluste, minimale Verzögerung und einen begrenzten Jitter vor.

Eine Reservierung wird von einem Terminal, einem Applikationsserver oder einem Nachbar-BB angestoßen. Im Gegensatz zum Standard ermittelt der BB zunächst die Verfügbarkeit der Ressourcen innerhalb seiner Domäne und überprüft dann, ob durch die Reservierung *SLS* mit Nachbar-Domänen verletzt werden. Danach konfiguriert er *Policer* im Ingress-Router und *Shaper* im Egress-Router. Für den Zugriff auf die Netzknoten werden standardisierte Verfahren wie COPS oder SNMP verwendet. Ein Zugangskontrollverfahren ist noch nicht definiert. Aus der Mission der Internet-2 Gemeinde geht jedoch hervor, dass die Einhaltung der Dienstklassenspezifikation mit geringen Mitteln (Komplexität) erreicht werden soll.

Zur Verifizierung der QoS-Ziele wird eine Messarchitektur definiert. Während des laufenden Betriebes soll anhand von aktiven (Senden von Test-Paketen) und passiven (Mithören des Verkehrs) Messungen die Einhaltung der Dienstklassenspezifikation überprüft werden. Die Ergebnisse der Messungen können einem BB als Grundlage für die Zugangskontrolle dienen.

Die BB-Architektur besitzt folgende Eigenschaften:

- Zentraler Ansatz
- Zugangskontrolle im BB
- Verkehrskontrolle (Policies) in den Edge-Routern
- standardisierte Konfigurationsschnittstelle zu den Routern
- proprietäre Schnittstelle zu aktiven oder passiven Messeinheiten in den Routern
- Teilnehmerschnittstelle nicht näher spezifiziert
- kein Realisierungsvorschlag für den *PS*

Die Schnittstelle zu den Terminals wird im Standard nicht näher spezifiziert. Die Internet2-Architektur kann wie folgt bewertet werden:

Vorteile	Nachteile
gute Skalierbarkeit	keine harte QoS-Garantie
	geringe Granularität
Inter-Domain Konzept (abschnittsweise)	vage Dienstklassenspezifikation hinsichtlich des *PDB*
	eingeschränkte Einführbarkeit

Tabelle 2-3: Bewertung der BB-Architektur

2.3.1.2 Resource Reservation Agents (RRA)

Das RRA-Konzept wurde von O. Schelen und S. Pink veröffentlicht [ScP98] und führt in jede DiffServ-Domäne einen zentralen Server (RRA) ein, der auf allen Links des Nutzdatenpfades eine Zugangskontrollfunktion durchführt. Das Zugangskontrollverfahren ist parameterbasiert und unterstützt Vorabreservierungen [SP98]. Dafür benötigt der RRA Informationen über Topologie und Konfiguration des Netzes. Er partizipiert passiv am Routing-Prozess und kann durch das Mithören der Routing-Nachrichten (z.B. *Link State Advertisement*) die notwendigen Topologieinformationen gewinnen. Die Konfigurationen der Links (Kapazität) erhält er durch gezieltes Abfragen der Router mit Netzmanagementprotokollen (z.B. SNMP). Es werden nur Schicht-3 Topologien betrachtet.
Um über mehrere Domänen eine Reservierung aufbauen zu können, ist ein sehr einfaches Protokoll zwischen den RRA-Instanzen vorgesehen. Die Nachrichten enthalten lediglich ein Quell- und Zieladressenpaar und einen Ratenparameter. Die Annahmeentscheidung wird pro Link auf der Basis des Ratenparameters, der Summe aller Raten der bereis auf einem Link angenommenen Reservierungen und der Linkkapazität getroffen. Die Ermittlung des Ressourcenbedarfes liegt allein bei der Quelle. Das Konzept sieht lediglich eine Dienstklasse vor (*Priority*). Eine Dienstgütespezifikation sowie der Nachweis der erzielbaren QoS fehlen.

Die RRA-Architektur besitzt folgende Eigenschaften:

- Zentraler Ansatz
- Zugangskontrolle im RRA
- Ende-zu-Ende Reservierung für alle Links (OSI-Schicht 3)
- proprietäre Schnittstelle zu den Terminals
- standardisierte Schnittstelle zu den Routern

Die RRA-Architektur kann wie folgt bewertet werden:

Vorteile	Nachteile
Ende-zu-Ende Reservierung	keine harte QoS-Garantie
einfaches Zugangskontrollver-fahren	kein Verfahren zur Berechnung des Ressourcenbedarfes
	keine Dienstgütespezifikation
Skalierbarkeit gegeben	eingeschränkte Einführbarkeit

Tabelle 2-4: Bewertung der RRA-Architektur

2.3.2 Ansätze mit verteilter Zugangskontrolle

2.3.2.1 Ende-zu-Ende Reservierung (SCORE)

Ion Stoica, Steve Shenker und Hui Zhang haben für DiffServ-Netze ein Architektur SCORE (*Scalable Core*) [SSZ98] mit dem Ziel entwickelt, den Übertragungsdienst einer zustandsbasierten Netzarchitektur mit einem weitestgehend zustandslosen Core-Netz nachzubilden. Dazu wurde in den Core-Routern eine Technik mit dem Namen DPS (*Dynamic Paket State*) eingeführt. DPS ist ein Verfahren, bei dem Ingress-Router Zustandsinformationen eines IP-Flows in den IP-Header (*TOS-, IP-Fragment-Field*) kodieren. Die Zustandsinformationen werden zusammen mit den Nutzdaten übertragen und von den Netzknoten entlang des Datenpfades ausgewertet. Zustandsinformationen pro IP-Flow werden somit nur im Ingress-Router und nicht im Core-Router gespeichert.
Der Teilnehmer signalisiert einen Reservierungswunsch mit RSVP. Die RSVP-Nachrichten werden transparent zwischen den Edge-Routern ausgetauscht. Ist die Reservierungsnachricht des Empfängers wieder am Ingress-Router angekommen, stößt dieser innerhalb der Domäne eine Reservierung an. Dazu wird eine proprietäre Signalisierung verwendet, welche alle Router entlang des Datenpfades veranlasst, eine Zugangskontrolle auszuführen. Die

Zugangskontrollfunktion ist sehr einfach und wird anhand einer zu reservierenden Rate r_{neu} durchgeführt. Jeder Router benötigt nur ein aggregiertes Wissen über alle bereits aktiven Reservierungen ($\Sigma r_{alt} + r_{neu} < C$). Nach erfolgreicher Ende-zu-Ende Reservierung werden die Ingress-Router mit den Verbindungsparametern konfiguriert, die für das Verarbeiten der Nutzdatenpakete in den Core-Routern benötigt werden. Sie werden vom Ingress-Router in den IP-Header eines jeden Paketes geschrieben. In [SZ99] wird das Scheduling Verfahren *Core-Jitter-Virtual-Clock* eingeführt, das von den Header-Informationen gesteuert wird und eine Ende-zu-Ende Garantie hinsichtlich der Verzögerung und der Bandbreite ermöglicht.

Die SCORE-Architektur besitzt folgende Eigenschaften:

- Verteilter Ansatz
- Zugangskontrolle in den Routern
- proprietäres Scheduling-Verfahren
- Ende-zu-Ende Reservierung auf allen Links
- Manipulation des IP-Headers
- proprietäres Reservierungsprotokoll zwischen Core-Routern
- modifiziertes RSVP auf der Teilnehmerschnittstelle

Die SCORE-Architektur kann wie folgt bewertet werden:

Vorteile	Nachteile
harte QoS-Garantie	geringe Granularität (Verkehrsklasse)
Skalierbarkeit gegeben (kein Management pro Flow im Kernnetz, einfaches Zugangskontrollverfahren)	erhebliche Probleme bei der Einführung (Migration)

Tabelle 2-5: Bewertung der SCORE-Architektur

2.3.2.2 Zugangskontrolle am Eingang (AQUILA)

In dem europäischen IST-Projekt AQUILA [Aqu00], [Eng03], wurde basierend auf DiffServ eine QoS-Architektur definiert und entwickelt, die im Zugangsbereich eine Teilnehmer-Netz Schnittstelle einführt. An den Eingängen findet in ACA-Instanzen (*Admission Control Agent*) eine Zugangskontrolle statt, die parameter- und/oder messbasiert arbeitet [Bran02], [Ba01]. Eine zentrale Steuerungsinstanz RCA (*Ressource Control Agent*) verfügt über Messwerte an verschiedenen Punkten des Netzes und begrenzt den maximalen Verkehrszufluss pro Eingang und Verkehrsklasse (Budget). Die Budgets werden von den ACA verwaltet und vom RCA an die Verkehrsverhältnisse im Kernnetz dynamisch angepasst. Das Reservierungsprotokoll läuft zwischen den Endgeräten und den Edge-Routern. In [Ba01] wird ein Scheduling-Verfahren definiert, mit dem man nur eine Dienstklasse für echtzeitkritische Verkehre realisieren kann, die unabhängig von den Paketankünften der übrigen Verkehrsklassen ist.

Die AQUILA-Architektur besitzt folgende Eigenschaften:

- Verteilter Ansatz
- Standard Scheduling-Verfahren
- Zugangskontrolle am Domänen-Eingang (parameter- /messbasiert)
- proprietäre Schnittstelle zu den Ingress-Routern
- proprietäre Schnittstelle zu den Teilnehmern
- modifiziertes RSVP auf der Teilnehmerschnittstelle

Die AQUILA-Architektur kann wie folgt bewertet werden:

Vorteile	Nachteile
gute Skalierbarkeit (keine Signalisierung und kein Management pro Flow im Kernnetz)	keine harte QoS-Garantie
	sehr geringe Granularität (Verkehrsklasse)
	Probleme bei der Einführung (Migration)

Tabelle 2-6: Bewertung der AQUILA-Architektur

2.3.2.3 Zugangskontrolle am Ausgang (Egress-AC)

Das folgende Verfahren von Edward Knightly *et al.* [Kni01] basiert auf einer ähnlichen Architektur wie die vorangegangene Zugangskontrollfunktion. Es verzichtet auf Zustände innerhalb der Domäne und sieht nur am Egress-Router eine Zugangskontrolle vor. Zur Signalisierung von Reservierungsanfragen verwendet es ein modifiziertes RSVP, dessen Meldungen jedoch nur von den Egress-Routern verarbeitet werden. In den Ingress-Routern wird in den Header der IP-Pakete (*TOS-*, *IP-Fragment-Field*) eine Kennzeichnung der Verkehrsklasse, eine Kennung des Ingress-Routers, eine Paketsequenznummer und ein Zeitstempel eingefügt. Anhand dieser Informationen kann der Egress-Router Verzögerungen der Pakete ermitteln und Verluste feststellen. Dazu werden die Systemzeiten der Ingress- und Egress-Router über NTP (*Network Time Protocol*) synchronisiert. Der Egress-Router kennt von jedem bereits übertragenen Paket einer Klasse und eines Pfades (Ingress-, Egress-Paares) den Zeitpunkt, zu dem es an der Domäne angekommen ist und wann es die Domäne wieder verlassen hat.
Die ganze Domäne kann somit als eine *Black-Box* modelliert werden. Die Bedienrate des ankommenden Verkehrs sowie die Verlustwahrscheinlichkeit des Systems können am Ausgang rückwirkend bestimmt werden. Am Ausgang eines Netzes kann man pro Zeitintervall den minimalen und maximalen Durchsatz (Bedienrate) sowie die Varianz messen. Aufgrund von Schwankungen der längs und quer zum betrachteten Pfad laufenden Verkehre, ergibt sich ein zeitlicher Verlauf der Bedienrate. Auf der Basis dieser statistischen Daten über die vergangene und aktuelle Verkehrssituation einerseits sowie den Verbindungsdaten der neuen Reservierung andererseits, kann eine Zugangskontrolle durchgeführt werden. Ein entsprechendes Verfahren wird in [CK00] vorgeschlagen, welches auf CSFQ-Schedulern [SSZ98] basiert. Aufgrund der gemessenen „Vergangenheit“ hinsichtlich der Lastsituation im Netz ermöglicht es allerdings nur unzuverlässige Prognosen hinsichtlich der in Zukunft zu erwartenden QoS. Das Signalisierungsverfahren ist gegenüber RSVP dadurch vereinfacht, dass eine Signalisierung nur für den Aufbau einer Reservierung benötigt wird. Ein Abbau wird hingegen implizit durch das messbasierte Zugangskontrollverfahren ermittelt.

Die Egress-AC Architektur besitzt folgende Eigenschaften:

- Verteilter Ansatz
- Manipulation des IP-Headers im *Ingress-Router*
- Zugangskontrolle am Domänen-Ausgang (messbasiert, aktiv)
- proprietäres Scheduling-Verfahren
- Architektur zur Synchronisation der Systemzeiten am Zugang (*Edge-Router*)

Die Egress-AC Architektur kann wie folgt bewertet werden:

Vorteile	Nachteile
kein zusätzlicher Verkehr durch Probenpakete	reaktives Zugangskontrollverfahren, keine QoS-Garantie
Skalierbarkeit gegeben (keine Signalisierung oder per-Flow Management im Kernnetz)	geringe Granularität (Verkehrsklasse, Pfad)
	Probleme bei der Einführung (Migration)

Tabelle 2-7: Bewertung der Egress-AC Architektur

2.3.2.4 Zugangskontrolle im Endgerät (EP-AC)

Neben der Zugangskontrolle im Netz (*Bandwidth Broker*, *Egress-Router*) gibt es auch Architekturen, die eine solche Funktion im Endgerät vorsehen. Sie arbeiten messbasiert und verwenden dabei Testpakete, die vor der Nutzdatenübertragung gesendet werden. Sie kontrollieren die Paketverluste einer Verkehrsklasse und erlauben lediglich Übertragungsdienste mit weicher QoS-Garantie, vergleichbar dem *Controlled Load Service* von IntServ.
In der Regel handelt es sich bei den Endgeräten um Host-Rechner, die durch das Aussenden von Probenpaketen mit der gewünschten Rate und der entsprechenden Markierung die Lastsituation im Netz überprüfen. Dies geschieht, indem sie die aktuelle Verlustwahrscheinlichkeit der Probenpakete feststellen. Anhand eines zuvor eingestellten Schwellenwertes entscheiden sie dann, ob die neue Verbindung zugelassen werden darf [Ele00].
Das Verfahren [BCP00] unterscheidet sich von dem vorherigen dahingehend, dass hier die Probenpakete in einer Verkehrsklasse mit niedrigerer Priorität gesendet werden (out-band), als die zugelassenen Nutzdatenpakete. Dadurch trifft dieses Verfahren eine defensivere Zugangsentscheidung und kann eine sicherere QoS-Vorhersage machen als das Vorherige.

Die QoS-Architektur mit einer Zugangskontrolle im Endgerät wird genauso bewertet wie der vorherige Ansatz (Egress-EP). Dieser Ansatz besitzt einen geringfügigen Vorteil hinsichtlich der Einführbarkeit, da die Komplexität der Zugangskontrolle vom Netz in die Endgeräte ausgelagert wird und keine IP-Header manipuliert werden müssen.

2.3.3 Traffic Engineering (TEQUILA)

Neben dem zentralen Ansatz des BB wurde in einem Projekt Namens TEQUILA ein erweitertes Konzept erarbeitet. TEQUILA steht für „*Traffic Engineering for Quality of Service in the Internet, at Large Scale*“ und ist ein IST-Projekt (*Information Society Technologies*) der EU (1998-2002) [TEQ98]. TEQUILA basiert auf dem DiffServ-Standard und legt den Schwerpunkt auf einen optimierten Netzbetrieb. Dazu wird eine Architektur spezifiziert, welche Methoden des *Traffic Engineerings* einführt und mit einem dynamischen Policy- und SLS-Management verbindet. Die Ziele sind eine optimale Auslastung des Netzes sowie das Erreichen quantitativer Ende-zu-Ende QoS-Garantien. Die Neuerungen gegenüber der DiffServ-Architektur bestehen in einem integrierten Betreiberkonzept, in dem folgende Aufgaben miteinander gekoppelt werden [Tri01]:

- Dimensionierung der Verkehrsklassen auf Basis der angenommenen SLS-Verträge (Verkehrsmatrix)
- Berechnung der Routen
- Einstellen der Routen und PHB (Puffer-, Scheduling-Parameter)
- Überwachen der SLS und dynamisches Anpassen der Routen und PHB

Das System arbeitet nicht nur statisch, sondern besitzt ein dynamisches SLS-Management (flow-basiert). Die Zugangskontrolle findet am Zugang der Domäne statt und beruht auf

einem messbasierten Verfahren. Die Architektur verwendet ein Netzüberwachungssystem [Asg02], welches mehrere Komponenten mit Zustandsinformationen (Netzknoten, Ende-zu-Ende Verhalten) versorgt und dadurch eine Anpassung der Routen (Routenmanagement) und PHB-Konfigurationen (Ressourcenmanagement) ermöglicht.

Die Themenschwerpunkte des Projektes sind im Einzelnen:

- Policy-Management
  - Spezifikation (Beschreibungssprache)
  - Speicherung (Datenmodellierung)
  - Verteilung
  - Verarbeitung in den Architekturkomponenten
- SLS-Management:
  - Kundenregistrierung (statisch)
  - Teilnehmersignalisierung (dynamisch)
  - Zugangskontrolle
- Netzüberwachung:
  - verteiltes, echtzeitfähiges Messsystem mit Knotenmonitoren, die von einem zentralen Netzmonitor gesteuert werden
- Traffic Engineering:
  - Netzdimensionierung
  - Einrichten von festen Pfaden, z.B. MPLS (statisch)
  - Routenmanagement (dynamisch)
  - Ressourcenmanagement (dynamisch)

Die TEQUILA-Architektur besitzt folgende Eigenschaften:

- Verteilter Ansatz
- Zugangs- und Verkehrskontrolle (Policies) in den Edge-Routern
- QoS-Routing und/oder Pfad-Management innerhalb der Domäne
- Netzüberwachungssystem
- standardisierte Schnittstellen zu den Netzknoten (SNMP, COPS)
- proprietäre Schnittstelle zu den Teilnehmern (modifiziertes RSVP)

Die TEQUILA-Architektur kann wie folgt bewertet werden:

Vorteile	Nachteile
weiche QoS-Garantie	harte QoS-Garantie nur in speziellen Fällen
integriertes Policy-, SLS-und TE-Konzept	hohe Komplexität
hohe erzielbare Netzauslastung	Probleme bei der Einführung

Tabelle 2-8: Bewertung der TEQUILA-Architektur

2.4 Weitere Ansätze

Neben den bisher diskutierten Ressourcenmanagement-Architekturen gibt es Projekte wie z.B. GLOBUS [Fos99], DARWIN [Cha98] oder CADENUS [Cor03] in denen Systemarchitekturen definiert werden, die den Begriff „Ressourcen“ weiter fassen als die bisherigen Ansätze. Sie sind keine reinen QoS-Architekturen, sondern unterstützen den Teilnehmer mit Funktionen, die bei den anderen Ansätzen als Teil der Dienststeuerung oder generell als Aufgabe des Dienstanbieters bei der Realisierung des Teilnehmerzugangs angesehen werden. Beispiele für solche Funktionen sind: Auswahl eines Dienstbetreibers, Verwaltung des Teilnehmerzugangs, Vermittlung von Diensten, Auswahl von dienstspezifischen Ressourcen (*Computation, Storage*), Abgleich der Fähigkeiten der Endgeräte (*Capabilities*) und Auswahl der an der Dienstrealisierung beteiligten Netzbetreiber. In diesem Abschnitt werden die Systemarchitekturen GLOBUS und DARWIN vorgestellt.

Der Schwerpunkt der nachfolgenden Betrachtungen liegt auf der jeweiligen QoS-Architektur als Teil der gesamten Systemarchitektur.

2.4.1 GLOBUS-Projekt

GLOBUS steht für ein Forschungsprojekt, an dem eine Vielzahl von Universitäten (Deutschland: Jülich, USA: Michigan, Chicago, Illinois, Southern California, etc.), Behörden (NASA, DARPA, NSF) und Firmen (IBM, Microsoft, Cisco) beteiligt sind. Die Ziele des Projektes lassen sich unter dem Begriff „*Grid Computing*“ zusammenfassen und betreffen sowohl den Aufbau einer Infrastruktur, als auch das Design und die Entwicklung von Anwendungen. Eines der betrachteten Anwendungsfelder für Grid-Computing ist das Management von Ressourcen. Es umfasst einheitliche Mechanismen zur Bezeichnung (*naming*), zum Auffinden (*locating*) und Belegen (*allocating*) von Rechen-, Speicher- und Transportkapazitäten in verteilten Systemen. Daraus wurde die QoS-Architektur GARA (*Global Architecture for Reservation and Allocation*) entwickelt.

GARA [Fos99] basiert auf keiner bestimmten Infrastruktur und ist daher grundsätzlich für DiffServ und IntServ Netze anwendbar. Sie besitzt einen zentralen *QoS-Resource Manager* (QoS-RM), der den Applikationen eine API (*Application Programming Interface*) zur Verfügung stellt, um Reservierungen zu etablieren, zu modifizieren oder zu terminieren. Der RM ermittelt und konfiguriert alle an einer Reservierung beteiligten „*Local Resource Manager*“ LRM, welche in den Routern angesiedelt sind. Die LRM üben eine Zugangskontrollfunktion aus, verwalten die Reservierungszustände und führen lokale Operationen durch. Diese Operationen umfassen die Konfiguration des Routers mit Verbindungsinformationen (*flow specification*) und Reservierungsinformationen (mittlere Rate, maximale Burstlänge) für die Verkehrskontrolle (Aktoren: *entities for classification and policing*). Außerdem liefern die LRM dem QoS-RM Statusinformationen des Routers, die von Sensoren angelegt werden (z.B. *number of dropped conforming packets*). Bei der Realisierung der RM wurden nur Cisco-Router (Typ: 7505) verwendet, die als Scheduling WFQ verwenden. Ein LRM nimmt die gewünschte Konfiguration über das proprietäre CLI (*Command Line Interface*) von Cisco vor.

Die GARA-Architektur besitzt folgende Eigenschaften:

- Zentraler Ansatz (keine Domänenbildung, nur ein Betreiber)
- Zugangskontrolle in den Routern und Endgeräten
- Vorabreservierungen
- proprietäre Schnittstelle zu den Terminals (inklusive *Feedback*-Mechanismus zu den Terminals)
- proprietäre Schnittstelle zu den Routern (inklusive Überwachung der Puffer in den Routern)

2.4.2 DARWIN-Projekt

DARWIN ist ein Forschungsprojekt mit dem Titel "*Resource Management for Application-Aware Networks*". Es läuft an dem Institut „ *School of Computer Science“* der Carnegie Mellon University (CMU), USA unter der Leitung der Ass. Professoren Peter Steenkiste und Hui Zhang. Gesponsert wird es von der Behörde für „*Defense Advanced Research Projects*“ (DARPA).
DARWIN beinhaltet eine QoS-Architektur und wurde für ein Multi-Provider Szenario entwickelt und bietet einen Satz an Mechanismen, die flexibel an die Bedürfnisse von beliebigen Anwendungen anpassbar sind [Cha98]. Ähnlich wie das Globus-Projekt sieht auch DARWIN einen zentralen Ressourcenmanagement Server (Service Broker „Xena“) vor. Xena nimmt die Reservierungsanfragen der Applikationen entgegen, ermittelt den Ressourcenbe-

darf (*Computation, Storage, Communication*) wählt einen oder mehrere Netzbetreiber aus, bindet, falls erforderlich, z.B. Transkodierer in den Verbindungsaufbau ein und stößt die Reservierung in den Netzknoten an. Dazu werden ein Protokoll (*Beagle*) und lokale Ressourcenmanager (LRM) in den Netzknoten definiert.

Das Konzept sieht vor, dass sich alle Applikationen bei einem Xena-Server mit ihren Fähigkeiten (*Capabilities*) registrieren. Reservierungsanfragen werden nur von registrierten Applikationen entgegen genommen und beinhalten den Ressourcenbedarf der Sitzung in Form eines Verbindungsgrafen. Xena interpretiert diesen Grafen und erkennt mögliche Freiheitsgrade bei der Ressourcenauswahl oder Inkompatibilitäten der verwendeten *Codecs* (insbesondere bei Mehrteilnehmerkonferenzen). Xena versucht nun, den Verbindungsgraphen mit den zur Verfügung stehenden Ressourcen zu realisieren und dabei die bestmögliche Qualität für den Anwender oder die geringsten Kosten für den Netzbetreiber zu erzielen.

Xena verwendet dazu ein grobes Abbild der Netztopologie (inklusive Lastzuständen) und greift auf einen internen Ressourcenregistrierungsdienst zum Auffinden von speziellen Servern, z.B. zum Transkodieren oder zum Mischen der Medienströme zu. Darüber hinaus ist ein externer Ressourcen-Erkennungsdienst zum Auffinden von Diensten und den zugehörigen Servern (z.B. *Simulation-, Content-Server*) vorgesehen. Die Verkehrskontrolle in den Netzknoten wird mit *Beagle* gesteuert. Xena konfiguriert einen Knoten im Netz mit dem Verbindungsgrafen und speziellen Anweisungen (*Policies*) hinsichtlich der Klassifizierung und des Schedulings. Dieser Knoten konfiguriert daraufhin mit Hilfe von Beagle alle anderen Netzknoten, die an der Kommunikation beteiligt sind. In jedem Netzknoten existiert ein LRM, der anhand des Verbindungsgrafen die Paketverarbeitung steuert und Ressourcen reserviert. Er wird dabei von einem dynamischen QoS-Management (*Delegate*) unterstützt. Ein *Delegate* ist ein Programmcode (JAVA), der z.B. von einem Dienstbetreiber auf die Netzknoten geladen wird („*Active Networks*“), um dort lokal Entscheidungen zu fällen (*Customized Runtime-Management*) [Ch98]. Er überwacht die Pufferzustände und führt eine Anpassung der Ressourcenvergabe durch, indem er lokale Policing- und Scheduling-Einheiten aktiviert bzw. adaptiert.

Für die Implementierung wurden keine kommerziellen Router sondern PC-basierte Systeme mit NetBSD verwendet. Als Scheduling-Verfahren kommt ein H-FSC (*Hierarchical Fair Service Curve*) zum Einsatz [SZE97], das sowohl harte als auch relative QoS-Garantien ermöglicht. Für echtzeitkritische Verkehre werden eine minimale Bedienrate und eine maximale Verzögerung garantiert. Bei der Realisierung des Reservierungsprotokolls *Beagle* wurde ein modifiziertes RSVP verwendet.

Die DARWIN-Architektur besitzt folgende Eigenschaften:

- Hierarchischer Ansatz auf Dienst- und Ressourcenebene
- Reservierung getriggert von Terminal oder Xena (*Third-Party Setup*)
- Zugangskontrolle in den Routern
- H-FSC Scheduling ermöglicht QoS in unterschiedlichen Granularitäten (Aggregat, Flow)
- QoS-Überwachung und Adaption von Policies in den Routern
- proprietäre Schnittstelle zu den Terminals (modifiziertes RSVP)
- proprietäre Schnittstelle zu den Routern

2.5 Diskussion

Zum Abschluss dieses Kapitels werden die vorgestellten QoS-Architekturen miteinander verglichen. Sie lassen sich hinsichtlich ihrer QoS-Garantie grob in drei Klassen einteilen: „**hart“, „weich“** und **„relativ“**.

Architekturen mit **harter** QoS-Garantie zeichnen sich durch eine quantitative Dienstgütespezifikation mit paketbezogenen Grenzwerten aus. Letztere dürfen nicht oder höchstenfalls nur geringfügig und kurzzeitig (ms) überschritten werden. Architekturen mit **weicher** QoS-Garantie besitzen ebenfalls eine quantitative Dienstgütespezifikation mit Grenzwerten, die allerdings nur im statistischen Mittel garantiert werden. Architekturen mit **relativer** QoS-Garantie arbeiten lediglich mit qualitativen Dienstgütespezifikationen.

In Tabelle 2-9 werden die verschiedenen Verfahren miteinander verglichen. Dazu werden Architekturklassen gebildet und alle QoS-relevanten Architekturmerkmale einander gegenübergestellt. Aus der Tabelle ist ersichtlich, welche QoS-Mechanismen an welcher Stelle im Netz (Zugang, Kern) angesiedelt sind und welche Granularität (Paket, Flow, Aggregat) sie besitzen.

Aus Gründen der Übersichtlichkeit wurde das Verfahren EP-AC aus Abschnitt 2.3.2.4 nicht aufgelistet. Es lässt sich auf das Verfahren Egress-AC abbilden, wenn man die Methoden aus dem Netzzugangsknoten in das Endgerät verlagert.

QoS-Garantie	QoS-Architektur	QoS-Mechanismen										Ende-zu-Ende Kontrolle
		Netzzugang					Kernnetz					
		Klassifizieren	Policing	AC	Scheduling	Monitore	Klassifizieren	Scheduling	Shaping	AC	Monitore	
hart	IntServ GS	MF	Flow	Flow (pb)	Flow	-	Flow-Label	Flow	Flow	Flow (pb)	-	✔ (+ Terminals)
	Darwin	MF	Flow	Flow (pb)	Flow	passiv	Flow	Flow	Flow	Flow (pb)	aktiv	✔
	SCORE	MF	Flow	Flow (pb)	Paket	-	Flow (DPS)	Paket	-	Flow (pb)	-	✔
weich	RRA	MF	Flow	Flow (pb)	PHB	-	DSCP	PHB	PHB	Flow (pb)	-	✔
	AQUILA	MF	Flow	Flow (pb)	PHB	passiv	DSCP	PHB	-	-	passiv	nur an Domänengrenzen
	Egress-AC	MF	Aggr.	Flow (mb)	PHB	aktiv (Pfad)	DSCP	PHB	PHB	-	-	✔ (Pfad)
	TEQUILA	MF	Flow, Aggr.	Flow, Aggr. (mb)	PHB, LSP	aktiv (Pfad)	DSCP, LSP	PHB, LSP	PHB, LSP	-	passiv (PHB)	✔ (Pfad: LSP)
relativ	BB	MF	Flow, Aggr.	Flow, Aggr. (k.A.)	PHB	-	DSCP	PHB	PHB	-	-	nur an Domänengrenzen
	DiffServ-AS	MF	Aggr.	-	PHB	-	DSCP	PHB	PHB	-	-	-

AC: Admission Control
DPS: Dynamic Packet State
DSCP: DiffServ Code Point
LSP: Label Switched Path
MF: Multi Field
pb: parameterbasiert
mb: messbasiert
PHB: Per Hop Behavior

Tabelle 2-9: Gegenüberstellung der QoS-Mechanismen

Es wird ersichtlich, dass Architekturen mit harter QoS-Garantie ein flowbasiertes Ende-zu-Ende Reservierungsverfahren und ein Scheduling pro Flow oder Paket sowohl im Zugangsbereich als auch im Kernnetz besitzen. Sie unterscheiden sich lediglich in den eingesetzten Verfahren (Scheduling, Zugangskontrolle) und Protokollen. Ferner fällt auf, dass

alle Architekturen mit weicher QoS-Garantie im Kernnetz Verkehrsaggregate bilden und eine Zugangskontrolle lediglich im Zugangsbereich durchführen. Häufig werden bei diesen Ansätzen die Zugangskontrollfunktionen durch verteilte Messsysteme unterstützt, die einen Einblick in den Zustand des Kernnetzes geben. Die Architekturen mit relativer QoS-Garantie unterscheiden sich von den vorherigen dahingehend, dass entweder keine Zugangskontrolle vorgesehen ist, oder diese nur am Netzzugang aufgrund lokaler Zustandsinformationen durchgeführt wird.

Schwierigkeiten bereitete die Klassifizierung und Bewertung des RRA-Verfahrens aus Abschnitt 2.3.1.2. Da weder die zu erreichende Dienstgüte noch eine Methode zur Ressourcenberechnung definiert wurde, ist eine genaue Klassifizierung des Verfahrens nicht möglich. Anhand der in [ScP98] publizierten Informationen wird davon ausgegangen, dass allein die Quelle die Höhe der zu reservierenden Ressourcen bestimmt und dadurch keine harten QoS-Garantien möglich sind.

In Tabelle 2-10 erfolgt eine Bewertung der jeweiligen Architekturen. Als Kriterien werden festgelegt:

- **QoS-Garantie**. Sie betrifft den Grad der Zusicherung (*Assurance*), mit der die vereinbarte Dienstgüte eines Flows eingehalten werden kann.
- **Granularität**, mit der ein Teilnehmer seine individuellen QoS-Anforderungen an das Netz stellen kann und betrifft die Anzahl der gleichzeitig realisierbaren Dienstgüteklassen sowie das Spektrum möglicher Dienstgütespezifikationen.
- **Komplexität**. Sie wird vor allem durch QoS-Mechanismen bestimmt, die Flow-basiert arbeiten. Sie betrifft die Anzahl der Verarbeitungsschritte sowohl bei der Paketverarbeitung als auch bei der Signalisierung (Aufbau einer Reservierung).
- **Einführbarkeit**. Sie gibt an, welcher Aufwand bei der Migration notwendig ist (z.B. Anzahl neuer Schnittstellen), um diese Technologie in eine bestehende Systemumgebung einzuführen und zu betreiben (Management-Overhead). Stand der Technik ist die von vielen Geräten bereits unterstützte DiffServ-Technologie.

Die Bewertung erfolgt in 5 Stufen: sehr gut (++), gut (+), befriedigend (o), bedingt ausreichend (-), ungenügend (--). Für einen hohen Gewährleistungsgrad ist ein Ende-zu-Ende Reservierungsverfahren sowie ein parameterbasiertes Zugangskontrollverfahren auf allen Links zwingend erforderlich. Eine gute Granularität des Dienstgüteangebotes wird durch ein Scheduling pro Flow/Paket sowie durch die Unterscheidung von Pfaden erreicht. Alle Flow-basierten QoS-Mechanismen führen insbesondere im Kernnetz zu einer hohen Komplexität und damit zu Skalierungsproblemen. Die Ergebnisse sind im Überblick in Tabelle 2-10 dargestellt.

Bei den Ansätzen wurde mit dem Architekturdesign und der Auswahl der Methoden jeweils ein anderer Kompromiss zwischen Granularität, QoS-Garantie und Komplexität eingegangen. Leider kann hinsichtlich der QoS-Garantie nicht für alle hier aufgeführten Architekturen eine exakte Aussage gemacht werden. Das liegt daran, dass weder eine genaue Ende-zu-Ende Dienstgütespezifikation vorgelegt noch ein Nachweis über die erzielbare QoS erbracht wurde. Die erzielbare Dienstgüte kann dann nur anhand der verwendeten QoS-Mechanismen geschätzt werden.

Tabelle 2-10 zeigt, dass die IntServ-Architektur die größte Granularität und Komplexität besitzt. Zwischen diesen beiden Kriterien gibt es einen direkt proportionalen Zusammenhang. Das Gegenstück zur IntServ-Architektur ist die DiffServ-Architektur: wo die Stärken des einen Ansatzes liegen, besitzt der andere Schwächen und umgekehrt. Anhand der Tabelle

können die konträren Design-Ziele der beiden IETF-Standards schnell erkannt werden. Die anderen Ansätze reihen sich zwischen diesen beiden Extrema ein.

	IntServ	DARWIN	SCORE	RRA	AQUILA	Egress-AC	TEQUILA	BB	DiffServ-AS
Dienstgütegarantie	++	++	++	o	o	o	o	-	--
Granularität	++	+	+	-	-	o	o	-	-
Komplexität	--	-	-	o	+	o	o	+	++
Einführbarkeit	--	--	--	o	-	-	-	o	++

Tabelle 2-10: Bewertung der QoS-Architekturen

Die Architekturen können zudem in einem zweidimensionalen Komplexitäts-/ QoS-Garantie-Diagramm dargestellt werden (siehe Abbildung 2-1).

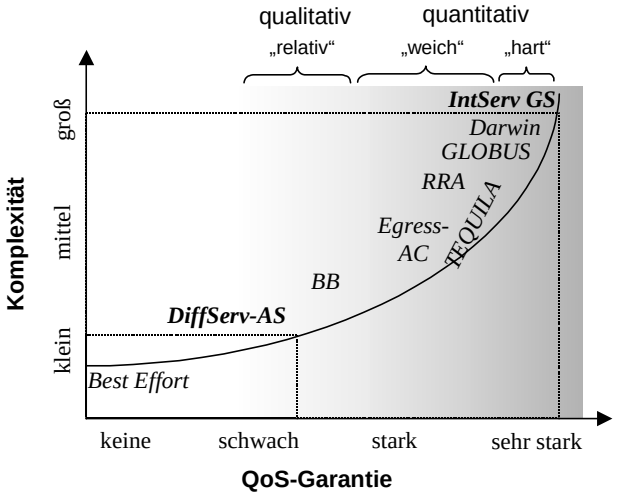


Abbildung 2-1: Architekturvergleich

Durch obige Grafik wird ein grundsätzlicher Zusammenhang zwischen der erzielbaren QoS-Garantie und der erforderlichen Komplexität ersichtlich: Die Komplexität steigt mit wachsender Dienstgütegarantie nicht linear, sondern überproportional an. Dieses Verhalten ist durch zwei Tatsachen erklärbar:

- für eine zuverlässige QoS-Garantie sind Zustandsinformationen pro Flow erforderlich;
- je größer der gewünschte Gewährleistungsgrad ist, desto größer ist zwangsläufig die Anzahl der Stellen im Netz, für die pro Flow Zustände gehalten werden müssen.

Zusammenfassend lässt sich feststellen, dass es zwar einige Ansätze mit harter QoS-Garantie gibt, diese jedoch sehr komplex sind und nur schwer in eine bestehende Systemumgebung eingeführt werden können. Die übrigen Architekturen bieten eine weitaus geringere QoS-Garantie.

Für einen Netzbetreiber sind vor allem Systemlösungen mit geringer Komplexität von Interesse, die auf einer standardisierten Basistechnologie beruhen und einen geringen Managementaufwand erfordern. Für die Betreiber von interaktiven Realzeitdiensten hingegen spielen primär ein hoher Gewährleistungsgrad und die leichte Einführbarkeit eine entscheidende Rolle. Auf eine hohe Granularität des Dienstangebotes hingegen kann in den meisten Fällen verzichtet werden.

Betrachtet man die eben vorgestellten Ansätze, so kann keiner eine Dienstgütegarantie sicher gewährleisten und zugleich leicht in eine bestehende Systemumgebung integriert werden. Es wird daher eine QoS-Architektur benötigt, die genau diese Anforderungen erfüllt.

3. Die Systemarchitektur im Überblick

In diesem Kapitel wird ein kurzer Überblick über die Prinzipien und Merkmale der Ressourcenmanagement-Architektur gegeben. Neben der Zielsetzung werden die technischen Rahmenbedingungen definiert. Das Kapitel endet mit einem ausführlichen Vergleich der RM-Architektur mit der IntServ- und DiffServ-Architektur sowie einer generellen Einordnung in die Reihe der in Kapitel 2 vorgestellten Ansätze.

3.1 Zielsetzung

Ziel der vorgestellten Arbeit ist es, eine RM-Architektur für Echtzeitverkehre in Intranets zu entwickeln. Das Ressourcenmanagement soll ermöglichen, echtzeitkritische Dienste mit verschiedenen garantierbaren Qualitätsstufen abzuwickeln. Der Nutzer eines solchen Dienstes soll dabei dieselbe Dienstgüte erfahren, wie er sie vom PSTN her gewohnt ist.

Ausgehend von einer IP-basierten Infrastruktur, bestehend aus Netzknoten mit denen Dienstklassen im Sinne eines *Class-of-Service* (CoS) Netzes realisiert werden können (siehe Abschnitt 3.2.2), ist die Aufgabe des Ressourcenmanagements, den Zugang der Teilnehmer zu gemeinsamen Netzressourcen einer Dienstklasse zu regulieren. Dabei soll für ein gegebenes Netz (Topologie, Dimensionierung, Routing) die Linkauslastung bei vorgegebener Übertragungsqualität optimiert werden. Dadurch können kurzfristig auftretende Überlastsituationen im Netz vermieden und harte **Dienstgütegarantien (QoS-Garantien)** gegeben werden. Die für eine neue Verbindung benötigten Ressourcen werden noch vor der Nutzdatenübertragung berechnet und mit den noch vorhandenen Ressourcen verglichen. Im Überlastfall wird der Verkehr abgewiesen (Blockierung) und somit die Dienstgüte bereits aktiver Verbindungen vor Qualitätseinbußen geschützt.

Bei dem Design der RM-Architektur soll eine weitestgehende **Entkopplung von den Diensten und Netztechnologien** erreicht werden. Dasselbe gilt auch auf Dienstebene. Das RM-System soll möglichst unabhängig von einer spezifischen Dienststeuerung sein und seine höherwertigen Übertragungsdienste beliebigen Anwendungen zur Verfügung stellen können.

Ferner soll auf eine gute **Skalierbarkeit** geachtet werden. Das RM-System soll nicht nur für Intranets geeignet sein, sondern auch auf größere IP-Netze übertragen werden können.

Darüber hinaus sollen Aspekte der **Migration**, d.h. der leichten Einführbarkeit des RM-Systems in eine bestehende Infrastruktur, von Anfang an berücksichtigt werden. Das Ressourcenmanagement ist dabei möglichst geschickt in seine Systemumgebung einzubinden, damit der Installations- und Managementaufwand so gering wie möglich bleibt. Wichtige Komponenten der Systemumgebung sind auf der einen Seite die Signalisierungsarchitekturen H.323 (SIP), auf der anderen Seite die zu verwaltenden Netzkomponenten (Switches, Router) und QoS-Konzepte (IEEE 802.q, IntServ, DiffServ).

3.2 Das zugrundeliegende Ressourcenmodell

Das Ressourcenmanagement basiert auf verschiedenen Annahmen bezüglich des zu verwaltenden Netzes. Diese werden im Folgenden kurz zusammengefasst. Zunächst wird das Umfeld eines Intranets beschrieben, anschließend werden die technischen Voraussetzungen im Netz definiert und daraus abgeleitet schließlich ein Knotenmodell entwickelt. Das Knotenmodell dient als Basis für die Zugangskontrollverfahren (Kapitel 5) und für die Datenmodellierung der Topologie (Kapitel 6).

3.2.1 Netzmodell: „Intranet“

Intranets sind private Firmennetze, die den IP-Verkehr der Mitarbeiter transportieren. Unter „internem“ Verkehr versteht man den Verkehr innerhalb eines Firmennetzes, d.h. zwischen den Arbeitsplätzen der Mitarbeiter einer Firma. Mit „externem“ Verkehr wird der Verkehr nach außen, d.h. in ein öffentliches Netz bezeichnet.

Bei kleinen Firmen beschränkt sich die räumliche Verteilung der Mitarbeiter häufig auf ein oder mehrere benachbarte Häuser. Der gesamte interne Verkehr kann über ein lokales Firmennetz geführt werden, das von der Firma selbst betrieben wird.

Größere Firmen dagegen besitzen meist mehrere nationale oder internationale Standorte. Sie können in der Regel nicht über ein eigenes Netz miteinander verbunden werden, sondern müssen an öffentliche Netze angeschlossen werden. Je nach Verkehrsaufkommen kann es sich um Wählverbindungen, fest angemietete Verbindungen (*Leased Lines*) oder um virtuelle private Netze (VPN) handeln. Ein VPN ist ein virtuelles Netz einer Benutzergruppe. Es entsteht durch die virtuelle Aufteilung der gemeinsamen Ressourcen eines öffentlichen Netzes auf mehrere Benutzergruppen. Handelt es sich um ein Wählnetz, wird diese Aufteilung durch die Vergabe und Umsetzung von öffentlichen und privaten Adressen erreicht. Für die Adressumsetzung wird die IN-Technologie (Intelligentes Netz IN) benötigt. In IP-Netzen kann ein VPN beispielsweise mit MPLS-Technologie realisiert werden. MPLS dient dabei dem Einrichten von dauerhaften Ende-zu-Ende Pfaden (*Label Switched Path* LSP) und der Einbettung (*Encapsulation*) der privaten Nutzdaten. Mit Hilfe von MPLS kann der öffentliche Internetverkehr zusammen mit dem privaten Intranetverkehr über ein Netz abgewickelt werden.

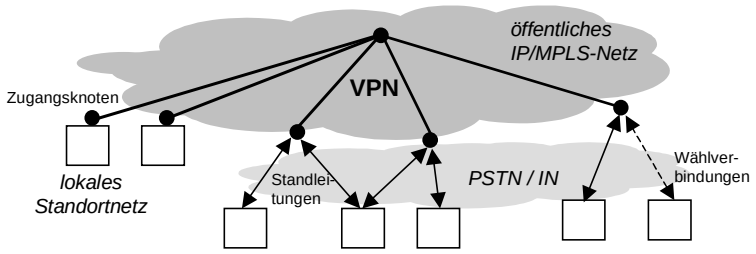


Abbildung 3-1: Beispielszenario eines Intranets

Bei der Anbindung von lokalen Firmennetzen an das VPN werden Service-Verträge mit einem öffentlichen IP-Netzbetreiber ausgehandelt. Der Vertragsinhalt kann z.B. ein Verkehrsvolumen sein, welches zu festen Konditionen zwischen den Standorten ausgetauscht werden kann. Dabei sind möglichst genaue Angaben zu den verschiedenen Verkehrstypen (Sprache, Video, Daten) hinsichtlich ihrer Verkehrscharakteristik und den QoS-Anforderungen zu machen. Vereinfachend kommt an dieser Stelle für Intranets hinzu, dass der Verkehr besser d.h. genauer abgeschätzt werden kann, als dies für das öffentliche Internet möglich ist. Zum einen ist die Anzahl der verwendeten Applikationen begrenzt, da jede

Applikation in der Regel vor ihrer Einführung hinsichtlich der Sicherheitsmerkmale, des Verkehrsverhaltens und der Qualitätsanforderungen geprüft wird. Dadurch ist ihr Systemverhalten bekannt und kann gegebenenfalls durch Messungen validiert werden. Zum anderen ist das Anwenderverhalten aufgrund der bekannten innerbetrieblichen Arbeitsprozesse besser vorhersagbar. In einem Intranet kann das Anwenderverhalten relativ einfach durch gezielte Messungen der Verkehrsströme im Netz ermittelt werden. So gilt es z.B. für die Abschätzung des Telefonverkehrs zu wissen, wie viele Arbeitnehmer einer Abteilung wie häufig und wie lange mit Arbeitnehmern einer anderen Abteilung kommunizieren. Ähnliches gilt auch für die Datenkommunikation zwischen Clients und Servern.

Der Betreiber eines VPN stellt jedem Firmenstandort einen Zugangsknoten zur Verfügung. Alle firmeninternen, standortübergreifenden Verbindungen laufen dann über dieses VPN. Die darunter liegende Transportnetzstruktur des VPN ist in der Regel für die Firma transparent.
Im Fall von externen Verbindungen bestehen in einem deregulierten Markt keine festen vertraglichen Verpflichtungen zu einem einzigen öffentlichen Netzbetreiber. Bei jedem Verbindungsaufbau kann ein anderer öffentlicher Netzbetreiber ausgewählt werden.
Abbildung 3-1 zeigt ein Beispielszenario für ein großes Firmennetz. Es besteht aus mehreren Standorten mit jeweils einem lokalen Firmennetz. Die lokalen Standortnetze der Firma sind teilweise mit Wählverbindungen und Standleitungen über einen öffentlichen PSTN-Betreiber oder direkt an einen öffentlichen IP-Netzbetreiber angebunden. Der IP-Netzbetreiber hat für die Firma ein VPN eingerichtet, über das sowohl Datenverkehr als auch echtzeitkritischer Multimediaverkehr abgewickelt werden können.

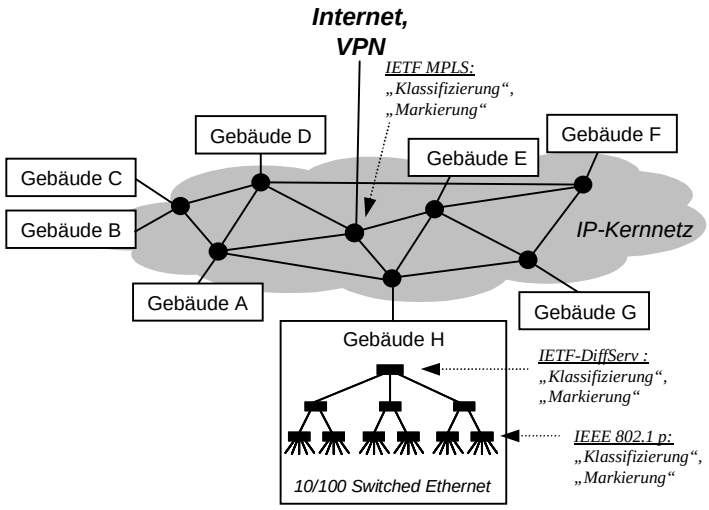


Abbildung 3-2: Szenario "Lokales Standortnetz"

Abbildung 3-2 zeigt ein Beispiel für ein lokales Standortnetz, in dem mehrere Gebäude über ein standortinternes IP-Kernnetz verbunden sind. Die hausinterne Vernetzung der Arbeitsplatzrechner erfolgt sternförmig über ein *Switched-Ethernet* LAN (*Lokal Area Network*). Ein *Switched-Ethernet* LAN ist ein spezielles *Ethernet* LAN (IEEE 802.3) [I802.3], bei dem jeder Teilnehmer über eine eigene Leitung (*Twisted Pair*, *Full-Duplex*) mit einem Port eines *Switches* (ISO Schicht-2 Netzknoten) verbunden ist. Jedem Teilnehmer stehen dadurch in beiden Richtungen Anschlussraten von 10 bzw. 100Mbit/s zur Verfügung. Die einzelnen Leitungen werden dabei in mehreren Stufen zusammengefasst: z.B. Arbeitsgruppe, Abteilung, Etage, Gebäude. Sowohl im Kernnetz als auch im Zugangsnetz wird das Prinzip der Verkehrsklassen verwendet. Die konsequente Trennung von echtzeitkritischen und nicht-echtzeitkritischen Datenströmen beginnend beim

Teilnehmeranschluss ist die Voraussetzung für Dienstgütegarantien beim Datentransport. Am Teilnehmerzugang werden die Schicht-2 Pakete nach dem IEEE Standard 802.1 D [I802.1] klassifiziert und entsprechend im Paketkopf markiert. Der Gebäude-Switch arbeitet auf OSI-Schicht 3 und klassifiziert und markiert die IP-Pakete nach dem DiffServ-Standard. Der externe Verkehr des Standortes wird am Zugangsknoten zum VPN entsprechend der DiffServ (DS-) Dienstklassen (*Codepoints*) und dem Zielort (VPN, Internet) auf MPLS-Pfade abgebildet.

Für das RM-System ergeben sich aus den Unterschieden zwischen dem öffentlichen Internet und dem nach außen hin geschlossenem System des Intranets folgende Vereinfachungen:

- **Sicherheit (*Security*):**
  Sie spielt für ein RM innerhalb von Firmennetzen eine untergeordnete Rolle. Auf Verschlüsselungs- und Tunnelmechanismen der Reservierungsnachrichten des RM-Systems wurde daher verzichtet. Eine Verschlüsselung der Nutzdaten kann unabhängig vom RM-System z.B. von den Endgeräten vorgenommen werden, solange mögliche Einflüsse auf die Verkehrscharakteristik in den Quellenparametern (Signalisierung) enthalten sind.
- **Überwachung (*Policies*):**
  Ein Intranet ist ein geschlossenes System (*trusted area*). Alle Netzknoten und Endgeräte werden von einer zentralen Netzadministration konfiguriert und verwaltet. Die Überwachung der Verkehrsquellen durch Filter (*policer*) oder die Kontrolle der vom Endgerät klassifizierten und markierten Pakete im Netz sind daher nicht zwingend erforderlich. Prinzipiell kann das RM-System dazu verwendet werden, Überwachungseinheiten in den Zugangsknoten anhand der Verbindungsparameter zu konfigurieren.
- **Verkehrsformung (*Shaping*):**
  Eine Formung von Verkehren ist erforderlich, da Pakete auf ihrem Weg durch das Netz verklumpen und am Netzausgang ein bursthafteres Verhalten als am Netzeingang aufweisen. Das veränderte Verhalten der Verkehre kann an der Grenze zwischen Netzbetreibern zu ungewollter Verletzung von Verträgen (*Service Level Agreements*) und auf der Gegenseite beim Einsatz von Verkehrsüberwachungseinheiten (*Policer*) zu Verlusten führen. Prinzipiell kann das RM-System auch hier dazu verwendet werden, Verkehrsformungseinheiten in Übergangsknoten zu anderen Netzbetreibern anhand der Verbindungsparameter zu konfigurieren. Eine Verkehrsformung auf der Granularität einzelner Paketströme an Netzübergängen erscheint jedoch nicht sinnvoll. Da Verklumpungseffekte insbesondere bei hoher Verkehrsaggregation in Kernnetzen auftreten, müssen zunächst Mechanismen gefunden werden, um aus den vielen Verkehrsbeschreibungen der einzelnen Paketströme eine Beschreibung für den aggregierten Paketstrom zu bilden. Diese Aufgabe ist jedoch nicht Gegenstand der vorliegenden Arbeit. Aufgrund der vergleichsweise niedrigen Verkehrsaggregation in Intranets spielt die Verkehrsformung für das RM-System eine untergeordnete Rolle. Darüber hinaus kann dem Problem der Verklumpung auch durch den Einsatz von non-work conserving Schedulern (siehe Abschnitt 3.2.3) begegnet werden.
- **Netzplanung:**
  Intranet-Verkehre, d.h. die Verkehrsbeziehungen und die Charakteristik (Verkehrsmatrix) sind besser vorhersagbar, als dies für das öffentliche Internet möglich ist. Zum einen ist die Vielfalt der verwendeten Applikationen begrenzt, zum anderen ist das Anwenderverhalten aufgrund der bekannten innerbetrieblichen

Arbeitsprozesse und der leichter zu generierenden Statistiken genauer bestimmbar. Dadurch können die Verkehrsklassen eines Intranets genauer dimensioniert werden. Zudem lassen sich leichter homogene Verkehrsgemische bilden und damit die QoS-Garantie (*Assurance*) verbessern.

3.2.2 Netztechnologie: "Class-of-Service" Netz

Um von einem zentralen Server aus die Ressourcen von Netzknoten verwalten zu können, müssen verschiedene Voraussetzungen gegeben sein. Eine Voraussetzung liegt in der Netztechnologie. In den Anfängen der DiffServ-Standardisierung wurde in [FJ95] der Begriff des **„Class-of-Service"** (CoS) Netzes geprägt. Ein CoS-Netz basiert auf dem Prinzip der Verkehrstrennung (*Traffic Discriminiation*) und einem statischen „*Link-Sharing*" Modell. Mit dem Begriff CoS werden im Folgenden Technologien sowohl auf ISO Schicht-2 (IEEE 802.1 D, *MAC-Tagging*: *Priority Bit*) als auch auf ISO Schicht-3 (IETF DiffServ: PHB *Per Hop Behavior, IP-Tagging*: TOS *Type of Service Field*) zusammengefasst.

Ein CoS-Netz unterscheidet mehrere Verkehrsklassen. Alle Pakete werden einer Verkehrsklasse zugeordnet. Pakete einer Verkehrklasse werden gleich behandelt, die Pakete unterschiedlicher Verkehrsklassen jedoch verschieden.

Bei der Konfiguration der Verkehrsklassen wird einem Knoten eine Vorschrift zum Klassifizieren, d.h. Zuweisen der eintreffenden Pakete zu einer bestimmten Verkehrsklasse gegeben. Die Klassifizierung erfolgt über die Steuerfelder im Paket-Header. Dabei können entweder einzelne Paketströme mit Hilfe von z.B. IP-Adressen, TCP/UDP Port-Nummern und RTP-Felder oder ganze Bündel von Paketströmen z.B. anhand von IP-Adressbereichen klassifiziert werden. Zudem werden für jede Verkehrsklasse explizit Ressourcen reserviert, die bei Bedarf ausschließlich den Paketen dieser Verkehrsklasse zur Verfügung stehen. Bei den reservierten Ressourcen handelt es sich um Leitungs- und Pufferkapazitäten.

Ein CoS-Netz besteht aus einem Basisnetz, das sich aus mehreren übereinander liegenden logischen Teilnetzen zusammensetzt. Alle logischen Teilnetze besitzen dieselbe Topologie wie das Basisnetz und verwenden das gleiche Routing. Die Leitungs- und Pufferkapazitäten des Basisnetzes werden jedoch auf die verschiedenen Teilnetze aufgeteilt. Jeder Paketstrom wird am Netzzugang einem der Teilnetze zugewiesen. Das Scheduling-Verfahren muss sicherstellen (siehe Abschnitt 3.2.3 oder 5.3.3), dass nur die Paketströme, die demselben Teilnetz zugewiesen werden, sich gegenseitig beeinflussen können. Die von einem Paketstrom erzielbare Übertragungsqualität hängt somit nicht mehr vom Lastzustand des gesamten Links, sondern lediglich vom Auslastungszustand der Ressourcen einer Verkehrsklasse ab. QoS-Garantien können jedoch nur gegeben werden, wenn der Auslastungszustand der Ressourcen dieser Verkehrsklasse z.B. durch eine Netzzugangskontrollfunktion begrenzt wird.

In einem CoS-Netz ist eine Zugangskontrollfunktion zu den Ressourcen einer Verkehrsklasse nicht vorgesehen. Das Ressourcenmanagement-System kann diese Funktion jedoch übernehmen und den Teilnehmerzugang zu den Ressourcen einer oder mehrerer Dienstklassen regeln. Der Betrieb eines RM-Systems setzt voraus, dass die Dienstklassen ausreichend dimensioniert sind. Andernfalls ist die Blockierungswahrscheinlichkeit der Teilnehmer sehr hoch.

Mit einem CoS-Netz, das von einem RM-System verwaltet wird, ist ein Netzbetreiber in der Lage, einen Transportdienst mit QoS-Garantien anzubieten. Ein RM-System stellt damit eine Differenzierungsmöglichkeit des Angebotes eines CoS-Netzbetreibers dar. Die Granularität des Angebotes hängt jedoch allein von der Anzahl der Dienstklassen des CoS-Netzes ab und wird vom RM-System nicht beeinflusst. Im Allgemeinen möchte der Netzbetreiber die

Anzahl der Dienstklassen so gering wie möglich halten, da jede Klasse einen Managementaufwand und damit Kosten verursacht. Wieviele Verkehrsklassen er im Netz dafür einrichtet, hängt stark von den Wünschen und Anforderungen seiner Kunden ab.

3.2.3 Knotenmodell

Ein wichtiger Baustein eines CoS-Netzes sind die Netzknoten. Sie sind der Ort in einem Netz, an dem es zu Paketverzögerungen und Verlusten kommen kann. Netzknoten verbinden mehrere Teile eines Netzes miteinander und besitzen daher mehrere Ein- und Ausgänge. Während des Betriebs kommt es häufig dazu, dass auf mehreren Eingängen gleichzeitig Pakete ankommen, die auf denselben Ausgang vermittelt werden müssen (äußere Blockierung). Betrachtet man die Ankunftsraten der Verkehre an einem Knotenausgang, so bedeutet dies, dass N Eingänge gleichzeitig mit der Rate R in einen Ausgangspuffer schreiben, der aber nur mit der Rate $C < N \cdot R$ ausgelesen werden kann. Die zwangsläufige Folge sind Paketstaus und gegebenenfalls Pufferüberläufe.

Um den Anforderungen von echtzeitkritischen Anwendungen gerecht werden zu können, benötigt man vor allem im Kernnetzbereich sehr leistungsfähige Hochgeschwindigkeits-Switches/Router, die mehrere Dienstklassen unterstützen. Die Leistungsfähigkeit eines Netzknotens zeichnet sich einerseits durch seine Portzahl, Anschlussraten sowie den maximalen Durchsatz bei der internen Paketvermittlung (*Forwarding*) und andererseits durch seine QoS-Mechanismen (Puffermanagement- und Scheduling-Verfahren) aus. Während die Wahl der Portzahlen und Portgeschwindigkeiten Aufgabe der Netzplanung und Netzdimensionierung ist, ist die Wahl der QoS-Mechanismen Aufgabe des QoS-Managements. Die QoS-Mechanismen sind für die Realisierung der Dienstklassen notwendig und ermöglichen im Fall von äußeren Blockierungen, dass echtzeitkritische Verkehre im Vergleich zu nicht-echtzeitkritischen Verkehren bevorzugt zu behandeln (bedienen).

Knotenarchitekturen
Der innere Aufbau (Switching-Architektur) ist für die Skalierbarkeit eines Knotens hinsichtlich der Portzahl sowie der Anschlussgeschwindigkeiten entscheidend. Es gibt verschiedene Arten von Architekturen [AM95], die auf unterschiedlichen Pufferstrategien basieren. So gibt es reine Eingangs- bzw. reine ausgangsgepufferte Systeme, Systeme mit einem einzigen zentralen Puffer (*shared buffer*), eingangsgepufferte Systeme mit virtuellen Ausgangspuffern oder Kombinationen aus Eingangs- und Ausgangspuffern. Die Ein-/Ausgänge sind durch ein Koppelnetz miteinander verbunden. Das Koppelnetz arbeitet in der Regel blockierungsfrei, d.h. es können nur äußere und keine inneren Blockierungen des Koppelnetzes auftreten. Die Aufgabe des Scheduling-Algorithmus ist es, möglichst viele konfliktfreie Ein-/Ausgangsportpaare in möglichst kurzer Zeit zu finden.

Für das Ressourcenmanagement kommt es weniger auf die Art der Realisierung an, als vielmehr auf seine Eigenschaften hinsichtlich der damit erzielbaren QoS. Daher werden bei dem Knotenmodell idealisierte Annahmen getroffen (siehe unten).

Modell
Für die weitere Arbeit wird für Router und Switches ein einheitliches Knotenmodell definiert. Dazu wird ein Netzknoten als ein „ideales“ ausgangsgepuffertes System vorausgesetzt. Idealisiert ist das hier vorgestellte Modell deshalb, weil vereinfachend davon ausgegangen wird, dass der interne Durchsatz beliebig erhöht werden kann. Paketverzögerungen aufgrund der Verarbeitung im Koppelnetz (*Forwarding*) werden nicht berücksichtigt. Allgemein kann man sagen, dass diese Annahme innerhalb von Firmennetzen weniger kritisch ist, da hier die Anforderungen hinsichtlich der benötigten Portzahlen und Portgeschwindigkeiten wesentlich geringer sind als z.B. in Weitverkehrsnetzen. Die Verzögerungen bei der Paketvermittlung

können daher im Vergleich zu den Wartezeiten in den Ausgangspuffern vernachlässigt werden.

Jeder Ausgang des Netzknotens besteht aus einer Klassifizierungseinheit, mehreren Puffern (ein Puffer pro Dienstklasse) und zwei hintereinander geschalteten Scheduling-Einheiten. Im Folgenden werden die Dienstklassen der Schicht-2 und Schicht-3 nicht weiter unterschieden und in Anlehnung an die DiffServ-Terminologie vereinfachend als **DS-Klassen** bezeichnet.

Ein Teil der Puffer ist für echtzeitkritische Verkehre vorgesehen, deren Auslastung vom Ressourcenmanagement-System kontrolliert wird. Die Puffer werden zunächst von einem WFQ-Scheduler und anschließend von einem *Simple-Priority*-Scheduler mit höchster Priorität bedient (siehe Abbildung 3-3). Der andere Teil der Puffer ist für nicht-echtzeitkritische Verkehre vorgesehen und wird nur vom *Simple-Priority*-Scheduler bedient. Durch den *Simple-Priority*-Scheduler wird eine Beeinflussung der echtzeitkritischen durch die nicht-echtzeitkritischen Verkehre ausgeschlossen. Durch den WFQ-Scheduler wird eine bestmögliche Trennung zwischen den Echtzeit-Verkehrsklassen erreicht. Durch ihn wird die Realisierung mehrerer DS-Klassen für echtzeitkritische Verkehre ermöglicht und dadurch eine bessere Granularität der Dienstgüte im Netz erzielt.

In diesem Modell verwaltet der WFQ nur einen Teil C^* der gesamten Linkkapazität C und garantiert jeder Dienstklasse einen bestimmten Mindestanteil ρ_i an einer virtuellen Linkrate C^*. Er begrenzt die Wartedauer eines zum Aussenden bereiten (*eligible*) Paketes der Länge L einer Klasse auf $t \leq L / (\rho_i \cdot C^*)$. Je kleiner der Quotient C^*/C ist, desto stärker ist die verkehrsformende Wirkung des Schedulers. Alle Pakete, die vom WFQ-Scheduler bereits bedient wurden, treffen anschließend als Verkehr mit höchster Priorität auf einen zweiten Scheduler. Dieser arbeitet nach dem *Simple-Priority* Prinzip und sendet echtzeitkritische Verkehre sofort, d.h. ohne zusätzliche Verzögerung aus.

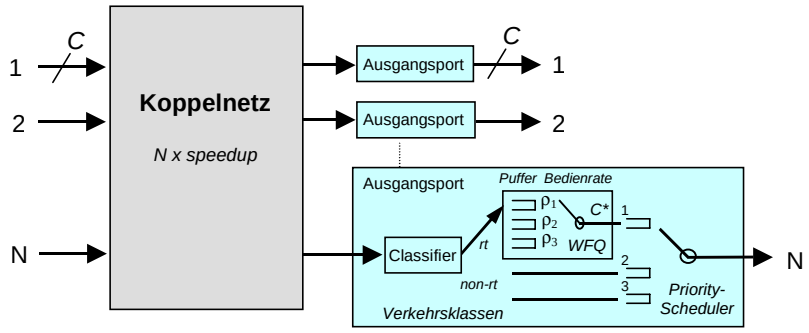


Abbildung 3-3: Modell eines CoS-Netzknotens

Ein solches Knotenmodell ist im Vergleich zu einem einzigen WFQ-Scheduler imstande, den Jitter der echtzeitkritischen Pakete zu reduzieren und den nicht-echtzeitkritischen Verkehren dennoch eine minimale Bedienrate $C - C^*$ zu garantieren.

Die Größe der Puffer und die Höhe der Bedienrate seien vom Konfigurationsmanagement für jede Dienstklasse separat einstellbar. Alle Konfigurationsdaten der DS-Klassen sowie der Verkehrslenkung (*IP-Forwarding* Tabelle) sind von außen zugänglich und über einen standardisierten Zugriffsmechanismus (z.B. SNMP) abfragbar. Das RM-System lernt die Konfiguration des Netzes und übt eine Zugangskontrolle für die Echtzeitverkehrsklassen aus.

3.3 Architekturmerkmale

Ausgehend von einem CoS-Netz steht das Konzept des Ressourcenmanagements auf drei Säulen:

- Einführung einer zentralen Netzzugangskontrollfunktion
- Einführung eines Ende-zu-Ende Reservierungsverfahrens
- Einführung von Verwaltungsdomänen bei großen Netzen

Die Umsetzung des Konzeptes beruht auf der

- Flexiblen Ankopplung des Ressourcenmanagements an eine Dienststeuerung
- Entkopplung von der Netztechnologie durch die Einführung von zwei Netzdiensten: Einem Topologieerkennungs- und einem Ressourcen-verwaltungsdienst

Zunächst wird das Prinzip der RM-Architektur erklärt, dann werden in mehreren Schritten die Architekturmerkmale eingeführt. Dabei werden die Komponenten der Architektur vorgestellt. Eine genauere funktionale Beschreibung der Komponenten ist in Abschnitt 6.3 zu finden.

3.3.1 Das Prinzip des Ressourcenmanagements

Der wesentliche Bestandteil dieses Konzeptes ([GME02], [PGM00]) ist die Einführung eines Reservierungsverfahrens auf Verbindungsebene in ein bestehendes CoS-Netz. Das Reservierungsverfahren ist an die Rufsteuerung einer Signalisierungsarchitektur (z.B. H.323, SIP, MEGACO) gekoppelt. Für jeden Verbindungswunsch eines Teilnehmers wird über eine Signalisierung eine Reservierungsprozedur im Netz angestoßen. Kernstück dieser Reservierungsprozedur ist eine Zugangskontrollfunktion, welche für jeden Link entlang des gesamten Pfades durch das Netz (Ende-zu-Ende) die notwendigen Ressourcen berechnet und danach entscheidet, ob die neue Verbindung angenommen werden kann oder abgelehnt werden muss.
Durch einen solchen Verbindungsannahme-Mechanismus kann das Netz dem Nutzer Garantien bezüglich der Einhaltung der geforderten *Quality-of-Service* Parameter (QoS) geben (siehe Kapitel 5). Die Entscheidung der Netzzugangskontrolle wird auf Basis von Effektiven Bitraten getroffen, welche in Abhängigkeit von der Quellencharakteristik, der geforderten QoS (maximale Verlustwahrscheinlichkeit, maximale Ende-zu-Ende Verzögerung) sowie den Eigenschaften des Pfades (Route, Link-Konfiguration, momentane Auslastung) durch das Netz berechnet werden. Durch das Ablehnen von Reservierungsanfragen (Blockieren) kann das Netz bei drohender Überlastung die Verbindungsgüte bereits aktiver Verbindungen aufrechterhalten.

3.3.2 Einführung einer Signalisierung

Zur Umsetzung eines solchen Reservierungsverfahrens wurde eine RM-Architektur mit einem zentralen Server entwickelt [PGM00], [GME02], [GM02]. Ziel dieses Ansatzes ist es, die Netzknoten nicht mit Signalisierungsverkehr zu belasten und dennoch eine Reservierung von Netzressourcen für alle Links des Ende-zu-Ende Datenpfades zu ermöglichen.
Das Konzept des Ressourcenmanagements basiert daher auf der strikten Trennung von Signalisierung und Nutzdatenübertragung. Die Verarbeitung der Signalisierungsnachrichten findet nicht in den Netzknoten (Routern, Switches) statt, sondern in speziellen Servern, den Ressourcen-Managern (siehe Abbildung 3-4).

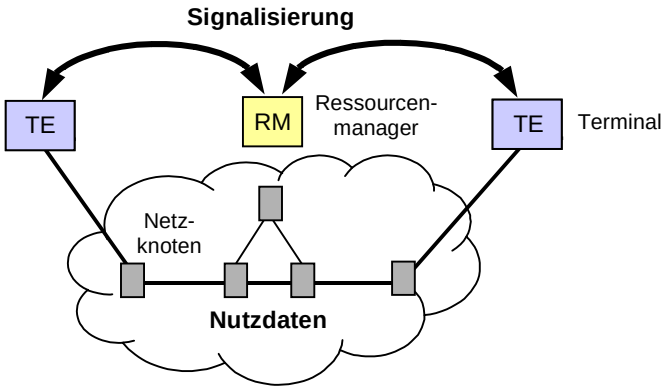


Abbildung 3-4: Konzept des Ressourcemanagement-Systems

Ein Ressourcen-Manager RM ist für eine bestimmte Anzahl von Netzknoten verantwortlich (Administrative Domäne) und übernimmt stellvertretend für diese die Zugangskontrollfunktion zu den lokal vorhandenen Ressourcen (Puffer-, Leitungskapazität). Um diese Zugangskontrollfunktion ausführen zu können, muss der RM die zur Verfügung stehenden Ressourcen und die eingestellten Pfade durch das Netz kennen.
Das Ressourcenmanagement arbeitet mit einem prädiktiven Zugangskontrollverfahren. Ein solches Verfahren basiert auf einer Signalisierung, mit der sich ein Teilnehmer vor oder während der Aufbauphase einer echtzeitkritischen Datenverbindung an das Ressourcenmanagement wendet, um notwendige Netzressourcen für die Dauer der Verbindung zu reservieren. Dabei teilt der Teilnehmer bzw. das Terminal TE dem Ressourcenmanagement den Verbindungswunsch in Form einer Reservierungsanfrage mit und gibt dabei einen Satz von Verbindungsparametern an. Der Verbindungsaufbauprozess des Teilnehmers wird solange unterbrochen, bis er eine Antwort vom RM erhält. Anhand der Verbindungsparameter kann der RM die benötigten Ressourcen entlang des Datenpfades zum Zielterminal berechnen und entscheiden, ob die vorhandenen Ressourcen im Netz ausreichen, um diese neue Verbindung zu tragen. Für eine bidirektionale Verbindung werden z.B. zwei Pfade ermittelt, d.h. der Pfad vom initiierenden Terminal zum gerufenen Terminal und der Pfad in die umgekehrte Richtung. In nicht sternförmig vermaschten IP-Netzen können diese Pfade unterschiedlich sein. Die Reservierung findet dann auf allen Links entlang beider Pfade statt. Kann die Verbindung zugelassen werden, aktualisiert der RM den Lastzustand aller beteiligten Ressourcen. Die Entscheidung des Ressourcenmanagements wird der Dienststeuerung des Teilnehmers mitgeteilt. Der Teilnehmer kann daraufhin den Verbindungsaufbau beenden und mit dem Senden von Nutzdaten beginnen oder er muss den Verbindungsaufbau abbrechen.

Das Reservierungsverfahren des Ressourcenmanagements ist zustandsbasiert und erfordert daher die explizite Freigabe der belegten Ressourcen nach Beendigung der Nutzdatenverbindung durch eine entsprechende Signalisierung. Eine Überwachung der Zustände kann sowohl durch die Dienststeuerung des Teilnehmers (siehe Abschnitt 3.3.4) als auch von Seiten des RM-Systems (Timer, Statusmeldungen) erfolgen.

Entscheidend für die Anwendbarkeit des RM-Ansatzes ist, dass zum Zeitpunkt der Reservierung die Routen vorhersagbar sind. Bei *Multipath Routing*-Verfahren muss dem RM der Mechanismus bekannt sein, nach dem die Router ankommende Pakete auf mehrere parallele Pfade aufteilen. Bei *Single Path Routing*-Verfahren gibt es keine dynamische Aufteilung. Sie sind einfacher zu handhaben und werden daher für die RM-Architektur als Routing-Verfahren empfohlen.

Der zentrale Ressourcenmanager stellt einen sogenannten „*Single Point of Failure*“ im System dar. Im Folgenden werden Komponentenausfälle jedoch nicht näher betrachtet, da

aus dem IT-Bereich bekannt ist, wie zuverlässige Server zu konzipieren sind. Die Prinzipien basieren auf der redundanten Auslegung des Servers sowie seiner redundanten Anbindung an das Netz (*Dual Homing*).

3.3.3 Entkopplung von der darunter liegenden Netztechnologie

Der RM selbst ist keine Komponente des IP-Netzes, sondern ein Zusatz-Dienst, der auf eine bestehende Infrastruktur aufsetzt. Aus funktionaler Sicht ist er daher oberhalb der Netzressourcen anzusiedeln. Die Funktionsverteilung zwischen dem Netz und dem RM sieht dabei wie folgt aus: Die vom RM verwalteten Ressourcen werden vom Netzplaner dimensioniert und vom Netzadministrator eingestellt. Der Prozess der Konfiguration von Netzknoten und das Einrichten der DS-Klassen erfolgt unabhängig vom Prozess der Ressourcenverwaltung.
Der RM hat daher die Aufgabe, die Konfiguration der DS-Klassen sowie die von den Routing-Protokollen etablierten Pfade in Erfahrung zu bringen. Dieser Erkennungsvorgang muss lediglich bei der Systeminitialisierung, bei Änderungen der Topologie oder bei der Konfiguration der Netzknoten stattfinden. Ferner sind für die Topologieerkennung spezifische Methoden notwendig, die zum einen von der verwendeten Technologie des Netzes abhängen, zum anderen in den Zuständigkeitsbereich der Netzadministration fallen. Aus diesem Grund wird der Topologieerkundungsdienst als ein eigenständiger Dienst beschrieben, der unabhängig vom Reservierungsvorgang ist [Gla00], [GM02], [GT02].

Der Topologieerkundungsdienst wird im Folgenden als Topologie-Manager TM bezeichnet. Aus Performance- und Stabilitätsgründen wird empfohlen, den TM und den RM auf unterschiedlichen Rechnern zu installieren. Der TM erkundet selbständig die Topologie (Schicht-2, Schicht-3) und die aktuelle Konfiguration (Routing, DS-Klassen) des Netzes, indem er direkt auf die Konfigurationsdateien der Netzknoten zugreift.

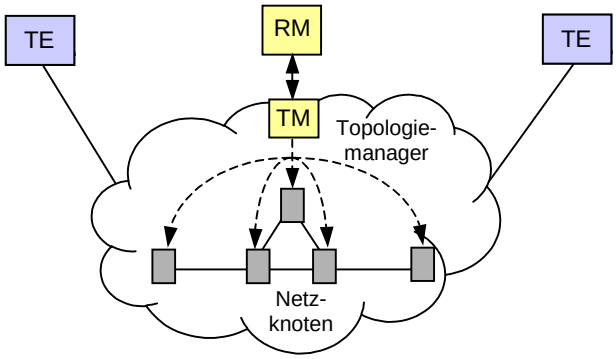


Abbildung 3-5: Entkopplung von der Netztechnologie

Er sammelt so alle Topologie- und Konfigurationsdaten der Netzknoten und speichert diese in einem Datenmodell ab. Der RM ist über ein Protokoll an den TM angebunden und dadurch in der Lage, auf die Topologiedaten des TM zuzugreifen. Prinzipiell können mehrere RM-Instanzen mit einem TM verbunden sein (siehe Abschnitt 3.3.5).
Normalerweise sind die Konfigurationsdaten der Netzknoten nicht frei zugänglich. Daher benötigt der TM in der Regel spezielle Zugriffsberechtigungen von Seiten der Netzadministration. Der TM ist folglich eher ein Teil des Netzes und stellt aus Sicht der Gesamtarchitektur eine Konvergenzschicht über den Netzkomponenten dar. In Form eines Datenmodells liefert er dem RM eine abstrakte Sicht auf das Netz. Auf der Basis dieses Datenmodells kann der RM dann die Zugangskontrollfunktion ausführen. Herstellerspezifische Besonderheiten beim Zugriff auf die Konfigurationsdateien bleiben

dadurch für das Ressourcenmanagement ebenso transparent wie die für den Zugriff notwendigen Passwörter (*Security Strings*).

Empfängt der RM eine Reservierungsanfrage, ermittelt er anhand der Zieladresse den Datenpfad, dem später die Nutzdaten zwischen den Terminals folgen werden. Durch das Datenmodell der Netztopologie kennt er alle Links und deren Dienstklassenspezifikationen, auf denen er eine Zugangskontrolle durchführen muss. Der RM führt, beginnend bei dem initiierenden Terminal, nach und nach für alle Links bis hin zum Zielterminal eine Zugangskontrollfunktion aus. Nur wenn die Zugangskontrollentscheidungen auf allen beteiligten Links positiv waren, kann die Reservierung durchgeführt und die Verbindung zugelassen werden.

3.3.4 Anbindung an eine Signalisierungsarchitektur

Der Teilnehmer bzw. die Applikation teilt vor oder während des Verbindungsaufbaus dem Ressourcenmanagement den Verbindungswunsch mit und gibt dabei einen Satz von Verbindungsparametern an.
Diese Mitteilung kann über ein eigenes Protokoll zwischen den Teilnehmern und dem Ressourcenmanagement oder indirekt über eine standardisierte Teilnehmer-Netz Schnittstelle erfolgen. Signalisierungsarchitekturen für Multimediadienste in IP-Netzen wie z.B. H.323 oder SIP bieten eine solche Schnittstelle an (*H.323: Gatekeeper, SIP: Registrar, Proxy, Redirect Server*). In beiden Fällen wendet sich ein Teilnehmer zunächst an eine **Dienststeuereinheit DS** im Netz, die für dienstspezifische Sonderaufgaben herangezogen werden kann. Sie umfasst Funktionen, wie sie im Internet bislang nicht vorgesehen waren. Beispiele für Funktionen einer DS sind Adressauflösung, Teilnehmerverwaltung, Endgeräteüberwachung oder Tarifierung eines Dienstes. Im Fall von Telefoniediensten kann eine solche Einheit auch stellvertretend für einen inaktiven Teilnehmer reagieren und eingehende Rufe auf eine *Voicebox* schalten oder den Ruf an ein anderes Endgerät weiterleiten. Für die Umsetzung dieser Funktionen enthält die standardisierte Teilnehmer-DS Schnittstelle bereits eine sehr mächtige Signalisierung, die alle Parameter beinhaltet, wie sie für die hier betrachteten Zugangskontrollverfahren benötigt werden. Dadurch besteht die Möglichkeit, den RM-Dienst an eine DS anzukoppeln. Der große Vorteil dieser Lösung liegt in der leichteren Einführbarkeit des RM-Dienstes in eine bestehende Systemumgebung. Da die Teilnehmer-Netz Schnittstelle davon unberührt bleibt, muss nicht eine Vielzahl von Endgeräten z.B. eines großen Firmennetzes mit einer neuen Schnittstelle zum RM-System versehen werden, sondern lediglich einige wenige Dienststeuereinheiten.

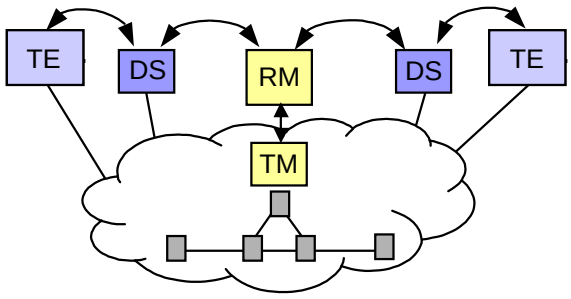


Abbildung 3-6: RM als Backend Service

Aus diesen Gründen wird das RM-System an eine Dienststeuereinheit angekoppelt und arbeitet für die Dienstarchitekturen im Hintergrund als *Backend Service*. Jeder Teilnehmer bzw. jedes Terminal ist bei einer bestimmten DS registriert und dieser dadurch fest zugeordnet. Die Zuordnung ist nicht starr an die Topologie des Netzes gebunden und hängt in

Firmen meistens von der internen Organisationsstruktur ab. Die DS im H.323 besitzt z.B. eine dynamische Adressverwaltung und unterstützt dadurch eine Mobilität der Teilnehmer innerhalb der Firma.

3.3.5 Domänenbildung

Da ein zentraler Ansatz für große Netze schlecht skaliert, sieht die RM-Architektur vor, größere Netze in mehrere RM- und TM-Domänen aufzuteilen. Jede dieser einzelnen Domänen wird dabei zentral von einem RM bzw. TM verwaltet. Durch einen verteilten Ansatz kann die in den jeweiligen Servern vorzuhaltende Datenmenge verringert und die Reaktionszeit eines Servers auf eine Anfrage reduziert werden. Reservierungen über mehrere RM-Domänen hinweg werden durch ein Inter-RM Protokoll koordiniert.

RM-Domäne

Ein RM ist für die Ressourcenverwaltung eines zusammenhängenden Netzbereiches (RM-Domäne) zuständig. Ein solcher Netzbereich umfasst Netzknoten und Terminals, die zu einem bestimmten Adressbereich (z.B. ein/mehrere IP-Subnetz/-e) gehören, sowie alle zugehörigen Links.
Die Aufteilung größerer Netze in mehrere RM-Domänen ist durch die Beschränkungen eines RM-Servers hinsichtlich seiner Performance erforderlich. Ein einzelner RM ist in den Signalisierungsablauf einer Dienststeuerung eingebunden und bearbeitet sequenziell die Reservierungsanfragen der Teilnehmer. Während die Anfrage eines Teilnehmers bearbeitet wird, ist der Prozess des Verbindungsaufbaus solange unterbrochen, bis der Teilnehmer vom RM die Erlaubnis zum Fortfahren bekommt. Die Aufteilung eines Netzes in RM-Domänen soll so erfolgen, dass die Zeitdauer eines Verbindungsaufbaues in bestimmten Grenzen gehalten werden kann. Dazu wurde ein entsprechendes Verfahren spezifiziert (siehe Abschnitt 6.7.1).

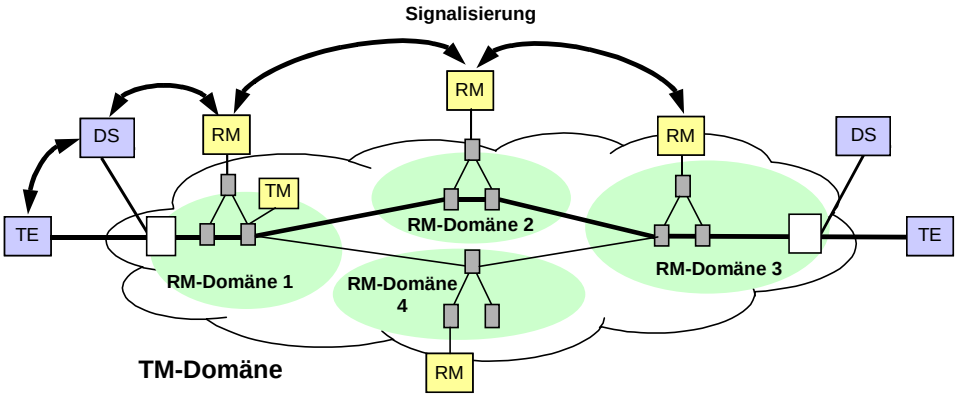


Abbildung 3-7: Signalisierungsbeispiel mit vier RM-Domänen

TM-Domäne

Eine TM-Domäne stellt ebenso wie die RM-Domäne einen Teilbereich eines größeren Netzes dar. Jeder Knoten eines Netzes wird genau einer TM-Domäne zugewiesen und von dem jeweiligen TM hinsichtlich seiner Konfiguration abgefragt. Im Gegensatz zur RM-Domänenbildung ist die Aufteilung größerer Netze in mehrere TM-Domänen mehr durch administrative Vorgaben bedingt, als durch Performance-Anforderungen an den TM.

Ein TM greift während der Topologieerkundung auf Konfigurationsdateien der Netzknoten zu. Ein solches Vorgehen erlaubt dem Zugreifenden einen tiefen Einblick in das Innerste des Netzes und gibt ihm unter Umständen sogar die Möglichkeit, die Konfiguration des Netzes zu verändern und dem Netzbetreiber großen Schaden zuzufügen.

3.3.6 Multi-Betreiber Szenario

Das Konzept der Domänenbildung schließt die Aspekte eines Multi-Betreiber Szenarios mit ein. Die jeweiligen RM- und TM-Domänen können von unterschiedlichen Dienst- oder Netzanbietern betrieben werden. Dies hat sowohl Auswirkungen auf die Domänenbildung als auch auf die Behandlung des Datenverkehres.

Eine Netzgrenze zwischen zwei Betreiber stellt in der Regel eine Grenze zwischen zwei TM-Domänen dar. Ein Netzbetreiber wird keinem anderen Netzbetreiber Zugriffsrechte auf seine Netzknoten geben. Daher wird ein TM nur innerhalb eines Netzes eines Providers eingesetzt.

Ob eine Netzgrenze zugleich auch eine Grenze zwischen zwei RM-Domänen darstellt, hängt davon ab, ob ein Netzbetreiber einem anderen Netz- oder Dienstbetreiber einen Zugriff auf seine Netztopologiedaten gewähren will. In den meisten Fällen wird dies nicht der Fall sein, sodass auch ein RM nur für die Ressourcen eines Netzbetreibers eingesetzt wird und diese lediglich bis an die Netzgrenze verwaltet.

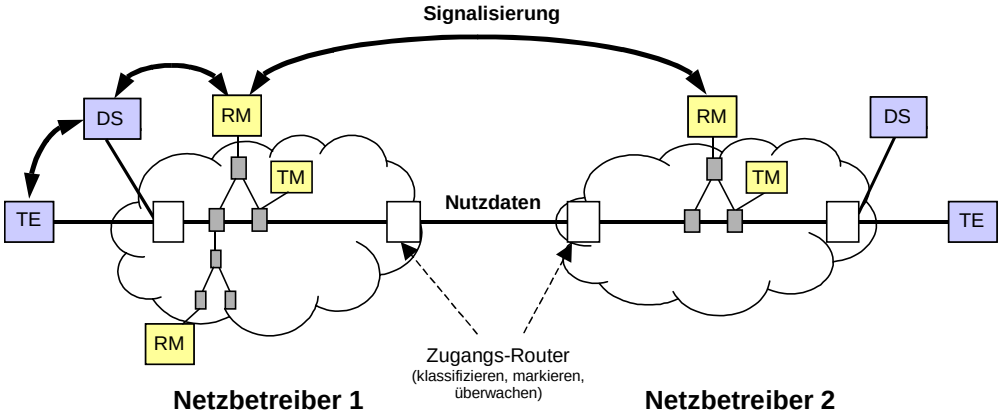


Abbildung 3-8: RM-System mit zwei Netzbetreibern (TM-Domänen)

Eine Netzgrenze zwischen zwei Betreibern stellt darüber hinaus auch eine Vertrauensgrenze hinsichtlich des zu transportierenden Paketverkehrs dar. Nur innerhalb einer einheitlichen administrativen Netzumgebung (TM-Domäne) kann den Endgeräten von Seiten des Netzes das notwendige Vertrauen entgegengebracht werden.

In einer „offenen“ Umgebung dagegen kann bei einer beliebigen Applikation nicht automatisch davon ausgegangen werden, dass sie sich an die bei der Reservierung mit dem Ressourcenmanagement getroffenen Abmachungen hält. Kritische Punkte dabei sind die Einhaltung der Verkehrsparameter der Quelle sowie die korrekte Klassifizierung und Markierung der IP-Pakete. Beim Übergang von zwei administrativen Domänen wollen daher beide Netzbetreiber am Zugang ihres Netzes die hereinkommenden Verkehrsströme kontrollieren. Dazu werden die Verkehrsparameter der jeweiligen Quelle überprüft (*Policing*). Darüber hinaus verwendet jeder Netzbetreiber normalerweise eine andere Anzahl von Dienstklassen und eine unterschiedliche Dienstgütespezifikation oder Kennzeichnung der Pakete, so dass an den Netzgrenzen erneut eine Klassifizierung und Markierung der IP-Pakete vorgenommen werden muss.

3.4 Signalisierungsbeispiel

Im Folgenden wird ein Signalisierungsbeispiel des Ressourcenmanagements für eine große Firma gegeben. Die Firma soll mehrere Standorte und Teilnetze besitzen, die jeweils vor Ort von einer eigenständigen Netzadministration verwaltet werden. Es soll eine unidirektionale Echtzeitdatenverbindung zwischen zwei Mitarbeitern über drei Teilnetze hinweg aufgebaut werden. Dazu wird eine Ende-zu-Ende Reservierung vorgenommen. An dem Verbindungsaufbau sind neben den Terminals der Teilnehmer im Zugangsbereich eine Dienststeuereinheit und für jedes Netz ein RM beteiligt. Der Daten- und der Signalisierungspfad des Beispielszenarios sind in Abbildung 3-9 dargestellt.

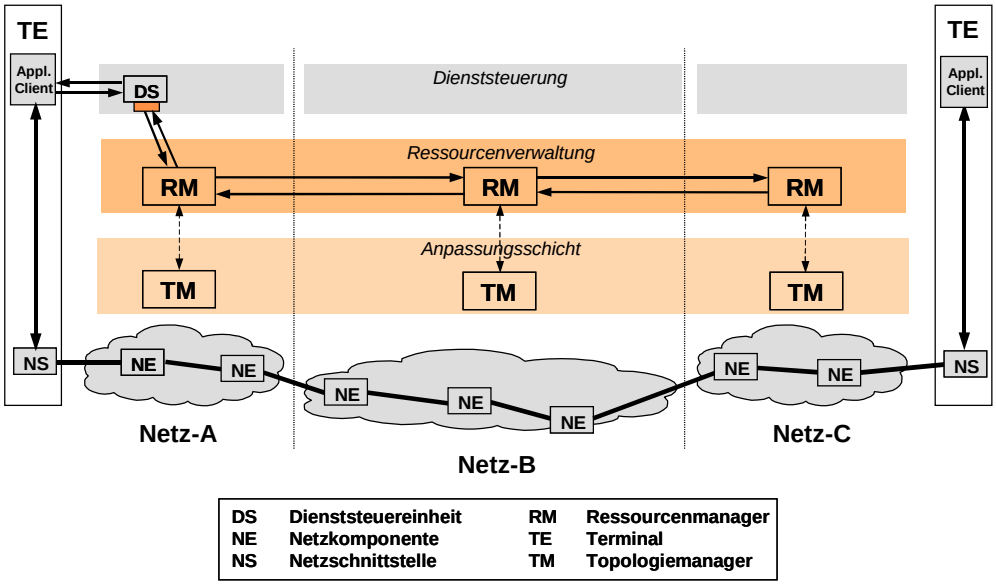


Abbildung 3-9: Beispielszenario

Das Signalisierungsbeispiel stellt den generellen Ablauf einer Ende-zu-Ende Reservierung und deren Einbettung in den Verbindungsaufbau einer Dienststeuerung dar. Das Beispiel wurde unabhängig von den protokollspezifischen Eigenschaften einer speziellen Dienststeuerung gewählt, so dass sich im Ablauf der Reservierung je nach gewählter Signalisierungsarchitektur Abweichungen ergeben können (siehe Kapitel 6). Ferner wurde in diesem Beispiel auf eine Überwachung der Verkehrsquelle im Netz verzichtet. Die Bezeichnungen von Nachrichten der Dienstsignalisierung zwischen den Teilnehmern sowie zwischen Teilnehmern und Dienststeuerung sind abstrahiert dargestellt.

Zum Zeitpunkt des Verbindungsaufbaus kennen alle Teilnehmer die ihnen zugeordnete Dienststeuereinheit DS, alle DS die ihnen zugeordneten RM und alle RM ihre Nachbar-RM. Ferner besitzen alle RM bereits eine vollständige Sicht auf die Topologie ihrer Domäne.

Die drei Phasen einer Verbindung „Aufbau“ (P_1), „Nutzdatentransfer“ (P_2) und „Abbau“ (P_3) sind in einem Nachrichtenflussdiagramm dargestellt (siehe Abbildung 3-10). Es zeigt zwei getrennte Signalisierungsvorgänge, die während des Auf- bzw. Abbaus einer Verbindung stattfinden. Im ersten Schritt teilt der initiierende Teilnehmer T_A seiner DS mit, den Dienst starten zu wollen. Der Zugang zu den Netzressourcen ist mit dem Dienstzugang unmittelbar gekoppelt. Im zweiten Schritt versucht der Teilnehmer T_A den Dienst zu dem gewünschten Zielteilnehmer zu etablieren.

Es wird nun der **Signalisierungsablauf** genauer betrachtet. Zunächst richtet der Teilnehmer T_A seinen Verbindungswunsch zu Teilnehmer T_B an seine DS. Der Verbindungswunsch enthält alle wichtigen Verbindungsparameter. Die DS überprüft daraufhin die Zugriffsberechtigung des Teilnehmers T_A auf den Dienst, ermittelt die IP-Adresse des Zielteilnehmers T_B und stößt bei dem ihr zugewiesenen RM_1 einen Reservierungsvorgang an. Dieser nimmt die Reservierungsanfrage der DS entgegen, bestimmt die erforderliche Dienstklasse, ermittelt den Datenpfad durch seine Domäne und überprüft die Netzressourcen dieser Dienstklasse entlang dieses Pfades. In diesem Beispiel liegt der Zielteilnehmer T_B außerhalb der Domäne von RM_1. RM_1 stellt fest, dass innerhalb seiner Domäne, d.h. vom Terminal des initiierenden Teilnehmers bis hin zur Domänengrenze noch ausreichend freie Ressourcen verfügbar sind, reserviert die notwenigen Ressourcen, ermittelt den Nachbar-RM (RM_2) und leitet die Reservierungsanfrage an diesen weiter. RM_2 bestimmt den Pfad von der Domänen-Grenze zu RM_1 in Richtung des Zielteilnehmers und stellt ebenfalls fest, dass dieser außerhalb seiner Domäne liegt. Er verfährt wie RM_1 und leitet dann die Reservierungsanfrage an RM_3 weiter. RM_3 erkennt nun, dass T_B in seiner Domäne liegt und kann die Reservierung bis zu seinem Terminal durchführen. Danach bestätigt er RM_2 die erfolgreiche Ende-zu-Ende Reservierung. Diese Bestätigung wird entlang des Reservierungspfades über RM_1 zurück an die DS_1 weitergeleitet. DS_1 signalisiert daraufhin dem T_A, dass dieser nun mit dem Verbindungsaufbau beginnen darf.

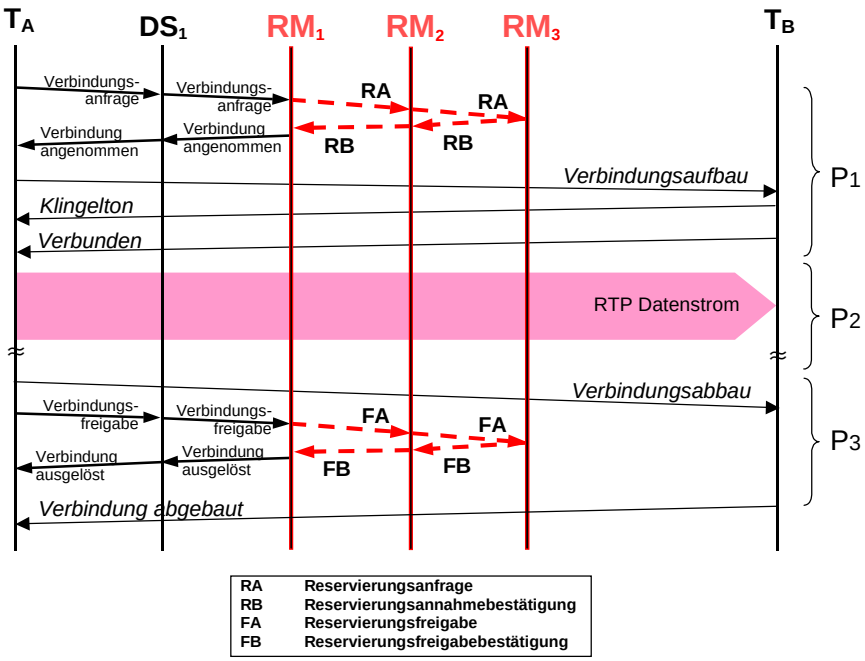


Abbildung 3-10: Nachrichtenflussdiagramm

Der Teilnehmer T_A etabliert nun eine Nutzdatenverbindung zum Teilnehmer T_B und beginnt im Anschluss daran mit der Nutzdatenübertragung. Dieser Teil der Signalisierung erfolgt direkt zwischen den Terminals.

Die Beendigung des Dienstes wird von T_A veranlasst und erfolgt in umgekehrter Reihenfolge. Zunächst wird T_B der Verbindungsabbauwunsch mitgeteilt, danach wird DS_1 das Dienstende signalisiert. Um die nicht mehr benötigten Ressourcen jedoch so schnell wie möglich anderen Netzteilnehmern zur Verfügung zu stellen, erfolgt die Signalisierung an die DS parallel zum Verbindungsabbau. T_A teilt der DS_1 das Dienstende zu einem frühestmöglichen Zeitpunkt mit und veranlasst dadurch die Freigabe der reservierten Netzressourcen. Die Prozedur zur Freigabe der Ressourcen erfolgt analog zur Reservierung.

3.5 Einordnung der RM-Architektur

Zum Abschluss des Kapitels soll die RM-Architektur zunächst mit den beiden Ansätzen der IETF, IntServ und DiffServ, verglichen werden. Die Ergebnisse des Vergleichs werden in Tabelle 3-1 und Tabelle 3-2 zusammengefasst. Im Anschluss daran wird die RM-Architektur in die Reihe der in Kapitel 2 diskutierten Architekturen eingeordnet.

Alle Architekturen verfolgen das gleiche Ziel, eine bessere Übertragungsqualität als den bislang im Internet realisierten „Best Effort"-Übertragungsdienst zu bieten. Wie aus Kapitel 2 jedoch bekannt ist, liegen den Architekturen unterschiedliche Vorstellungen hinsichtlich der zu erzielenden Dienstgüte zugrunde.

3.5.1 DiffServ

Während die RM-Architektur harte und weiche Dienstgütegarantien gewähren kann, ist das Dienstgüteverständnis von DiffServ ein völlig anderes. DiffServ hat kein absolutes Qualitätsmaß, sondern ein relatives. Dienstgütegarantien sind demzufolge in einer DiffServ-Architektur nicht möglich.

Bei der RM-Architektur wird von der DiffServ-Architektur das Prinzip der Dienstklassen übernommen. Wie bei DiffServ werden in den Netzknoten nur Funktionen ausgeführt, die zur Weiterleitung der Pakete notwendig sind. Alle Funktionen der Verkehrskontrolle wie z.B. Klassifizierung, Markierung oder Überwachung der Paketströme werden ebenso wie bei DiffServ am Netzzugang durchgeführt. Im Gegensatz zu DiffServ bezieht sich die Überwachung der Datenströme jedoch auf eine einzelne Verbindung und nicht auf ein Aggregat einer Verkehrsklasse.

Das RM-Konzept ist dem des Bandwidth Brokers ähnlich. Im Gegensatz zum Bandwidth Broker findet die Zugangskontrolle nicht allein am Netzzugang statt, sondern auf allen Links entlang des Datenpfades. Nur so ist es für das Ressourcenmanagement möglich, den Teilnehmern echtzeitkritischer Anwendungen harte Dienstgütegarantien auf Basis der DiffServ-Technologie in den Netzknoten zu gewähren.

3.5.2 IntServ

Die RM- und die IntServ-Architektur sind sich hinsichtlich der erzielbaren QoS-Garantie sehr ähnlich, unterscheiden sich jedoch stark bezüglich der zugrunde liegenden Konzepte. Daher werden für diesen Vergleich die bereits aus Kapitel 2 bekannten Kriterien herangezogen: Granularität, Komplexität/Skalierbarkeit, Migration/Management-Overhead.

Granularität
Ein wesentlicher Unterschied zwischen der IntServ- und RM-Architektur besteht in der Flexibilität und Granularität, mit der die individuellen Dienstgüteanforderungen der Teilnehmer berücksichtigt werden können. Während bei der IntServ-Architektur ein Teilnehmer die Dienstgüte frei wählen kann, ist die Granularität des Dienstgüteangebotes bei der RM-Architektur aufgrund des Dienstklassenkonzeptes geringer.

Komplexität/Skalierbarkeit
Im Gegensatz zur IntServ-Architektur erfolgt die Reservierung nicht in den Netzknoten, sondern „virtuell" in einem eigenen Server (RM), der nur für die Verarbeitung der Signalisierungsnachrichten verantwortlich ist. Dadurch müssen keine Zustände pro Verbindung (Flow) oder Paket in den Netzknoten gespeichert werden. Für große Netze wird zudem ein Domänen-Konzept mit einer entsprechenden inter-RM Signalisierung eingeführt. Durch den zentralisierten Ansatz des RM können ferner Ende-zu-Ende Reservierungen

schneller etabliert werden, da in der Regel weniger Signalisierungsinstanzen beim Aufbau beteiligt sind als bei IntServ. Zudem ist die Reaktionszeit des Systems nicht mehr so stark von der Auslastung des Netzes abhängig.

Ein weiteres Unterscheidungsmerkmal ist das Reservierungsprotokoll (siehe nachfolgenden Protokollvergleich). Eine Reservierungsnachricht der RM-Architektur bezieht sich auf den gesamten Kontext einer Sitzung und nicht nur auf einen einzelnen IP-Flow wie beispielsweise bei RSVP. Dies führt zu einer Reduktion des Signalisierungsverkehrs und damit zu einer besseren Skalierbarkeit der RM-Architektur.

Migration / Management-Overhead
Die Einführbarkeit einer QoS-Architektur hängt maßgeblich von der Anzahl zu modifizierender und zu installierender Systeme ab.

Die IntServ-Technologie muss sowohl auf allen Knoten des Netzes als auch auf allen Endgeräten eingeführt werden. Sie basiert auf komplexen Prozessen der Paketverarbeitung und der Signalisierung. Die DiffServ-Technologie hingegen basiert auf robusten und einfachen Mechanismen, die zur Zeit von einer wesentlich größeren Anzahl von Herstellern und Geräten unterstützt werden als die IntServ-Technologie.

Die RM-Architektur setzt auf der DiffServ-Technologie auf, entkoppelt aber das Ressourcenmanagement von dem darunter liegenden Netz. Dies wird durch die Trennung der Funktionen „Topologieerkundung“ und „Ressourcenverwaltung“ erzielt. Der in Kapitel 6 näher vorgestellte Topologieerkundungsdienst lässt sich leicht in einem Intranet installieren, da er für die Kommunikation mit den Netzknoten standardisierte Mechanismen verwendet. Die RM-Architektur basiert ferner auf einer flexiblen Anbindung an Dienststeuereinheiten (SIP, H.323). Sie kommt dadurch ohne eine Schnittstelle zum Teilnehmer aus und kann an eine beliebige Dienstsignalisierung angepasst werden (siehe DS-Proxy: Abschnitt 6.6.1). Das verbessert die Einführbarkeit, bewirkt jedoch zugleich, dass die RM-Architektur nicht mehr unabhängig von der Dienststeuerung ist.

Die Anzahl der zu verwaltenden Komponenten im Netz und damit der Aufwand bezüglich der Installation, Versionsverwaltung und Fehlerbehebung bei möglichen Komponentenausfällen ist durch den zentralisierten RM-Ansatz stark reduziert. Auf der Seite des Netzes können zur Konfiguration der Knoten und Endgeräte Autokonfigurationsmechanismen *(Policies*) verwendet werden. Die RM-Architektur selbst besitzt ebenfalls entsprechende Verfahren zur Erkennung und Registrierung von benachbarten Komponenten (siehe Systeminitialisierung: Abschnitt 6.4.4) sowie eine Überwachungsfunktion der Reservierungszustände (siehe z.B. Request Handler: Abschnitt 6.6.3).

Dienstgütegarantie
Auf Signalisierungsebene wird von der IntServ-Architektur das Prinzip einer Ende-zu-Ende Reservierung pro Verbindung übernommen. Beide Architekturen können für die Zugangskontrolle dieselben Methoden verwenden.
Ein Unterschied zwischen der IntServ- und der RM-Architektur allerdings liegt in dem Puffermodell. Bei IntServ werden Puffer und Leitungskapazitäten explizit pro Flow reserviert und dadurch eine maximale Trennung der einzelnen Verkehrsströme erreicht. Im Gegensatz dazu verwendet die RM-Architektur das Prinzip der Dienstklassen. Das Zugangskontrollverfahren kann nur die Bedienrate des gemeinsamen Puffers kontrollieren, aber nicht einem einzelnen Flow des Verkehrsgemisches einen bestimmten Bedienratenanteil garantieren. Um dieselbe Dienstgütegarantie wie IntServ gewähren zu können, müssen daher homogene Verkehrsgemische gebildet und Flow-basierte Filtereinheiten am Netzzugang eingerichtet werden.

Die Bildung homogener Verkehrsgemische erfordert im Netz die Einrichtung vieler Dienstklassen, was wiederum die Komplexität der Architektur erhöht.

Zusammenfassend werden die Architekturmerkmale einander gegenübergestellt. Betrachtet werden Dienstmerkmale, erzielbare Dienstgüte, Skalierbarkeit, Einführbarkeit und der Managementaufwand (Tabelle 3-1). Ein Vergleich der Reservierungsverfahren von RM- und IntServ-Architektur findet im Anschluss statt (Tabelle 3-2).

	RM-Architektur	IntServ	DiffServ
Reservierungsprotokoll	ja	ja	nein
Dienstmerkmale			
Abhängigkeit von einer Dienstarchitektur	ja	nein	nein
Vorabreservierungen	ja	nein	-
Dienstgüte			
Harte QoS-Garantie	ja	ja	nein
Weiche QoS-Garantie	ja	ja	nein
Granularität	gering	hoch	gering
Reservierungsaufbauzeiten	sehr kurz	kurz	-
Skalierbarkeit			
Trennung von Paketverarbeitung und Zugangskontrolle	ja	nein	ja
Anzahl der Puffer in den Knoten	gering	sehr hoch	gering
Domänenkonzept	ja	nein	ja
Signalisierungsverkehr	gering	hoch	-
Migration / Management			
Teilnehmerschnittstelle	nein	ja	nein
Autokonfigurationsmechanismen	ja	ja	ja
Anzahl der zu wartenden Instanzen	wenige	viele	wenige

Tabelle 3-1: Architekturvergleich

Protokollvergleich

Wird in einem IntServ-System das RSVP-Protokoll für die Reservierung eingesetzt, ist das durch die Signalisierung verursachte Verkehrsaufkommen wesentlich größer als bei einem RM-System. Das liegt zum einen an den weichen Reservierungszuständen (*soft-states*), die periodisch durch neue Reservierungsnachrichten aufgefrischt werden müssen (*refresh*). Zum anderen wird für jede Multimediaverbindung einer Sitzung ein eigener Reservierungsvorgang angestoßen. Ein Domänenkonzept zur Reduzierung der Reservierungsnachrichten ist beim RSVP-Protokoll nach RFC 2205 nicht vorgesehen. Ein Vergleich der Reservierungsprotokolle ist in Tabelle 3-2 aufgeführt.

Reservierungsverfahren	RM-Architektur	IntServ (RSVP, RFC 2205)
Verbindungen		
Punkt-zu-Punkt	ja	ja
Punkt-zu-Mehrpunkt	ja	ja
Multicast-Unterstützung	nein	ja
Reservierungszustände		
Typ	hart	weich
Abbau	explizit	explizit / Timeout
Signalisierung		
Initiator einer Reservierung	Initiator einer Sitzung	Sender eines Datenstroms
Kontext einer Transaktion	Sitzung	einzelner Datenstrom
Festlegung der QoS-Parameter	Initiator	Empfänger
Weitere Merkmale		
Automatische Reaktion auf Routenänderung	ja	ja
Modifikation einer Reservierung	ja	ja

Tabelle 3-2: Protokollvergleich

3.5.3 Fazit

Zusammenfassend kann man sagen, dass für echtzeitkritische Anwendungen sowohl die IntServ- als auch die RM-Architektur für einen Netzbetreiber in Frage kommen. Die RM-Architektur lässt sich jedoch in bestehende Systemumgebungen leichter einführen und ist zudem für größere Netze besser skalierbar.

Die RM-Architektur sieht eine Trennung zwischen der Netz- und der Reservierungsebene vor. Darüber hinaus wird das Reservierungsprotokoll flexibel an die Dienstsignalisierung angebunden. Für die Einführung eines RM-Systems sind nur minimale Veränderungen an wenigen Servern notwendig. Das RM-System stellt seinem Betreiber einen erweiterbaren Satz von Zugangskontrollverfahren für harte und weiche QoS-Garantien zur Verfügung. Es erlaubt eine flexiblen Zuordnung der Verfahren zu einzelnen DS-Klassen und somit die Realisierung unterschiedlicher Dienstgütespezifikationen.

Die RM-Architektur lässt sich daher in das Bewertungsschema aus Kapitel 2 zusammen mit den anderen Ansätzen wie folgt einreihen:

	IntServ	DARWIN	SCORE	RM	RRA	AQUILA	Egress-AC	TEQUILA	BB	DiffServ-AS
Dienstgütegarantie	++	++	++	++	o	o	o	o	-	--
Granularität	++	+	+	o	-	-	o	o	-	-
Komplexität	--	-	-	o	o	+	o	o	+	++
Einführbarkeit	--	--	--	+	o	-	-	-	o	++

Tabelle 3-3: Einordnung der RM-Architektur

Die RM-Architektur erfüllt damit die gesetzten Anforderungen. In den nachfolgenden Kapiteln wird die Realisierung der hier vorgestellten Konzepte näher beschrieben.

4. Messung und Charakterisierung von Echtzeitverkehrsquellen

Dieses Kapitel liefert die Grundlagen für die Auswahl geeigneter Zugangskontrollverfahren für die RM-Architektur. Dazu wurden Messungen zur Charakterisierung von realen Echtzeitverkehrsquellen in IP-Netzen durchgeführt. Die Messungen werden anhand verschiedener statistischer Methoden analysiert. Die Ergebnisse der statistischen Untersuchungen können als Basis für weitere Arbeiten zur Quellenmodellierung dienen.

4.1 Motivation

Bei der Betrachtung von Datenquellen sind grundsätzlich zwei Arten der Modellierung zu unterscheiden: Die Modellierung auf Netzebene und die Modellierung auf Teilnehmerebene. Die Modellierung auf Netzebene betrachtet das Sendeverhalten einer einzelnen Datenquelle mit dem Ziel, den Ressourcenbedarf im Netz für eine solche Datenverbindung abzuschätzen. Die Modelle beschreiben häufig Ankunftsprozesse in einem Netzknoten und können beispielsweise für Planungs- und Dimensionierungsaufgaben verwendet werden. Die Modellierung auf Teilnehmerebene (Quellenmodellierung) hingegen betrachtet das Verhalten einer Applikation mit dem Ziel, eine allgemeine, d.h. nicht zweckgebundene, mathematische Beschreibung der Eigenschaften der Quelle zu finden.

Im Rahmen des Ressourcenmanagements liegt der Schwerpunkt der Betrachtungen in der Modellierung auf Netzebene. Es werden vorkonfigurierte Ressourcen im Netz über eine Zugangskontrolle verwaltet, um die Verbindungsqualität zu sichern. Dazu ist es notwendig, den Ressourcenbedarf der verschiedenen Quellentypen bestimmen zu können. Den Ressourcenbedarf wiederum kann man nur bestimmen, wenn man das Verkehrsverhalten der Quellen kennt und entsprechend modellieren kann.

In der Literatur sind Verkehrsquellen bereits mehrfach untersucht worden. Seitdem es Telefonnetze gibt, wurden Modelle von Verkehrsquellen entwickelt. Zunächst lag der Schwerpunkt der Forschung auf reinen Sprachquellen. Mit der Standardisierung des ATM rückten zunehmend auch Video- und Datenquellen in den Fokus des Interesses. In den meisten Fällen wird ein Kodierungsverfahren in einer idealen Systemumgebung beschrieben und modelliert. Die Modellierung bezieht also das System, auf dem die Anwendung läuft, nicht mit ein. Das liegt vor allem daran, dass ATM-Quellen und -Endgeräte für den praktischen Einsatz nie realisiert wurden und dadurch auch nicht untersucht werden konnten. Nachdem die ATM-Technologie noch vor ihrer Einführung als dienstintegrierendes Netz von der Internet-Technologie überrollt wurde, gilt es nun zu überprüfen, ob die bislang getroffenen Annahmen auch für Quellen in IP-Netzen gültig sind.

Betrachtet man spezielle Endgeräte wie das klassische Telefon, so wurden diese für einen speziellen Echtzeitdienst, z.B. „Sprache“ konzipiert. Betrachtet man hingegen PC-basierte Echtzeitanwendungen, kann es sein, dass man im Netz zu optimistische Annahmen über das Verhalten der Verkehrsquelle trifft. Auf einem Rechner sind normalerweise mehrere Prozesse gleichzeitig aktiv und diese Prozesse konkurrieren um die gemeinsamen Systemressourcen (Speicher, CPU-Zyklen). Daraus können sich Verzerrungen (Jitter) des idealen Verkehrsverhaltens ergeben, welche das Verhalten der Quelle nachhaltig prägen.

Ferner ist der Paketierungsprozess, d.h. die Aufbereitung der kodierten Nutzdaten für die Übertragung, in einem IP-Netz ein anderer als z.B. in einem ATM-Netz. Beim Vorgang der Paketbildung werden den Nutzdaten sehr viele Steuerinformationen hinzugefügt. Bei der Übertragung ergibt sich häufig ein sehr ungünstiges Verhältnis von Nutzdaten zu tatsächlich übertragenen Daten auf dem Netz. Dies führt dazu, dass manche Anwendungen größere Nutzdateneinheiten bilden und diese als Ganzes übertragen. Betrachtet man einen kontinuierlichen Strom von echtzeitkritischen Daten einer Applikation, so führt dieser Vorgang bereits im Endgerät zu einer Art „Verklumpung“ des Verkehrs.

Daraus ergeben sich für IP-Verkehrsquellen zahlreiche neue Eigenschaften, die bislang nur unzureichend untersucht wurden.

4.2 Zielsetzung

Die Zielsetzung der Messungen ergibt sich aus den Anforderungen an die Zugangskontrolle der RM-Architektur. Aufgabe der Netzzugangskontrolle ist es, den Nutzern von Echtzeitanwendungen für die Übertragung ihrer IP-Pakete harte oder weiche QoS-Garantien zu gewährleisten. Zwei wichtige QoS-Parameter sind die Paketverlustwahrscheinlichkeit und die Paketverzögerung im Netz.

Für die Verzögerungen und Verluste im Netz sind Pufferfüllstände in den Netzknoten entscheidend. Wachsen die Pufferfüllstände an, kann es bei Erreichen der Pufferkapazität zu Paketverlusten kommen. Um die Pufferfüllstände kontrollieren zu können, muss dem RM-System zum Zeitpunkt der Verbindungsannahme die maximale Datenmenge bekannt sein, die pro Zeitintervall an einem Puffer eintreffen kann. Dazu muss das RM von jeder einzelnen Verkehrsquelle im Netz die maximale Datenmenge kennen, die diese pro Zeitintervall in das Netz senden kann.

Das Sendeverhalten realer Verkehrsquellen wird anhand von Messungen mit kommerziellen Applikationen und Systemen ermittelt. Ziel der Messungen ist es, die Grundlagen zu schaffen, um das Verkehrsverhalten von realen Echtzeitverkehrsquellen in IP-Netzen untersuchen zu können.

Anhand der Messungen können:

- wesentliche Merkmale dieser Verkehre identifiziert,
- eine geeignete Verkehrsbeschreibung in Form eines Parametersatzes gefunden und
- die Anwendbarkeit von Zugangskontrollverfahren auf diese Art von Verkehren untersucht werden.

In diesem Kapitel liegt der Schwerpunkt auf der Planung und Durchführung von Messungen, der Analyse der gemessenen Verkehre sowie der Identifikation von charakteristischen Merkmalen.

4.3 Vorgehensweise

Das Sendeverhalten realer Verkehrsquellen wird anhand von Messungen mit kommerziellen Applikationen und Systemen ermittelt. Für die Planung und Durchführung der Messungen müssen die Ziele genauer definiert werden.

Aus Sicht des Netzes (RM-Systems) wird das Verhalten einer Verkehrsquelle durch den Teilnehmer, die Applikation und das System bestimmt (siehe Abbildung 4-1).

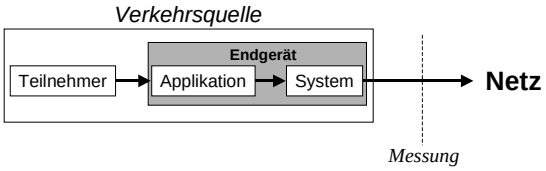


Abbildung 4-1: Einflüsse auf das Verhalten einer Verkehrsquelle

Das Sendeverhalten einer Verkehrsquelle hängt von vielen Parametern ab. Primär wird es durch die Eigenschaften der Applikation geprägt. Der Teilnehmer und das System können das Sendeverhalten jedoch mehr oder weniger stark beeinflussen.

Die Applikationen bieten den Teilnehmern verschiedene Einstellungsmöglichkeiten (Codec, Codec-Parameter). Für jede Einstellung besitzt eine Applikation feste Vorgaben bezüglich der Erzeugung von Nutzdatenpaketen (Paketierung). Dabei existieren für den Applikationsentwickler Freiheitsgrade, wieviele kodierte Datenblöcke (*Frames*) in einem Nutzdatenpaket übertragen werden. Der Teilnehmer beeinflusst z.B. eine Videoapplikation durch die Bestimmung der Beleuchtungsverhältnisse und sein Bewegungsverhalten vor der Kamera. Das System schließlich beeinflusst das Verhalten der Verkehrsquelle durch schwankende Verzögerungen beim Aussenden der IP-Pakete.

Um ein möglichst breites Spektrum an unterschiedlichen Verkehrsquellen zu bekommen, wurden verschiedene Applikationen verwendet (PC-basierte Softwarepakete mit und ohne Hardware-Unterstützung) und zudem die Einstellungen variiert. Um die Einflüsse des Systems untersuchen zu können, wurden als Endgeräte Rechner mit unterschiedlichen Prozessoren und Betriebssystemen verwendet.

Zur Bestimmung der worst case Quellencharakteristik muss zu jeder Einstellung der Applikation das worst case Nutzerverhalten gefunden werden. Dazu wurden pro Applikation und Einstellung mehrere Messungen mit unterschiedlichem Nutzerverhalten durchgeführt.

Bei den Messungen wird neben dem Nutzdatenverkehr auch der Signalisierungsverkehr betrachtet. Der Signalisierungsverkehr in IP-Netzen wurde bislang in der Literatur vernachlässigt. Dies kann jedoch insbesondere bei niederbitratigen Sprachquellen (Datenrate < 6 kbit/s) zu einer Unterschätzung des Quellverkehrs führen (siehe Abschnitt 4.5.6).

Die erzeugten Messreihen wurden anschließend (*offline*) nach verschiedenen statistischen Verfahren analysiert. Ziel der Analyse ist es, alle wesentlichen Merkmale dieser Verkehre zu identifizieren.

In Abbildung 4-2 wird die genaue Vorgehensweise dargestellt. Im Vorgriff auf Kapitel 5 ist im unteren Teil der Abbildung ersichtlich, wie die Messungen später verwendet werden.

Liegen mehrere Messreihen mit identischer Applikation, Einstellung und System vor, kann nach erfolgter Charakterisierung das worst case Verhalten der Verkehrsquelle ermittelt werden. Danach werden Verfahren zur Berechnung Effektiver Bitraten gesucht, anhand derer eine Netzzugangskontrollfunktion für die RM-Architektur realisiert werden kann (siehe Kapitel 5).

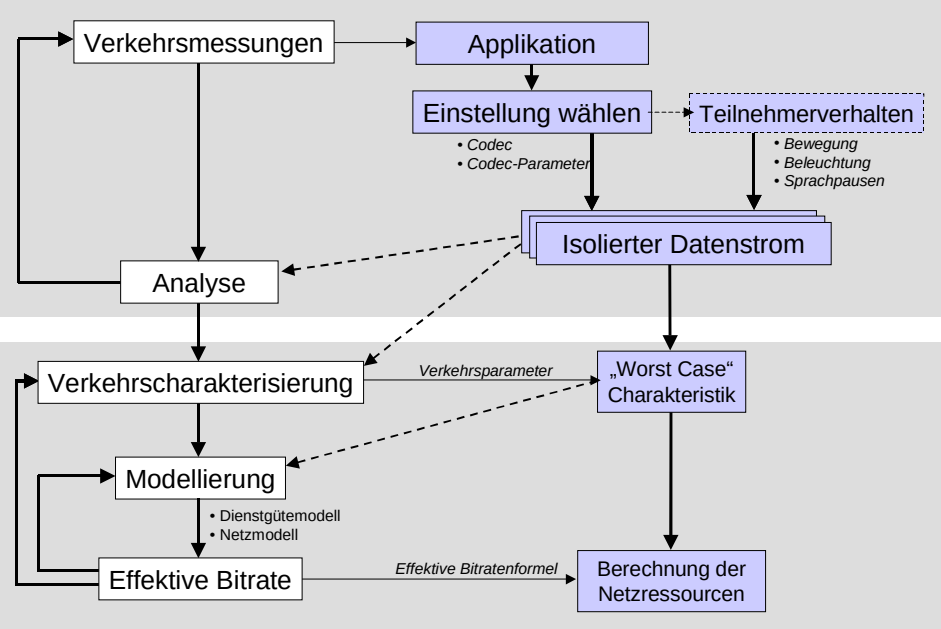


Abbildung 4-2: Vorgehensweise bei der Charakterisierung von Echtzeitverkehren

4.4 Messungen

In den vorangegangenen Abschnitten wurden Ziele und Vorgehen definiert. In diesem Abschnitt wird nun die Durchführung der Messungen beschrieben.

Aus Abbildung 4-1 geht hervor, dass die Messungen im Zugangsbereich des Netzes erfolgen müssen, um alle Eigenschaften einer Verkehrsquelle erfassen zu können.

Führt man die Messungen auf demselben Rechner durch, auf dem die Applikation läuft, beeinflussen sich die Anwendungsprozesse gegenseitig (Jitter). Dies macht erforderlich, dass die Applikation und das Messprogramm auf unterschiedlicher Hardware laufen.

Führt man die Messungen dagegen im Netz durch, so kann es zu Überlagerungen mit anderen Verkehren im Netz kommen. Der Messaufbau ist dabei so zu wählen, dass ein Einfluss anderer Verkehre ausgeschlossen werden kann.

Die Anforderungen an die Durchführung können wie folgt zusammengefasst werden:

- die Messungen müssen im Zugangsbereich des Netzes durchgeführt werden und
- der Einfluss anderer Verkehre im Netz muss minimiert werden.

Gemäß dieser Vorgaben wurde 1998 am Lehrstuhl für Kommunikationsnetze eine Serie von ca. 120 Einzelmessungen durchgeführt [SS98] und ausgewertet [Kli98]. In den Jahren 1999 und 2000 folgten erneut Messungen mit verbesserter Hardware [Wal00].

4.4.1 Messaufbau

Um das Verhalten einer einzelnen Verkehrsquelle erfassen zu können, müssen Störeinflüsse durch andere Verkehre im Netz ausgeschlossen werden.

Durch den zeitgleichen Mehrfachzugriff unterschiedlicher Quellen auf das gemeinsame Medium (Ethernet, 10 Mbit/s) werden die zeitlichen Abstände der Pakete im Netz verzerrt (z.B. CSMA-CD: Kollisionen, *Backoff*-Algorithmus, *Capture*-Effekt).

Die Messreihen wurden daher in einem isolierten Subnetz durchgeführt, welches aus zwei Endgeräten, einem Messrechner und einem Ethernet-Hub besteht. Ferner wurde pro Messreihe und Medium (Audio, Video) nur eine Datenverbindung zwischen einem Quell- und Zielrechner aufgebaut. Die Messreihe wurde nur dann ausgewertet, wenn während der Messdauer kein Verkehr von anderen Anwendungen (Hintergrundverkehr) auf dem Netz vorhanden war.

Als Verkehrsquellen wurden die in den Jahren 1997 und 1998 verfügbaren Videotelefonieanwendungen der bekanntesten Firmen eingesetzt: LiveLAN von PictureTel, Armada Escort25 Pro von Vcon und Netmeeting von Microsoft. Alle diese Systeme basieren auf dem ITU-Standard H.323. Während bei LiveLAN und Armada der Kodier-/Dekodiervorgang von einer Hardware unterstützt wird, handelt es sich bei Netmeeting um eine rein softwarebasierte Lösung. Die Client-Applikationen liefen unter Microsoft Windows'95 auf Intel Pentium-II Rechnern.

4.4.2 Messdatenerfassung

Eine Verkehrsquelle sendet einen Paketstrom in das Netz, der mit Hilfe eines Rechners und eines entsprechenden Programms gemessenen werden kann. Für die Charakterisierung eines Paketstromes werden zu jedem Paket folgende Informationen benötigt:

- Paketlänge und
- zeitlicher Abstand zum darauffolgenden Paket.

Um einzelne Pakete einem Paketstrom zuordnen zu können (siehe unten) sind darüber hinaus weitere Informationen erforderlich:

- IP Quell- und Zieladresse;
- TCP/UDP Quell- und Ziel-Portnummer.

Jedes Paket muss daher zeitlich erfasst und zusammen mit obigen Informationen für die anschließende Analyse gespeichert werden.

Für die Messdatenerfassung wurden verschiedene Messprogramme verwendet:

- *Snoop* unter *Solaris X86,*
- *Tcpdump* und *Ksnuffle* unter *Linux, SuSE 6.4*

Als Hardware kamen Sun Workstations (*Spark 10*), PC-Workstations Pentium-Pro200 (128 MB-RAM, 128 MB-Swap auf eigener 1GB-Meßplatte, SCSI-Hostadapter) und ein *AMD-K6 II 300MHz* zum Einsatz. Während des Messvorgangs wurde darauf geachtet, dass keine anderen Prozesse aktiv waren (z.B. *Linux: Single User Mode*).

Die Messprogramme detektieren Ethernet-Pakete auf dem Netz. Sie besitzen Filterfunktionen und können ganze Pakete oder nur bestimmte Teile (*Header*) auf die Festplatte des Rechners kopieren. In den meisten Messprogrammen kann die Menge der zu speichernden Information explizit beim Programmaufruf angegeben werden. Die Angabe erfolgt in einer Anzahl von Bytes, beginnend mit dem Ethernet-Paketkopf. Mit jedem erfassten Paket wird ein Zeitstempel (Systemzeit) vergeben und zusammen mit den anderen Paketdaten in einer Datei gespeichert.

Bei einer Sitzung werden von den Anwendungen in der Regel mehrere Datenverbindungen (TCP/UDP) aufgebaut. Grundsätzlich kann man zwischen Steuerkanälen und Nutzdatenkanälen unterscheiden. Die Anzahl der Steuer- und Nutzdatenkanäle hängt von der Signalisierungsarchitektur (H.323, SIP) sowie von den Kommunikationsbeziehungen zwischen den Teilnehmern ab.

Bei einer H.323 Protokollarchitektur ergeben sich für die Messungen eines einzelnen H.323-Rufes zwischen zwei Teilnehmern, bestehend aus einer bidirektionalen Sprachverbindung und einer bidirektionalen Videoverbindung, folgende Datenkanäle (Abbildung 4-3): zwei RTP-Nutzdatenkanäle pro Medium, ein RTCP-Kontrollkanal pro RTP-Kanal und zwei H.323 Signalisierungskanäle (H.225.0, H.245). Jeder Kanal ist durch seine IP-Adressen und seine Port-Nummern von Quelle und Ziel eindeutig definiert.

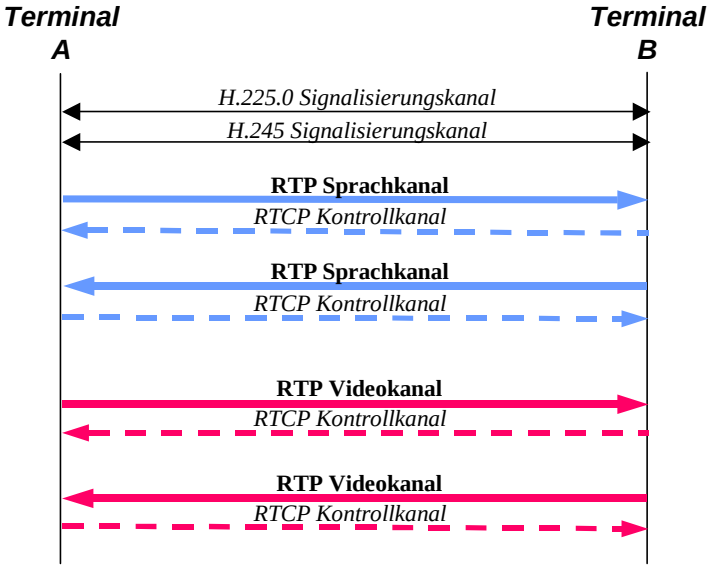


Abbildung 4-3: Datenverbindungen während eines H.323 Rufes

Die Ethernet-Pakete aller Verbindungen werden vom Messprogramm detektiert und als ein Ereignis zusammen mit einem Zeitstempel (Systemzeit) in einer Datei abgelegt. Nach Abschluss der Messung muss für die Analyse der gesamte Verkehr in einzelne Paketströme aufgetrennt werden. Um die einzelnen Pakete ihren unidirektionalen Datenverbindungen zuordnen zu können, werden Verbindungsparameter (IP-Adressen, Port-Nummern) benötigt. Um Signalisierungsdaten von Nutzdaten unterscheiden zu können, ist die Angabe des verwendeten Transportprotokolls (TCP, UDP) hilfreich.

Aus diesen Gründen wurde bei der Messung der komplette Paketkopf, bestehend aus Ethernet-, IP-, UDP/TCP und RTP-Header aufgezeichnet. Filterfunktionen wurden nicht verwendet, um auch den Hintergrundverkehr kontrollieren zu können. Für die durchgeführten Messreihen wurde in den meisten Fällen parallel zu einer Sprachverbindung auch eine Videoverbindung aufgebaut.

4.4.3 Messgenauigkeit

Die Anzeige des Zeitstempels bei der Zeiterfassung mit *Snoop* oder *tcpdump* erfolgt mit einer Auflösung von 10µs. Der Zeitstempel wird erst nach dem vollständigen Empfang des Paketes vergeben. Dadurch werden die Angaben des Empfangszeitpunktes um die Empfangsdauer der Pakete verzögert. Folglich tritt bei der Messung ein deterministischer Messfehler auf, der von der Paketlänge abhängt und nachträglich korrigiert werden kann.

Problematischer sind lastabhängige Fehler bei der Zeitstempelvergabe. Diese konnten mit den zur Verfügung stehenden Geräten nicht ausgeschlossen werden. Je höher die zu messende Paketrate ist und je mehr Information im laufenden Betrieb abgespeichert wird, desto höher ist die Wahrscheinlichkeit von Messfehlern. Das Problem ist dabei der langsame Speicherzugriff auf die Festplatte.

Bei allen Messungen lag die mittlere Auslastung des Netzsegmentes unter 17% und die mittlere Paketrate unter 50 Paketen pro Sekunde. Die Rechnerauslastung lag bei den Messungen mit hohen Nutzdatenraten im Mittel bei 10%. Belastungstests der Messrechner haben ergeben, dass erst ab einer Rechnerauslastung von 80% Fehler auftreten. Als Referenzmessgerät wurde dabei ein LAN-Analyzer von Hewlett Packard (HP 4972A) verwendet. Für die Messreihen konnte jedoch der LAN-Analyzer nicht eingesetzt werden, da er in der vorliegenden Version weder einen ausreichenden lokalen Speicher noch eine Ausgabefunktion der Messdaten auf ein externes Gerät besitzt.

4.4.4 Einflüsse der Hardware

Will man die Einflüsse der Hardware auf das zeitliche Verhalten eines Paketstroms untersuchen, müssen drei Arten von Anwendungen unterschieden werden. Neben den reinen softwarebasierten Systemen (Netmeeting) gibt es auch PC-basierte Anwendungen mit Hardwareunterstützung (LiveLan, Vcon). Der Kompressionsvorgang findet auf eigenen DSPs (Digital Signaling Processor) statt, ohne den Hauptprozessor des PCs zu belasten. Neben den PC-basierten Systemen gibt es auch reine IP-Telefone, auf deren Hardware nur das für die IP-Telefonie notwendige Anwenderprogramm läuft.

Messungen mit IP-Telefonen vom Typ HighPath OptiPoint 500 der Firma Siemens hat ergeben, dass das Verkehrsverhalten der Quelle ziemlich genau dem Verhalten des Kompressionsalgorithmus entspricht. PC-basierte Systeme hingegen zeigen ein wesentlich bursthafteres Verhalten [SRG03]. Im Folgenden werden nur PC-basierte Systeme betrachtet.

Die ersten Messungen wurden mit drei Pentium 100 MHz Rechnern unter Windows95 durchgeführt. Zu einem späteren Zeitpunkt wurden noch einmal Messungen mit einem Pentium-II Rechner unter Windows98 vorgenommen. Ein Vergleich mit den alten Messungen ergab, dass die neuen Messreihen einen viel genaueren Einblick in das Verhalten der Anwendungen ermöglichen. Die größere Prozessorleistung sowie das Betriebssystem Windows98 mit verbessertem Task-Management haben großen Einfluss auf das Quellenverhalten und die Zeitstempelvergabe (siehe Abbildungen 4-4 und 4-5).

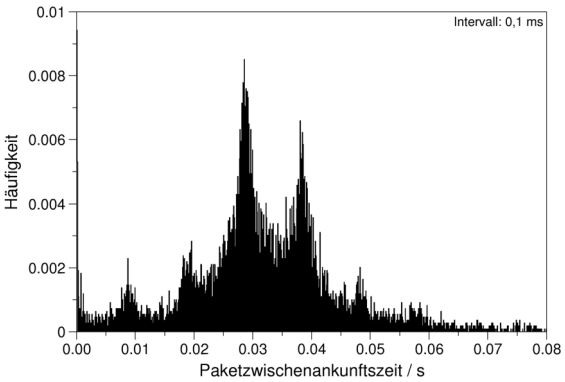


Abbildung 4-4: Verteilung der PZA bei LiveLan Video unter Windows'95 (alte Messungen)

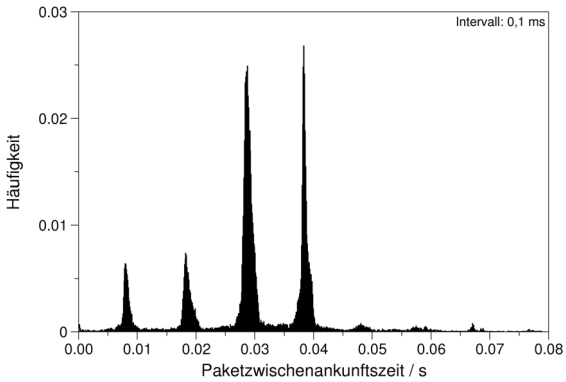


Abbildung 4-5: Verteilung der PZA bei LiveLan Video unter Windows'98 (neue Messungen)

Aufgrund dieser Erkenntnisse wurden alle Messreihen noch einmal mit den moderneren Gerätschaften durchgeführt. Alle nachfolgenden Analysen beziehen sich ausschließlich auf diese Messreihen.

Bei den Hardware-unterstützen Systemen Vcon und LiveLan wirkte sich die verbesserte Rechenleistung des PC-Systems relativ gering auf das Sendeverhalten aus. Demzufolge war

der Einfluss der Hardware bei Netmeeting größer, da sich bei den rein softwarebasierten Systemen die Hardware auch auf das Sendeverhalten auswirkte.

4.5 Statistische Analyse

Im folgenden Kapitel werden mehrere statistische Standardverfahren auf die Messreihen angewendet und folgende Größen ermittelt:

- Mittelwert, Varianz
- Verteilungsfunktion
- Autokorrelationsfunktion
- Hurst-Parameter

Sie dienen jeweils der Charakterisierung von Teilaspekten des Quellenverhaltens. Für die Auswertung der Messdaten wurde das kommerzielle Statistik-Programm SPLUS der Firma *MathSoft* [SPL01] verwendet.

In der Literatur werden zur Charakterisierung und Modellierung von kontinuierlichen Paketströmen häufig zeitliche Intervalle gebildet. Innerhalb dieser Intervalle werden dann die Paketlängen aufsummiert und der Verkehr in Bytes pro Intervall oder als mittlere Datenrate pro Zeitintervall angegeben. Für die Genauigkeit des Modells ist die Wahl der Intervallgröße ausschlaggebend. Je größer das Intervall ist, desto ungenauer werden die möglichen Aussagen z.B. hinsichtlich der zu erwartenden Paketverzögerung in einem Puffer. Ist das Intervall sehr klein (< 10ms), ist die Anzahl der übertragenen Pakete pro Intervall sehr gering und die Verkehrscharakteristik hängt stark davon ab, ob ein gemessenes Paket dem einen oder anderen Intervall zugewiesen wird. Geringfügige Schwankungen beim Aussenden der Pakete oder Störungen beim Messvorgang haben großen Einfluss auf die Verkehrscharakteristik.

Aus diesen Gründen bezieht sich ein Großteil der nachfolgenden Analysen direkt auf die Paketreihen ohne vorherige Intervallbildung.

4.5.1 Kurzbeschreibung der Quellen

Im Folgenden werden die verwendeten Echtzeitverkehrsquellen kurz vorgestellt. In Tabelle 4-1 sind die verschiedenen Anwendungen (*LiveLan LL, VCON, Netmeeting NM*) einander gegenübergestellt. Sie werden nach dem verwendeten Kodierverfahren und den wählbaren Einstellungen miteinander verglichen.

Die Einstellung bei LiveLan und Vcon bezieht sich auf die Bildgröße des übertragenen Videobildes (QCIF:176 x 144 pixel, CIF: 352 x 288 pixel) und die Senderate. Dabei werden die Nutzdatenraten der Video- und Sprachverbindung zusammen gerechnet. Die Senderate ist bei LiveLan und Vcon zwischen 64 kbit/s und 768 kbit/s in mehreren Stufen einstellbar. Diese Raten beziehen sich auf den Ausgang des Coders.

Bei Netmeeting kann das verwendete Kodierverfahren nicht angegeben werden, da in dieser Anwendung mehrere proprietäre Kodierverfahren realisiert sind und der Nutzer nicht direkt ein bestimmtes Verfahren auswählen kann. Auch eine Analyse des Datenfeldes *PayloadType* im RTP-Paketkopf [Sc96] führte zu keinem Ergebnis (*encoding name: „unassigned“).* Es können nur die Bildgröße und die Bildqualität als Parameter gewählt werden.

Anwendung	LL		VCON		NM		
Hardware	analoge Kamera, PCI Karte		digitale Kamera, PCI Karte		digitale Kamera, Parallelport		
Medium	Video	Audio	Video	Audio	Video	Audio	
Einstellungen	64 - 768 Kbit/s		64 - 768 Kbit/s		picture size quality	-	
	QCIF, CIF	-	QCIF, CIF	-			
Kodierverfahren	H.261	G.711	H.261	G.728	proprietär, Windows 98	G.723.1 Silence Suppr.	G.711
Quantisierung (Bit pro Pixel bzw. Sample)	24 Bit	8 Bit	24 Bit	8 Bit	k.a.	16 Bit	8 Bit
Kompressionsfaktor	100	1	100	4	k.a.	20.31	1
Framegröße	< 32 kbyte	8 Bit	< 32 kbyte	10 Bit	k.a.	189 Bit	8 Bit
Framerate	7.5 - 30 Hz	8000 Hz	7.5 - 30 Hz	1600 Hz	k.a.	33 Hz	8000 Hz
Spitzenbitrate (kbit/s)	7758	64	7758	16	k.a.	6.4	64
∅ Rate des Coders (kbit/s)	110 - 704	64	48 - 752	16	k.a.	≤ 6.4	64

Tabelle 4-1: Beschreibung der Quellen

4.5.2 Paket-Overhead

Bei der Übertragung von kontinuierlichen Datenströmen über Paketnetze ist das Verhältnis von Nutzdaten zu den insgesamt übertragenen Daten auf der Leitung (Übertragungsmedium) besonders kritisch. Den Nutzdaten werden Steuerinformationen des Netzes und der Applikation vorangestellt. Die Abbildung 4-6 zeigt den Aufbau eines Ethernet Paketes. Alle Angaben zu den einzelnen Feldern erfolgen in Bytes.

Abbildung 4-6: Aufbau eines Ethernet Paketes

Die Menge der Steuerinformationen ist in IP-Netzen unabhängig von dem verwendeten Kodierverfahren. Somit wird die Übertragung hinsichtlich ihres Ressourcenverbrauches umso ineffizienter, je niedriger die Nutzdatenrate ist. In nachfolgender Tabelle ist der Anteil der Nutzdaten für mehrere Kodierverfahren dargestellt.

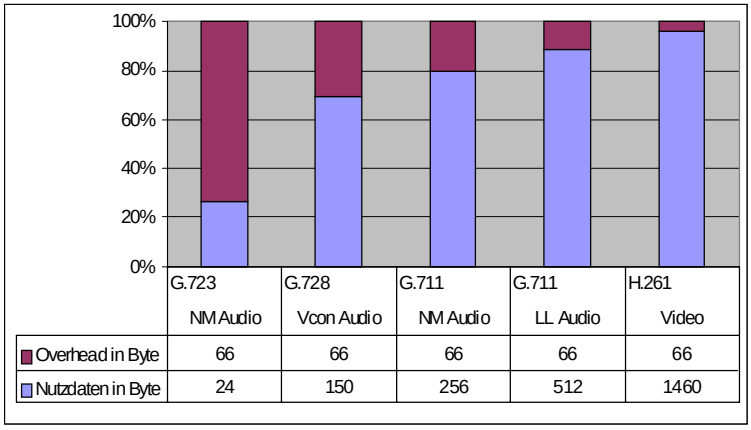


Abbildung 4-7: Paket Overhead

Verfahren zur Komprimierung des Paketkopfes, wie sie von der IETF im Zusammenhang mit drahtlosen Netzen standardisiert wurden [CJ99], [DNP99], werden hier nicht betrachtet.

4.5.3 Analyse des Videoverkehrs

Die Senderate einer Videoverkehrsquelle ist eine statistische Größe und hängt von verschiedenen Faktoren ab. Unabhängig von dem verwendeten Kompressionsverfahren hat die Bildqualität, d.h. die Bildgröße und die Bildwiederholfrequenz (Framerate), einen großen Einfluss auf die zu übertragende Datenmenge. Jeder übertragene Bildpunkt wird in einer bestimmten Form kodiert. Welche Informationsmenge zur Darstellung eines Bildpunktes benötigt wird (Quantisierung), hängt von dem verwendeten Kodierverfahren ab und kann in der Regel dynamisch verändert werden.

Neben der Bildqualität haben auch die Bildinhalte einen mehr oder weniger großen Einfluss auf die Effizienz der Kompression und damit auf die Senderate einer Videoquelle. Zu den Bildinhalten zählen z.B. die Beschaffenheit und Bewegung der Bildobjekte, Helligkeits- oder Farbverteilung innerhalb eines Bildes, Häufigkeit und Art der Szenenwechsel. Allgemein kann man über eine Sequenz, bestehend aus mehreren hintereinanderfolgenden Bildern, sagen: Je geringer die Änderungen der Bildinhalte von Bild zu Bild ausfallen, umso effizienter arbeiten die Kompressionsverfahren.

Während die Bildqualität in der Regel vom Anwender einstellbar ist, hat man auf die Bildinhalte normalerweise keinen oder nur geringen Einfluss. Man kann also sagen, dass die mittlere Senderate einer Videoquelle von der Bildqualität vorgegeben wird und die Bildinhalte für die statistischen Schwankungen verantwortlich sind.

Nun sind die Implementierungen von Videokompressionsverfahren nicht statischer Natur, sondern können dynamisch auf die jeweiligen Bildinhalte angepasst werden [Oht94]. Dabei gibt es für den Kompressionsalgorithmus zwei unterschiedliche Strategien:

- **A**: Es gibt eine Zielvorgabe hinsichtlich der Bildqualität, die auf Kosten einer stärker schwankenden Senderate eingehalten werden soll (siehe Abbildung 4-8a).
- **B**: Es gibt eine Zielvorgabe hinsichtlich der Senderate, die auf Kosten der Bildqualität (Quantisierung, Framerate) so stabil wie möglich eingehalten werden soll (siehe Abbildung 4-8b).

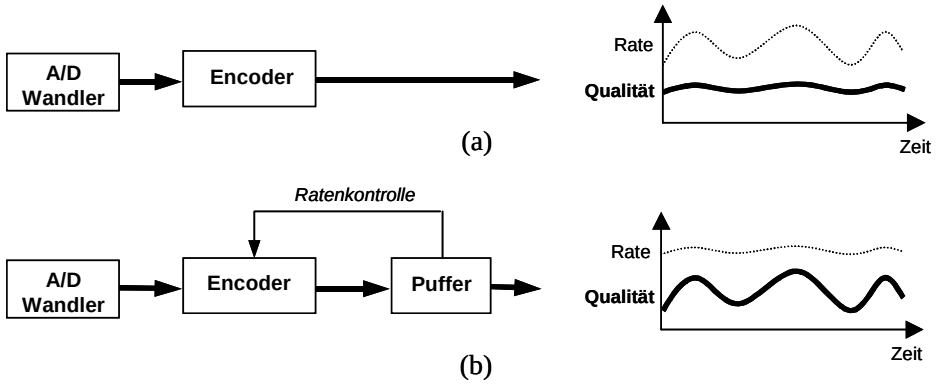


Abbildung 4-8: Übertragungskonzepte für Kanäle mit variabler (a) und konstanter (b) Bitrate

Bei den hier untersuchten Videokonferenzsystemen wird vorwiegend Strategie B verwendet, da diese Systeme kompatibel zu anderen Systemen in leitungsvermittelten Netzen sein sollen. Stark schwankende Senderaten sind in diesem Zusammenhang unerwünscht, da sie in Netzkopplungselementen zu großen Verzögerungen oder gar Verlusten führen können.

Abbildung 4-9 zeigt eine Messung, bei der ein Coder mit Ratenkontrolle eingesetzt wurde. Sie stellt den Verlauf der Senderate über der Zeit, die gemessene mittlere Senderate der Videoquelle und die maximal zulässige Rate des Coders dar. Der Ratenverlauf ergibt sich aus

dem gemessenen Paketstrom durch Bildung von Fünf-Sekunden-Intervallen und der Bestimmung der mittleren Senderate pro Intervall. Die maximale Rate des Coders kann in der Regel vom Anwender eingestellt werden. Die mittlere Rate einer Messung hängt jedoch neben der gewählten Einstellung stark von den Bildinhalten ab. Je nach dem Bewegungsmuster und den Helligkeitsverhältnissen kann die mittlere Rate einer Messreihe geringfügig oder deutlich unterhalb der maximalen Rate liegen.

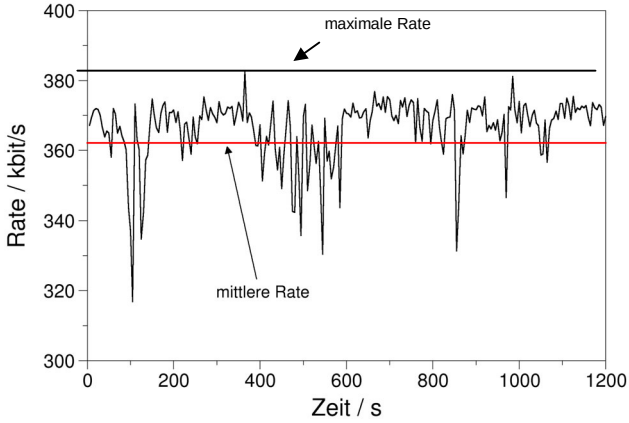


Abbildung 4-9: Typischer Ratenverlauf bei Intervallbildung von 5 Sekunden

4.5.3.1 Darstellung der Messreihen

Für die Erkennung von Fehlern und die weitere Auswertung der Daten ist es von Vorteil, die Messergebnisse grafisch darzustellen.

Da alle Daten im ASCII-Format vorliegen, können sie von beliebigen Programmen bearbeitet werden. Große Datenmengen und aufwendigere Untersuchungen lassen sich mit speziell für statistische Auswertungen und Modellierung ausgelegten Programmen wie z.B. SPLUS durchführen.

Um über das Verhalten der Quelle einen ersten Eindruck zu bekommen, kann man eine Messreihe sehr einfach als Sequenz von Paketlängen und Zeitstempeln auftragen (SPLUS: High Density Line Plot). Anhand einer solchen Darstellung erkennt man z.B. sehr schnell Ausreißer hinsichtlich hoher Werte (z.B. siehe Abbildung 4-10). Auch „Aussetzer“ bei der Übertragung kann man bei einer Darstellung der Paketabstände über der Zeit sehr einfach erkennen.

Abbildung 4-10 stellt die Paketlängen über der Zeit dar und zeigt, wann welche Paketlängen bevorzugt aufgetreten sind. Die Darstellung gibt zudem Aufschluss darüber, ob dem Paketerzeugungsprozess der Quelle über die gesamte Messdauer eine einzige Paketlängenverteilung zugrunde liegt oder ob sich diese ändert. Durch eine Verkleinerung des Bildausschnittes kann man feststellen, mit welcher Framerate gesendet wurde und wieviele Pakete pro Frame übertragen wurden.

Bei Betrachtung von Abbildung 4-10 fällt auf, dass bei Netmeeting mit den Einstellungen Qualität „schnell“, Bildgröße „mittel“ und bei wenig Bewegung im Bild vorwiegend kurze Pakete (< 250 Bytes) gesendet werden. Gelegentlich treten jedoch Phasen auf, in denen auch längere Pakete (> 600 Bytes) übertragen werden.

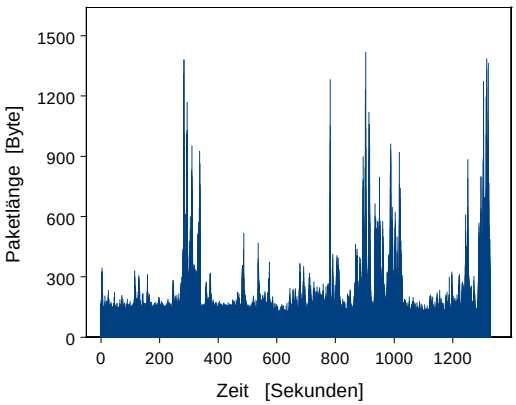


Abbildung 4-10: Sendeverhalten Netmeeting (Bildgröße: mittel, Qualität: schnell)

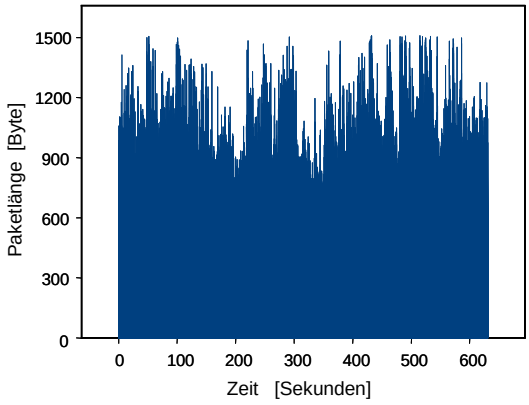


Abbildung 4-11: Sendeverhalten LiveLan (174 kbit/s, CIF)

Aus Abbildung 4-11 geht auf den ersten Blick hervor, dass bei LiveLAN (CIF, 174 kbit/s) längere Pakete gebildet und insgesamt mehr Daten pro Zeit übertragen werden als bei der zuvor betrachteten Netmeeting Verbindung. Ferner fällt auf, dass die Senderate im Gegensatz zu Netmeeting nicht so stark schwankt, was auf einen Coder mit Ratenkontrolle zurückzuführen ist. Bei genauerem Hinsehen stellt man zudem fest, dass an zwei Stellen der Messreihe die maximalen Paketlängen abgesunken sind. Das Verhalten von LiveLAN, bei sich wenig ändernden Bildinhalten die Framerate nicht zu reduzieren, sondern kürzere Pakete zu senden, wird später bei der Betrachtung der Verteilungsdichtefunktionen deutlicher.

Die Vcon-Applikation (ohne Abbildung) weist ein ähnliches Verhalten wie LiveLan auf. Einbrüche bei der maximalen Paketlänge sind jedoch, unabhängig von den gewählten Einstellungen, nicht zu erkennen. Diese Anwendung sendet die Nutzdaten schon bei niedrigen Bitraten vorzugsweise in längeren Paketen als dies beispielsweise bei LiveLan oder Netmeeting der Fall ist.

Neben der getrennten Darstellung von Paketlängen und Paketabständen geben zweidimensionale Grafiken (z.B. SPLUS: Scatter Plot) Aufschluss über die Korrelationen zwischen den beiden Größen. Festzuhalten ist, dass bei LiveLAN und Vcon die mittleren Paketlängen mit wachsendem Paketabstand zunehmen (Abbildung 4-12). Bei Vcon kann darüber hinaus der Einfluss des Bewegungsverhaltens auf den Paketgenerierungsprozess gezeigt werden (siehe Abbildung 4-12 und Abbildung 4-13). Bei ansonsten identischen Einstellungen kommen bei hohen Bewegungsanteilen Frameraten unter 30 Hz sowie Paketlängen unter 600 Bytes kaum mehr vor.

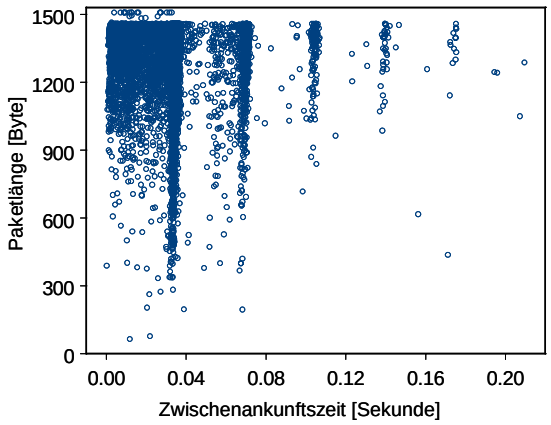


Abbildung 4-12: Vcon 384 kbit/s, CIF (wenig Bewegung)

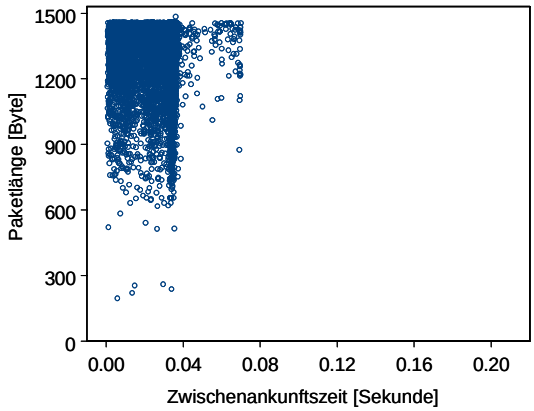


Abbildung 4-13: Vcon 384 kbit/s, CIF (viel Bewegung)

Für detailliertere Aussagen bezüglich des Zeitverhaltens und der Paketlängen mussten weitere Untersuchungen vorgenommen werden. Dazu dienten im Wesentlichen die Bitratenanalyse und statistische Methoden wie die Verteilungsdichtefunktionen.

4.5.3.2 Bitratenanalyse

Die Bitrate der Applikationen wurde in Abhängigkeit der wählbaren Bildparameter, des Bewegungsverhaltens der Videokonferenzteilnehmer und der Helligkeitsverhältnisse untersucht. Die Bildparameter werden von den Applikationen vorgegeben und umfassen die Bildgröße und die Bildqualität. Das Bewegungsverhalten der Teilnehmer reichte von „wenig Bewegung“ d.h. möglichst wenigen und langsamen Bewegungen des Kopfes und der Hände bis hin zu schnellen Bewegungen des Konferenzteilnehmers sowie Bewegungen im Bildhintergrund durch andere Personen. Szenenwechsel wurden durch Änderungen der Helligkeitsverhältnisse simuliert. Dabei wurde die Raumbeleuchtung verändert.

Die untersuchten Anwendungen werden im Folgenden mit Hilfe von Abbildung 4-14 näher erläutert. Die Abbildung zeigt typische Zeitverläufe der mittleren Senderaten der jeweiligen Anwendungen. Die mittleren Senderaten werden aus den Messreihen durch die Bildung von Sekundenintervallen, Berechnung der mittleren Rate pro Intervall und anschließender Bestimmung des gleitenden Durchschnitts über 10 Intervalle hinweg ermittelt.

Es traten zwei Klassen von Kompressionsverfahren auf: Verfahren mit Ratenkontrolle (H.261: LiveLan, Vcon) und ohne Ratenkontrolle (Netmeeting). Die Verfahren mit Ratenkontrolle zeichnen sich dadurch aus, dass sie unabhängig vom Szenenwechsel (Helligkeitsverhältnisse) eine obere Grenze der Senderate nicht überschreiten. Die Senderate kann zwar deutlich (~30%) unterhalb dieser Grenze liegen (weißer Bildhintergrund, keine Bewegung); im Normalfall weicht sie jedoch nur geringfügig (~10%) von der eingestellten Ratengrenze ab und erreicht diese nur bei hohen Bewegungsanteilen und starken Änderungen der Raumhelligkeit. Im letzteren Fall nimmt die Bildqualität merklich ab. Die Verfahren ohne Ratenkontrolle hingegen zeigen viel stärkere Schwankungen der Senderate bei nahezu gleichbleibender Bildqualität. Die Senderate kann bei bewegten Bildszenen im Vergleich zu ruhigen Bildszenen auf den vierfachen Wert ansteigen.

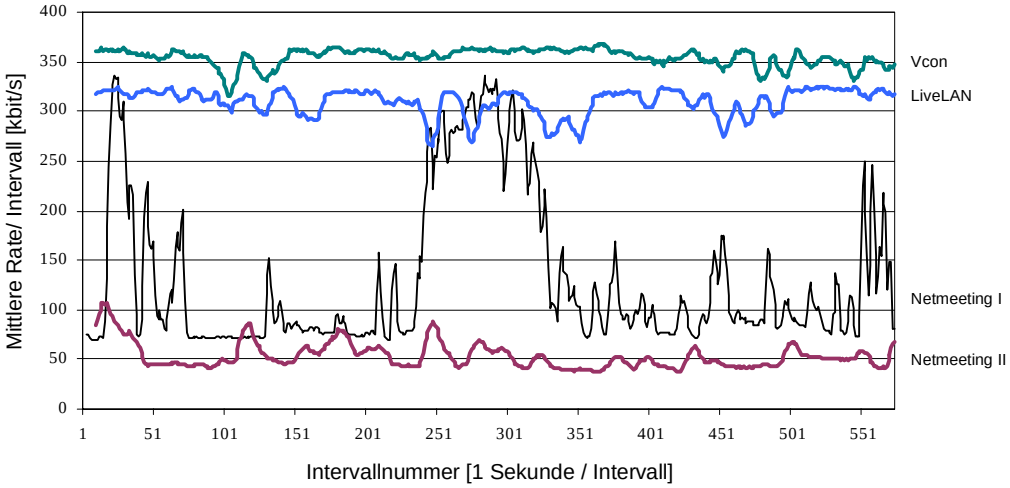


Abbildung 4-14: Ratenverläufe nach Bildung des gleitenden Durchschnitts (Fenstergröße 10)

Abbildung 4-14 zeigt darüber hinaus einige grundlegende Charakteristika der Messreihen. Vcon Anwendungen verwenden standardmäßig das G.728 Sprachkodierverfahren, LiveLAN Anwendungen hingegen ausschließlich den G.711 Sprachcodec. Demzufolge senden die Videoquellen von Vcon bei identischen Rateneinstellungen (hier: 384 kbit/s) mit einer entsprechend höheren Rate als die LiveLan Videoquellen. Ferner weisen Videoverbindungen von Vcon eine geringere Varianz der Senderaten auf als die von LiveLAN. Die minimale Senderate der Videoquelle fällt bei wenig Bewegung und geringen Helligkeitsveränderungen bei Vcon nicht so stark ab wie bei LiveLAN. Eine separate Untersuchung mit einer Standbildübertragung hat diesen Eindruck bestätigt und bei der Einstellung von 384 kbit/s eine minimale Senderate für Vcon von 240 kbit/s und für LiveLAN von 100 kbit/s ergeben.
Das Verhalten von Netmeeting Video wird in Abbildung 4-14 mit zwei unterschiedlichen Kurven dargestellt. Dabei wurden dieselben Geräte mit denselben Einstellungen verwendet. Während die Kurve Netmeeting-I durch gezielte Manipulation der Beleuchtungsverhältnisse die enorme Schwankungsbreite einer solchen Videoverbindung aufzeigt, stellt die Kurve Netmeeting-II das Verhalten einer Videoverbindung unter normalen Bedingungen einer Videokonferenz dar.

Die Messung Netmeeting-I zeichnet sich durch eine größere Helligkeit des Raumes, eine Phase mit extremen Helligkeitsschwankungen sowie heftigen Bewegungen im Bild aus. Dadurch kann die mittlere Senderate von 70 kbit/s auf zeitweise über 300 kbit/s ansteigen. Unter normalen Videokonferenzbedingungen in leicht abgedunkelten Räumen mit geringer Bewegung schwankt die mittlere Rate der Videoquelle lediglich zwischen 40 kbit/s und 100 kbit/s. Bei dem von Netmeeting verwendeten proprietären Videocodec fällt auf, dass das Sendeverhalten viel stärker von den Helligkeitsverhältnissen des Raumes und dem Bewegungsmuster der Teilnehmer als von den Systemparametern (Einstellungen) abhängig ist.

4.5.3.3 Verteilungen

Paketlängen

Im Folgenden wird der Einfluss der verschiedenen Einstellungen an den Applikationen sowie der Bewegungsanteile im Bild auf die Paketlängenverteilung untersucht. Die Paketlängenverteilung gibt Aufschluss über die zu übertragende Informationsmenge pro Bild. Die Framegröße ist jedoch häufig nicht direkt aus der Paketlängenverteilung ablesbar, da die Bildinformation eines Frames oft in mehreren Paketen übertragen wird. Daher wird bei der Analyse die zeitliche Abfolge der Paketlängen mit in die Betrachtung einbezogen. Die

gemessenen Paketlängen liegen alle in einem Bereich zwischen 73 Bytes und 1526 Bytes. Jedes gemessene Paket besitzt insgesamt 66 Bytes Header-Informationen: Ethernet-Header (26 Bytes), IP-Header (20 Bytes), UDP-Header (8 Bytes) und RTP-Header (12 Bytes). Im Allgemeinen hängt bei allen Applikationen die Paketlängenverteilung sehr stark von der Bildgröße, der Quantisierung und den Bewegungsanteilen ab. Eine verstärkte Bewegung im Bild erhöht zwar die Anzahl sehr langer Pakete, der Mittelwert hingegen ändert sich oft nur um wenige Prozentpunkte.

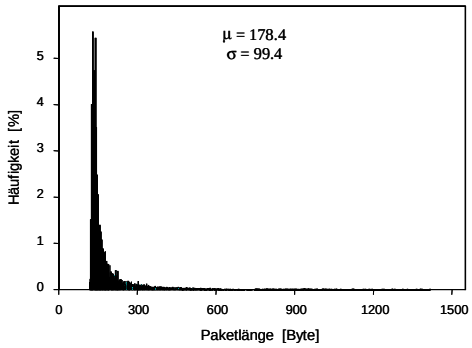


Abbildung 4-15: Paketlängenverteilung Netmeeting (Bildgröße: mittel, Qualität: schnell)

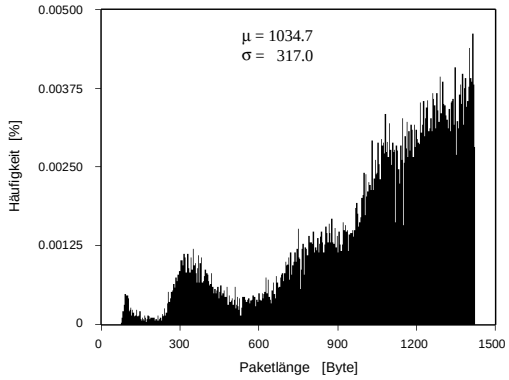


Abbildung 4-16: Paketlängenverteilung Netmeeting (Bildgröße: mittel, Qualität: gut)

Abbildung 4-15 und Abbildung 4-16 zeigen zwei sehr unterschiedliche Paketlängenverteilungen der Applikation Netmeeting. Die Framegröße bei Netmeeting hängt sowohl von der Bildgröße als auch von der Bildqualität und dem Bewegungsverhalten der Teilnehmer ab. Bei Netmeeting wird die Bildqualität über eine Quantisierung der Bildpunkte (Anzahl der kodierten Bytes pro Pixel) geregelt. Das Bewegungsverhalten ist für die Effizienz des Coders bei der Datenreduktion ausschlaggebend.

Bei allen Messreihen wurde in Phasen mit sehr geringen Bewegungsanteilen ein Paket pro Frame gesendet. Damit konnten in diesen Fällen die Framegrößen anhand der Paketlängen bestimmt werden. In Fällen mit sehr geringen Bewegungsanteilen liegt die mittlere Länge der Videopakete bei den Einstellungen Qualität „schnell“ und Bildgröße „klein“ bzw. „mittel“, bei etwa 140 bzw. 180 Bytes (Abbildung 4-15). Wird dagegen die Qualität auf „mittel“ bzw. „gut“ angehoben, so steigt die mittlere Paketlänge auf Werte um 400 Bytes bzw. 600 Bytes an. Bei zunehmenden Bewegungsanteilen wächst die Framegröße kontinuierlich auf Werte bis zu 2500 Bytes an (Abbildung 4-16). Kurze Pakete treten vor allem dann auf, wenn die Framegröße die maximale Paketlänge im Ethernet (*MTU-Size: 1500 Bytes*) um einen kleineren Wert übersteigt.

Spitzenwerte bezüglich der Framegröße wurden bei Bildgröße „groß“ gemessen. Sie betrugen bis zu 6000 Bytes und verteilten sich auf 5 Pakete. Ferner fällt bei Netmeeting auf, dass die Anwendung die maximale Paketlänge auf 1421 Bytes begrenzt.

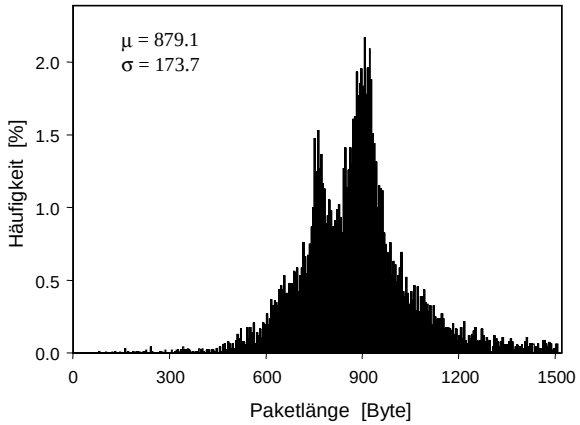


Abbildung 4-17: Paketlängenverteilung LiveLan (174 kbit/s, CIF)

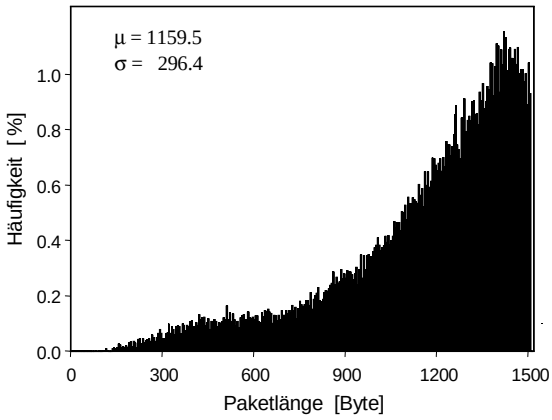


Abbildung 4-18: Paketlängenverteilung LiveLan (384 kbit/s, CIF)

Die Abbildung 4-17 und Abbildung 4-18 zeigen, dass wie schon zuvor bei Netmeeting auch bei der LiveLan-Applikation je nach Einstellung große Unterschiede in der Paketlängenverteilung festzustellen sind. Im Gegensatz zu Netmeeting ist die Qualität der Videoübertragung bei der LiveLan-Applikation in Form der maximalen Senderate von 174 kbit/s oder 384 kbit/s einstellbar. Mit zunehmender Bitrate nimmt die Framegröße zu und es ergeben sich für jede gewählte Bitrate charakteristische Framelängen von ca. 900 Bytes bei 174 kbit/s und ca. 1600 Bytes bei 384 kbit/s. Auf Paketebene bedeutet dies, dass die mittlere Paketlänge ansteigt und gleichzeitig die Anzahl der Pakete pro Frame zunimmt. Bei Phasen mit viel Bewegung konnten allgemein noch längere Pakete und mehrere kurz hintereinanderfolgende Pakete gemessen werden. Während bei 174 kbit/s in der Regel ein Paket pro Frame übertragen wird, konnten bei 384 kbit/s vorwiegend 2 Pakete pro Frame, in Phasen mit sehr viel Bewegung sogar drei und mehr Pakete pro Frame festgestellt werden. Das ist insofern verständlich, da die Kompressionsverfahren bei viel Bewegung nicht so effektiv arbeiten wie bei wenig Bewegung. Die Ratenkontrolle sorgt dafür, dass die eingestellte Bitratenobergrenze nicht überschritten wird und kann im Bedarfsfall den Coder veranlassen, die Bildpunkte gröber zu quantisieren.

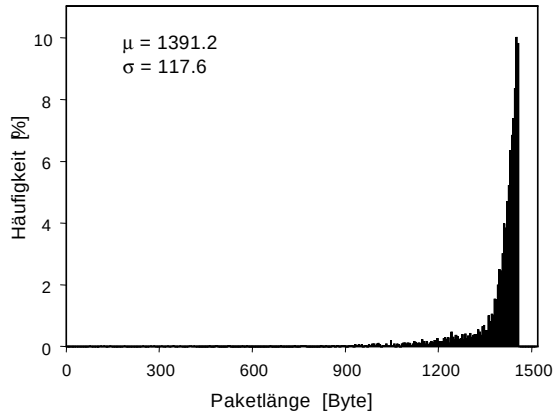


Abbildung 4-19: Paketlängenverteilung Vcon (128 kbit/s, CIF, 15 Frames/s)

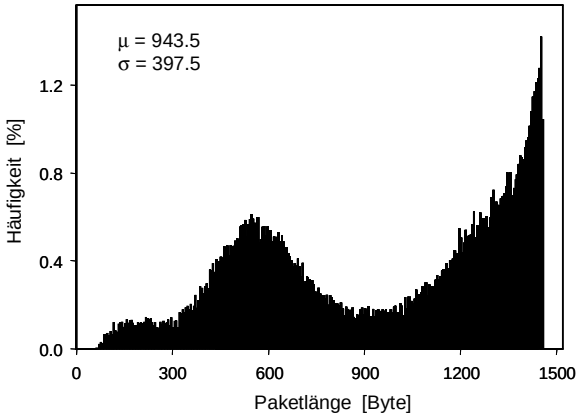


Abbildung 4-20:Paketlängenverteilung Vcon (384 kbit/s, CIF, 30 Frames/s)

Bei Vcon ist die maximale Senderate in mehreren Stufen von 64 kbit/s bis 384 kbit/s einstellbar. Wie die obigen Abbildungen zeigen, bestätigt sich bei Vcon die Vermutung aus der Durchsatzanalyse. Diese Applikation verwendet schon bei niedrigeren Bitraten vorzugsweise längere Pakete und niedrigere Frameraten als LiveLan. Bei der Einstellung wird, wie in Abbildung 4-19 angegeben, mit wenigen Ausnahmen immer genau ein Paket pro Frame

gesendet. Stellt man eine größere Senderate ein, steigen die Frameraten an und gleichzeitig werden die Frames größer. Bei viel Bewegung und hoher Senderate werden meistens zwei Pakete pro Frame gesendet. Einem längeren Paket mit Mittelwert von ca. 1300 Bytes folgt in der Regel ein kürzeres Paket mit einem Mittelwert von ca. 580 Bytes (Abbildung 4-20). Dadurch sinkt die mittlere Paketlänge im Vergleich zu kleineren Senderaten. Ferner fällt bei Vcon auf, dass die Anwendung die maximale Paketlänge auf 1458 Bytes begrenzt.

Paketabstände

Die Analyse der Paketabstände gibt nun einen genaueren Einblick in das zeitliche Verhalten des Paketgenerierungsprozesses. Gerade die Abstände zwischen den Paketen sind stark von der Rechnerauslastung bzw. der Lastsituation im Netz abhängig und verlangen entsprechende Vorkehrungen beim Messaufbau (siehe Kapitel 4.4.1). Die Verteilung der Paketabstände gibt Aufschluss über die Bildwiederholfrequenz (Framerate). Die Framerate ist jedoch häufig nicht direkt aus der Paketabstandsverteilung ablesbar, da die Bildinformation eines Frames oft in mehreren Paketen übertragen wird. Daher wird bei der Analyse die zeitliche Abfolge der Paketabstände mit in die Betrachtung einbezogen.

Das Verhalten von Netmeeting zeigt den Charakter einer Quelle mit variabler Senderate ohne jegliche Ratenkontrolle. Die mittlere Senderate hängt sehr stark von den Helligkeits- und Bewegungsverhältnissen ab und ändert sich daher über der Zeit.
Die hier betrachtete Messreihe für Bildgröße „mittel“ und Qualität „schnell“ hat eine Länge (Messdauer) von 22 Minuten und eine mittlere Senderate von 21,5 kbit/s, die Messreihe für Bildgröße „mittel“ und Qualität „maximal“ hat eine Länge (Messdauer) von 45 Minuten und eine mittlere Senderate von 132 kbit/s. Bei beiden Messreihen wurde das Bewegungsverhalten während der Messung variiert.

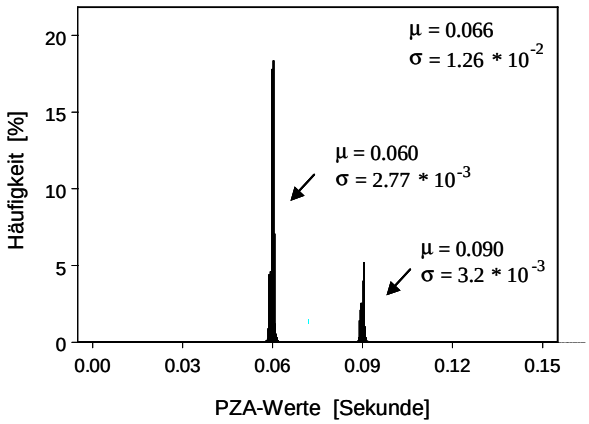


Abbildung 4-21: Verteilung der Paket-Zwischenankunftszeiten Netmeeting (Größe: mittel, Qualität: schnell)

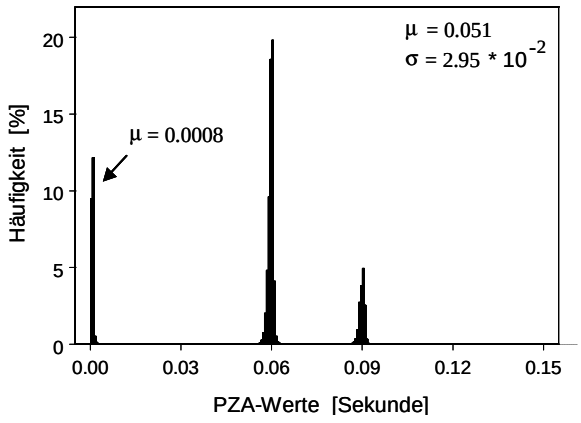


Abbildung 4-22: Verteilung der Paket-Zwischenankunftszeiten Netmeeting (Größe: mittel, Qualität: maximal)

Auffallend bei Netmeeting ist, dass die Paketabstände einem strikten periodischen Raster unterliegen. Das Muster hängt ausschließlich von der gewählten Bildgröße ab und wird weder vom Qualitätsparameter noch vom Bewegungsverhalten beeinflusst. Bei der Einstellung Bildgröße „mittel“ werden vier Pakete in Folge mit einem nahezu konstanten Abstand von 60ms gesendet, was einer Framerate von 16,7 Hz entspricht. Danach folgt ein Paket mit einem Abstand von 90ms und dann wieder vier Pakete mit 60ms. Vergrößert man das Bild (Einstellung: „groß“), vergrößert sich der Frame-Abstand und man erhält eine andere Sequenz, bestehend aus sich periodisch abwechselnden Frame-Abständen von 120ms und 150ms. Erhöht man die Bildqualität oder den Bewegungsanteil im Bild so erhöht sich lediglich die

Framegröße. Es werden dadurch häufiger mehrere Pakete pro zu übertragendem Frame geschickt, das Paketabstandsmuster bleibt jedoch ansonsten unberührt.

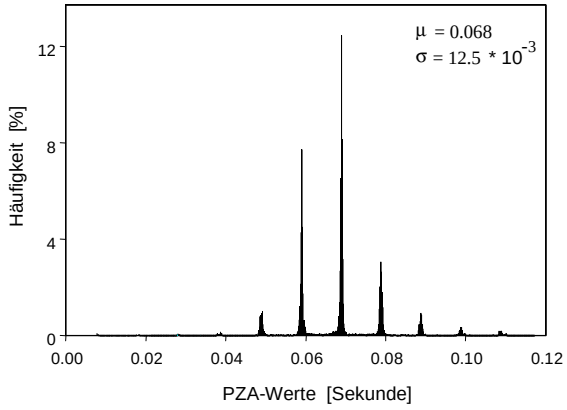


Abbildung 4-23: Verteilung der Paketzwischenankunftszeiten LiveLan (174 kbit/s, CIF)

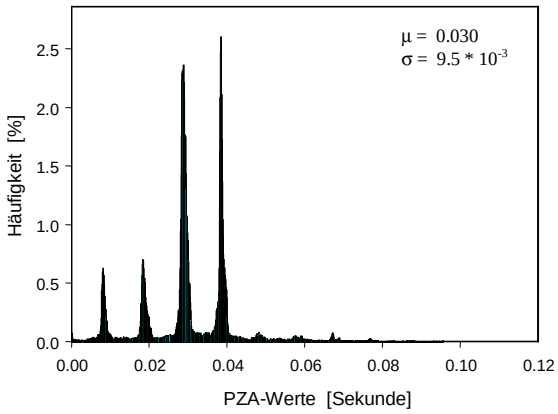


Abbildung 4-24: Verteilung der Paketzwischenankunftszeiten LiveLan (384 kbit/s, CIF)

Die obigen Abbildungen zeigen den Einfluss der eingestellten Senderate auf den Paketabstand bei LiveLan. Auf den ersten Blick fällt auf, dass ähnlich wie bei Netmeeting nur ganz bestimmte Paketabstände vorkommen. Abbildung 4-23 weist 7 Peaks zwischen 48,5ms und 108,5ms in einem festen Abstand von exakt 10ms auf. Die Abstände entsprechen Frameraten zwischen 20,6 Hz und 9,2 Hz. Bevorzugt verwendet wurden in diesem Fall jedoch die Frameraten 17,1 Hz und 14,6 Hz. Zudem fällt auf, dass alle Frames in einem einzigen Paket übertragen werden konnten. Abbildung 4-24 hingegen zeigt, dass die Framerate bei hoher Senderate fast ausschließlich zwei bestimmte Werte einnimmt. Der eine Wert liegt bei 35 Hz und der andere bei 26 Hz. Sehr selten kommen Frameraten von 20,6 Hz bzw. von 17,1 Hz vor. Auffallend ist ferner, dass mehrere Pakete, die zu einem Frame gehören, nicht als ein Burst, sondern verzögert mit einem festen Paketabstand von durchschnittlich 8ms gesendet werden. Die Anwendung nimmt hier eine Glättungsfunktion des Verkehrs vor. Erhöht man den Bewegungsanteil, erhöht sich die mittlere Framegröße. Es werden mehrere Pakete (2 bis 5) pro Frame gesendet, während der mittlere Paketabstand nahezu unberührt bleibt.

Bei Vcon mit niedriger Senderate ist festzustellen, dass pro Frame in der Regel nur ein Paket versendet wird. Im Gegensatz zu LiveLan bleibt jedoch die Framerate konstant (siehe Abbildung 4-25). Sie ändert sich mit den eingestellten maximalen Senderaten sowie mit den Bewegungsanteilen. Bei niedrigen Senderaten liegt die Framerate vorwiegend bei 15 Hz (192 kbit/s) bzw. 7,5 Hz (128 kbit/s). Bei hohen Senderaten (≥ 384 kbit/s) liegt sie im Regelfall bei 30 Hz und kann bei konstanten Lichtverhältnissen und geringem Bewegungsanteil auf 15 Hz und kurzzeitig sogar auf 7,5 Hz abfallen (siehe Abbildung 4-26). In Momenten hoher Bewegung werden des öfteren auch zwei oder drei Pakete pro Frame erzeugt, wodurch häufiger kurze Paketabstände (< 10ms) auftreten. Dadurch entsteht oberhalb der maximalen Framerate eine Art Badewannenkurve (siehe Abbildung 4-26). Im Gegensatz zu LiveLan findet hier kein Glätten der Intraframe-Bursts statt.

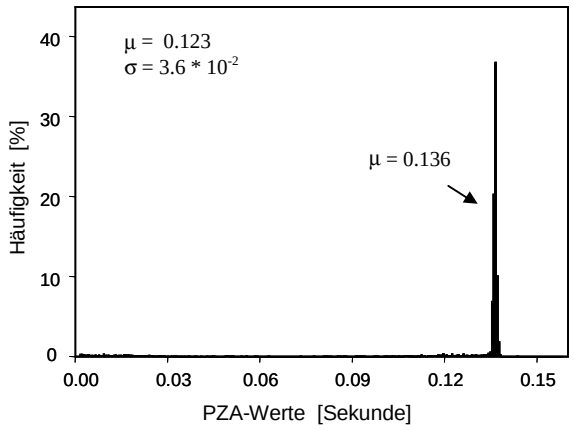


Abbildung 4-25: Verteilung der Paket-Zwischenankunftszeiten Vcon (128 kbit/s, CIF)

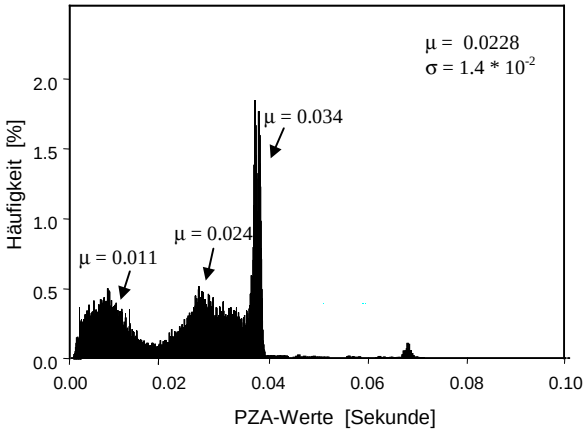


Abbildung 4-26: Verteilung der Paket-Zwischenankunftszeiten Vcon (384 kbit/s, CIF)

4.5.4 Zeitreihenanalyse

Im Folgenden wird bei den untersuchten Videodatenquellen das zeitliche Verhalten näher betrachtet. Zur Analyse der zeitlichen Zusammenhänge im Sendeverhalten werden Autokorrelationsfunktionen sowie zur Erkennung von Langzeitabhängigkeiten die Methoden der Varianzanalysen, R/S-Analyse (*Rescaled Adjusted Range Statistic*) und MLE (*Maximum Likelyhood Estimation*) verwendet.

Die statistischen Methoden können sowohl auf die Paketlängen als auch auf die Paketabstände der Messreihen angewendet werden. Alle in diesem Kapitel grafisch dargestellten Messreihen haben eine Länge von 20 bis 30 Minuten und umfassen zwischen 20000 und 60000 Pakete.

4.5.4.1 Autokorrelationen

Anhand der Autokorrelationsfunktion AKF kann der zeitliche Zusammenhang von Ereignissen x_t veranschaulicht werden. Die Autokorrelationsfunktion setzt sich aus den Korrelationskoeffizienten γ_k zusammen, welcher aus einer Sequenz von Ereignissen x_t mit dem Mittelwert μ wie folgt berechnet werden kann:

$$\gamma_k = E[(x_t - \mu)(x_{t-k} - \mu)]$$

Die Korrelationskoeffizienten γ_k sind eine Maßzahl für die Wahrscheinlichkeit, dass in einer Messreihe ähnliche Ereignisse zu einem späteren Zeitpunkt k erneut aufgetreten sind. Sie geben somit Aufschluss darüber, ob über mehrere aufeinanderfolgende Ereignisse einer Messreihe hinweg ein kausaler Zusammenhang besteht. Für eine einheitliche Darstellung werden die Korrelationskoeffizienten γ_k bei Bildung der AKF mit γ_o normiert. Dadurch besitzt eine AKF für $k = 0$ immer den Wert $\sigma_k = 1$.
Existiert kein zeitlicher Zusammenhang zwischen den einzelnen Ereignissen eines stochastischen Prozesses z.B. den einzelnen Paketen einer Messung, fällt die AKF für $k = 1$ bereits auf einen Wert um 0 ab. Weisen die Ereignisse ein kurzzeitabhängiges Verhalten auf, d.h. geht der kausale Zusammenhang verloren, so fällt die AKF exponentiell ab. Ein stationärer Prozess gilt als kurzzeitabhängig, wenn für die Autokorrelationskoeffizienten gilt:

$$\sigma_k \sim c^{-k}$$

Weist der Verkehr hingegen langzeitabhängiges Verhalten auf, bleibt ein kausaler Zusammenhang über sehr lange Zeiträume bestehen und die AKF fällt lediglich hyperbolisch ab. Es gilt der Zusammenhang:

$$\sigma_k \sim k^{-\beta} \qquad k \to \infty, \quad 0 < \beta < 1$$

Betrachtet man die AKF von deterministischen Prozessen, kann diese ebenso einen hyperbolischen Abfall besitzen. In diesem Fall liegt dem Paketgenerierungsprozess jedoch kein stochastischer Prozess zugrunde und es darf nicht auf ein langzeitabhängiges Verhalten geschlossen werden.

Zum Phänomen der Langzeitabhängigkeit und den damit verbundenen Folgen für z.B. das Pufferverhalten gehören noch andere Eigenschaften, die im nachfolgenden Kapitel näher untersucht werden.

Bei vielen der im Kapitel 4.5.3 untersuchten Messreihen wurde ein deterministisches Verhalten der Applikationen bei der Paketerzeugung festgestellt. Besitzt eine Messung ein deterministisches Verhalten z.B. in Form von bestimmten, periodisch wiederkehrenden Mustern, kann dies sehr anschaulich anhand der Autokorrelationsfunktion grafisch dargestellt werden. Die folgenden beiden Abbildungen zeigen zwei AKF der Paket-Zwischenankunftszeiten von zwei Messungen mit Netmeeting bei identischen Einstellungen, jedoch sehr unterschiedlichem Bewegungsverhalten. Während bei wenig Bewegungsanteilen das deterministische Verhalten des Coders mit periodisch wiederkehrenden Frameabständen klar zu erkennen ist, geht bei viel Bewegung der kausale Zusammenhang zwischen Frame-Abstand und Paketabstand nahezu völlig verloren. Wie man an Abbildung 4-28 gut erkennen kann, steigt bei viel Bewegung die Wahrscheinlichkeit stark an, dass einem großen Paketabstand ein sehr kurzer folgt ($\sigma_1 = -0.5$).

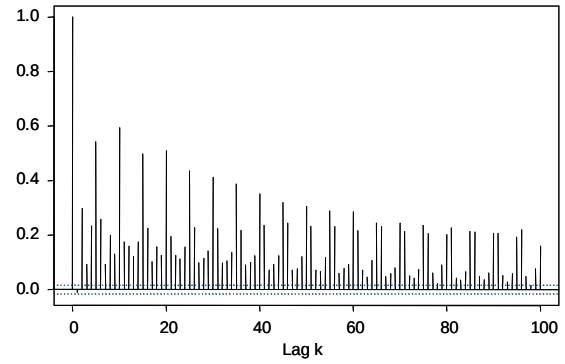


Abbildung 4-27: AKF der Paket-Zwischenankunftszeiten Netmeeting Video: wenig Bewegung

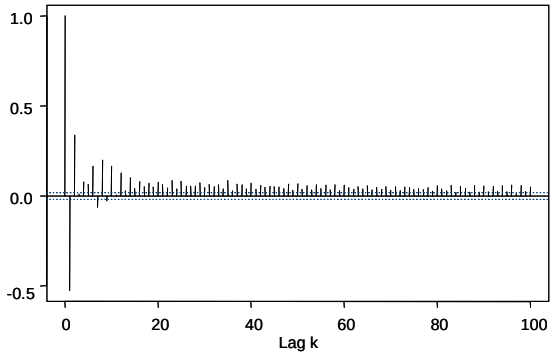


Abbildung 4-28: AKF der Paket-Zwischenankunftszeiten Netmeeting Video: viel Bewegung

Bei den Paketlängen sieht es ähnlich aus (Abbildung 4-29, Abbildung 4-30). Auch hier geht das deterministische Verhalten des Coders mit zunehmender Bewegung verloren. Bei wenig Bewegung wird ein Frame in einem Paket übertragen und die Paketgröße schwankt nur geringfügig um ihren Mittelwert. In Momenten hoher Bewegungsanteile im Bild wachsen die Framegrößen stark an. Dadurch werden in der Regel pro Frame mehrere Pakete mit sehr unterschiedlichen Längen gebildet.

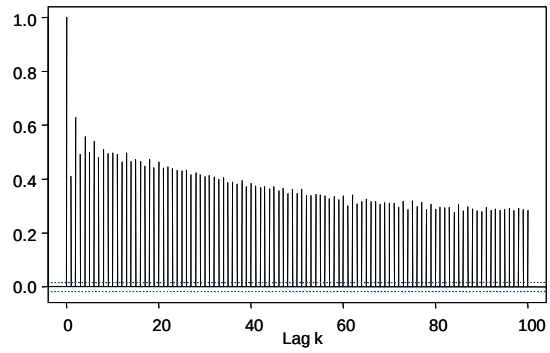


Abbildung 4-29: AKF der Paketlängen Netmeeting Video: wenig Bewegung

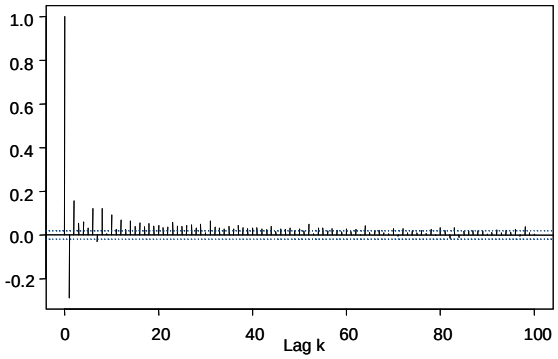


Abbildung 4-30: AKF der Paketlängen Netmeeting Video: viel Bewegung

4.5.4.2 Hurst-Parameter (Langzeitabhängigkeiten)

Das Phänomen der Langzeitabhängigkeit ist seit den 50-er Jahren aus den Gebieten der Hydrologie, Biochemie und Ökonomie bekannt und fand Anfang der 90-iger Jahre mit der Verkehrsanalyse von IP-Netzen seine Renaissance [Lel94]. Bislang wurden Telekommunikationsnetze anhand von Verkehrsmodellen dimensioniert und verwaltet, die Langzeitabhängigkeiten nicht berücksichtigen. Tritt jedoch bei Verkehren in IP-Netzen das Phänomen der Langzeitabhängigkeit auf [GW94], [CB96], hat dies einen großen Einfluss auf den Ressourcenbedarf dieser Verkehre. Das Pufferverhalten langzeitabhängiger Verkehre ist völlig anders im Vergleich zu dem von kurzzeitabhängigen Verkehren. Werden über ein nach herkömmlichen Methoden dimensioniertes Netz Verkehre mit langzeitabhängigen Verhalten übertragen, so werden die Paketverluste und Verzögerungen (insbesondere bei großen Puffern) teilweise erheblich unterschätzt [PGC97], [ENW96], [Ven97], [Nor94]. Im Folgenden werden die gemessenen Verkehre nach solchen Eigenschaften untersucht, da diese gegebenenfalls völlig andere Quellenmodelle und Zugangskontrollverfahren erfordern.

In der Literatur gibt es für den Nachweis von Langzeitabhängigkeiten mehrere Methoden wie die Varianz-Zeit-, Spektral-, R/S-Analyse und die Whittle'sche Näherung der MLE-Methode [Ber94]. Für einen Großteil der stochastischen Prozesse kann anhand dieser Analysen der Grad der Langzeitabhängigkeit einer Messreihe bestimmt und als ein Parameter angegeben werden. Zur Berechnung dieses sogenannten Hurst-Parameters (*H*-Parameter) werden möglichst lange Messreihen benötigt. Je länger die Messreihe ist, desto genauer kann der *H*-Parameter berechnet werden und desto genauer stimmen in der Regel auch die Ergebnisse der einzelnen Berechnungsmethoden überein. Seine möglichen Werte liegen zwischen 0,5 und 1. Der Wert von 1 steht dabei für die maximale Langzeitabhängigkeit, der Wert von 0,5 hingegen für die reine Kurzzeitabhängigkeit.

Bei allen Analysenmethoden zur Bestimmung des H-Parameters werden die Messreihen unter verschiedenen Skalierungen betrachtet. Bei feinster Skalierung werden alle Messwerte direkt verwendet. Bei gröberer Skalierung werden mehrere Einzelwerte zusammengefasst und durch ihren Mittelwert repräsentiert. Dazu teilt man eine Sequenz x_k in Blöcke gleicher Länge m und berechnet die Repräsentanten $x_k^{(m)}$ der Blöcke aus:

$$x_k^{(m)} = (x_{km-m+1} + \ldots + x_{km}) / m$$

Für langzeitabhängige Prozesse kann ein H-Parameter angegeben werden, wenn für die Varianz bei $m \to \infty$ gilt: $VAR(x^{(m)}) \sim c(m^{-\beta})$; $c = konstant, 0 < \beta < 1$

Für die AKF gilt: $\sigma_k^{(m)} \to \sigma_k \qquad m \to \infty$

Der *H*-Parameter ergibt sich dann per Definition aus β und wird über folgende Beziehung ermittelt:

$$H = 1 - (\beta / 2) \qquad 0.5 < H < 1$$

Bei der R/S-Analyse und der Varianzanalyse handelt es sich um grafische Methoden zur Bestimmung von H. Durch Anlegen einer Tangente wird das asymptotische Verhalten der Messwerte ermittelt. Die Schwierigkeit dieser Analysen besteht darin, die Tangente geeignet an die Stützwerte anzulegen. Die Idee hinter der MLE-Methode zur Bestimmung des H-Parameters hingegen ist, ausgehend von einem langzeitabhängigen Prozeß, z.B. einem fr-ARIMA Prozess, die Parameter so zu wählen, dass die zugehörige Log-Likelihood Funktion maximiert wird [Bre73]. Für die Berechnungen der Parameter AR, MA und d des fr-ARIMA Modells wurde die Implementierung von SPLUS verwendet. Um die Berechnung zu beschleunigen, basiert die SPLUS-Implementierung auf der Näherung von Haslett und Raftery [HR89].

Für die in Tabelle 4-2 angegebenen H-Parameter wurde der M-Parameter der Haslett-Näherung gleich *n/10* gesetzt, wobei n die Anzahl der gemessenen Pakete der betrachteten Messreihe ist. Der H-Parameter ergibt sich dann aus dem d-Parameter über die Beziehung:

$$H = d + 0.5$$

Im Folgenden werden obige Methoden auf die Messreihen angewendet. Dazu wurden zunächst aus den Paketabständen und Paketlängen Raten gebildet (Paketlänge/Paketabstand). Auf die Sequenz aus Raten wurden dann verschiedene Verfahren zur Bestimmung des Hurst-Parameters angewendet. Neben der Varianzanalyse kamen die R/S-Analyse und die Whittel'sche Näherung der MLE-Methode zum Einsatz. Da der *H*-Parameter nur geschätzt werden kann, liefern die jeweiligen Verfahren leicht unterschiedliche Ergebnisse.

In Tabelle 4-2 sind die *H*-Parameter der Raten für niedrige (<192 kbit/s) und hohe Bitraten (≥ 384 kbit/s) angegeben. Bei allen Messreihen wurde das Bewegungsverhalten der Teilnehmer während der Messung stark variiert.

Die Ergebnisse zeigen, dass Vcon die niedrigsten und Netmeeting die höchsten H-Parameter liefert. Das war zu erwarten, da Netmeeting keine Ratenkontrolle und Vcon die stabilste Ratenkontrolle verwendet hat. Für einen Anstieg des *H*-Wertes sind bei Codecs mit Ratenkontrolle und bei wenig Bewegung die lang anhaltenden, starken Abfälle der Senderate und bei Codecs ohne Ratenkontrolle und bei viel Bewegung der lang anhaltende, starke Anstieg der Senderate ausschlaggebend. Vcon hat schon bei der Bitratenanalyse gezeigt, dass in Phasen mit wenig Bewegung die Rate nicht so stark absinkt wie bei LiveLan.

H-Parameter	Netmeeting				LiveLan				Vcon			
Methode	Var	R/S	MLE	∅	Var	R/S	MLE	∅	Var	R/S	MLE	∅
Bitrate niedrig	0.68	0.70	0.75	0.71	0.89	0.95	0.99	0.94	0.50	0.53	0.56	0.53
Bitrate hoch	0.95	0.92	0.93	0.93	0.59	0.61	0.59	0.59	0.56	0.54	0.55	0.55

Tabelle 4-2: H-Parameter der Videoquellen

LiveLan-Messreihen haben bei längeren Zeitphasen ohne Bewegung regelrechte Einbrüche der Senderate um bis zu 35%, während bei Vcon in vergleichbaren Fällen ein Rückgang der Senderate von lediglich 20% festzustellen war. Die auf den ersten Blick extrem wirkenden Unterschiede bei LiveLan zwischen „Bitrate niedrig" und „Bitrate hoch" sollen verdeutlichen, wie stark insbesondere bei LiveLan der H-Parameter von der Zeitdauer der Phasen mit geringer Bewegung abhängt. Während bei der Messung „Bitrate hoch" diese Phase ca. 10 Sekunden angehalten hat, dauerte sie bei „Bitrate niedrig" über 50 Sekunden. Entfernt man

nachträglich die zu dem Einbruch gehörigen Ratenwerte aus der Sequenz und berechnet den *H*-Parameter neu, ergibt sich ein Wert von 0,57.

Bei Netmeeting hingegen war eine starke Abhängigkeit der Senderate von den Lichtverhältnissen erkennbar. Während die Messreihe „Bitrate niedrig“ nur Änderungen der Teilnehmerbewegung enthält, wurde in der Messreihe „Bitrate hoch“ die Raumhelligkeit für einen längeren Zeitraum erhöht (siehe Abbildung 4-14, Netmeeting-I). Bei Netmeeting konnte durch ein normales Bewegungsverhalten der Teilnehmer ein *H*-Wert um 0,7 und bei lang anhaltenden Veränderungen der Raumhelligkeit ein *H*-Wert um 0,9 erzeugt werden.

4.5.4.3 Zusammenfassung

Die Zeitreihenanalyse hat gezeigt, dass bei geringer Bildbewegung und gleichbleibenden Helligkeitsverhältnissen anhand der Autokorrelationsfunktion das deterministische Verhalten der Codecs sehr gut zu erkennen ist. Dieses Verhalten geht jedoch verloren, wenn die zufällig über der Zeit verteilten Bewegungsanteile starken Schwankungen unterliegen.

Die Untersuchungen hinsichtlich langzeitabhängiger Eigenschaften der Videoverkehre haben gezeigt, dass ein solches Verhalten nicht nur reine VBR-Quellen (VBR: Variable Bit Rate) wie die z.B. hier betrachtete Netmeeting-Applikation aufweisen, sondern auch bei Videoquellen mit einer Ratenkontrolle unter gewissen Voraussetzungen langzeitabhängige Effekte auftreten können.

Dies setzt jedoch voraus, dass der Videokonferenzteilnehmer z.B. seinen Sitzplatz für längere Zeit verlässt, die Kameraposition sich unterdessen nicht ändert und zudem keine Bewegungen im Bildhintergrund oder Änderungen der Raumbeleuchtung stattfinden. Nur in diesen Fällen sinkt die Senderate für einen längeren Zeitraum deutlich unter den kontrollierten Maximalwert ab, was dann wiederum den H-Parameter ansteigen lässt.

Im „Normalfall“ d.h. bei normalem Bewegungsverhalten der Teilnehmer und relativ konstanten Beleuchtungsverhältnissen ist jedoch festzuhalten, dass weder bei LiveLan noch bei Vcon H-Werte größer als 0,60 festgestellt wurden. Das lässt den Schluss zu, dass Videodatenströme, die mit Codecs mit Ratenkontrolle wie dem H.261 kodiert sind, im Regelfall kein langzeitabhängiges Verhalten aufweisen.

Bei Netmeeting hingegen wird bei dem proprietären Kodierverfahren keine Ratenkontrolle verwendet. Diese Applikation reagiert auf Helligkeitsveränderungen sehr stark, so dass die Senderate über längere Zeiträume sowohl sehr große als auch sehr niedrige Werte annehmen kann. Folglich weisen auch die ermittelten H-Parameter von den meisten Messreihen mit Werten $\geq 0{,}7$ auf ein langzeitabhängiges Verhalten hin. Das ist insofern überraschend, da bislang in der Literatur langzeitabhängiges Verhalten in Videoverkehren nur bei der Übertragung von Spielfilmen festgestellt wurde [Ber96], [GW94], [Lam95]. Bei Spielfilmen werden dramaturgische Höhepunkte, die sich über viele Minuten hinweg aufbauen, für die langzeitabhängigen Effekte verantwortlich gemacht. Für MPEG-1 Videoquellen konnten diese Effekte in einer Diplomarbeit am Lehrstuhl für Kommunikationsnetze reproduziert werden [Bre00].

4.5.5 Analyse des Sprachverkehrs

Alle untersuchten Applikationen verwenden unterschiedliche Standard-Sprachcodecs. Besondere Einstellungen, wie bei Video, können hier nicht gewählt werden. Während LiveLan und Vcon die konstantbitratigen Codecs G.711 (64 kbit/s) bzw. G.728 (16 kbit/s) verwenden, setzt Microsoft bei Netmeeting eine Vielzahl von Codecs ein. Im Folgenden werden bei Netmeeting zwei Einstellungen untersucht: der sehr niederbitratige Codec G.723.1 (6,4 kbit/s) mit Sprechpausenunterdrückung (*Silent-Suppression*) und der G.711 Codec. Alle untersuchten Applikationen verwenden konstante Paketgrößen für die Übertragung der Audiodaten.

Entsprechend der Codecs und der unterschiedlichen Optimierung zwischen Delay und Paket-Overhead sind die Paketlängen sehr verschieden. So betragen die Paketlängen 578 Bytes bei LiveLAN, 216 Bytes bei Vcon und 90 Bytes bzw. 322 Bytes bei Netmeeting. Der Overhead, verursacht durch die Paketierung, beläuft sich bei LiveLAN auf lediglich 13%, bei Vcon auf 44% und bei Netmeeting G.723.1 sogar auf bis zu 375%. Das bedeutet, dass die mittleren Bitraten im Ethernet entsprechend höher liegen als die Nutzdatenraten der Codecs. Somit kommen Netmeeting mit G.723.1 und Vcon mit G.728 auf etwa dieselben Raten, obwohl der G.723.1 eine viel stärkere Komprimierung der Sprachinformation vornimmt als der G.728.

Die maximale Gesamtverzögerung von Audio Samples in der Applikation, verursacht durch Kodierung und Paketierung, beträgt bei Netmeeing lediglich 37.5ms. Dabei handelt es sich rein um die Kodierungsverzögerung, da jeder Frame sofort gesendet werden kann (Paketierung: 1 Frame pro Paket). Bei LiveLAN G.711 hingegen beträgt die Gesamtverzögerung 64ms. Dabei handelt es sich jedoch mehr um die Verzögerung bei der Paketierung (512 Frames pro Paket). Um den Paket-Overhead zu reduzieren, werden mehrere Frames gepuffert und dann gemeinsam in einem Paket übertragen. Die Applikationen können bei Verwendung desselben Codecs unterschiedliche Strategien wählen. So beträgt die Gesamtverzögerung bei Netmeeting G.711 bei einer Paketierung von 256 Frames pro Paket lediglich 32ms. Bei Vcon G.728 beträgt die Gesamtverzögerung bei einer Paketierung von 120 Frames pro Paket 75ms.

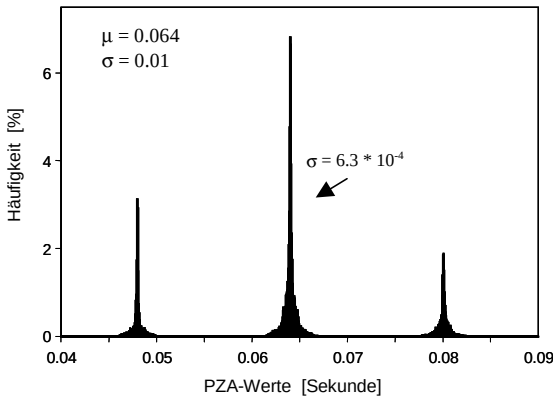


Abbildung 4-31: Verteilung der Paket-Zwischenankunftszeiten LiveLan

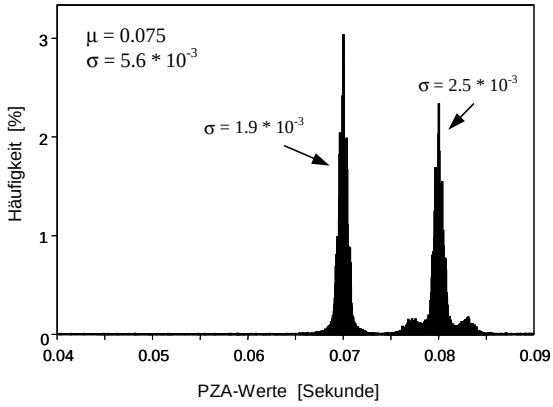


Abbildung 4-32: Verteilung der Paket-Zwischenankunftszeiten Vcon

Im Folgenden wird das zeitliche Verhalten der Applikationen näher untersucht. Die Verteilung der PZA-Werte von LiveLan Audio weist drei *Peaks* bei 48ms, 64ms und 80ms auf. Sie teilen sich im Verhältnis 1:3:1 auf diese drei *Peaks* auf und folgen zeitlich gesehen einem festem Muster (80ms, 64ms, 48ms, 64ms, 64ms). Die Werte schwanken nur in einem sehr kleinen Bereich um ihre jeweiligen Mittelwerte. Die Standardabweichung beträgt jeweils ca. $6{,}3 \cdot 10^{-4}$. Die geringfügigen Abweichungen sind auf das nicht echtzeitfähige Betriebssystem, auf Einflüsse des Netzes und auf Messfehler bei der Zeitstempelvergabe zurückzuführen.
Laufen auf einem Rechner mehrere Prozesse gleichzeitig, steigt insbesondere bei Betriebssystemen ohne *Task*-Priorisierung mit zunehmender Rechnerauslastung die Standardabweichung stark an. Um den Einfluss des Betriebssystems genauer zu untersuchen, wurden Vergleichsmessungen unter *Windows'95* herangezogen. Als kleinster Paketabstand wurde bei LiveLan und geringer Rechnerauslastung (< *5%*) unter Windows'98 ein minimaler Paketabstand von 35ms (Abbildung 4-31), unter *Windows'95* einer von 2ms gemessen (Abbildung 4-4). Bei *Windows'95* konnten sogar bei bis zu drei aufeinanderfolgenden Paketen sehr kurze Abstände festgestellt werden.

Die Verteilung von Vcon Audio besitzt nur zwei *Peaks* bei 70ms und 80ms. Die PZA-Werte teilen sich im Verhältnis 1:1 auf. Einem langen Paketabstand folgt immer ein kurzer und umgekehrt. Die Standardabweichung um diese *Peaks* ist etwas größer als bei LiveLan.

Bei NetMeeting Audio ergeben sich für die verschiedenen Codecs große Unterschiede im zeitlichen Verhalten. Obwohl die Verteilungen der Paket-Zwischenankunftszeiten aufgrund der nahezu identischen Mittelwerte um 30ms auf den ersten Blick sehr ähnlich aussehen, sind die Unterschiede bei Betrachtung der Standardabweichungen erheblich. Die Schwankungen der Paketabstände um ihren Mittelwert sind bei G.723.1 ca. um den Faktor 100 größer als bei G.711. Da bei G.723.1 eine Sprechpausenunterdrückung eingesetzt wird, kommen bei Gesprächspausen PZA-Werte von mehreren Sekunden vor.

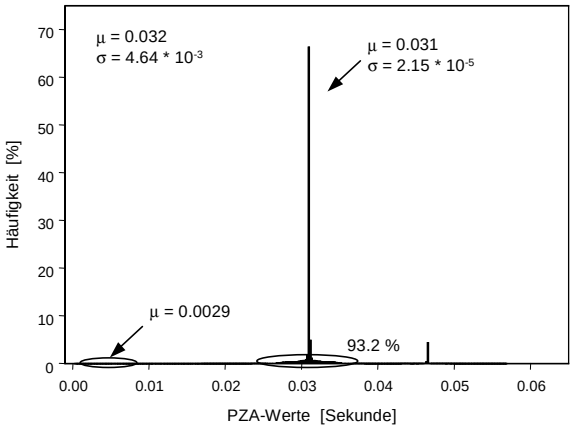


Abbildung 4-33: Verteilung der Paket-Zwischenankunftszeiten Netmeeting G.711

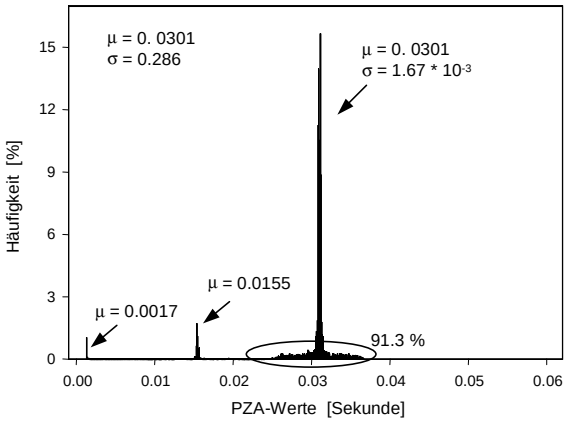


Abbildung 4-34: Verteilung der Paket-Zwischenankunftszeiten Netmeeting G.723.1

Über der Zeit gesehen, treten diese Ereignisse nur sehr selten auf, haben jedoch einen großen Einfluss auf die Standardabweichung der Paketzwischenankünfte. Diese hohe Varianz der Paketabstände, verursacht durch das Sprechverhalten der Teilnehmer, hat über der Zeit gesehen starke Schwankungen der Senderaten zur Folge. Betrachtet man ein Gespräch zwischen zwei Teilnehmern, das aus Monologen variabler Länge besteht und berücksichtigt darüber hinaus noch die Tatsache, dass während eines Monologes sehr kurze Paketabstände von weniger als 2ms auftreten können, kann der G.723.1-Verkehr durchaus als bursthaft bezeichnet werden.

Will man einen solch variablen Verkehr auf Langzeitabhängigkeiten testen, stellt sich zuerst die Frage, ob ein Verkehr, der aus zwei Zuständen besteht, überhaupt sinnvoll mit einem einzigen stochastischen Prozess beschrieben werden kann. Versucht man es dennoch und berechnet den *H*-Parameter mit Hilfe der Varianzanalyse direkt aus den Paketabständen, so ergeben sich Werte um $H = 0{,}65$. Dabei werden jedoch die wenigen, lang anhaltenden Gesprächspausen jeweils nur durch einen einzigen, sehr großen Wert berücksichtigt. Diskretisiert man hingegen die Zeit und berechnet pro Intervall eine mittlere Rate, werden pro Gesprächspause mehrere sehr kleine Werte generiert. Damit fließen diese stärker in den *H*-Parameter ein. Bestimmt man nun den *H*-Parameter über die zeitdiskreten Ratenwerte, werden *H*-Werte um 0,75 ermittelt. Als Basis für diese Berechnungen wurde ein sehr lange Sequenz von über 50 Minuten, bestehend aus 100 000 Paketen und 218 Gesprächspausen, verwendet. Ein Paketabstand länger als eine Sekunde wurde dabei als Gesprächspause gewertet. Abbildung 4-35 zeigt die Verteilung der Gesprächspausenlängen, die im Mittel bei 2,5 Sekunden lagen. 50% der Werte waren kleiner als 1,7 Sekunden, 10% der Werte waren größer als 5 Sekunden und das Maximum lag bei 17,5 Sekunden. Ein Vergleich der Verteilung der Gesprächspausenlängen mit einer *Weibull*-Verteilung in Abbildung 4-36 zeigt bei einem Shape-Parameter von 0,65 eine sehr gute Übereinstimmung.

Untersuchungen aus der Literatur mit ON-OFF Modellen haben ergeben, dass ein hyperbolischer Abfall der Verteilungsfunktion der ON- oder OFF-Phasen ein langzeitabhängiges Verkehrsverhalten hervorruft. Modelliert man den G.723.1-Verkehr mit einem ON-OFF Prozess, bestätigt die Weibull-Verteilung der OFF-Phasen mit ihrem hyperbolischen Abfall („*heavy-tail*“) das zuvor schon festgestellte langzeitabhängige Verhalten.

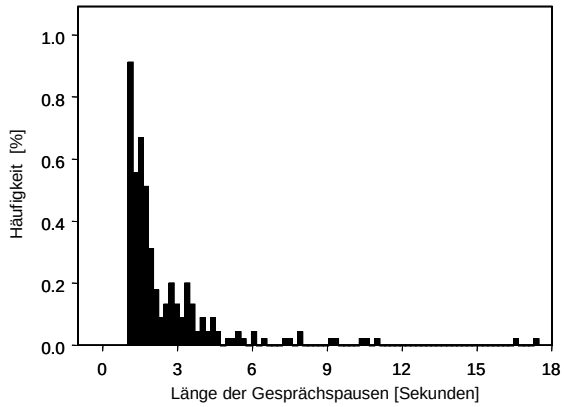


Abbildung 4-35: Histogramm der Gesprächspausen bei G.723.1

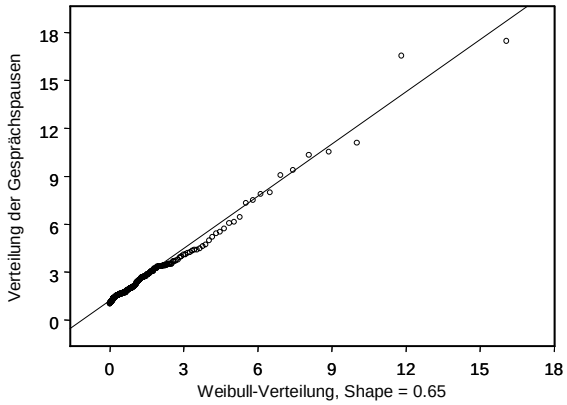


Abbildung 4-36: QuantileQuantile-Darstellung zur Ermittlung der Verteilungsfunktion

Zusammenfassend kann man sagen, dass einem Sprachverkehr mit Sprechpausenunterdrückung trotz niedriger Senderaten ein langzeitabhängigen Verhalten nachgewiesen werden kann.

4.5.6 Analyse des Signalisierungsverkehrs

Neben den Nutzdaten tauschen die Endgeräte auch Signalisierungsdaten aus. Es gibt zwei Arten von Signalisierungsverkehr: den zum Steuern eines Rufes und den zum Überwachen einer bestehenden Nutzdatenverbindung. Die Steuerung findet zum Auf- bzw. Abbau eines Rufes statt und darüber hinaus bei Veränderungen der Verbindungsparameter während eines bestehenden Rufes. Die Überwachung einer Verbindung hingegen findet während des Nutzdatenaustausches statt.

Im H.323-Standard der ITU sind neben dem RTP-Protokoll zur Nutzdatenübertragung mehrere Steuerprotokolle vorgesehen. Sie dienen der Netzzugangskontrolle (RAS-Signalisierung des H.225.0-Protokolls) [H.225], dem Auf-/Abbau einer Multimediasitzung durch die Rufsignalisierung (Q.931-Signalisierung des H.225.0-Protokolls), dem Auf-/Abbau von Nutzdatenkanälen (H.245-Protokoll) [H.245] und der Überwachung der Übertragungsqualität (RTCP-Protokoll) während der Verbindung. Bis auf die RTCP-Pakete (UDP) werden allen anderen Steuerpakete mit dem Transportprotokoll TCP übertragen.

4.5.6.1 RTCP-Verkehr

Zu jeder RTP-Verbindung existiert bei einer bidirektionalen Videokommunikation genau ein entsprechender RTCP-Verbindung. Die Datenmenge, die über diese Verbindung versendet wird, ist nur von der Verbindungsdauer abhängig. Bei LiveLan ist die Länge der RTCP-Pakete konstant und beträgt 122 Bytes. Die Abstände, mit der RTCP-Pakete gesendet werden, liegen zwischen 2,5 und 7,0 Sekunden. Im Mittel ergibt sich ein Paketabstand von ca. 5,2 Sekunden und eine mittlere Senderate von ca. 0,188 kbit/s. Bei Vcon hingegen ist nicht nur die Länge der RTCP-Pakete sondern auch der Abstand konstant. Die Länge beträgt 94 Bytes, der Abstand 5,0 Sekunden und die sich daraus ergebende konstante Senderate 0,154 kbit/s. Die Länge der RTCP-Pakete schwankt zwischen 90 und 122 Bytes und der Mittelwert liegt bei ca.

110 Bytes. Die Paketabstände bewegen sich ähnlich wie bei LiveLan zwischen 2,5 und 7,0 Sekunden. Aus dem mittleren Paketabstand von ca. 4,8 Sekunden ergibt sich eine mittlere Senderate von ca. 0,183 kbit/s.

Zusammenfassend kann man sagen, dass die mittleren Senderaten der RTCP-Quellen unter 0,2 kbit/s und die Spitzenbitraten unter 0,4 kbit/s liegen. Die Übertragungsraten der RTCP-Steuerkanäle machen im Vergleich zu den Übertragungsraten der RTP-Nutzdatenkanäle in allen Fällen weniger als 1% und im Regelfall sogar weniger als 0,3% aus. Mit einem minimalen Paketabstand von 2,5 Sekunden und den sehr kleinen Paketlängen enthält er auch keine Bursts, die in irgendeiner Form andere Verkehre im Netz beeinflussen würden. Somit ist der RTCP-Verkehr im Vergleich zum RTP-Verkehr sehr einfach abschätzbar und muss nicht modelliert werden.

4.5.6.2 TCP-Verkehr

Der TCP-Verkehr hängt von der Signalisierungsarchitektur (H.323, SIP) und im Falle des H.323 zudem von der verwendeten Rufaufbauvariante ab. Neben dem Standard Rufaufbau gibt es seit der Version 3 des H.323 eine Sonderform „*Fast Connect*“, bei der zur Beschleunigung des Rufaufbaus mehrere Nachrichten der Signalisierung zu einer Nachricht zusammengefasst werden. Alle hier betrachteten H.323-Implementierungen entsprechen nur der Variante „*Standard*“.

Bei der Variante „*Standard*“ werden nacheinander 4 TCP-Ports geöffnet. Zwei Ports werden zum Zeitpunkt des Verbindungsauf- bzw. Verbindungsabbaus für die H.225.0 Signalisierung und für die H.245 Signalisierung benötigt. Die anderen zwei Ports dienen dem Austausch von Signalisierungsdaten während der laufenden Verbindung. Somit wird im Folgenden zwischen der Phase des Rufaufbaus, der Phase des Nutzdatenaustausches und der Phase des Rufabbaus unterschieden. Die Angaben zu Paketlängen und Senderaten beziehen sich wie bei den Analysen in den vorherigen Kapiteln auf Ethernetpakete.

Der Aufbau einer Punkt-zu-Punkt Sprachverbindung dauert nahezu genauso lange wie der Aufbau einer Sprach- und Videoverbindung, nämlich im Mittel 4 Sekunden. Dabei werden zwischen dem Initiator der Verbindung und dem gerufenen Teilnehmer im ersten Fall 37 und im zweiten Fall 49 Pakete ausgetauscht. Zwei Drittel der Pakete sind lediglich 72 Bytes lang. Nur 4 - 6 Pakete sind länger als 100 Bytes und konnten dem H.225.0 Verbindungsaufbau (SETUP - CONNECT) sowie einem Teil der H.245 Signalisierung („*Capability-Exchange*“) zugeordnet werden. Die maximale Paketlänge beträgt 231 Bytes bei Vcon und 376 Bytes bei LiveLan. Für den Rufaufbau ergibt sich somit für alle TCP-Verbindungen eine mittlere Senderate von 8 kbit/s.

Der Rufabbau dauert im Mittel nur 3 Sekunden und umfasst ca. 16 Pakete, die alle kürzer sind als 90 Bytes. Für den Rufabbau ergibt sich somit für alle TCP-Verbindungen zusammengenommen eine mittlere Senderate von 3 kbit/s.

Während der Verbindung wurden sehr unterschiedliche Raten gemessen. Bei Vcon werden wenige Pakete gesendet, so dass sich im Mittel Raten von weniger als 0,5 kbit/s für den TCP-Verkehr ergeben. Bei LiveLan sind die mittleren Übertragungsraten deutlich höher und liegen bei ca. 2 kbit/s. Darüber hinaus ist der Verkehr wesentlich bursthafter als bei Vcon. Im Abstand mehrerer Sekunden treten immer wieder einmal Bursts auf. Innerhalb der ersten 30 Sekunden einer Verbindung sind die Bursts größer als zu einem späteren Zeitpunkt. Die größeren Bursts besitzen nahezu konstante Dauer von durchschnittlich 1 Sekunde und setzten sich aus 15 – 35 in der Regel sehr kurzen Paketen (< 100 Byte) zusammen. Berechnet man die mittlere Rate solcher Bursts, so erhält man Werte zischen 10 und 20 kbit/s. Nach ca. 30 Sekunden treten nur noch Bursts von 5 – 8 Paketen auf. Auf der Suche nach dem maximalen Burst muss man noch viel kürzere Zeitintervalle als die zuvor untersuchten

Einsekundenintervalle betrachten. In einem zwei Millisekunden großen Zeitfenster konnten Burst-Längen von bis zu 730 Bytes ermittelt werden.

Netmeeting weist ein ähnliches Verhalten auf wie LiveLan. Bursts treten ebenfalls während der gesamten Verbindungsdauer auf, insbesondere jedoch innerhalb der ersten 30 Sekunden einer Verbindung. Bei genauerer Analyse fällt auf, dass die mittlere Paketlänge und der mittlere Paketabstand insgesamt größer sind. Bei der Betrachtung von Sekundenintervallen können die Bursts aufgrund der größeren Paketlängen noch stärker ausfallen und im Mittel sogar 30 kbit/s betragen. Die maximale Burstlänge in einem zwei Millisekunden großen Zeitfenster betrug 780 Bytes. Zu einem späteren Zeitpunkt der Verbindung wächst der Abstand der Bursts von 1-2 Sekunden auf 15 Sekunden an. Gleichzeitig umfasst ein solcher Burst lediglich 3 kurze Pakete (< 90Bytes).

4.5.6.3 Zusammenfassung

Zusammenfassend kann man sagen, dass der RTCP-Verkehr mit Übertragungsraten von weniger als 0,4 kbit/s, sehr kurzen Paketen von höchstens 122 Bytes und sehr großen Paketabständen von mehr als 2 Sekunden im Vergleich zum TCP-Verkehr zu vernachlässigen ist. Die maximale Burstlänge beträgt bei allen untersuchten Applikationen lediglich 122 Bytes. Der TCP-Verkehr erreicht beim Rufaufbau Raten von 8 kbit/s und während der Verbindungsphase mittlere Raten von ca. 2 kbit/s. Während sich die TCP-Pakete des Verbindungsaufbaus aufgrund ihres zeitlichen Versatzes nur unwesentlich mit den UDP-Paketen der Nutzdatenübertragung überlagern, können die TCP-Bursts während der Nutzdatenübertragung den UDP-Verkehr doch nachhaltig beeinflussen. Die maximale Burstlänge liegt bei Vcon bei 120 Bytes und bei LiveLan und Netmeeting bei 800 Bytes.

4.6 Ergebnisse im Überblick

In diesem Kapitel wurden die Sprach-, Video- und Signalisierungsverkehre der kommerziellen Videokonferenzapplikationen von Microsoft, PictureTel und Vcon analysiert. Dabei wurden der Durchsatz, die Verteilungen und die Korrelationen der Paketlängen und Paketabstände sowie die Langzeitabhängigkeiten genau untersucht. Dabei haben sich große Unterschiede sowohl zwischen den Applikationen als auch zwischen den verschiedenen Einstellungen ergeben. Die Ergebnisse sind in Tabelle 4-3 kurz zusammengefasst und werden abschließend noch einmal diskutiert.

Die Analyse der **Sprachcodecs** hat ergeben, dass bei allen Applikationen die Paketlängen während einer Verbindung konstant bleiben. Sie variieren von Applikation zu Applikation und hängen nicht nur vom Codec, sondern auch von dessen Implementierung ab. So konnten bei zwei verschiedenen Implementierungen des G.711 Standards unterschiedliche Strategien bei der Paketierung festgestellt werden. Die größten Unterschiede jedoch wiesen die Applikationen hinsichtlich ihrer Paket-Zwischenankunftszeiten auf.

Zusammenfassend kann man sagen, dass die untersuchten Implementierungen der Codecs G.711 und G.728 ein sehr deterministisches Verhalten aufweisen. Die Paket-Zwischenankunftszeiten folgen einem festen, periodisch wiederkehrenden Muster. Betrachtet man jedoch eine Messung des Sprachcodecs G.723.1 mit Sprechpausenunterdrückung, schwankt die mittlere Senderate, je nachdem wie intensiv das Gespräch geführt wurde, d.h. wie häufig Sprechpausen aufgetreten sind und wie lange diese gedauert haben. Für diese Verkehre konnten aufgrund der Verteilungen der Sprechpausen sogar Langzeitabhängigkeiten nachgewiesen werden. Eine Untersuchung des Einflusses der Rechnerauslastung auf das Sendeverhalten hat gezeigt, dass selbst die vermeintlich konstantbitratigen Sprachcodecs wie z.B. G.711 oder G.728 einen bursthaften Verkehr liefern können, wenn sie auf einem PC unter Windows’95/’98 ohne Task-Priorisierung in Kombination mit anderen gleichzeitig aktiven

Tasks laufen. Im Verkehrsverhalten der Quelle traten gelegentlich „Störungen“ auf. Diese Störungen zeigten sich auf Paketebene in Form einer längeren Sendepause gefolgt von einem Burst aus bis zu 3 Paketen.

Neben den Sprachcodecs wurden auch zwei Implementierungen des **Videocodecs** H.261 von PictureTel und Vcon sowie ein proprietärer Videocodec von Microsoft untersucht. Bei Videocodecs kann die aktuelle Senderate der Quelle von vielen Faktoren abhängen: vom Frametyp, der Framerate, der Bildauflösung, der Quantisierung der Bildpunkte, dem Informationsgehalt des Bildes sowie den Unterschieden der Bildinhalte zum jeweils vorangegangenen Bild. Neben der gewählten Einstellung wird die Senderate von den Bewegungen der Bildinhalte und Änderungen der Lichtverhältnisse über der Zeit beeinflusst. Insbesondere das Bewegungsverhalten der Objekte im Bildvordergrund hat großen Einfluss auf die Senderate. Das liegt daran, dass der Bildausschnitt bei Desktop-Videokonferenzsystemen in der Regel sehr klein ist und der Teilnehmer das Bild dadurch nahezu ausfüllt. Bewegungen von Teilnehmern stellen demzufolge nicht nur eine Veränderung von Objektpositionen im Bildausschnitt der Kamera dar, sondern beeinflussen auch die Helligkeitsverhältnisse im Bild.

Die Analyse der Implementierungen des Videocodecs H.261 bei LiveLan und Vcon hat sehr unterschiedliche Ergebnisse geliefert. Allgemein kann man sagen: Je größer die gewählte Auflösung und Bitrate, desto größer werden die Frames. Je geringer die Bewegungsanteile sind, desto stärker sinkt die mittlere Senderate unter den eingestellten Maximalwert ab und umso deutlicher ist das periodische Verhalten des Codecs erkennbar.

Bei niedrigeren Bitraten bildeten die Vcon-Applikation im Vergleich zu LiveLan größere Frames. Ferner haben die Paket-Zwischenankunftszeiten von Vcon bei allen Messungen eine größere Varianz aufgewiesen als die von LiveLan. Unterschiede haben sich auch bei starken Änderungen der Bildinhalte z.B. durch Bewegung ergeben. Während dies bei Vcon eine Erhöhung der Framerate zur Folge hatte, war bei LiveLan eine Zunahme der Framegröße festzustellen. Bei wenig Bewegung sank bei Vcon die Framerate mittelfristig um 20%, kurzfristig sogar auf bis zu ein Viertel der maximalen Framerate ab. Bei LiveLan hingegen fiel bei wenig Bewegung die mittlere Framegröße um 30% ab. Bei normalem Bewegungsverhalten konnte den Messreihen kein langzeitabhängiges Verhalten nachgewiesen werden.

Die Analyse des proprietären Codecs von Netmeeting hat ergeben, dass es sich hierbei um eine typische VBR-Quelle ohne Ratenkontrolle handelt. Die Senderate kann sich je nach Bewegungs- und Lichtverhältnissen des Bildes verdoppeln, in Extremfällen bei „Szenenwechseln“ sogar vervierfachen. Als Einstellparameter konnten nur die Bildgröße und die Bildqualität verändert werden. Je größer und besser die Bilder waren, desto größer wurden auch die Frames. Bei größeren Bildern konnte ein größerer Frame-Abstand festgestellt werden. Bildqualität und Bildinhalt hingegen hatten nur Einfluss auf die Framegröße, nicht jedoch auf die Framerate. Der Test auf langzeitabhängiges Verhalten lieferte Hurst-Werte zwischen 0,7 und 0,9. Der H-Parameter war umso größer, je extremer die Helligkeitsverhältnisse im Raum während der Messung verändert wurden.

Neben dem Nutzdatenverkehr wurde auch der **Signalisierungsverkehr** untersucht. Während sich die RTCP-Verkehre für die Modellierung als vernachlässigbar erwiesen, waren bei den TCP-Verkehren wesentlich höhere Übertragungsraten festzustellen. Insbesondere bei LiveLan- und Netmeeting-Applikationen wurden auch während der Nutzdatenverbindung Steuerinformationen ausgetauscht, denen ein bursthaftes Verhalten nachgewiesen wurde.

Anwendung	**LL**		**VCON**		**NM**		
Hardware	analoge Kamera, PCI Karte		digitale Kamera, PCI Karte		digitale Kamera, Parallelport		
Medium	**Video**	**Audio**	**Video**	**Audio**	**Video**	**Audio**	
Einstellungen	64 - 768 Kbit/s		64 - 768 Kbit/s		picture size		
	QCIF, CIF	-	QCIF, CIF	-	quality	-	
Kodierverfahren	**H.261**	**G.711**	**H.261**	**G.728**	**proprietär,** Windows 98	**G.723.1** Silence Suppr.	**G.711**
Quantisierung (Bit pro Pixel bzw. Sample)	24 Bit	8 Bit	24 Bit	8 Bit	k.a.	16 Bit	8 Bit
Kompressionsfaktor	100	1	100	4	k.a.	20.31	1
Framegröße	< 32 kbyte	8 Bit	< 32 kbyte	10 Bit	k.a.	189 Bit	8 Bit
Framerate	7.5 - 30 Hz	8000 Hz	7.5 - 30 Hz	1600 Hz	k.a.	33 Hz	8000 Hz
Spitzenbitrate (kbit/s)	7758	64	7758	16	k.a.	6.4	64
∅ **Rate des Coders** (kbit/s)	110 - 704	64	48 - 752	16	k.a.	≤ 6.4	64
Messung im Ethernet (incl. Ethernet-Header)							
Paketlänge incl. Ethernet-Header (bytes)	∅ 880 - 1160	578 konstant	∅ 934 - 1390	216 konstant	∅ 178 -1100	90 konstant	322 konstant
∅ **Rate der Anwendung incl. Paket-Overhead** (kbit/s)	-	72.25	-	23.04	-	≤ 24	80.5
∅ **Bitrate gemessen** (kbit/s)	128 - 620	72.25	64 - 730	21.76	20 - 300	11.83 - 19.92	80.5
Spitzenbitrate	0.3 - 9.5 Mbit/s	130 kbit/s	3.0 - 9.9 Mbit/s	45 kbit/s	4.5 - 9.9 Mbit/s	425 kbit/s	130 kbit/s
Bitratenvariation	gering	mittel	gering	sehr gering	hoch	sehr hoch	sehr gering
Framerate	15 - 30 Hz	-	7.5 - 30 Hz	-	3 - 17Hz I-Frames: 1-3 Hz	-	-
Paketanzahl pro Frame	1-3	-	1-2	-	1-6	-	-
Framegröße (Bytes)	430 - 3130 ∅ 1160	-	1100 - 2500 ∅ 1400	-	I-Frames 3450 - 6980 ∅ 5546 P/B-Frames: 180 - 1150	-	-
Hurst-Parameter	~ 0.55	-	~ 0.6	-	~0.7	~ 0.75	-

Tabelle 4-3: Ergebnisse im Überblick

Für die weitere Verwertung der Ergebnisse ist letztendlich der Verwendungszweck der Verkehrsuntersuchung ausschlaggebend. Die Netzarchitektur (DiffServ) und das Klassifizierungskonzept des Netzbetreibers beispielsweise entscheiden darüber, ob sich die Signalisierungs- und Nutz-Daten dieselben Netzressourcen teilen und sich dadurch gegenseitig beeinflussen können oder nicht. Kann eine Beeinflussung nicht ausgeschlossen werden, müssen bei allen Applikationen die Bursts des TCP-Verkehrs mit in die Entscheidung bei der Verbindungsannahme und damit auch in die Verkehrsbeschreibung der Quelle einbezogen werden. Werden sie hingegen unterschiedlichen Verkehrsklassen zugewiesen, genügt es, wenn die Verkehrsbeschreibung der Quelle nur den Nutzdatenanteil beinhaltet.

Während für die Netzplanung alle Verkehre berücksichtigt werden müssen, kann für das Ressourcenmanagement unter gewissen Umständen lediglich der Nutzdatenverkehr von Bedeutung sein. Aus der Sicht des Ressourcenmanagements müssen alle diejenigen Verkehrseigenschaften in der Verkehrsbeschreibung und im Verkehrsmodell zur Verbindungsannahmekontrolle berücksichtigt sein, welche für die Abschätzung der Verbindungsqualität benötigt werden.

Die statistischen Analysen haben einen sehr detaillierten Einblick in das Sendeverhalten der untersuchten Sprach- und Video-Quellen gegeben. Im Folgenden werden die Messungen dazu verwendet, geeignete Zugangskontrollverfahren für die RM-Architektur zu bestimmen.

5. Zugangskontrollverfahren

Viele Anwendungen stellen gewisse Mindestanforderungen an die IP-Netze hinsichtlich der Übertragungsqualität. Die Übertragungsqualität wird durch Verzögerungen und Verluste der IP-Pakete auf dem Weg durch das Netz bestimmt. Diese wiederum hängen von den zur Verfügung stehenden Ressourcen entlang des Ende-zu-Ende Datenpfades ab.

In heutigen IP-Netzen wird die Auslastung der Netzknoten nicht begrenzt, so dass es aufgrund der Aggregation von Verkehren immer zu spontanen Stausituationen kommen kann, deren Dauer und Intensität nur schwer vorhersagbar sind. Gerade die unkontrollierten Stausituationen machen die Garantie z.B. einer maximalen Ende-zu-Ende Verzögerung unmöglich.

Die Auslastung der Netzknoten kann nur über eine Zugangskontrollfunktion geregelt werden, welche die ankommenden Verkehre begrenzt. Sie erkennt Überlastsituationen im Voraus, indem sie Zustände des Netzes überwacht und entscheidet, ob eine neue Verbindung zugelassen werden darf oder nicht (Blockierung).

5.1 Stand der Technik

Netzzugangskontrollfunktionen wurden bereits in verbindungsorientierten Netzen wie z.B. den ATM- und Frame-Relay-Netzen realisiert. In verbindungslosen Paketnetzen wie dem Internet ist dagegen zur Zeit noch keine Netzzugangskontrolle realisiert. Es gibt aber bereits verschiedene Ansätze, um eine solche Zugangskontrolle einzuführen (siehe Kapitel 2).

Eine Klasse von Ansätzen ist messbasiert (***Measurement Based***) [Kni01], [Ele00], [Kel91], [Kel96]. Bei ihnen wird die Auslastung des Netzes entweder punktuell an den Netzknoten selbst (passiv) oder über mehrere Hops hinweg durch das Versenden von Testpaketen (aktiv) gemessen. In der Regel arbeiten diese Verfahren verteilt und kommen ohne eine zentrale Steuerungsinstanz aus. Wird an einem Knoten eine Überlast festgestellt, kann der Knoten selbst nach einem vorkonfigurierten Muster reagieren und eine entsprechende Reaktion (z.B. *dropping, shaping, re-marking*) einleiten. Alle diese Verfahren haben gemeinsam, dass sie spontan auftretende Stausituationen nicht vermeiden können, sondern nur zeitlich versetzt auf diese reagieren. Sie werden daher in der Literatur auch als reaktive Ansätze bezeichnet. Mit ihnen kann man nur statistische QoS-Garantien gewährleisten, dafür jedoch eine höhere mittlere Netzauslastung erzielen. Sie werden immer dann eingesetzt, wenn dem Anwender kurzfristige Beeinträchtigungen der Übertragungsqualität zugemutet werden können.

Will man als Netzbetreiber für Sprachdienste wie z.B. IP-Telefonie einen dem herkömmlichen Telefonnetz vergleichbaren oder sogar noch besseren Übertragungsdienst anbieten, sind reaktive Ansätze ungeeignet. Dafür gibt es die Klasse der parameterbasierten Ansätze (***Parameter Based***) [SPG97], [SSZ98], [EM93], [EMW95], [RR97]. Diese Verfahren beinhalten

eine Teilnehmer-Netz Signalisierung und basieren auf einem FIFO Scheduling. Sie werden auch als präventiv bzw. A-priori Verfahren bezeichnet, da sie zum Zeitpunkt des Verbindungsaufbaus, d.h. noch vor der Datenübertragung, eine Annahmeentscheidung treffen. Dazu schätzen sie den erforderlichen Ressourcenbedarf mit Hilfe der signalisierten Verbindungs- und Dienstgüteparameter sowie den gespeicherten Lastzuständen des Netzes ab. Sie gehen davon aus, dass abgelehnte Verbindungen am Netzzugang blockiert (verworfen) werden. Die parameterbasierten Verfahren kann man in zwei Gruppen unterteilen mit:

- **Harter QoS-Garantie:**
  Sie basieren auf einer worst case Verkehrsbeschreibung und einem deterministischen Quellenmodell. Sie garantieren eine maximale Ende-zu-Ende Verzögerung oder eine maximale Verlustwahrscheinlichkeit.
- **Weicher QoS-Garantie:**
  Sie basieren auf einem statistischen Quellenmodell und auf einer Verkehrsbeschreibung, welche die Variabilität des Verkehrs berücksichtigt.
  Das zugrundeliegende Puffermodell sieht einen Puffer vor, den sich mehrere Verbindungen teilen (*shared buffer*). Sie berechnen den Ressourcenbedarf einer Verbindung in Abhängigkeit von der Lastsituation des Puffers und können eine statistische Garantie der mittleren Verlustwahrscheinlichkeit geben.

Verfahren mit **harter QoS-Garantie** gehen häufig von einem verbindungsorientierten Netz aus, betrachten jeden einzelnen Datenstrom isoliert von den anderen Datenströmen und berechnen eine Effektive Bitrate für den gesamten Ende-zu-Ende Pfad [SPG97] (siehe Abschnitt 5.5.1.1). Um den gesamten Kapazitätsbedarf mehrerer Verbindungen auf jedem Link zu bestimmen, müssen die Effektiven Bitraten nur aufsummiert werden. Eine andere Gruppe von Verfahren geht von einem Puffermodell aus, in dem sich mehrere Verbindungen einen Puffer teilen. Sie überwachen den maximalen Verkehrszufluss, bei dem keine Verluste auftreten und garantieren somit eine Verlustwahrscheinlichkeit $P_{loss} = 0\%$ [EMW95], [RR97] (siehe Abschnitt 5.5.1.2).

Verfahren mit **weicher QoS-Garantie** berücksichtigen den statistischen Multiplexgewinn bei der Aggregation mehrerer Verbindungen auf einem Link. Sie überlagern mehrere Verkehre in einem Puffer z.B. einer Dienstklasse und bestimmen die Verlustwahrscheinlichkeit P_{loss} für den gesamten Puffer B in Abhängigkeit des Verkehrsgemisches [EM93], [EMW95], [RR97]. Bei diesen Verfahren gibt es wiederum mehrere Untergruppen, je nachdem, welche Berechnungsmethode zugrunde liegt:

- Berechnung einer Effektiven Bitrate E_j pro Verbindung j und Link [EM93].
- Berechnung der Verlustwahrscheinlichkeit P_{loss} für ein Verkehrsgemisch [EMW95].
- Berechnung einer aggregierten Bitrate C_{aggr} für alle aktiven Verbindungen pro Link [RR97].

Unter einer **Effektiven Bitrate** versteht man die virtuelle Bedienrate, die am Puffer benötigt wird, um eine bestimmte, garantierte Verlustwahrscheinlichkeit einhalten zu können. Kennzeichen eines Zugangskontrollverfahrens auf der Basis von Effektiven Bitraten ist, dass die Annahmeentscheidung durch eine einfache Summenbildung getroffen werden kann, und die Komplexität der Berechnung unabhängig ist von der Anzahl der bereits zugelassenen Verbindungen.

Die Vertreter der anderen beiden Gruppen berechnen entweder die zu erwartende Verlustwahrscheinlichkeit P_{Loss} oder eine Gesamtkapazität C_{aggr}, die am Ausgang des Puffers für das Verkehrsgemisch benötigt wird. Eine Klassifizierung verschiedener Zugangskontrollverfahren ist in Tabelle 5-1 dargestellt.

CAC-Architektur	Quellenmodell	Ankunftsprozess	Puffermodell (*B, C*)	Berechnungsmethode	Beispiele
parameterbasiert (pro-aktiv)	**deterministisch** *(siehe Abschnitt 5.5.1)*	**- ON/OFF** (*worst case*)	ungeteilt *(flow-based)*	**Effektive Bitrate** $C_{aggr} = \Sigma E_j(D_{ete}, r_j, b_j, p_j, H)$	[SPG97]
			geteilt *(class-based)*	**Effektive Bitrate** $C_{aggr} = \Sigma E_j(r_j, b_j, p_j, B, C)$ **Berechnung von C_{aggr}** $C_{aggr} = F(N_j, r_j, b_j, p_j, B)$	[EMW95] [RR97]
	statistisch *(siehe Abschnitt 5.5.2)*	**- PP** **- IPP / IFP** (+ *Bursts*) **- MMPP / MMFP** (+ *Korrelationen*) **(TES, ARIMA, fr-ARIMA, fr-BM)**	geteilt *(class-based)*	**Effektive Bitrate** $C_{aggr} = \Sigma E_j(r_j, p_j, C, P_{loss})$ $C_{aggr} = \Sigma E_j(r_j, b_j, p_j, P_{loss})$ **Berechnung von P_{loss}** $P_{loss} = F(N_j, r_j, b_j, p_j, C, B, P_{loss})$ **Berechnung von C_{aggr}** $C_{aggr} = F(N_j, r_j, b_j, p_j, P_{loss})$ $C_{aggr} = F(N, r, \sigma, H, B, P_{loss})$	[Lin94] [EM93], [GAN91] [EMW95], [Elv95] [RR97] [Gio96]
messbasiert (reaktiv)	**keines** *(siehe Abschnitt 5.5.2.4)*	**beliebig** (+ *Langzeitabhängigkeiten*)	geteilt *(class-based)*	**Effektive Bitrate** $E_j(s,t) = 1/st \cdot \log E[e^{sA_j[1,t]}]$ *large deviation approximation*	[Kel96]

IFP: Interrupted Fluid Process
IPP: Interrupted PP
MMPP: Markov Modulated PP
MMFP: MM Fluid Process

Tabelle 5-1: Klassifizierung der Zugangskontrollverfahren

Tabelle 5-1 enthält neben den Zugangskontrollverfahren die zugehörigen Puffer- und Quellenmodelle. Letztere werden in der Literatur für die Modellierung von VBR-Videoquellen verwendet. Es gibt eine Vielzahl von Verkehrsmodellen, die sich nach Quellentypen (Sprach-, Video-, Datenquellen) unterscheiden. Um die Komplexität möglichst gering zu halten, werden nicht alle Verkehrseigenschaften modelliert, sondern nur bestimmte Aspekte herausgegriffen. Bei der Analyse von bursthaften IP-Verkehren (Datenquellen, VBR-Videoquellen) kristallisierte sich heraus, dass die bislang in der Telekommunikation verwendeten *Poisson Prozesse (PP)* mit ihrer negativ-exponentiellen Verteilung der Ankünfte nicht geeignet sind, um das Pufferverhalten solcher Quellen hinreichend genau zu beschreiben. Es wurden daher ON-/OFF-Verteilungen (*IPP: Interrupted PP*) verwendet. Um den Korrelationen der Verkehre (VBR-Videoquellen) besser gerecht zu werden, wurden TES und *Fluid-Flow* Modelle entwickelt. Noch komplexere Verteilungen konnten mit Hilfe von *Markov Modulated Fluid* Prozessen (*MMFP*) gebildet werden. Weitere Modelle wurden entwickelt, um dem Phänomen der Langzeitabhängigkeit gerecht zu werden (*fr-ARIMA, fr-BM*).

5.2 Zielsetzung

In Kapitel 4 dieser Arbeit wurden die Verkehre von Videokonferenzsystemen gemessen und unter Anwendung verschiedener statistischer Methoden analysiert.

In diesem Kapitel werden nun die durchgeführten Messungen dazu verwendet, geeignete Netzzugangskontrollverfahren für das Ressourcenmanagementsystem zu finden. Mit Hilfe dieser Verfahren sollen zwei Arten von **Dienstklassen** eingeführt werden:

- **harte QoS-Garantie**: Diese Dienstklasse ist für besonders kritische Anwendungen mit harten Echtzeitanforderungen hinsichtlich der maximalen Paketverzögerung vorgesehen. Paketverluste werden ausgeschlossen.
- **weiche QoS-Garantie**: Diese ist für Anwendungen mit weniger kritischen Echtzeitanforderungen vorgesehen. Sie bietet eine statistische Garantie bezüglich des zu erwartenden Paketverlustes und der maximalen Paketverzögerung. Überlastsituationen müssen somit nicht hundertprozentig vermieden werden. Dies bedeutet, dass die Übertragungsqualität zeitlich schwankt. In seltenen Fällen können für die Teilnehmer kurzfristig Qualitätseinbußen auftreten.

Das Zugangskontrollverfahren soll bei jeder Anfrage eines Teilnehmers bezüglich einer neu aufzubauenden Videokonferenzverbindung die zusätzlich benötigten Ressourcen abschätzen können (siehe Abbildung 5-1). Als Eingabeparameter benötigt es vom Teilnehmer Informationen hinsichtlich der Verkehrscharakteristik (Quellenmodell) und der geforderten Dienstqualität (Dienstklasse).

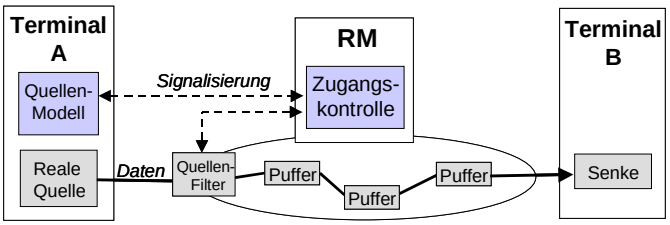


Abbildung 5-1: Quellenmodellierung als Teil des Ressourcenmanagements

Aus Sicht des Ressourcenmanagements ist es wünschenswert, einen einheitlichen Satz von Verkehrsparametern für alle Arten von Verkehrsquellen zu verwenden. Das macht Signalisierungs- und Zugangskontrollverfahren unabhängig von der Art der Quelle.

Als Quellen werden einzelne IP-Paketströme von Sprach- oder Videodaten verstanden, wie sie in Kapitel 4 gemessen wurden.

Die beiden Ziele dieses Kapitels sind in Abbildung 5-2 noch einmal zusammengefasst:

- Finden eines einheitlichen Verkehrsmodells für die zuvor untersuchten Quellen.
- Ermittlung geeigneter Zugangskontrollverfahren, welche auf diesem Verkehrsmodell basieren und harte und weiche QoS-Garantien gewährleisten können.

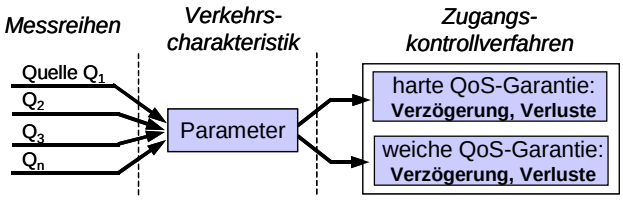


Abbildung 5-2: Zielsetzung

In den folgenden Abschnitten werden zunächst die Anforderungen an die Zugangskontrollverfahren definiert, woraus dann ein Verkehrs- und ein Netzmodell abgeleitet wird. Entsprechend dem Verkehrsmodell wird eine Methodik zur Charakterisierung von gemessenen IP-Verkehren entwickelt. Abschließend werden aus der Literatur geeignete Zugangskontrollverfahren ausgewählt, die auf den zuvor definierten Modellen basieren.

Die Verfahren werden mit Hilfe der charakterisierten Messreihen untersucht und miteinander verglichen. Dazu werden mit dem jeweiligen Verfahren die für die Einhaltung einer gewissen Dienstgüte benötigten Netzressourcen berechnet. Die Ergebnisse der Berechnungen werden mit Simulationen überprüft. Abbildung 5-3 stellt dieses Vorgehen noch einmal schematisch dar:

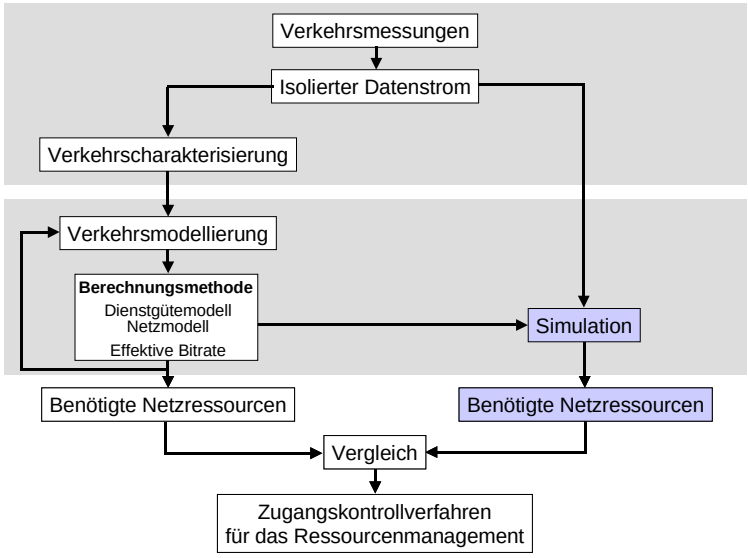


Abbildung 5-3: Vorgehen bei der Auswahl eines geeigneten Verfahrens

5.3 Modellierung

Jedem parameterbasierten Zugangskontrollverfahren liegen verschiedene Modelle zugrunde. Sie betreffen den Verkehr, die Dienstgüte und das Netz. Die einzelnen Komponenten der Modellierung werden in Abbildung 5-4 in einen funktionellen Zusammenhang gestellt.

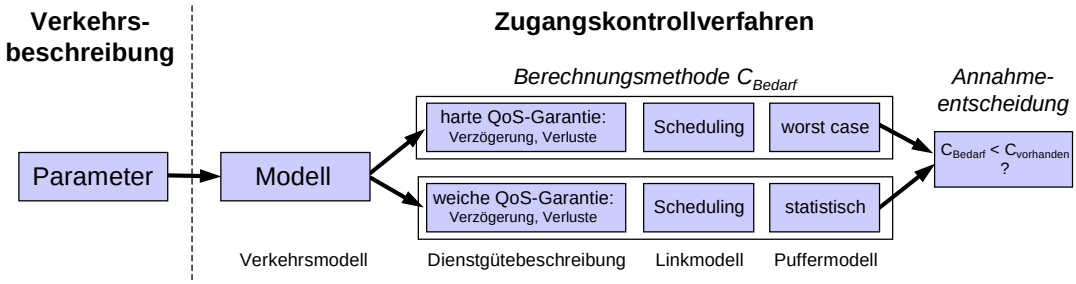


Abbildung 5-4: Komponenten der Modellierung

Ein Verfahren ist dann für die RM-Architektur geeignet, wenn die zugrundeliegenden Modelle mit den Anforderungen der RM-Architektur übereinstimmen. Die Anforderungen der RM-Architektur an die Modellierung können wie folgt formuliert werden:

- die Modellierung muss den Zugangskontrollverfahren ermöglichen, harte und weiche QoS-Garantien zu gewährleisten.
- die Modellierung soll sich für Sprach- und Videoverkehre gleichermaßen eignen.
- alle verwendeten Quellenmodelle sollen auf einer einheitlichen Verkehrsbeschreibung basieren.

Zudem soll die Berechnungsmethode eine möglichst geringe Komplexität besitzen, da sie zentral in den RM durchgeführt wird. Außerdem muss das zugrundeliegende Link- und Puffermodell mit dem in Abschnitt 3.2.3 definierten Knotenmodell übereinstimmen.

Aufbauend auf den Ergebnissen der Messungen von Kapitel 4 werden nachfolgend zunächst die Verkehrsparameter der RM-Architektur definiert und ein geeignetes Verkehrsmodell vorgestellt. Verschiedene Scheduling-Verfahren (Linkmodelle) werden hinsichtlich ihrer Komplexität für Netzknoten und Zugangskontrollverfahren diskutiert und damit die Begründung für die Wahl des WFQ-Schedulers im Knotenmodell geliefert (siehe Abschnitt 3.4.3).

5.3.1 Verkehrsbeschreibung

In diesem Abschnitt soll eine geeignete Charakterisierung der hier betrachteten Verkehre ermittelt werden. Sie stellt die Basis für die Verkehrsmodellierung und die Abschätzung des Ressourcenbedarfs eines Zugangskontrollverfahrens dar. Um die beiden Dienstklassen aus Abschnitt 5.2 realisieren zu können, muss die Verkehrscharakteristik sowohl das worst case Verhalten als auch das statistische Verhalten der IP-Quellen beschreiben.

Ein Zugangskontrollverfahren, das harte QoS-Garantien bezüglich der Verzögerung eines Paketes geben soll, benötigt eine worst case Beschreibung des Verkehrs (*traffic envelope*). Eine andere Art der Verkehrsmodellierung kann bei nachfolgender Berechnung oder Abschätzung des Pufferverhaltens zu Abweichungen von der Realität und damit zur Verletzung der QoS-Parameter führen.

Das worst case Verhalten einer Quelle wird üblicherweise in Form einer maximalen Senderate (Spitzenbitrate) oder als maximale Datenmenge angegeben, die pro Zeitintervall gesendet werden kann. Für die Modellierung des statistischen Verhaltens einer Quelle benötigt man

neben der Spitzenbitrate auch die mittlere Senderate und einen Parameter, der die Variabilität des Verkehrs (Ratenverteilung) beschreibt. Die Variabilität kann z.B. in Form eines Varianz-Parameters angegeben werden.

Für bursthafte Verkehrsquellen wurde im Zuge der ATM-Standardisierung eine andere Form der Verkehrsbeschreibung definiert. Ihr liegt das GCRA-Modell (*Generic Cell Rate Algorithm*) [CCITT Rec. I.371] zugrunde, welches besser in der Version des *Continuous State Leaky Bucket* bekannt ist. Dieser Ansatz wurde von der IETF im Rahmen der IntServ-Standardisierung als *Token Bucket* Modell übernommen (TSPEC) [Wr97]. Beide Modelle definieren Verkehrsparameter nach einer Regel, die eine eindeutige Unterscheidung zwischen konformen und nicht-konformen Zellen bzw. Paketen ermöglicht.

Im Gegensatz zum ATM wurden bei der IETF die Verkehrsparameter unabhängig von einer spezifischen Dienstklasse definiert. In der nachfolgenden Tabelle sind die Parameter der beiden Technologien einander gegenübergestellt.

	ATM	**IP**
Maximale Burstgröße	BT *Burst Tolerance*	b *Maximum Burst Size*
Maximale mittlere Rate	SCR *Sustainable Cell Rate*	r *Token Fill Rate*
Spitzenbitrate	PCR *Peak Cell Rate*	p *Peak Rate*
Maximale Schwankung der Verzögerung	CDVT *Cell Delay Variation Tolerance*	-

Tabelle 5-2: Verkehrsbeschreibungen von ATM- und IP-Netzen

Der Parameter “Maximale Schwankung der Verzögerung“ wurde nur bei ATM definiert. Er bezieht sich auf die maximal zulässige Abweichung von der Spitzenbitrate der Quelle im Netz und wurde für die Beschreibung von konstantbitratigen Quellen (CBR) eingeführt. Die Parameter “Burstgröße”, “maximale mittlere Rate” und “Spitzenbitrate“ haben in beiden Fällen dieselbe Bedeutung. Sie beschreiben sowohl das worst case Verhalten als auch das statistische Verhalten einer Verkehrsquelle hinreichend gut.

Für das Ressourcenmanagement wird in Anlehnung an die IETF-Standardisierung eine Verkehrscharakterisierung nach dem Token-Bucket Modell verwendet. Sie ist sowohl für konstantbitratige als auch für extrem bursthafte Quellen anwendbar. Dadurch kann das Ressourcenmanagement mit einem einheitlichen Satz von Parametern arbeiten und gleichzeitig flexibel für verschiedene Verkehrstypen (Verkehrsklassen) eingesetzt werden. Ferner ist die Interoperabilität mit anderen Reservierungsverfahren wie RSVP [BRA97] oder YESSIR [PS98] gegeben, da diese auf derselben Art der Verkehrscharakterisierung basieren.

Token-Bucket (TB) Modell

Das Token-Bucket (TB) Modell ist das Modell eines Netzfilters, der die maximale Burstgröße einer Quelle begrenzt. Er überwacht einen vorbeifließenden Paketstrom, indem er Sendeberechtigungen (Token) verwaltet, die zum Passieren des Filters benötigt werden. Er wird durch zwei Parameter definiert: *Token Fill Rate r*, *Bucket Size b*.

Abbildung 5-5 zeigt die Arbeitsweise eines *Simple Token Bucket* Filters (STB-Filter). Dieser verwaltet ein Konto mit Sendeberechtigungen (Token [*Bytes*]). Das Konto wird ständig mit der Füllrate *r* gespeist und hat eine maximale Größe *b*. Will ein Paket der Größe *L* [*Bytes*] den Filter passieren, wird überprüft, ob für die entsprechende Anzahl *L* an Sendeberechtigun-

gen vorliegen. Falls ja, werden die Sendeberechtigungen vom Konto abgebucht und das Paket darf den Filter passieren. Falls nein, wird es verworfen.

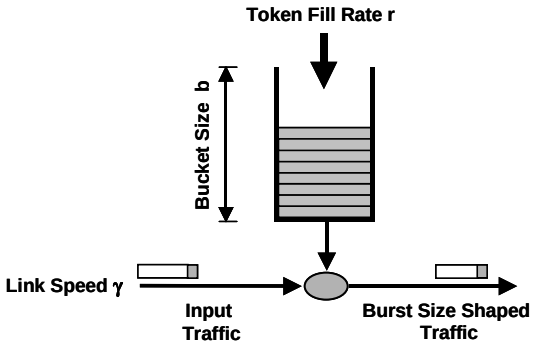


Abbildung 5-5: Modell des Simple Token Bucket

Durch die *Bucket Size b* wird das Datenvolumen eines Bursts beschränkt, während der zeitliche Abstand zweier aufeinanderfolgender Bursts durch *r* geregelt wird. Sendet der Teilnehmer mehr Bytes als zuvor in der Verkehrsbeschreibung signalisiert, werden die nichtkonformen Pakete verworfen.

Neben dem *STB* gibt es einen *Complex Token Bucket (CTB)*. Der *CTB*-Filter kann in Form von zwei hintereinandergeschalteten *STB*-Filtern implementiert werden. Er ist dadurch in der Lage, nicht nur die Burstgröße, sondern auch die Spitzenbitrate *p* der Quelle zu limitieren. Der erste Token Bucket begrenzt die maximale Burstgröße, der zweite die Spitzenbitrate mit einer *Bucket Size b* gleich der maximalen Paketlänge und einer Füllrate *r* gleich der Spitzenbitrate *p*.

Das TB-Modell kann nicht nur als Filter sondern darüber hinaus auch zur Charakterisierung gefilterter Verkehre verwendet werden. In diesem Fall stellt es ein worst case Verkehrsmodell dar, das die maximale Burstlänge der Quelle sowie den minimalen Abstand zwischen zwei maximalen Bursts charakterisiert.

5.3.2 Verkehrsmodell

Gesucht ist ein geeignetes Quellenmodell, welches auf der TB-Verkehrsbeschreibung beruht. Von diesem Quellenmodell sollen sich Zugangskontrollverfahren ableiten lassen, die harte und weiche QoS-Garantien ermöglichen. Um harte QoS-Garantien geben zu können, benötigt ein Zugangskontrollverfahren ein worst case Verkehrsmodell. Das Modell wird dazu verwendet, die maximale Verzögerung im Netz sowie den Pufferbedarf zu bestimmen.

In der IETF wurde im Zuge der IntServ-Architektur für den *Guaranteed Service* ein Zugangskontrollverfahren vorgeschlagen. Dieses gewährt eine harte QoS-Garantie und basiert auf einer TB-Charakterisierung der Quelle. Es verwendet ein ON-/OFF-Verkehrsmodell, wie viele andere Verfahren (siehe Tabelle 5-1, Abschnitt 5.1), die auf einer Verkehrsbeschreibung des TB-Modells bzw. Leaky Bucket Modells beruhen.

Ein ON-/OFF-Modell ist ein Quellenmodell, welches sich sehr einfach aus den TB-Parametern ableiten lässt. Eine Quelle sendet während der ON-Phase mit maximaler Rate *p*. Während der OFF-Phase hingegen sendet die Quelle keine Pakete. Die maximale Dauer einer ON-Phase sowie die minimale Dauer der OFF-Phase sind durch die Bucket Size *b* und die Token Füllrate *r* definiert (siehe Abschnitt 5.5.1.2).

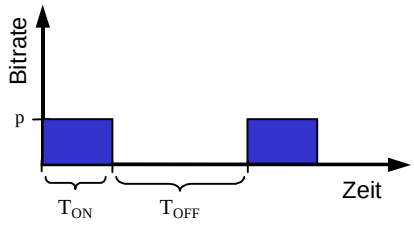


Abbildung 5-6: Periodisches ON/OFF-Modell

Ein worst case ON-/OFF-Modell basiert auf der Annahme, dass die maximale Burstgröße durch einen TB-Filter begrenzt wird. Das erste Paket eines solchen Bursts trifft dabei auf einen vollen Bucket. Der Bucket wird während des Bursts mit der Rate *p-r* geleert. Ist der Bucket leer, kann die Quelle höchstens mit der mittleren Rate weiter senden. Das letzte Paket eines maximalen Bursts ist für die Bestimmung der maximalen Verzögerung ausschlaggebend. Geht man davon aus, dass im Netz eine größere Übertragungsrate als die Token-Füllrate reserviert wird, ist es für die Betrachtung des Pufferbedarfs und der maximalen Verzögerung unerheblich, ob nach einem maximalen Burst die Quelle mit der mittleren Rate weitersendet oder nicht. Für diese Berechnungen werden allein die ON-Phasen der Quelle(n) herangezogen.

In einem periodischen ON-/OFF-Modell (siehe Abbildung 5-6) wird davon ausgegangen, dass die Quelle keine Pakete sendet, bis der Bucket erneut gefüllt ist. Nach der OFF-Phase sendet die Quelle dann wieder einen Burst von maximaler Intensität und Länge.

5.3.3 Link-Modell

Das hier vorgestellte Linkmodell ergibt sich aus dem in Abschnitt 3.4.2 bzw. 3.4.3 beschriebenen Netz- und Knotenmodell. Dort wurde ein Modell von unabhängigen Teilnetzen erstellt, die durch eine Kombination aus WFQ- und SP-Schedulern in allen Netzknoten realisiert werden. Damit kann eine maximale Entkopplung sowohl der RM-Verkehre von den nicht RM-Verkehren als auch mehrerer RM-Dienstklassen untereinander erreicht werden. In diesem Abschnitt werden Scheduling-Verfahren vorgestellt, ihre Auswirkungen auf die Zugangskontrollverfahren diskutiert und eine Begründung für die Auswahl des WFQ-Schedulers für das Knotenmodell geliefert.

Eine wesentliche Komponente des Knotenmodells aus Abschnitt 3.4.3 ist die Scheduling-Einheit. Sie ist notwendig, da auf einem Netzknoten mehrere Dienstklassen realisiert werden sollen. Der Scheduler entscheidet, wann welches Paket auf dem Link übertragen wird. Er verwaltet die Systemzeit des Ausgangs und bestimmt somit den Anteil, der einer Verkehrsklasse an der gemeinsamen Ressource „Linkkapazität“ zugeteilt wird. Er hat somit großen Einfluss auf die Verzögerungen eines Paketes bei der Paketverarbeitung.

In der Literatur gibt es verschiedene Arten von **Schedulern**: *First Come First Serve* (FCFS), *Fixed Priority*- (FP), sowie die Gruppe der *Delay-Based* (BD) und *Rate-Based* (RB) Scheduler. Will man in einem Netz mehr als einen Dienst (z.B. *Best-Effort)* anbieten, dürfen FCFS-Scheduler nicht verwendet werden. Will man zudem mehr als eine Dienstklasse mit einer Dienstgütegarantie ausstatten, benötigt man einen DB- oder RB-Scheduler. Ein DB-Scheduler arbeitet prioritätsbasiert. Er ordnet jedem Paket einen Zeitpunkt (*Deadline*) zu, bis zu dem es bedient sein muss und priorisiert die Pakete entsprechend ihrer *Deadline*. Ein RB-Scheduler basiert auf dem theoretischen Modell des *Generalized Processor Sharing* (GPS). Er arbeitet ratenbasiert und ist imstande, einer Dienstklasse eine minimale Bedienrate zu garantieren. Das GPS-Modell ist insbesondere für *Fluid-Flow* Verkehrsmodelle zur Berechnung

von Ende-zu-Ende Verzögerungsobergrenzen über mehrere Hops hinweg geeignet. Diese Art von Scheduler wird häufig auch als *Fair Queuing* Scheduler bezeichnet, wenn sie nicht nur eine gewisse Mindestbedienrate garantieren sondern darüber hinaus versuchen, nicht belegte Kapazitäten entsprechend ihrer Reservierung auf aktive Dienstklassen aufzuteilen. Das bekannteste Verfahren ist das *Weighted Fair Queuing* (WFQ).

Die **Komplexität** eines DB-Schedulers z.B. EDF (*Earliest Deadline First*) steigt mit der Anzahl *K* der zu bedienenden Pakete im Puffer *O(log K)* an. Sie kann jedoch auf eine ähnliche Komplexität wie die eines RB-Schedulers reduziert werden [LW94], die bei *N* Verbindungen im schlimmsten Fall *O(N)* beträgt. Eine Garantie hinsichtlich der Verzögerung kann mit DB-Schedulern nur gegeben werden, wenn vor dem eigentlichen Scheduling-Vorgang eine Ratenkontrolle der eingehenden Verkehre (*shaping*) durchgeführt wird. RC-EDF Scheduler wurden in [ZF94] eingeführt. In [GPS96] erfolgte der Nachweis, dass sie imstande sind, die Performance z.B. eines GPS-Schedulers nachzubilden und hinsichtlich der Ende-zu-Ende Verzögerung sogar zu übertreffen. Allgemein kann man jedoch sagen, dass DB-Scheduler aufgrund ihrer paketbasierten Arbeitsweise immer komplexer sind als RB-Scheduler.

Ein einzelner Scheduler arbeitet lokal in einem Netzknoten und stellt somit nur eine Verzögerungskomponente des Ende-zu-Ende Pfades dar. Ende-zu-Ende Dienstgütegarantien setzen eine **Zugangskontrollfunktion** für jeden Link voraus. Dazu muss die Ende-zu-Ende gültige Dienstgütespezifikation auf pro Netzknoten gültige Dienstgütespezifikation heruntergebrochen werden. Wird der Verkehrszufluss durch ein entsprechendes Zugangskontrollverfahren auf den Netzknoten geregelt (begrenzt), kann eine maximale Verzögerung pro Netzknoten und damit Ende-zu-Ende garantiert werden. Das Scheduling-Verfahren (*Schedulability Region)* gibt für die Einhaltung der Qualitätsvorgaben die maximalen Ankunftsraten der Pakete vor. Das Zugangskontrollverfahren führt dazu einen Test durch, der bei DB-Schedulern im allgemeinen wesentlich komplexer ist als bei RB-Schedulern [GGP97], [Geo96], [GAN91]. Das liegt daran, dass bei DB-Schedulern im Gegensatz zu RB-Schedulern nicht nur eine Überwachung der Bedienrate am Ausgang sondern auch ein zweiter Test für die Kontrolle der Verzögerungen (*Deadlines*) erfolgen muss [FKT97], [CS98].

Das **Ressourcenmanagement** kann prinzipiell sowohl für ein Netz mit DB- als auch mit RB-Schedulern eingesetzt werden. Ziel der RM-Architektur ist es, den Teilnehmern QoS-Garantien zu geben und gleichzeitig skalierbar zu sein. Die Komplexität bei der Paketverarbeitung in den Netzknoten ist daher ebenso gering zu halten wie in den RM-Instanzen selbst. Aus diesem Grund wurden für das Knotenmodell WFQ-Verfahren vorgeschlagen. Das hat drei entscheidende Vorteile hinsichtlich der Skalierbarkeit des RM-Systems:

- einfachere Zugangskontrollverfahren im RM-System
- einfachere Paketverarbeitung in den Netzknoten
- weite Verbreitung des WFQ-Schedulers in Routern

Für die Performance eines Ressourcen-Managers und damit für die Skalierbarkeit der Ressourcenmanagement-Architektur ist ein einfaches Zugangskontrollverfahren von entscheidender Bedeutung. Ein Ressourcen-Manager nimmt für alle Teilnehmer seiner Domäne die Verbindungsanfragen entgegen und durchläuft die Annahmeprozedur stellvertretend für alle beteiligten Netzknoten des Datenpfades. Alle in Abschnitt 5.1 aufgeführten parameterbasierten Verfahren kommen somit prinzipiell als Zugangskontrollverfahren in Frage.

Das **Knotenmodell** des Ressourcenmanagements sieht einige wenige Dienstklassen vor. Die Netzknoten verwalten keine Zustände pro Verbindung, sondern lediglich pro Dienstklasse.

Die Komplexität der WFQ-Scheduler muss daher im Vergleich zu einer IntServ Architektur mit mehreren zehntausend Zuständen nicht weiter betrachtet werden. Im RM-System ist jeder Dienstklasse ein Zugangskontrollverfahren zugewiesen, welches die Zustände dieser Dienstklasse verwaltet. Ein physikalischer Link (Knotenausgang) des Netzes kann somit aus einem Satz von **virtuellen Links** L_i, bestehend aus einem Puffer mit Kapazität B_i und einer garantierten Bedienrate C_i modelliert werden. Ist die Bedienrate eines virtuellen Links unabhängig von den Ankunftsprozessen der anderen Puffer, kann das Zugangskontrollverfahren allein auf der Basis der virtuellen Links arbeiten. Für jede Ende-zu-Ende Reservierung existieren entsprechende Pfade, die aus mehreren verketteten virtuellen Links mit vorhersagbaren Eigenschaften bestehen. Kennt das Zugangskontrollverfahren die Ankunftsprozesse auf allen Links des Pfades, kann es das Pufferverhalten des gesamten Pfades vorhersagen.

5.4 Charakterisierung der Quellen nach dem TB-Modell

5.4.1 Vorgehen

Für jede untersuchte Applikation wurden pro Einstellung verschiedene Messungen mit unterschiedlichen Umgebungsparametern (Bewegung, Beleuchtung) durchgeführt (siehe Abbildung 5-7). Jede dieser Messungen A_x soll über eine Abbildungsfunktion $F(A_x)$ ein Satz an TB-Parameter S_x gebildet werden, der das gemessene Quellenverhalten exakt beschreibt. Bei der Charakterisierung $F(A_x)$ sollen die Parameter möglichst klein d.h. nur so groß gewählt werden, dass bei einer Filterung des gemessenen Verkehrs gemäß diesen TB-Parametern gerade keine Verluste auftreten. Für jede Quelle, d.h. Applikation und Einstellung, soll aus den verschiedenen Parametersätzen einer ermittelt werden, der das worst case Verhalten beschreibt S_{max}.

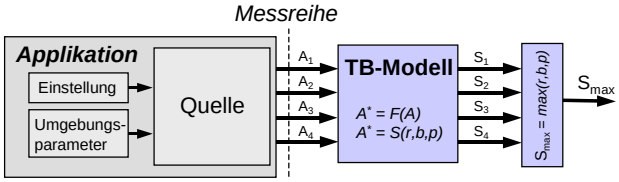


Abbildung 5-7: Vorgehen bei der Bestimmung der TB-Parameter (Modell)

5.4.2 Methodik zur Ermittlung der TB-Parameter

Im folgenden Kapitel wird die Methodik erläutert, mit der die Parameter Burstgröße (Bucket Size), Token-Füllrate und Spitzenbitrate aus den Messreihen ermittelt werden können. Es wird von einer unbekannten Quelle ausgegangen, die anhand von Messungen charakterisiert werden soll. Dieses Verfahren kann damit auch im laufenden Netzbetrieb zur Charakterisierung von Verkehren verwendet werden.

Bei der Charakterisierung sollen die TB-Parameter für jede Anwendung bei verschiedenen Einstellungen ermittelt werden. Dabei wird der gemessene Verkehr A auf einen Satz von TB-Parametern $A^* = S(r,b,p)$ abgebildet. Die Abbildung $F(A)$ ist so vorzunehmen, dass bei entsprechender Filterung der jeweiligen Messreihe mit einem TB-Filter keine oder zumindest nahezu keine Paketverluste auftreten würden, d.h. für den Verkehr gilt die Voraussetzung:
$A^* = A$.

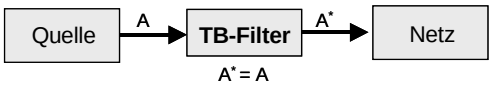


Abbildung 5-8: Verkehrsüberwachung im Netz

Die Spitzenbitrate *p* kann unabhängig von der Bucket Size *b* und der Token-Füllrate *r* bestimmt werden. Die Bucket Size *b* jedoch ist von der Token-Füllrate *r* abhängig. Der Zusammenhang zwischen der Bucket Size und der Token-Füllrate ist in nachfolgender Abbildung schematisch dargestellt.

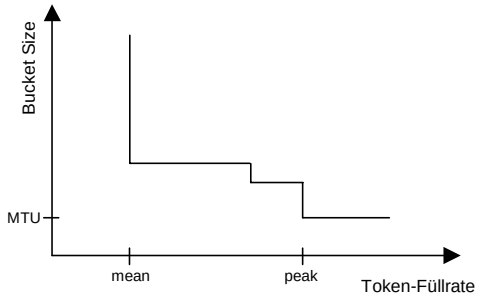


Abbildung 5-9: Zusammenhang zwischen Bucket Size und Token-Füllrate

Wird die Token-Füllrate kleiner als die mittlere Senderate gewählt, ist ein unendlich großer Bucket notwendig, um die Paketverluste zu begrenzen. Erhöht man schrittweise die Token-Füllrate, bleibt die erforderliche Bucket Size zunächst gleich groß und fällt dann in mehreren Stufen auf die maximale Paketlänge (MTU) ab, welche bei der Spitzenbitrate erreicht wird. Beide Parameter, *r* und *b*, stehen für den Ressourcenbedarf der Quelle. Sie sollen daher so bestimmt werden, dass der Ressourcenbedarf minimiert wird. Eine solche Optimierung kann natürlich nur für ein bestimmtes Zugangskontrollverfahren durchgeführt werden. Abbildung 5-10 zeigt im Vorgriff auf die in Abschnitt 5.5 untersuchten Verfahren eine Methode, wie mit Hilfe von Messungen die TB-Charakterisierung der Verkehre hinsichtlich des Ressourcenbedarfs optimiert werden kann.

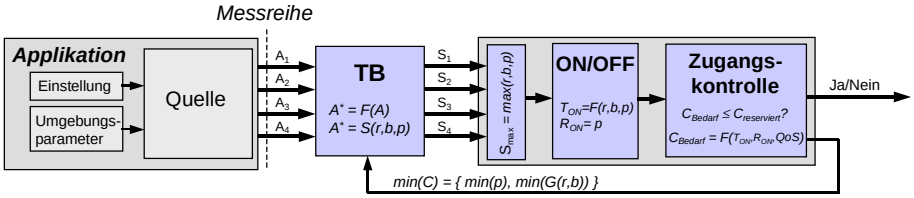


Abbildung 5-10: Vorgehen zur optimierten Parameterbestimmung

5.4.2.1 Spitzenbitrate

Die eindeutige Bestimmung der Spitzenbitrate anhand eines gemessenen Paketstromes, bestehend aus einer Sequenz von Paketen P_i mit dem Abstand PZA_i und Länge PL_i, bringt einige Schwierigkeiten mit sich. Die höchste gemessene Rate eines Paketstromes ist die Übertragungsrate, mit der jedes Paket im Ethernet gesendet wird (Linkkapazität z.B. 10 Mbit/s).

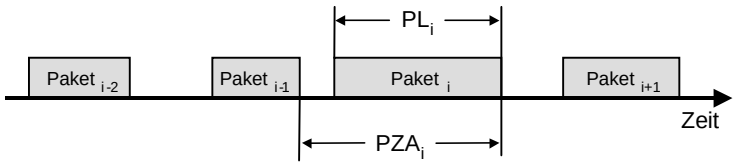


Abbildung 5-11: Paketstrom

Berücksichtigt man bei Bestimmung der Spitzenbitrate p_z den zeitlichen Abstand eines Paketes zu seinem Vorgänger-Paket, kann man diese aus den Messreihen wie folgt ermitteln (Abbildung 5-11):

$$p_z = \max\left(\frac{PL_i}{PZA_i}\right)$$

Liegt p_z nahe der Linkkapazität, ist zu prüfen, wodurch die kurzen Paketabstände zustande kommen. Es kann sich dabei sowohl um Störungen (*Jitter*) bedingt durch konkurrierende Prozesse auf dem Quellrechner als auch um völlig reguläre Vorgänge bei der Datenerzeugung bzw. Datenaufbereitung handeln. Bei einer Videoübertragung z.B. kann es sein, dass gelegentlich Bilder mit einem Datenvolumen von mehr als 1450 Bytes erzeugt werden (z.B. bei viel Bewegung). Der Bildinhalt muss dann auf mehrere IP-Pakete aufgeteilt werden. Diese Pakete werden unmittelbar hintereinander als Burst gesendet.

Als Erstes wird daher ermittelt, wie häufig und in welchen Abständen die hohen Spitzenwerte aufgetreten sind. Kamen sie sehr selten ($P < 10^{-3}$) und unregelmäßig vor, so handelt es sich um Störungen oder Sonderfälle. Ergibt sich durch Verwerfen solcher Pakete eine wesentlich niedrigere Spitzenbitrate p_q, werden diese nicht für die Ermittlung herangezogen. Zur Berechnung der Spitzenbitrate p_q wurde in diesen Fällen das 99,9% *Quantile* aller gemessenen Spitzenbitraten gebildet. Besonders bei LiveLan konnte bei der Einstellung von 384 kbit/s ein Absenken der Spitzenbitrate von 9.9 Mbit/s auf 2.0 Mbit/s und bei 174 kbit/s von 9,9 Mbit/s auf 400 kbit/s erzielt werden (siehe Tabelle 5-4). Betrachtet man die sich daraus ergebenden Konsequenzen, so begeht man bei der Ressourcenabschätzung anhand der veränderten p_q-Parameter keinen Fehler, solange der Verkehr am Netzzugang durch einen entsprechenden TB-Filter überwacht wird und alle nicht-konformen Pakete auch tatsächlich verworfen werden. Nur dann kann eine Beeinträchtigung des anderen Verkehrs ausgeschlossen werden.

Treten jedoch regelmäßig hohe Raten-Werte auf, kann eine Reduktion der Spitzenbitrate durch das Verwerfen einiger weniger Pakete nicht erzielt werden. In den Messreihen traten Bursts insbesondere bei Videoverbindungen mit Framegrößen von mehr als 1500 Bytes auf. Die Pakete eines solchen Bursts werden dann für die Berechnung der Spitzenbitrate p_b zusammengefasst:

$$p_b = \textit{Burstgröße [Bytes] / Burstsdauer [Sekunden]}$$

Zur Bestimmung der Burstgröße werden paketbezogene Raten berechnet und aufeinanderfolgende Pakete mit Raten-Werten größer eines bestimmten Grenzwertes als Burst gezählt. Die

Burstdauer wird aus der Zeitdifferenz aus letztem und erstem Paket und die Burstgröße aus der Summe der Paketlängen berechnet. Zur Bestimmung des Grenzwertes hat sich bei Codern mit Ratenkontrolle die Spitzenbitrate des Coders und bei solchen ohne Ratenkontrolle das Maximum der mittleren Raten nach Bildung von gleitenden 15 Sekunden-Intervallen bewährt.

Die Charakterisierung der Videoverkehre hat gezeigt (siehe Tabelle 5-4), dass sich in den Fällen die Spitzenbitrate deutlich absenken lässt, in denen häufig ein Frame auf mehrere Pakete aufgeteilt wird. Für das Ressourcenmanagement-System ergeben sich daraus zwei Konsequenzen:

- Mit p_b konfigurierte TB-Filter dürfen nicht zur Überwachung der Spitzenbitrate eingesetzt werden, da ansonsten die Verluste viel zu hoch wären und viele Bilder gestört werden könnten.
- Zum Schutz bereits zugelassener Verbindungen müssen am Netzzugang anstelle der TB-Filter *Shaper* mit entsprechend dimensionierten Puffern eingesetzt werden.

5.4.2.2 Token-Füllrate

Die Token-Füllrate *r* eines TB-Filters begrenzt die mittlere Senderate der Quelle. Ist der Token-Füllstand des TB kleiner als die Bucket-Size, werden mit einer konstanten Rate *r* Token nachgefüllt. Für die Charakterisierung einer Verkehrsquelle anhand der TB-Parameter bedeutet dies, dass die Token-Füllrate nicht kleiner sein darf als die mittlere Senderate der Quelle.

Bei der Verkehrscharakterisierung kann die Füllrate nicht unabhängig von der Bucket Size bestimmt werden. Da die Token-Füllrate während einer Verbindung konstant bleibt, die Senderate der Quelle jedoch über der Zeit schwankt, ergeben sich Abhängigkeiten zwischen den beiden Größen. Bei Verkehrsquellen mit variabler Senderate zeigt sich ein nichtlinearer Zusammenhang zwischen der Bucket Size und der Füllrate *B(r)*. Setzt man die Füllrate gleich der mittleren Bitrate der Messung, erhält man im allgemeinen sehr große *b*-Werte. Erhöht man die Füllrate, fällt die erforderliche Bucket Size zunächst sehr schnell ab. Ab einer gewissen Füllrate *r** kann jedoch die Bucket Size nicht oder nur geringfügig weiter reduziert werden.

Alle untersuchten Verfahren zur Berechnung des Ressourcenbedarfes eines Verkehrsstromes im Netz (siehe Abschnitt 5.5) reservieren für eine Verbindung mit einer etwas höheren Füllrate und dafür deutlich niedrigerer maximaler Burstgröße weniger Ressourcen als für eine Verbindung mit einer etwas geringeren Füllrate und dafür deutlich größerer maximaler Burstgröße. Für die Parametrisierung einer Quelle ist die Füllrate *r* daher gleich *r** zu setzen.

Nachfolgende Abbildungen zeigen zwei typische Verläufe von *B(r)*. Abbildung 5-12 steht dabei stellvertretend für eine Quelle mit Ratenkontrolle, Abbildung 5-13 für eine VBR-Quelle ohne Ratenkontrolle. Die Untersuchung der Messreihen mit der mittleren Rate r_{mean} hat ergeben, dass Quellen mit Ratenkontrolle ihre Sättigung mit $r^* = 1{,}3\ r_{mean}$ früher ereichen als reine VBR-Quellen. Eine VBR-Quelle wie Netmeeting erreicht ihre Sättigung in mehreren Stufen. Konstantbitratige Sprachquellen hingegen erreichen ihre Sättigung sehr schnell ($r^* = 1{,}05\ r_{mean}$).

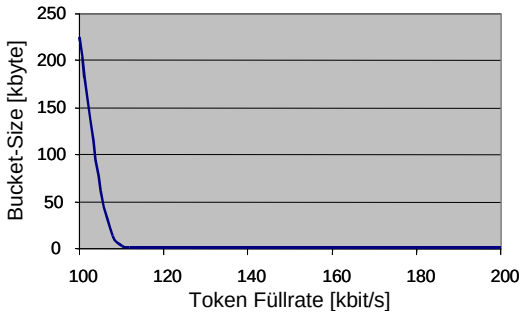


Abbildung 5-12: Charakterisierung LiveLan (174 kbit/s, CIF)

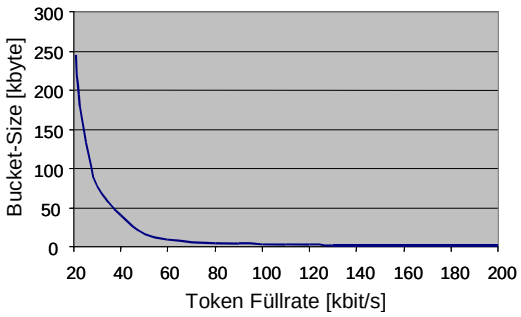


Abbildung 5-13: Charakterisierung Netmeeting (Bildgröße: mittel, Qualität: schnell)

Zur grafischen Bestimmung von *r** werden die relevanten Ausschnitte aus obigen Grafen in den Abbildung 5-14 und Abbildung 5-15 mit einer größeren Auflösung noch einmal dargestellt.

Die Bestimmung der Token-Füllrate zur Charakterisierung eines Videocoders mit Ratenkontrolle ist relativ einfach, da man anhand weniger berechneter *b*-Werte *r** schnell bestimmen kann. Abbildung 5-12 zeigt, dass *r** nur geringfügig über der mittleren Rate von 100 kbit/s der Messung liegt und einen Wert um 110 kbit/s erreicht. Abbildung 5-14 bestätigt dies bei höherer Auflösung und man kann aus der Grafik einen aufgerundeten Wert von 111 kbit/s ermitteln. Je nach Bewegungsverhalten der Teilnehmer können sich jedoch geringfügige Unterschiede ergeben. Untersuchungen haben gezeigt, dass sich bei starken Bewegungsänderungen höhere Werte für *r** ergeben als bei geringer Bewegung.

Ist die Quelle bekannt, kann man bei Codecs mit Ratenkontrolle den *r**-Parameter direkt aus der eingestellten Rate des Coders berechnen. Man benötigt für die gewählte Einstellung (Codec, Senderate) lediglich die mittlere Paketlänge der Applikation, kann daraus den Paketierungs-Overhead abschätzen und aus der Senderate des Coders eine Maßzahl für die mittlere Übertragungsrate auf dem Ethernet erhalten, die auf Dauer nicht überschritten wird. Aufgerundet ergibt sich für eine Videoverbindung von LiveLan bei der Einstellung 174 kbit/s und ein *r*-Wert von 120 kbit/s (siehe Tabelle 5-4).

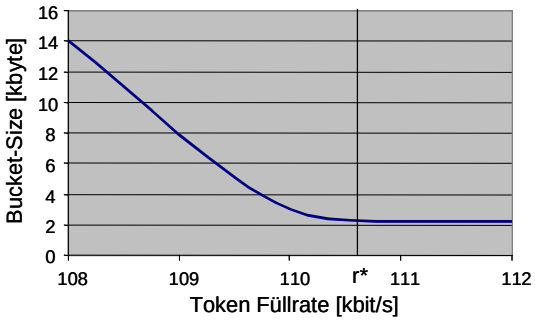


Abbildung 5-14: Bestimmung von r* LiveLan (174 kbit/s, CIF)

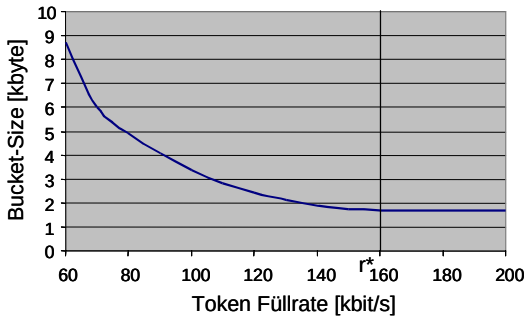


Abbildung 5-15: Bestimmung von r* Netmeeting (Bildgröße: mittel, Qualität: schnell)

Für die Bestimmung der Token-Füllrate zur Charakterisierung eines Videocoders ohne Ratenkontrolle werden mehr Stützwerte benötigt. Der Übergang in die Sättigung erfolgt in mehreren Stufen. Aus Abbildung 5-13 würde man vermuten, dass *r** einen Wert von ca. 80 kbit/s hat.

Aus Abbildung 5-15 geht hervor, dass bei ca. 70 kbit/s eine weitere Abflachung der Kurve *B(r)* stattfindet, eine Sättigung jedoch erst bei einem viel höheren Wert von *r** = 160 kbit/s

eintritt. Dieser Wert entspricht allerdings nahezu dem zuvor als Spitzenbitrate ermittelten Wert p_b.

Um den Parameter *r* in einer möglichst geeigneten Form zu bestimmen, wird der Aspekt der Ressourcenreservierung herangezogen. Beide Parameter, *r* und *b*, werden vom Ressourcenmanagement für die Bestimmung des Ressourcenbedarfs der Quelle verwendet. Die Verkehrsparameter *r* und *b* sind dabei so festzulegen, dass der berechnete Ressourcenbedarf minimiert wird. Die Applikationen sollen dabei unabhängig von einem bestimmten Zugangskontrollverfahren charakterisiert werden.

Daher wird eine allgemeingültige Berechnungsmethode der TB-Parameter eingeführt. Sie basiert auf der Annahme, dass die Quelle eine mittlere Senderate *r* und einen Varianzparameter *b** besitzt, der angibt, wie stark die Senderate um ihren Mittelwert schwankt. Er wird analog zur Bucket Size *b* ermittelt. Der Ressourcenbedarf wird hier einer zu reservierenden Rate gleichgesetzt und aus der Summe der mittleren Rate und einem mit einem Faktor *k* gewichteten *b**-Parameter bestimmt:

$$R = r + k \cdot b^*(r).$$

Unabhängig von *k* ergeben sich charakteristische Verläufe der Ressourcen, anhand derer der *r*-Parameter eindeutig bestimmt werden kann. Der Parameter *r* ergibt sich aus dem globalen Minimum der Funktion *R(r)*. Der Vorgang zur optimierten Parametrisierung ist in Abbildung 5-16 schematisch dargestellt.

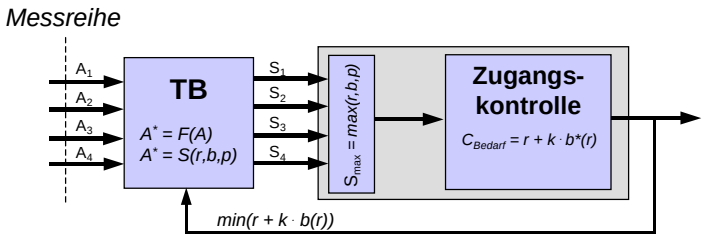


Abbildung 5-16: Vorgehen zur optimierten Parameterbestimmung

Abbildung 5-17 zeigt den Verlauf der Ressourcen *R* in Abhängigkeit von *r*. Aus dem Grafen geht hervor, dass bei der Messung mit Netmeeting und bei der Einstellung Bildgröße „mittel“, Qualität „schnell“ das Minimum bei einem *r*-Wert von 70 kbit/s liegt. Nach dieser Methode wurden alle Verkehre von Videocodern ohne Ratenkontrolle charakterisiert.

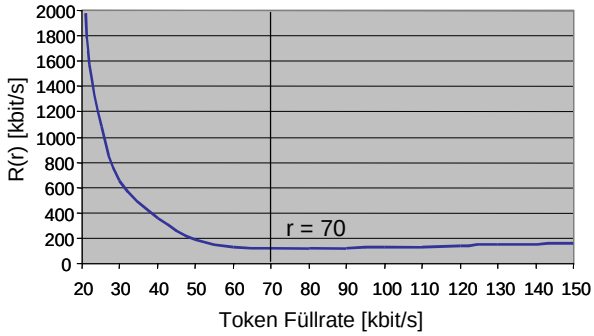


Abbildung 5-17: Ressourcenfunktion R(r) zur Bestimmung von r, Netmeeting (Bildgröße: mittel, Qualität: schnell)

5.4.2.3 Bucket Size

Entsprechend der Beschreibung des TB-Modells schwankt der Bucket Füllstand zwischen voll und leer. Zu Beginn einer Übertragung ist der Bucket gefüllt und enthält *b* Token. Sendet die Quelle mehrere Bursts mit einer höheren Rate als *r*, leert sich der Bucket. Ist er vollständig entleert, werden alle ankommenden Pakete verworfen. Ziel der Charakterisierung einer Quelle anhand von Messreihen ist nun, den Bucket so zu dimensionieren, dass ein entsprechend konfigurierter TB-Filter für diese Paketsequenz nicht leer läuft.

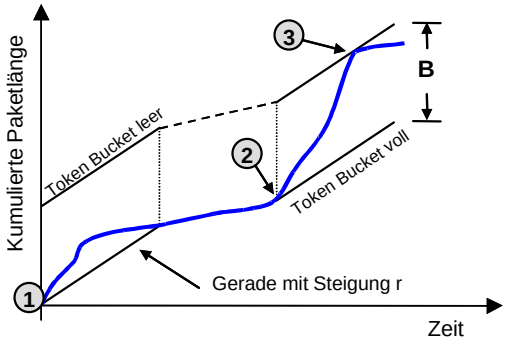


Abbildung 5-18: Füllstand des Token Bucket Filters

Die Bucket Size *b* kann aus den Messdaten und einer vorgegebenen Füllrate *r* durch die Simulation bestimmt werden. Dabei wird der Füllstand eines virtuellen Puffers bestimmt. Der Puffer kann unendlich große negative und positive Füllstände annehmen. Der Pufferfüllstand wird durch sequenzielles Abarbeiten der Messdaten wie folgt bestimmt:

Gestartet wird bei einem Füllstand 0. Wird ein Paket mit der Paketzwischenankunftszeit *t'* gemessen, wird zum Zeitpunkt *t'* eine gewisse Anzahl an Bytes $r \cdot t'$ hinzugefügt. Gleichzeitig wird eine gewisse Zahl an Bytes vom Puffer entfernt, welche der Länge des gemessenen Paketes entspricht. So wird für jedes Paket der Messung ein Pufferfüllstand berechnet. Am Ende der Simulation wird der zeitliche Verlauf eines Pufferfüllstandes bestimmt, wie er exemplarisch in Abbildung 5-19 gezeigt wird. Aus diesem Verlauf kann die erforderliche Bucket Size ermittelt werden, welche dem maximalen Abfall der Füllstandskurve entspricht.

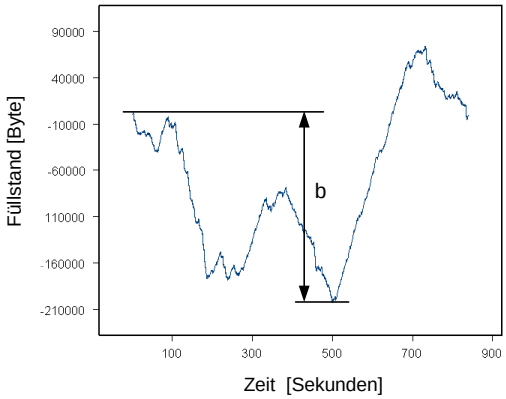


Abbildung 5-19: Simulation des Pufferfüllstands zur Bestimmung der Buket Size, r = r_{mean}

5.4.3 Charakterisierung der Quellen

Im Abschnitt 5.4.2 wurde die Methodik zur Bestimmung der TB-Parameter anhand von Messreihen erklärt. Diese Methodik wird nun auf alle Messreihen angewendet. Ziel der Charakterisierung ist es, für jede Anwendung und wählbare Einstellung einen Satz an Parametern zu definieren.

Jede Messreihe liefert zunächst ihren eigenen Satz an Parametern. Mehrere Messungen mit derselben Applikation bei identischer Einstellung führten aufgrund veränderter Systemparameter zu teilweise sehr unterschiedlichen Ergebnissen. Zu diesen Systemparametern zählen bei Videoanwendungen die Raumbeleuchtung sowie das Bewegungsverhalten der Teilnehmer. Bei Audioanwendungen mit Sprechpausenunterdrückung die Häufigkeit und Länge der Sprechpausen. Die Einflüsse dieser Systemparameter auf den Paketverkehr wurden bei den Messungen untersucht und schlagen sich folglich auch auf die Charakterisierung nieder.

Bei nicht echtzeitfähigen Betriebssystemen zählt auch die Auslastung des Rechners zu den Systemparametern. Die Auslastung des Endgerätes wurde jedoch nicht bei der Modellierung berücksichtigt, da dieser Einfluss in Zukunft durch den Einsatz echtzeitfähiger Betriebssysteme und größerer Prozessorleistungen vernachlässigt werden kann.

Bei den konstantbitratigen **Sprachcodecs** G.711 und G.728 fiel auf, dass die TB-Parameter der einzelnen Messungen nahezu identisch waren. Lediglich die Bucket Size variierte zwischen einem und zwei Paketen. Wurde jedoch der Rechner, auf dem die Applikation lief, zusätzlich mit anderen Aufgaben belastet, lag die Bucket Size bei bis zu 7 Paketen. Um einen einzigen Satz an Parametern zu erhalten und das worst case Verhalten der Quelle bei der Modellierung zu berücksichtigen, wurde die Bucket Size auf zwei Pakete festgesetzt. Höhere b-Werte traten nur in Sonderfällen mit der Wahrscheinlichkeit $P[b > 2\ Pakete] < 10^{-2}$ auf.

Berücksichtigt man diese *Sonderfälle* bei der Charakterisierung, beschreibt man den Verkehr in den allermeisten Fällen zu pessimistisch und vergeudet im Normalfall Netzressourcen. Das gilt insbesondere beim Multiplexen von vielen einzelnen Verbindungen über eine Leitung. Die Ergebnisse der Charakterisierung der Sprachquellen sind in Tabelle 5-3 zusammengefasst [GRC00].

Audio-Applikation		Einstellungen		Token Bucket			Paket-Länge
		$r_{codec,au}$ [kbit/s]	$r_{link,au}$ [kbit/s]	r [kbit/s]	b (r) [kbyte]	p [kbit/s]	L_{max} [byte]
LiveLan	G.711	64	72.25	73	1.16	130	578
Vcon	G.728	16	23.04	24	0.44	45	216
Netmeeting	G.711	64	80.50	82	0.65	110	322
	G.723.1	6.4	24.00	24	0.45	42	90

Tabelle 5-3: Sprachquellen

Bei den variabelbitratigen **Videoquellen** lieferte die Charakterisierung der Messreihen sehr unterschiedliche Ergebnisse. Je nach Teilnehmerverhalten ergaben sich für die H.261-Videoanwendungen große Unterschiede bei den Token-Füllraten, den Bucket Sizes und den Spitzenbitraten. Pro Applikation und Einstellung wurden mehrere Messungen mit unterschiedlichem Teilnehmerverhalten durchgeführt. Für jede einzelne Messung wurden die TB-Parameter bestimmt und aus allen Parametersätzen die größten *r*-, *b*- und *p*-Werte zur Charakterisierung der Applikationen in Tabelle 5-4 übernommen [GRC00]. Die Parameter enthalten somit nicht nur ein normales Teilnehmerverhalten, sondern auch Extremsituationen. Diese Extremsituationen können, wie bei LiveLan oder Vcon, einen kurzfristigen oder aber wie bei Netmeeting einen sehr starken Anstieg der TB-Parameter zur Folge haben. Eine Filte-

rung der so charakterisierten Verkehre mit Policern oder Shapern würde im Regelfall zu keinem, in extremen Ausnahmefällen lediglich zu Paketverlusten mit $P_{loss} < 10^{-2}$ führen.

Video-Applikation	Einstellungen		Token Bucket					Paket-Länge
	$r_{set,all}$ [kbit/s]	$r_{link,vi}$ [kbit/s]	r [kbit/s]	b (r) [kbyte]	p_z [Mbit/s]	p_q [Mbit/s]	p_b [Mbit/s]	L_{max} [byte]
LiveLan	768	740	750	9.0	9.9	9.5	2.0	1 520
	384	338	350	7.0	9.9	2.0	1.0	1 520
	174	117	120	2.5	9.9	0.4	0.4	1 520
Vcon	384	390	400	8.0	9.9	9.9	0.9	1 470
	128	119	120	3.2	9.9	6.2	0.4	1 470
Netmeeting	P: mittel Q: bestens	-	350	8.0	9.9	9.9	0.8	1 430
	P: mittel Q: mittel	-	100	2.5	9.9	9.9	0.4	1 430
	P: mittel Q: schnell	-	70	1.8	9.9	0.9	0.2	1 430

Tabelle 5-4: Videoquellen

5.5 Vergleich mehrerer Verfahren zur Ressourcenabschätzung

In diesem Kapitel werden verschiedene aus der Literatur bekannte Verfahren zur Ressourcenabschätzung beschrieben und untersucht. Sie berechnen die benötigte Puffer- oder Leitungskapazität im Netz anhand einer Verkehrsbeschreibung der Quelle und eines QoS-Kriteriums. Alle hier betrachteten Verfahren basieren auf Verkehrsmodellierungen, denen die TB-Parameter zugrunde liegen (siehe Tabelle 5-1, Abschnitt 5.1). Als QoS-Kriterien werden Verlustwahrscheinlichkeiten und Ende-zu-Ende Verzögerungen definiert. Sie stellen entweder eine harte oder weiche Grenze dar, welche unter keinen Umständen oder nur kurzfristig überschritten werden darf.

Im Folgenden werden zunächst Verfahren zur Ressourcenabschätzung betrachtet, die eine harte Garantie bezüglich der Einhaltung des QoS-Kriteriums geben können und anschließend Verfahren, die eine im statistischen Sinn weiche Garantie geben.

5.5.1 Verfahren mit harter QoS-Garantie

Die Verfahren mit harter Garantie werden vor allem bei echtzeitkritischen Verkehren verwendet, die sehr empfindlich auf Paketverzögerungen und Paketverluste reagieren. Sie müssen für einen Verkehr mit bekannter Verkehrsbeschreibung die minimal notwendigen Ressourcen so abschätzen, dass das QoS-Kriterium unter allen Umständen eingehalten werden kann. Dazu müssen zum einen die zur Verfügung stehenden Netzressourcen entlang des Datenpfades durch das Netz und zum anderen das Verhalten aller bereits aktiven Quellen bekannt sein. Reichen die zur Verfügung stehenden Ressourcen entlang des Datenpfades nicht aus, muss der neue Verkehr am Netzzugang blockiert werden.

Alle Verfahren mit harter Garantie basieren auf sogenannten worst case Szenarien. Dies gilt sowohl für die Verkehrsbeschreibung und das Verkehrsmodell der Quelle als auch für das Puffermodell der Netzknoten. So kann die maximale Verzögerung eines Paketes z.B. in einem Netzknoten nur dann bestimmt werden, wenn das worst case Sendeverhalten dieser Quelle bekannt ist. Kann eine gegenseitige Beeinflussung mit anderen Datenströmen nicht ausgeschlossen werden, gilt dies auch für alle anderen Datenströme, die gleichzeitig über diesen Netzknoten übertragen.

Im Folgenden werden drei verschiedene Verfahren betrachtet. Sie unterscheiden sich in ihrem QoS- und Netzmodell. Das Verfahren von IntServ [SPG97] gibt als QoS-Kriterium eine maximale Ende-zu-Ende Verzögerung der IP-Pakete an und zieht dabei alle Links des Netzes von der Quelle bis zur Senke in Betracht. Für jede neue Verbindung werden die benötigten Ressourcen (Puffer, Leitungskapazität) explizit berechnet und im Netz Ende-zu-Ende exklusiv für diese Verbindung reserviert. Die anderen beiden Verfahren von NEC [RR97] und Lucent [EMW95] betrachten lediglich einen einzelnen Link und garantieren eine verlustlose Übertragung auf diesem Link. In dem Knotenmodell teilen sich alle Verbindungen einen gemeinsamen Puffer.

5.5.1.1 Verfahren nach IntServ

Die IntServ Architektur basiert auf einem verbindungsorientierten Ansatz für das Internet mit dem Ziel, eine maximale Ende-zu-Ende Verzögerung der Pakete zu garantieren. Für jeden Paketstrom, der über das Internet übertragen werden soll, werden zuvor von der Anwendung Netzressourcen reserviert. Die zu reservierenden Ressourcen beinhalten Puffer- und Linkkapazitäten. Das Netzmodell von IntServ setzt voraus, dass diese Ressourcen dem Paketstrom exklusiv zur Verfügung stehen. Für jeden Datenstrom werden Bitraten und Pufferspeicher auf allen Knoten entlang des Datenpfades reserviert und dadurch wie bei einem leitungsvermittelten Netz ein Kanal mit einer festen Übertragungsrate Ende-zu-Ende aufgebaut.

Kommt ein Nutzdatenpaket an dem Eingangsport eines Netzknotens an, wird es ohne Verzögerung an den entsprechenden Ausgang vermittelt. Dort wird es in einen zuvor für diesen Verkehr reservierten Puffer geschrieben und mit der zuvor reservierten Kanalrate bedient. Betrachtet man die maximale Verzögerung eines Paketes, die bei der Übertragung auftreten kann, so muss man neben der reinen Laufzeit und der Sendedauer auf jedem Link zwei weitere Arten von Verzögerung unterscheiden. Zum einen das **Queuing-Delay**, welches von anderen zuvor gesendeten Paketen desselben Paketstromes verursacht wird, und zum anderen das **Scheduling-Delay**, welches durch Pakete anderer Paketströme bei der Aufteilung der Linkkapazität auf mehrere Datenströme auftritt.

Kommt ein Paket in einem Puffer an, können andere Pakete vor ihm im Puffer liegen. Diese werden bei FIFO-Puffern noch vorher bedient (Queuing-Delay). Rückt ein Paket an die erste Stelle im Puffer vor, sodass es als nächstes bedient werden kann, muss es dennoch eine gewisse Zeit bis zum Versenden warten (Scheduling-Delay).

Während das Queuing-Delay durch die Verkehrsbeschreibung vorgegeben wird, kann das Scheduling-Delay anhand der Paketlänge, der reservierten Rate des Paketstromes sowie der Linkrate bestimmt werden. Bezüglich der zu reservierenden Pufferkapazitäten wird im IntServ-Modell bei der Berechnung der maximalen Ende-zu-Ende Verzögerung davon ausgegangen, dass der Paketstrom einer Quelle irgendwo im Netzknoten gepuffert und auftretende Bursts dabei auf die Kanalrate R ge-shaped werden (Queuing-Delay). Wo dieses Shaping des Quellverkehrs auf die Rate R im Netz stattfindet, hängt vom verwendeten Scheduling-Verfahren ab. Entscheidend ist, ob ungenutzte Linkkapazitäten auf die gerade aktiven Verbindungen aufgeteilt werden (*work conserving*) oder nicht (*non-work conserving*).

Bei *work conserving* Schedulern kann es sein, dass einem Verkehr an einem weniger ausgelasteten Netzknoten eine höhere Rate zugewiesen wird als zuvor reserviert wurde. Das bedeutet, dass derjenige Netzknoten des Ende-zu-Ende Pfades das maximale Queuing-Delay verursacht, welcher am stärksten ausgelastet ist. Zum Zeitpunkt der Reservierung ist für die Dauer der Verbindung nicht vorhersagbar, wo im Netz dieser Shaping-Vorgang stattfindet. Daher müssen auf allen Netzknoten des Pfades entsprechend große Puffer reserviert werden. Zudem

kann es vorkommen, dass die maximale Burstlänge einer Quelle entlang des Datenpfades von Knoten zu Knoten zunimmt. Früher gesendete Pakete eines Teilnehmers können dabei auf ausgelastete Links treffen und verzögert werden, während nachfolgende Bursts ohne Verzögerung mit der Linkrate übertragen werden. Diese Verklumpung des Verkehrs auf dem Weg durch das Netz muss bei der Pufferdimensionierung berücksichtigt werden.

Bei *non-work conserving* Schedulern werden alle Puffer mit einer festen Rate bedient. Sie arbeiten dadurch wie ein Shaper. Verklumpungseffekte des Verkehrs auf dem Weg durch das Netz können somit ausgeschlossen werden. Das Queuing-Delay tritt in diesem Fall also nur am Netzzugang, das Scheduling-Delay in allen Netzknoten des Datenpfades auf. Daher sind am Netzzugang größere Puffer als innerhalb des Netzes erforderlich.

Das IntServ-Verfahren kann beide Arten von Schedulern bei der Ressourcenberechnung berücksichtigen. Für die Ressourcenberechnung müssen der Ende-zu-Ende Pfad durch das Netz und damit alle beteiligten Netzknoten bekannt sein. Dazu gehören die Anzahl der Knoten, die verwendeten Scheduling-Verfahren sowie die Leitungslängen (Laufzeiten). Darüber hinaus werden die Verkehrsparameter der Anwendung und das QoS-Anforderungen des Teilnehmers benötigt.

Der Standard IntServ sieht zwei unterschiedliche Arten der **Verkehrsbeschreibung** vor, die im Folgenden näher untersucht werden. Der einfachere Ansatz basiert auf dem Simple Token Bucket, der zweite auf dem Complex Token Bucket.

Variante I: Charakterisierungsmodell „Simple Token Bucket“ (STB)

Das Simple Token Bucket Modell geht davon aus, dass alle Pakete eines Bursts mit der Linkrate γ gesendet werden. Eine Begrenzung der Spitzenbitrate p der Quelle auf einen Wert $p < \gamma$ findet nicht statt. Gesucht ist nun eine Formel, anhand der die zu reservierende Rate R und der zu reservierende Puffer P berechnet werden können. Gegeben ist ein nach dem STB-Modell charakterisierter Verkehr mit Bucket Size b und Token Füllrate r, die maximale Paketlänge L_{max} des Paketstromes, ein Datenpfad von der Quelle zur Senke, bestehend aus N-Knoten, und eine maximal zulässige Ende-zu-Ende Verzögerung D_{e2e}.
Welche maximale Ende-zu-Ende Verzögerung auftreten kann ist in Gleichung 5-1 angegeben. Anhand dieser Methode kann man z.B. für eine bestimmte zu reservierende Bitrate R die maximal mögliche Verzögerung exakt bestimmen. Dabei entspricht der erste Term b/R dem nur einmal entlang des Pfades auftretenden maximalen Queuing-Delay. Der zweite Term enthält die Verzögerungen, die auf jedem Link auftreten. Dieser setzt sich aus dem Scheduling-Delay (C/R), der Sendedauer (L/γ) sowie der Laufzeit (D) zusammen. Die Link-Parameter C und D sind in Gleichung 5-3 für einen WFQ-Scheduler angegeben. Die Variable L des Parameters C steht dabei für die maximale Paketlänge des betrachteten Datenstromes und ist folglich für alle Links konstant. Die Variablen T_{prop}, γ und L_{max} des Parameters D beziehen sich auf den jeweiligen Link als Ganzes und stehen für die Laufzeit, die Linkkapazität und die maximale Paketlänge (MTU-Size).

$$D_{ete} \le \frac{b}{R} + \sum_{j=1}^{N}\left[\frac{C_j}{R} + D_j\right]$$

Gleichung 5-1: Maximale Ende-zu-Ende Verzögerung (STB)

$$R = \max\left[r, \frac{b + N \cdot L}{D_{ete} - \sum_{j=1}^{N}\left(\frac{L_{max,j}}{\gamma_j} + T_{prop}\right)}\right]$$

Gleichung 5-2: Effektive Bitrate R (STB)

Ist die maximal zulässige Verzögerung gegeben, lässt sich die zu reservierende Rate mit Gleichung 5-2 berechnen. Gleichung 5-2 ergibt sich aus Gleichung 5-1 durch Auflösen nach *R*. In Gleichung 5-2 sind die Link-Parameter aus Gleichung 5-3 bereis eingesetzt. Die Regel zur Pufferdimensionierung ist in Gleichung 5-4 gegeben. Während bei *work conserving* Schedulern der Pufferbedarf entlang des Datenpfades von Knoten zu Knoten wächst, ist er bei *non-work conserving* Schedulern konstant.

$$C_j = L_{\max,\ flow}$$

$$D_j = \frac{L_{\max\ j}}{\gamma_j} + T_{prop,\ j}$$

Gleichung 5-3: Link Parameter für WFQ-Scheduler

$$P_k = b + r \cdot \sum_{i=j}^{k} \left[\frac{L_{\max,\ flow}}{R} + \frac{L_{\max\ i}}{\gamma_i} \right] \qquad \text{(I)}$$

$$P_k = b; \quad \forall k \qquad \text{(II)}$$

Gleichung 5-4: Pufferdimensionierung (STB) (I) work conserving (II) non-work conserving

Variante II: Charakterisierungsmodell „Complex Token Bucket“ (CTB)

Das Complex Token Bucket Modell geht davon aus, dass alle Pakete eines Bursts mit der Spitzenbitrate *p* der Quelle gesendet werden, welche in der Regel kleiner als die Linkrate γ ist. Dadurch kann im Vergleich zum STB-Modell der Ressourcenbedarf gesenkt werden. Gesucht ist nun eine Formel, anhand der die zu reservierende Rate *R* und der zu reservierende Puffer P berechnet werden können. Gegeben ist ein nach dem CTB-Modell charakterisierter Verkehr mit der Spitzenbitrate *p*, Bucket Size *b* und Token Füllrate *r*, die maximale Paketlänge L_{max} des Paketstromes, ein Datenpfad von der Quelle zur Senke, bestehend aus N-Knoten, und eine maximal zulässige Ende-zu-Ende Verzögerung D_{e2e}.
Nachfolgende Gleichungen geben die maximale Ende-zu-Ende Verzögerung bei gegebener zu reservierender Rate *R*, die zu reservierende Rate *R* bei gegebener maximaler Verzögerung ($R \geq r$) und die zu reservierende Pufferkapazität *P* an.

$$D_{e2e} \leq \begin{cases} \dfrac{(b-L)}{R}\dfrac{(p-R)}{(p-r)} + \dfrac{L}{R} + \sum_{j=1}^{N}\left[\dfrac{L}{R} + \dfrac{L_{\max\ j}}{\gamma_j} + T_{prop,\ j} \right], & p > R \\ \dfrac{L}{R} + \sum_{j=1}^{N}\left[\dfrac{L}{R} + \dfrac{L_{\max\ j}}{\gamma_j} + T_{prop,\ j} \right] & p \leq R \end{cases}$$

Gleichung 5-5: Maximale Ende-zu-Ende Verzögerung (CTB)

$$R = \begin{cases} \dfrac{(b + L \cdot N) \cdot p - (N+1) \cdot L \cdot r}{b - L + (p-r) \cdot \left[D_{e2e} - \sum_{J=1}^{N} \dfrac{L_{\max,\ j}}{\gamma_j} + T_{prop,\ j} \right]}, & p > R \\ \dfrac{\sum_{J=1}^{N+1} L}{D_{e2e} - \sum_{J=1}^{N} \dfrac{L_{\max,\ j}}{\gamma_j} + T_{prop,\ j}}, & p \leq R \end{cases}$$

Gleichung 5-6: Effektive Bitrate R (CTB)

$$P_k = L_{max} + (b - L_{max}) \cdot \frac{p - X}{p - r} + X \cdot \sum_{i=j}^{k} \left[\frac{L_{max,\,flow}}{R} + \frac{L_{max}}{\gamma_i} \right]$$

Dabei ist X definiert mit:

$$X = \begin{cases} r\ ; & \frac{b - L_{max}}{p - r} < \sum_{i=j}^{k} \left[\frac{L_{max,\,flow}}{R} + \frac{L_{max,i}}{\gamma_i} \right] \\ R\ ; & \frac{b - L_{max}}{p - r} \geq \sum_{i=j}^{k} \left[\frac{L_{max,\,flow}}{R} + \frac{L_{max,i}}{\gamma_i} \right], \quad p > R \\ p\ ; & sonst \end{cases}$$

Gleichung 5-7: Pufferdimensionierung (CTB)

Will man die CTB-Formel in Abhängigkeit der STB-Formel darstellen (Z: Zähler, N: Nenner), ergeben sich für die Fälle $p > R$ aus:

$$R_{STB} = \frac{Z_{STB}}{N_{STB}}$$

folgende Beziehung für die CTB-Formel:

$$R_{CTB} = \frac{Z_{STB} - \frac{N \cdot L \cdot r}{p}}{N_{STB} + \frac{b - L}{p}}$$

Gleichung 5-8: R_{CTB} (STB)

Aus Gleichung 5-8 geht hervor, dass die CTB-Formel für $b \sim L$ und $p >> r$ in die STB-Formel übergeht. Am meisten Ressourcen lassen sich mit Hilfe der CTB- gegenüber der STB-Formel einsparen, wenn $p << \gamma$ und $b > L$. Ferner wächst das Einsparungspotential mit N.

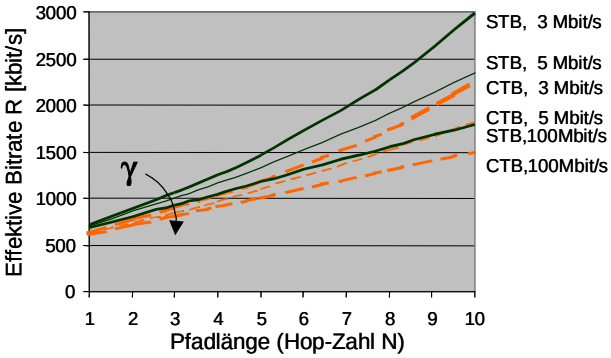
Bestimmung der Effektiven Bitraten nach IntServ

Im Folgenden sollen die untersuchten Quellen anhand ihrer Effektiven Bitrate miteinander verglichen werden. Die Effektive Bitrate hängt nicht nur von den Quellenparametern sondern auch von Netzparametern ab. Zur Berechnung der Effektiven Bitrate muss von einem Beispielnetz ausgegangen werden und die Netzparameter müssen dabei so gewählt werden, dass man den unterschiedlichen Ressourcenbedarf der jeweiligen Quellen gut erkennen kann.

In einem ersten Schritt soll dazu die Abhängigkeit der Effektiven Bitrate von der Länge des Datenpfades (Knotenzahl N) untersucht werden. In einem zweiten Schritt soll dann die Abhängigkeit der Effektiven Bitrate von dem QoS-Kriterium (D_{e2e}) untersucht werden. Ziel beider Untersuchungen ist es, eine geeignete Wahl der Netzparameter für den nachfolgenden Quellenvergleich zu finden.

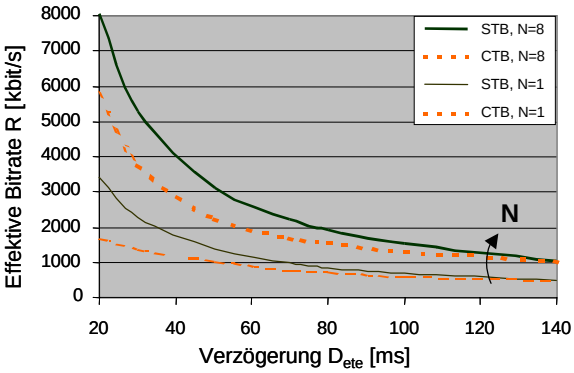
Abbildung 5-20 zeigt die Abhängigkeit der Effektiven Bitrate R von der Pfadlänge N und der Linkrate γ. Für hohe Linkraten und nicht allzu große N-Werte besteht ein nahezu linearer Zusammenhang zwischen R und N. Entscheidend dafür ist das Verhältnis von D_{e2e} und dem

Term $N(L/\gamma)$ aus den Nennern von Gleichung 5-2 und Gleichung 5-6. Solange dieses Verhältnis kleiner oder gleich 0.25 beträgt, kann die Funktion R(N) mit einer Gerade angenähert werden. Das gilt für die STB-Formel unabhängig von den Verkehrsparametern und für die CTB-Formel in den Fällen, in denen $(b-L) / (p-r) << D_{e2e}$ ist. Aus Abbildung 5-20 geht hervor, dass die *R*-Werte aus der CTB-Formel in allen Fällen kleiner als die R-Werte der STB-Formel sind und dass der Unterschied zwischen den Effektiven Raten der STB und der CTB für große N zunimmt. Ferner fällt auf, dass in einem 100 Mbit/s-LAN bei einer Hop-Zahl von 2 die Effektive Bitrate das Doppelte der mittleren Senderate der Quelle beträgt und bei einer Hop-Zahl von 5 bereits das Dreifache.
Um die verschiedenen Quellentypen miteinander vergleichen zu können, sollen Einflüsse des Netzes weitgehend vermieden werden. Bei einer Hop-Zahl von 1 gibt es nur einen Scheduler, der einen fixen Verzögerungsanteil (Scheduling-Delay) zur maximalen Ende-zu-Ende Verzögerung hinzufügt. Ansonsten dominiert das Queuing-Delay, welches von den Verkehrseigenschaften bestimmt wird. Für den späteren Vergleich der Effektiven Bitraten wird demzufolge die Hop-Zahl gleich 1 gesetzt. Für die Wahl der Linkgeschwindigkeit gilt Ähnliches. Auch hier soll der Netzeinfluss minimiert werden. Dabei ist zu berücksichtigen, dass die Linkraten im Kernnetz (Backbone) wesentlich höher liegen als im Zugangsbereich. Allgemein gilt, dass für die Verzögerung bedingt durch die Sendedauern der Pakete entlang des Pfades der Link mit der niedrigsten Linkrate dominiert. Bei einer Hop-Zahl von 1 wird eine Linkrate von 100 Mbit/s gewählt, wie sie im Bereich von Firmennetzen üblich ist.



Quelle: (r,b,p,D_{ete}) = (350 kbit/s, 7.0 kbyte, 2.0 Mbit/s, 100ms)

Abbildung 5-20: $R_{STB}(N,\gamma)$ und $R_{CTB}(N, \gamma)$



Quelle: (r,b,p,γ) = (350 kbit/s, 7.0 kbyte, 2.0 Mbit/s, 100 Mbit/s)

Abbildung 5-21: $R_{STB}(D_{ete}, N)$ und $R_{CTB}(D_{ete}, N)$

Ein weiterer wichtiger Parameter zur Bestimmung von *R* ist die maximale Ende-zu-Ende Verzögerung D_{e2e}. Der Zusammenhang zwischen *R* und D_{e2e} ist in Abbildung 5-21 für $N=1$ und $N=8$ dargestellt. Aus Abbildung 5-21 geht hervor, dass *R* mit größeren D_{e2e}-Werten hyperbolisch abnimmt. Für den betrachteten Verkehr ist für $N=8$ bei einem D_{e2e}-Wert von 80ms ewa das Fünffache, bei einem D_{e2e}-Wert von 140ms immer noch das Dreifache der mittleren Senderate zu reservieren. Bei der Wahl des D_{e2e}-Wertes ist zu beachten, dass sich zur Verzögerung der echtzeitkritischen Daten im Netz noch die Verzögerung bei der Signal- und Datenverarbeitung in den Endgeräten addieren. Um zu vermeiden, dass der Anwender die Folgen der Paketverzögerung zu spüren bekommt, sollen Gesamtverzögerungen von mehr als 150ms vermieden werden. Je nach Codec kann die Verzögerung in den Endgeräten bis zu 50ms betragen. Daher wird für die nachfolgenden Vergleiche von einer maximal zulässigen Netzverzögerung von $D_{e2e} = 100ms$ ausgegangen.
Für die folgenden Vergleiche der Videokonferenzsysteme werden die Netzparameter $N = 1$, $\gamma = 100$ *Mbit/s*, D_{e2e} = 100ms gesetzt und die Effektiven Bitraten nach den Gleichungen 5-2

(R_{STB}) und 5-6 (R_{CTB}) berechnet. Die Token-Bucket Parameter der Quellen werden aus Tabelle 5-2 und Tabelle 5-3 übernommen.

Die Ergebnisse sind in Tabelle 5-4 und Tabelle 5-5 zusammengefasst [GRC00]. Bei den Audioapplikationen (Tabelle 5-4) fällt auf, dass bis auf Netmeeting-G.711 die zu reservierende Rate ca. zweimal so groß ist wie die mittlere Rate r und zudem R_{STB} immer größer bzw. R_{CTB} immer kleiner ist als p. Netmeeting-G.711 hingegen lieferte einen so konstanten Datenstrom, dass die Reservierung der Token-Füllrate r ausreichend ist.

Das relativ gleichmäßige Sendeverhalten der Audioapplikationen mit absolut gesehen sehr niedrigen b- und p-Werten liefert auf den ersten Blick hohe Effektive Bitraten nahe den Spitzenbitraten. Das liegt zum einen an den niedrigen Spitzenbitraten und den sich daraus ergebenden geringen Kapazitätsanforderungen der Audioapplikationen, zum anderen an den im Verhältnis zu den kleinen TB-Parametern relativ großen Paketlängen. Ein Vergleich der beiden G.711 Implementierungen zeigt, dass bei Netmeeting ca. 30% geringere Kapazitäten benötigt werden als bei LiveLan. Das liegt an den größeren Paketlängen bei LiveLan, die sich auf alle TB-Parameter auswirken. Während r aufgrund des geringeren Paketierungs-Overheads kleiner ist, sind die Parameter p und insbesondere b wesentlich größer. Zusammenfassend kann man zum Paketierungsprozess sagen, dass bei der Generierung von vielen kurzen Paketen die Ressourcenbedarf trotz des größeren Paketierungs-Overheads geringer sind als bei der Bildung von wenigen langen Paketen.

Audio-Applikation		Einstellungen		Token Bucket			Paket-Länge	Effektive Bitrate	
		$r_{codec,au}$ [kbit/s]	$r_{link,au}$ [kbit/s]	r [kbit/s]	b (r) [kbyte]	p [kbit/s]	L_{max} [byte]	STB [kbit/s]	CTB [kbit/s]
LiveLan	G.711	64	72.25	73	1.16	130	578	140	110
Vcon	G.728	16	23.04	24	0.44	45	216	53	40
Netmeeting	G.711	64	80.50	82	0.65	110	322	82	82
	G.723.1	6.4	24.00	24	0.45	42	90	44	32

Tabelle 5-5: Effektive Bitraten (Audio-Applikationen)

Bei den Videoapplikationen ergeben sich aus den jeweiligen Spitzenbitraten p_z, p_q, p_b gemäß der CTB-Formel verschiedene Effektive Bitraten $R_{CTB,z}$, $R_{CTB,q}$, $R_{CTB,b}$ (Tabelle 5-5). Da die Raten p_z sehr hoch sind, liefern die entsprechenden Raten $R_{CTB,z}$ ähnliche Werte wie R_{STB}. Verwirft man wenige Pakete, können die Spitzenbitraten teilweise deutlich gesenkt und daraus niedrigere $R_{CTB,q}$–Werte berechnet werden. Wird der Verkehr zuvor ge-shaped, d.h. durch einen Puffer auf eine maximale Senderate p_b geglättet, können im Durchschnitt noch einmal 20% an Kapazität eingespart werden.

Video-Applikation	Einstellungen $r_{set,all}$ [kbit/s]	Token Bucket					Paket-Länge L_{max}, [byte]	Effektive Bitrate [kbit/s]			
		r [kbit/s]	b (r) [kbit/s]	p_z [Mbit/s]	p_q Mbit/s]	p_b [Mbit/s]		STB	CTB		
									p_z	p_q	p_b
LiveLan	768	750	9.0	9.9	9.5	2.0	1 520	843	837	837	813
	384	350	7.0	9.9	2.0	1.0	1 520	683	668	613	549
	174	120	2.5	9.9	0.4	0.4	1 520	322	320	278	278
Vcon	384	400	8.0	9.9	9.9	0.9	1 470	759	740	740	576
	128	120	3.2	9.9	6.2	0.4	1 470	375	371	369	290
Netmeeting	P: mittel Q: bestens	350	8.0	9.9	9.9	0.8	1 430	756	735	735	537
	P: mittel Q: mittel	100	2.5	9.9	9.9	0.4	1 430	315	313	313	267
	P: mittel Q: schnell	70	1.8	9.9	0.9	0.2	1 430	259	258	253	224

Tabelle 5-6: Effektive Bitraten (Video-Applikationen)

Die Ergebnisse sind in den Abbildungen 5-3 und 5-4 noch einmal grafisch dargestellt. Die Effektiven Bitraten der jeweiligen Applikationen werden der mittleren Bitrate (Token-Füllrate) und der Spitzenbitrate *p* gegenübergestellt. Dabei werden in Grafik 5-4 neben R_{STB} die Effektiven Bitraten $R_{CTB.q}$ und $R_{CTB,b}$ verwendet. Ferner wurde auf die Darstellung der Spitzenbitraten verzichtet, da sie wesentlich höher sind als die Effektiven Raten.

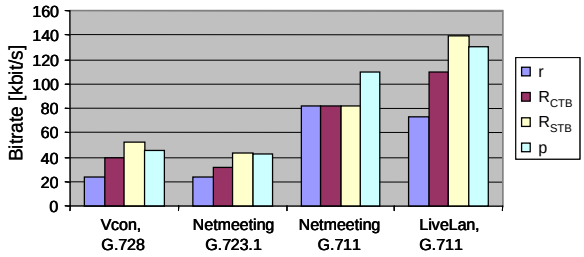


Abbildung 5-22 : Vergleich der Sprachcodecs
(Token-Füllrate r, Spitzenbitrate p)

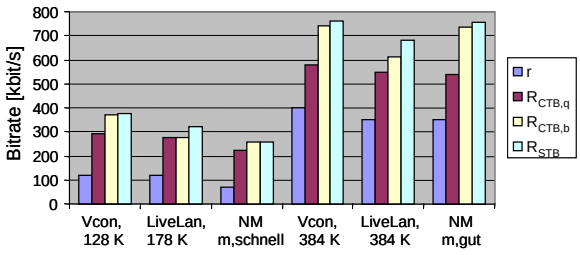


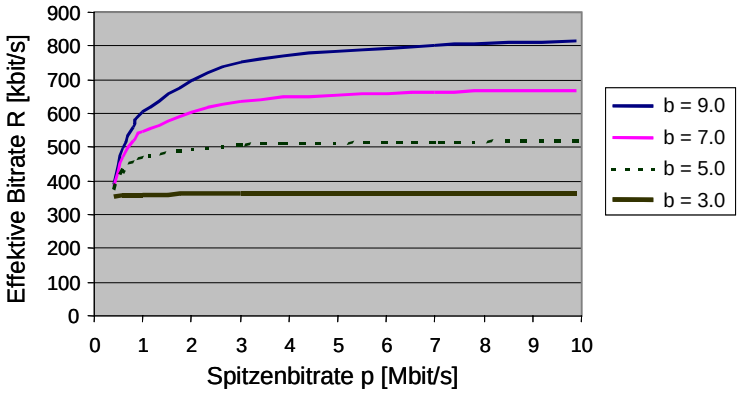
Abbildung 5-23: Vergleich der Videocodecs
(Token-Füllrate r)

Ein Vergleich obiger Abbildungen zeigt, dass bei Quellen mit konstanter Senderate (Abbildung 5-22) alle Ratenwerte sehr nahe beieinander liegen. Die Spitzenbitrate ist in der Regel nicht größer als das Doppelte der mittleren Rate. Während R_{CTB} in allen Fällen zwischen der mittleren und der Spitzenbitrate *p* liegt, nimmt R_{STB} häufig einen höheren Wert an als *p*. Das liegt daran, daß *b* im Verhältnis zu *L* klein ist und *L* im Vergleich zu *R* sehr groß. Daher liefert das Scheduling-Delay bereits bei einer Hop-Zahl von 1 einen relativ großen Beitrag zur gesamten Verzögerung.

Bei den Videoquellen (Abbildung 5-23) gilt in allen Fällen der Zusammenhang: $r < R_{CTB,q} \leq R_{CTB,b} \leq R_{STB} < p$. Im Gegensatz zu den Audioapplikationen ist R_{STB} wesentlich kleiner als *p*. Dies gilt auch bei den Videoquellen mit niedriger mittlerer Senderate. Da *b* wesentlich größer ist als bei den Audioanwendungen, ergeben sich höhere Werte für *R*. Da *R* wesentlich größer ist als *r*, liefert das Scheduling-Delay trotz der größeren Paketlängen *L* einen geringeren Beitrag zur Gesamtverzögerung.

Ferner fällt beim Vergleich der Raten $R_{CTB,q}$ und $R_{CTB,b}$ auf, dass durch das Absenken der Spitzenbitrate erheblich weniger Ressourcen reserviert werden müssen. Dieses Verhalten liegt in der Natur der CTB-Formel und soll genauer untersucht werden. Dazu ist der Zusammenhang zwischen der Effektiven Bitrate und den Parametern *p* und *b* in Abbildung 5-24 exemplarisch für eine Quelle mir $r = 350\ kbit/s$ dargestellt. Dabei wurde für verschiedene b-Werte die Funktion *R(p)* gezeichnet. Es fällt auf, dass die Funktion *R(p)* einen parabolischen Verlauf besitzt. Beginnend bei $p = r$ steigt *R* zunächst sehr stark an und nähert sich bei großen b-Werten langsam, bei kleinen *b*-Werten schnell einem Maximum an. Das Maximum selbst hängt auch von *b* ab und lässt sich näherungsweise aus Gleichung 5-6 wie folgt angeben:

$$R_{\max} \approx \frac{b + L \cdot N}{D_{e2e} - N \cdot \frac{L_{max,j}}{\gamma_j}}; \qquad p \to \infty$$



Quelle: r = 350 kbit/s, b [kbyte], Netz: N=1, γ = 100 Mbit/s)

Abbildung 5-24: R_{CTB} (p,b)

Das bedeutet, dass gerade für Videoverkehre mit großen *b*-Werten die Bestimmung der Spitzenbitraten einen sehr großen Einfluss auf die zu reservierenden Netzressourcen besitzen. Es wurden bei der Charakterisierung zwei verschiedene Varianten zur Ermittlung der Spitzenbitrate eingeführt. Während bei der Methode zur Bestimmung von p_q keine zusätzliche Pufferung des Verkehrs notwendig ist, werden für die Bestimmung von p_b Shaper vorausgesetzt. Die Rate p_b wurde so bestimmt, dass ein Frame mit maximaler Größe innerhalb des minimalen Frameabstandes übertragen werden kann. Das wiederum setzt voraus, dass der Quellverkehr gepuffert und der Puffer mit p_b ausgelesen wird. Nur so kann sichergestellt werden, dass die Senderate der Quelle immer kleiner p_b ist und dabei keine Pakete durch Filterung verloren gehen. Dadurch werden jedoch die Pakete eines Frames um maximal die Dauer eines minimalen Frameabstandes verzögert. Berücksichtigt man diese Verzögerung durch einen entsprechend verringerten D_{e2e}–Wert bei der Berechnung der Effektiven Bitrate $R_{CTB,b}$, so ergeben sich wieder dieselben Raten wie bei $R_{CTB,q}$. Durch den Einsatz von Shapern am Netzzugang kann lediglich die Pufferanforderung im Netz reduziert werden, nicht jedoch der Bandbreitenbedarf.

5.5.1.2 Verlustlose Verfahren nach NEC und Lucent

Die hier betrachteten Verfahren basieren auf einem Linkmodell und einem worst case ON-OFF Verkehrsmodell. Im Gegensatz zu der IntServ-Architektur wird hier von Verkehrsklassen ausgegangen, deren Verkehre sich gemeinsame Netzressourcen teilen. Auf jedem Link *j* des Netzes wird beim Einrichten der Verkehrsklassen eine Pufferkapazität B_j und eine Bedienrate C_j reserviert. Die Verfahren überwachen die Auslastung des Puffers, indem sie für das aktuelle Verkehrsgemisch den maximal möglichen Pufferfüllstand bestimmen. Beide Ansätze garantieren eine Paket-Verlustwahrscheinlichkeit von 0%. Die maximale Verzögerung D_j auf einem Link *j* ergibt sich dann aus dem maximalen Pufferfüllstand Q_j geteilt durch die Bedienrate C_j.
Das Verkehrsmodell ist ein periodisches ON-OFF Modell, welches in beiden Fällen aus den TB-Parametern abgeleitet wird (siehe Abbildung 5.5, Abschnitt 5.3.5). Während der ON-Phase sendet die Quelle mit der Rate *p*, während der OFF-Phase sendet sie keine Daten. Die Länge der ON- und OFF-Phasen ist für einen Verkehr konstant und wird bei den zwei betrachteten Verfahren unterschiedlich bestimmt.

Berechnungsmethode nach NEC_{vl} (verlustloses Muliplexen)
Die Verfahren wurden von den Autoren G. Ramamurthy und Q. Reng in [RR97] veröffentlicht. Beim NEC-Verfahren werden die Dauern T_{ON} und T_{OFF} wie folgt bestimmt:

$$T_{ON} = \frac{b}{p}; \qquad T_{OFF} = \frac{b}{p} \cdot \left(\frac{p-r}{r} \right); \qquad T_{ON} + T_{OFF} = \frac{b}{r}$$

Gleichung 5-9: Verkehrsmodell - NEC

Eine Verkehrsquelle *i* diesen Typs verursacht in einem Puffer mit Bedienrate *C* maximal den Füllstand q_i:

$$q_i = (p_i - C) \cdot T_{ON} = (p_i - C) \cdot \frac{b_i}{p_i}; \qquad p_i \geq C \geq r_i$$

Gleichung 5-10: Pufferfüllstand q_i, verursacht durch eine Quelle i

Dabei wird von nur einer aktiven Quelle im Puffer ausgegangen. Ein Puffer einer Verkehrsklasse muss imstande sein, gleichzeitig mehrere Verbindungen aufzunehmen, ohne dabei überzulaufen. Es wird für die Abschätzung des Pufferbedarfes von einem bereits gut ausgelasteten Puffer ausgegangen, der Pakete von N-Verbindungen enthält und nun einen weiteren Verkehr b_{neu} aufnehmen soll. Die dafür notwendige Kapazität q_{neu} wird näherungsweise bestimmt aus der Relation:

$$\frac{q_{neu}}{B} = \frac{b_{neu}}{b_{neu} + \sum_{j=1}^{N} b_j}$$

Gleichung 5-11: Näherung für q

Ferner wird davon ausgegangen, dass sich für jeden einzelnen Verkehr i im Puffer ein Anteil c_i an der gesamten Bedienrate *C* angeben lässt. Aus Gleichung 5-10 kann man für jede aktive Quelle einen Pufferfüllstand q_i in Abhängigkeit der Rate c_i angeben. Eingesetzt in Gleichung 5-11 ergibt sich daraus die für den neuen Verkehr benötigte Bedienrate c_{neu}:

$$c_{neu} = \begin{cases} p_{neu} \cdot \left(1 - \dfrac{B}{\sum_{J=1}^{N} b_j} \right); & c_{neu} > r_{neu} \\ r_{neu}; & c_{neu} \leq r_{neu} \end{cases}$$

Gleichung 5-12: Benötigte Bedienrate pro Verbindung, NEC_{vl}

Die benötigte Bedienrate hängt von der aktuellen Auslastung ab. Bei niedriger Auslastung des Puffers, d.h. $\Sigma b \leq b^* = B/(1 - r_{neu}/p_{neu})$ wird c_{neu} gleich r_{neu} gesetzt. Lässt man nun nach und nach weitere Verbindungen zu, steigt die zu reservierende Rate pro Verbindung an. Das gilt auch für die bereits zugelassenen Verbindungen. Die Effektive Bitrate $R_{N,v}$ hängt somit von der Auslastung des Puffers ab. Der Zusammenhang zwischen $R_{N,v}$ und der Pufferauslastung ist schematisch in nachfolgender Abbildung dargestellt.

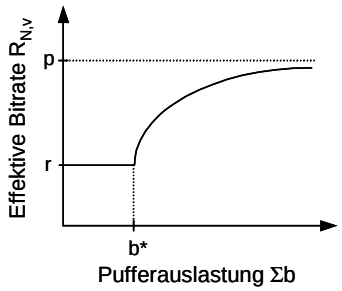


Abbildung 5-25: Effektive Bitrate $R_{N,v}$ (Σb)

Aus Gleichung 5-12 ergibt sich für den Kapazitätsbedarf des gesamten Puffers:

$$C_{Bedarf} = \max\left[\left(\sum_{j=1}^{N} p_j\right)\cdot\left(1-\frac{B}{\sum_{j=1}^{N} b_j}\right);\ \sum_{j=1}^{N} r_j\right]$$

Gleichung 5-13: Kapazitätsabschätzung NEC_{vl}

Mit jeder neu hinzugekommenen Verbindung ändert sich *Σb* und damit der Bedarf an Bedienrate jeder bereits zugelassenen Quelle. Daher kann keine Effektive Bitrate pro Verbindung angegeben werden. Ferner muss der Gesamtbedarf hinsichtlich der Bedienrate jedesmal komplett neu berechnet werden. Um den Puffer nicht zu überlasten, muss C_{Bedarf} immer kleiner gleich C sein. Die Zugangskontrolle des Netzes muss sicherstellen, dass zu jedem Zeitpunkt gilt:

$$C_{Bedarf} = \sum_{j=1}^{N} c_j \le C$$

Gleichung 5-14: Zugangskontrollfunktion NEC_{vl}

Handelt es sich um N Verbindungen desselben Typs, vereinfacht sich die Gleichung 5-13 zu:

$$C_{Bedarf} = \max\left[(N\cdot p)\cdot\left(1-\frac{B}{N\cdot b}\right);\quad N\cdot r\right]$$

Bestimmung der Effektiven Bitraten nach $Lucent_{vl}$ (verlustloses Multiplexen)
Beim Lucent-Verfahren [EMW95] werden die Dauern T_{ON} und T_{OFF} wie folgt bestimmt:

$$T_{ON} = \frac{b}{p-r};\qquad T_{ON}+T_{OFF} = \frac{b\cdot p}{r\cdot(p-r)};\qquad T_{OFF} = \frac{b}{r}$$

Gleichung 5-15: Verkehrsmodell – $Lucent_{vl}$

Die Modellierung der ON-Phase erfolgt analog zum IntServ CTB-Modell. Ein Vergleich mit dem Verkehrsmodell von NEC zeigt, dass bei Lucent die Zeitdauer der ON-und OFF-Phasen länger sind. Das Lucent-Modell geht somit von längeren Bursts aus als NEC. Es berücksichtigt bei der Bestimmung der ON-Dauer, dass der Bucket sich während eines Bursts weiter mit r füllt. Bestimmt man die maximale Warteschlangenlänge q_i in einem Puffer (B,C), verursacht durch den Verkehr einer Quelle i, ergibt sich folgende Beziehung:

$$q_i = (p_i - C) \cdot T_{ON} = (p_i - C) \cdot \frac{b_i}{p_i - r_i}; \qquad p_i > C$$

Gleichung 5-16: Pufferfüllstand q, verursacht durch eine Quelle i

Will man für den Verkehr *i* einen Anteil an der gesamten Bedienrate des Puffers angeben, wird dieser anhand folgender Beziehung angenähert:

$$\frac{q_i}{B} = \frac{c_i}{C}$$

Gleichung 5-17: Näherung für q

$$c_{neu} = \begin{cases} \dfrac{p_{neu}}{1 + \left(\dfrac{B}{C \cdot b_{neu}} \right) \cdot (p_{neu} - r_{neu})}; & r_{neu} < \dfrac{b_{neu} \cdot C}{B} \\ r_{neu}\ ; & \dfrac{b_{neu} \cdot C}{B} < r_{neu} < p_{neu} \end{cases}$$

Gleichung 5-18: Benötigte Bedienrate pro Verbindung, Lucent$_{vl}$

Die benötigte Bedienrate kann bei diesem Verfahren für jede Verbindung unabhängig von der aktuellen Pufferauslastung angegeben werden. Die Effektive Bitrate $R_{L,v}$ kann in Abhängigkeit der TB-Parameter und der Pufferparameter *B* und *C* berechnet werden. Die maximale Verzögerung D_{max} eines Paketes in einem Puffer (*B,C*) beträgt: $D_{max} = B/C$. Abbildung 5-26 zeigt die Effektiven Bitraten beispielhaft für einige Messreihen der untersuchten Videoquellen in Abhängigkeit von D_{max}.

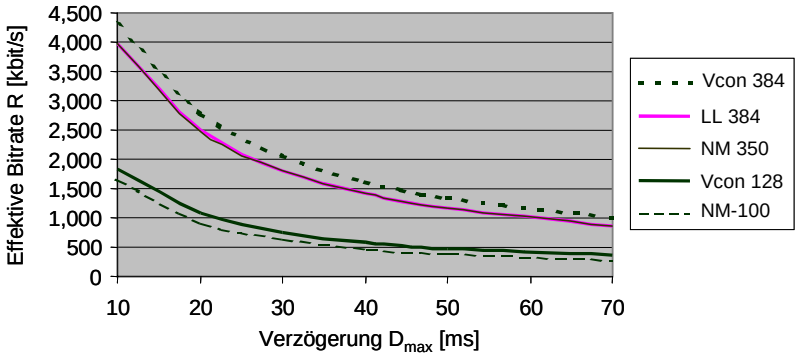


Abbildung 5-26: Effektive Bitrate $R_{L,v}$ (D_{max})

Vergleicht man die Effektiven Raten $R_{L,v}$ mit denen der CTB-Formel R_{CTB} bei identischen *D*-Werten, so fällt auf, dass die R_{CTB}–Werte deutlich (10-15%) über den $R_{L,v}$-Werten liegen. Im Gegensatz zu dem hier betrachteten Verfahren berücksichtigt die CTB-Formel neben dem reinen Queuing auch das Scheduling und liefert dadurch zwangsläufig höhere Raten. Vernachlässigt man das Scheduling ($L/R = 0$, $L/\gamma = 0$), gehen die beiden Gleichungen ineinander über.

Für eine Kontrollfunktion zur Verbindungsannahme auf einem Link gilt folgende Beziehung:

$$C_{Bedarf} = \sum_{j=1}^{N} c_j \leq C$$

Gleichung 5-19: Zugangskontrollfunktion Lucent$_{vl}$

5.5.1.3 Vergleich der Verfahren mit harter QoS-Garantie

Zum Abschluss sollen die Verfahren mit harter QoS-Garantie miteinander verglichen werden. Da das NEC-Verfahren für alle Verbindungen eine Gesamtrate berechnet, die sehr stark von der Auslastung abhängt, wird für den Vergleich nicht die Effektive Bitrate, sondern die erzielbare Auslastung für ein Beispielnetz herangezogen.

Gegeben sei ein Netz, welches aus einem einzelnen Knoten besteht, auf dem mehrere Dienstklassen realisiert sind. Die Linkkapazität C beträgt 100 Mbit/s. Für die Dienstklasse der Sprachverbindungen wird eine maximale Verzögerung $D = 30ms$ angesetzt. Dafür steht netzseitig eine Kapazität C^* von 15 Mbit/s zur Verfügung, die nicht überschritten werden darf. Anhand der Verfahren soll nun die maximale Anzahl von Verbindungen N ermittelt werden, die zugelassen werden kann. Dabei werden nur immer Verkehre desselben Typs überlagert. Die Puffergröße B wird aus D und C^* ermittelt: $B = D \cdot C^* = 56{,}25\ kbyte$.

Die Ergebnisse sind in Abbildung 5-27 grafisch dargestellt. Tabelle 5-6 enthält die berechneten Werte für die maximal zulässige Anzahl an Verbindungen.

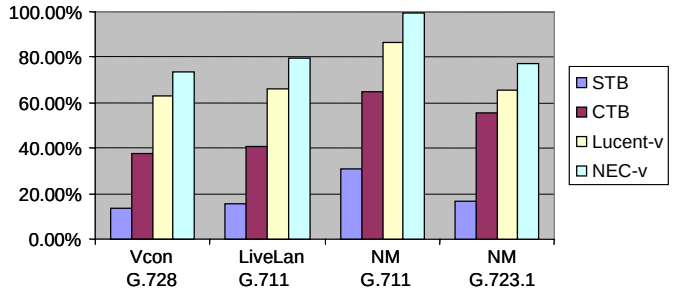


Abbildung 5-27: Erzielbare Linkauslastung

Audio-Applikation		Token Bucket r [kbit/s]	b (r) [kbyte]	p [kbit/s]	Anzahl N STB	CTB	Lu	NEC	Auslastung [%] STB	CTB	Lu	NEC
LiveLan	G.711	73	1.16	130	32	84	136	163	15.6	40.9	66.2	79.3
Vcon	G.728	24	0.44	45	85	236	392	461	13.6	37.8	62.7	73.8
Netmeeting	G.711	82	0.65	110	57	119	158	182	31.2	65.1	86.4	99.5
	G.723.1	24	0.45	42	103	349	411	482	16.5	55.8	65.6	77.1

Tabelle 5-7: Berechnete Auslastungswerte

Die Ergebnisse zeigen, dass bei den beiden verlustlosen Verfahren von NEC und Lucent aufgrund der Vernachlässigung des Schedulings eine wesentlich höhere Auslastung des Netzes erzielt werden kann. Dadurch kann bei diesen Verfahren die tatsächliche Verzögerung um den Wert $D^* = N \cdot (L/C^* + L/C)$ größer sein als 30ms. Verzögerungen werden bei den verlustlosen Verfahren nicht explizit berücksichtigt. Die maximale Verzögerung im Puffer hängt vom Auslastungszustand des Puffers ab und wird durch die Puffergröße B begrenzt. Verzögerungen, die beim Bedienen des Puffers auftreten, werden nicht berücksichtigt.
Um das Scheduling-Delay zu berücksichtigen, müsste man B bei der Konfiguration der Dienstklasse für eine maximale Anzahl von Verbindungen N_{max} so dimensionieren, dass in

diesem Fall die maximale Verzögerung in einem Netzknoten den Soll-Wert *D* nicht überschreitet. Vernachlässigt man den Term *L/C*, kann man näherungsweise *B(N)* ansetzen mit:

$B = C^* \cdot (D - N_{max} \cdot L/C^*) = C^* \cdot D - N_{max} \cdot L$.

Vergleicht man die erzielbare Auslastung von NEC und Lucent, so fällt auf, dass bei NEC die erzielbare Auslastung noch einmal um ca. 10% höher ist als bei Lucent. Der Unterschied liegt in der vereinfachten Modellierung durch eine verkürzte Dauer der ON-Phasen. Je größer die Differenz aus *p* und $r \cdot (p-r)$ ist, desto geringer ist der Unterschied.

Zusammenfassend kann man sagen, dass die höheren Auslastungen der verlustlosen Verfahren im Vergleich zur CTB-Formel auf Vereinfachungen bei der Modellierung zurückzuführen sind. Will man eine harte QoS-Garantie bezüglich der maximalen Verzögerung und der Verluste gewähren, bietet sich die CTB-Formel an. Genügt es, eine harte QoS-Garantie bezüglich der Verluste zu gewähren und die maximale Verzögerung grob abzuschätzen, empfiehlt sich die Lucent-Formel. Aufgrund der zu optimistischen Bestimmung der Burst-Dauern kann die NEC-Formel nicht als ein Zugangskontrollverfahren für echtzeitkritische Verkehre empfohlen werden.

5.5.2 Statistische Verfahren mit weicher QoS-Garantie

Die Verfahren mit einer im statistischen Sinne weichen QoS werden immer dann eingesetzt, wenn harte Garantien nicht unbedingt erforderlich und Netzressourcen knapp sind. Im Kapitel 5.5.1 wurde deutlich, welche hohen Bitraten für harte Garantien reserviert werden müssen. Die zu reservierenden Ressourcen übersteigen den mittleren Ressourcenbedarf einer Quelle um ein Vielfaches. Nicht bei allen Anwendungen muss eine vereinbarte Verbindungsqualität für die gesamte Verbindungsdauer unter allen Umständen sichergestellt sein. Wie viele Ressourcen durch das Lockern der Dienstgütegarantie eingespart werden können, wird in diesem Kapitel anhand zweier Verfahren näher untersucht. Beide Verfahren versuchen den statistischen Multiplexgewinn bei der Aggregation zahlreicher Verbindungen möglichst gut abzuschätzen. Beide basieren auf einer Verkehrscharakterisierung der TB-Parameter.

5.5.2.1 Verfahren nach NEC_{stat}

Das NEC-Verfahren [RR97] basiert auf einem Markov-Modell, bestehend aus den zwei Zuständen *High* und *Low*. Im Zustand High sendet die Quelle mit der Rate λ_H, im Zustand *Low* mit λ_L. Darüber hinaus ist das Modell definiert durch seine Zustandswahrscheinlichkeiten P_H und P_L. Nachfolgend wird gezeigt, wie die Modellparameter aus den TB-Parametern *r, b, p* und den Linkparametern *B* und *C* bestimmt werden können:

$$\lambda_H = \min\left(1, \frac{T_{ON}}{T_N}\right) \cdot p + \max\left(0, 1 - \frac{T_{ON}}{T_N}\right) \cdot r \qquad \lambda_L = \max\left(0, 1 - \frac{T_{ON}}{T_N}\right) \cdot r$$

$$P_H = \frac{T_{ON}}{T_{ON} + T_{OFF}} = \frac{r}{p} \qquad P_L = 1 - \frac{r}{p} = \frac{p - r}{p}$$

Gleichung 5-20: Markov Modell (NEC_{stat})

Wobei T_N und T_{ON} (siehe NEC_{vl}) gegeben sind mit:

$$T_N = \frac{B}{2 \cdot C}; \qquad T_{ON} = \frac{b}{p}$$

Mehrere solcher Quellen werden nun überlagert und der Ressourcenbedarf des Aggregats geschätzt. Dazu wird das Modell der *Markov-Modulated Fluid Sources* [AMS82] und eine Approximation der Verlustwahrscheinlichkeit nach Gauß verwendet. Das Ergebnis ist eine Wahrscheinlichkeitsdichtefunktion über der Rate $C_{Aggregat}$, welche gaußverteilt ist. Daraus läßt sich für eine vorgegebene maximale Verlustwahrscheinlichkeit ε die zu reservierende Gesamtrate C_{neu} berechnen: $P(C_{Aggregat} > C_{neu}) < \varepsilon$

$$M_{neu} = M_{alt} + r$$
$$\sigma_{neu}^2 = \sigma_{alt}^2 + \left(\lambda_H - \lambda_L\right)^2 \cdot P_{ON} \cdot \left(1 - P_{ON}\right)$$
$$C_{neu} = M_{neu} + \varsigma \cdot \sigma_{neu}$$

Gleichung 5-21: Kapazitätsabschätzung NEC_{stat}

Der Term $\varsigma \cdot \sigma$ *($\varsigma > 1$)* beschreibt dabei den über die mittlere Rate hinausgehenden Kapazitätsbedarf. Dieser kann aufgrund der Abschätzung der Gesamtrate mit Hilfe einer Gaußverteilung und einer erlaubten Verlustwahrscheinlichkeit angegeben werden mit:

$$\zeta = \sqrt{2 \cdot \left(-\log\left(-2 \cdot \eta \cdot \log(\eta)\right) - \log\left(1 - \frac{\log\left(-2 \cdot \log(\eta)\right)}{1 - \log(\eta)}\right)\right)} \approx 1.8 - 0.46 \cdot \log_{10}(\eta)$$

Gleichung 5-22: Näherung 1 für Spreizfaktor ς

Wobei η gegeben ist mit:

$$\eta = \frac{M_{new} \cdot \sqrt{2\pi}}{\sigma_{new}} \cdot \varepsilon$$

In [GAN91] wird eine weitere Näherung für ς angegeben mit:

$$\zeta = \sqrt{-2 \cdot \ln(\varepsilon) - \ln(2\pi)}$$

Gleichung 5-23: Näherung 2 für Spreizfaktor ς

Betrachtet man den Sonderfall einer homogenen Aggregation, d.h. einer Aggregation von Verkehren desselben Typs, ergibt sich aus den Gleichungen 5-21 und 5-23:

$$C_{aggr} = N \cdot r + \varsigma \cdot \sqrt{N} \cdot \sigma_0 \qquad c_{eff} = \frac{C_{aggr}}{N} = r + \frac{\varsigma \cdot \sigma_0}{\sqrt{N}}$$

Gleichung 5-24: Kapazitätsabschätzung NEC_{stat}, homogene Aggregation

Die nachfolgenden Abbildungen zeigen den Verlauf der Effektiven Bitraten in Abhängigkeit von *N*. In Abbildung 5-28 wurde der Verlauf für zwei maximale Verzögerungen *D=B/C* bei einer Verlustwahrscheinlichkeit von $\varepsilon = 10^{-3}$ gezeichnet. Die Abbildung 5-29 hingegen zeigt den Verlauf *c(N)* bei einer maximalen Verzögerung von *D = 10ms* für verschiedene Verlustwahrscheinlichkeiten ε.

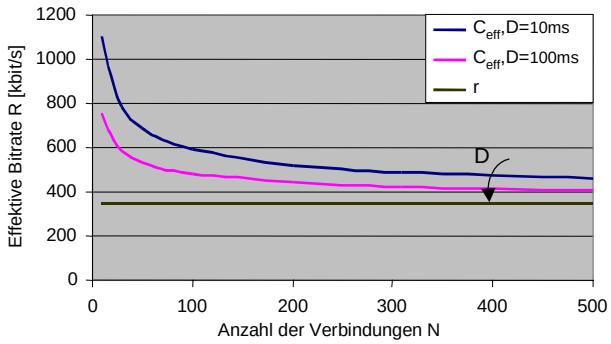


Abbildung 5-28: Effektive Bitrate $R_{NEC,stat}$ (N,D) $\varepsilon=10^{-3}$, Quelle: LiveLan 384 kbit/s

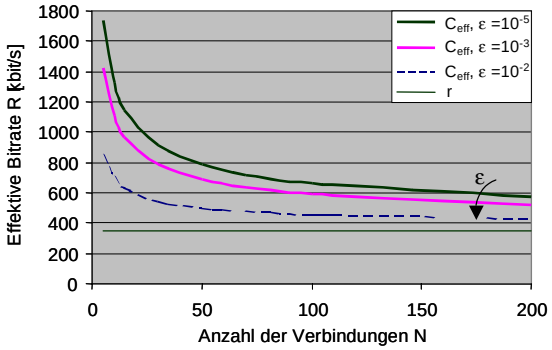


Abbildung 5-29: Effektive Bitrate $R_{NEC,stat}$ (N, ε) D=10ms, Quelle: LiveLan 384 kbit/s

Was den Einfluss der maximalen Verzögerung *D* auf die Effektive Bitrate anbelangt, ist das Verhältnis $2 \cdot b/p$ zu *D* entscheidend (siehe Gleichung 5-20). Solange $D < 2 \cdot b/p$ ist, ist die Effektive Bitrate unabhängig von *D*. Steigt *D* über diesen Wert an, nimmt die Effektive Bitrate ab. Dieser Zusammenhang ist in Abbildung 5-30 für drei verschiedene Quellen dargestellt. Die Quellenparameter sind Tabelle 5-5 bzw. Tabelle 5-6 zu entnehmen.

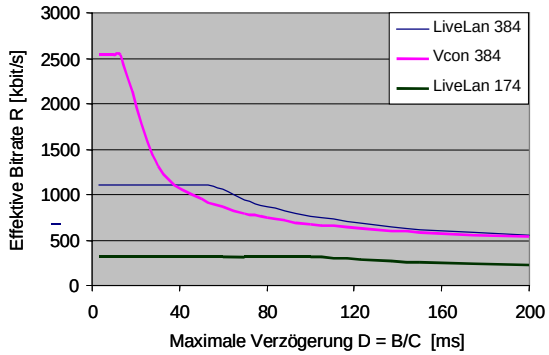


Abbildung 5-30: Effektive Bitrate $R_{NEC,stat}$ (D), N=10, $\varepsilon=10^{-3}$

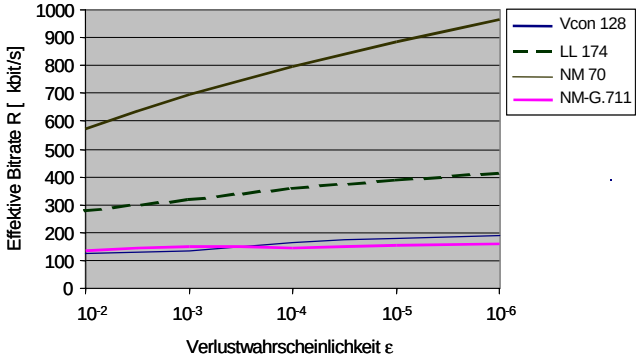


Abbildung 5-31: Effektive Bitrate $R_{NEC,stat}$ (e), N=10, D=10ms

Abbildung 5-31 zeigt den Verlauf der Effektiven Bitrate in Abhängigkeit von der Verlustwahrscheinlichkeit für verschiedene Quellen. Die Sprachquelle Netmeeting (NM) G.711 und die Videoquelle NM-70 besitzen eine ähnliche mittlere Senderate von ca. 80 kbit/s. Ebenso die Videoquellen Vcon-128 und LiveLan-174 mit Senderaten von ca. 120 kbit/s. Während die ersten beiden Quellen darüber hinaus eine ähnliche Spitzenbitrate von 100-200 kbit/s besitzen, liegt diese bei den beiden zuletzt genannten Quellen höher (> 400 kbit/s). Die mit Abstand höchste Spitzenbitrate von 6 Mbit/s besitzt jedoch die Vcon-128 Quelle. Demzufolge weist sie auch die größte Effektive Bitrate für alle Verlustwahrscheinlichkeiten auf.

5.5.2.2 Verfahren nach $Lucent_{stat}$

Dieses Verfahren wurde von Answar Elwalid und Debasis Mitra [EM93], [EMW95] bei den Bell Labs von AT&T als Zugangskontrolle für VBR-Quellen in ATM-Netzen entwickelt. Das Verkehrsmodell ist dasselbe periodische ON-/OFF-Modell, das schon im verlustlosen Fall verwendet wurde. Die Schätzfunktion der Verlustwahrscheinlichkeit von ATM-Zellen wird als Chernoff-Grenze [Bil86] angegeben. Dem Verfahren liegt die Bildung von Verkehrsklassen zugrunde. Dabei werden Verkehre mit identischer Verkehrscharakteristik, d.h. mit einem einheitlichen Satz von TB-Parametern, zu einer Verkehrsklasse zusammengefasst.

Das Verfahren betrachtet die von einer ON-/OFF-Quelle *i* zu einem bestimmten Zeitpunkt genutzte Bedienrate des Puffers u_i. Mehrere Quellen derselben Verkehrsklasse unterscheiden sich nur in ihrer Phasenlage. Überlagert man mehrere solcher Quellen, ergibt sich aus der Summe aller u_i eine Zufallsgröße *U*, die insgesamt genutzte Bedienrate. Das Verfahren beinhaltet eine Transformation der Zufallswerte mit einem Parameter *s*. Mittels der Chernoff-Grenze kann dann für die Verlustwahrscheinlichkeit eine obere Grenze in Form einer Funktion *F(s*)* angegeben werden:

$$P_{loss} = P\left(U \geq C\right) \leq e^{-F(s^*)}$$

Gleichung 5-25: Chernoff-Grenze

Die Funktion F ist eine Schätzfunktion des Ressourcenbedarfes und hängt von *C, B, s*, N_j,* sowie den TB-Parametern r_j, b_j, p_j ab. Die Parameter *B* und *C* geben dabei die Größe und Bedienrate des Puffers, *N* die Anzahl der Verbindungen pro Verkehrsklasse *j* an. Zur Bestimmung der Grenze muss *s** so gewählt werden, dass *F(s)* ihr Maximum erreicht. In diesem Schritt liegt die Komplexität des Verfahrens begründet, welche mit der Anzahl der Verkehrsklassen stark ansteigt.
Für den vereinfachten Fall einer Verkehrsklasse kann s^* angegeben werden mit:

$$s^* = \frac{1}{c_o} \cdot \ln\left[\frac{C \cdot (c_o - r)}{r \cdot (N \cdot c_o - C)}\right]; \qquad N \cdot c_o > C, \quad c_o > r$$

Der Parameter c_o steht dabei für die Effektive Bitrate im verlustlosen Fall (siehe Abschnitt 5.5.1.2). Für die Funktion *F(s*)* ergibt sich dann:

$$F(s^*) = N \cdot \left[\frac{C}{N \cdot c_o} \cdot \ln\left(\frac{C}{N \cdot r}\right) + \left(1 - \frac{C}{N \cdot c_o}\right) \cdot \ln\left(\frac{N \cdot c_o - C}{N \cdot \left((c_o - r)\right)}\right)\right]; \qquad N \cdot c_o > C, \quad c_o > r$$

Gleichung 5-26: Schätzfunktion des Ressourcenbedarfes für N-Verbindungen

Die Funktion *F(s*)* hängt also im homogenen Fall nur noch von der Anzahl der Verbindungen *N* und c_o ab. Aus der Näherung $P_{Loss} = \varepsilon \approx e^{-F(s^*)}$ (*Asymptotic Large Deviations Approximation*) ergibt sich ferner:

$$\ln\left(\frac{1}{\varepsilon}\right) = F\left(N, c_0\right)$$

Gleichung 5-27: Schätzfunktion der Verlustwahrscheinlichkeit

Zur Bestimmung der Effektiven Bitrate bei homogener Aggregation ist nun die Variable *N* in Gleichung 5-27 so groß wie möglich zu wählen, damit das Dienstgütekriterium in Gleichung 5-26 erfüllt ist und zudem gilt: $C/c_o < N_{max} < C/r$
Die Effektive Bitrate kann dann angegeben werden mit: $c_{eff} = C / N_{max}$

In [Hui90] wurde die Methode für mehrere Verkehrsklassen untersucht. Wie bereits zuvor erwähnt, ist die Komplexität dieses Verfahrens für ein heterogenes Verkehrsgemisch sehr groß. In [EMW95] ist daher eine Vereinfachung des Verfahrens für den heterogenen Fall angegeben. Eine optimistische obere Grenze für die Anzahl zulässiger Verbindungen wird dabei aus dem homogenen Fall abgeleitet mit:

$$\sum_{j=1}^{J} N_j \cdot c_j^h \leq C$$

Gleichung 5-28: Vereinfachtes Lucent$_{stat}$ Verfahren für mehrere Verkehrsklassen

Die Werte von c_j^h ergeben sich aus den Berechnungen, wie bei der homogenen Aggregation mit Verbindungen der Verkehrsklasse j:

$$c_j^h = C / N_{max,j}.$$

Der Kapazitätsbedarf eines Verkehrsgemisches ist nach dieser Methode wesentlich einfacher und schneller zu berechnen. Im folgenden wird diese Methode für alle Berechnungen mit heterogenen Verkehrsgemischen angewendet.

5.5.2.3 Kombination der Verlustlosen und Statistischen Verfahren

Die Statistischen Verfahren haben gezeigt, dass bei niedriger Aggregationsstufe, d.h. für kleine Werte von *N*, die Effektiven Bitraten größer werden können als im verlustlosen Fall. Sowohl das Lucent- als auch das NEC-Verfahren sehen daher eine Kombination beider Berechnungsmethoden vor.

Das Lucent-Verfahren sieht vor, die Effektive Bitrate sowohl nach der verlustlosen Methode als auch nach der statistischen Methode zu berechnen. Für die Zugangskontrolle wird dann das Minimum der beiden Ratenwerte verwendet.

Für das NEC-Verfahren wurde in [RR97] eine andere Art der Kombination des verlustlosen- und statistischen-Verfahrens vorgestellt, die im folgenden mit NEC-Kombi bezeichnet wird. Bei niedriger Verkehrsaggregation wird auch hier das verlustlose Verfahren und bei höherer Verkehrsaggregation das statistische Verfahren eingesetzt. Nur der Übergang erfolgt nicht sprunghaft, sondern asymptotisch. Bei NEC-Kombi wird für jede neue Verbindung zunächst immer die Gesamtbitrate nach der verlustlosen und der statistischen Methode berechnet. Danach wird jeweils die Änderung im Vergleich zum alten Zustand bestimmt und anschließend das Minimum gebildet. Die Gesamtrate ergibt sich dann aus der Summe der minimalen Änderungswerte.

$$\Delta\, Verlustlos = C_{Ges,verlustlos,neu} - C_{Ges,verlustlos,alt}$$
$$\Delta\, Statistisch = C_{Ges,\ statistisch,neu} - C_{Ges,statistisch,alt}$$

$$C_{Gesamt,Kombi} = \sum \min\left(\Delta Verlustlos - \Delta Statistisch\right)$$

Gleichung 5-29: NEC-Kombi

In nachfolgender Grafik ist der Verlauf der Effektiven Bitrate $R(N) = C_{Ges} / N$ für die verschiedenen NEC-Varianten exemplarisch dargestellt. Die Linkkapazität blieb dabei unverändert. Man sieht, dass das NEC-Kombi Verfahren für kleine *N* zunächst dieselben Werte liefert wie das Verlustlose Verfahren. Für größere N hingegen nähern sie sich den Ratenwerten des Statistischen Verfahrens asymptotisch an.

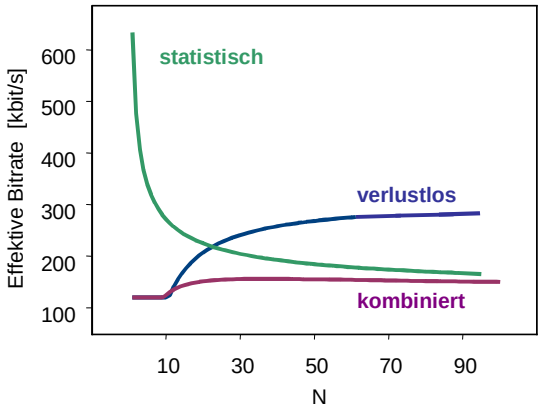


Abbildung 5-32: Effektive Bitraten R(N) der NEC-Varianten „statistisch“, „verlustlos“ und „kombiniert“

5.5.2.4 Verfahren nach Kelly

Aus der Literatur ist bekannt, dass messbasierte, reaktive Zugangskontrollverfahren eine wesentlich höhere Auslastung des Netzes ermöglichen als zustandsbasierte, pro-aktive Verfahren. Erstere basieren nicht auf einer Verkehrsbeschreibung, sondern auf aktuellen Messdaten. Daher enthalten die Kapazitätsberechnungen keine worst case Modellierung der Quellen. Die Berechnung bezieht sich direkt auf das gerade vorliegende Sendeverhalten der Quellen (den anfallenden Kapazitätsbedarf), welcher im statistischen Mittel deutlich niedriger liegt als im worst case. Bei der Verkehrscharakterisierung wurde festgestellt, dass die Messungen in der Regel nur einen Burst maximaler Größe enthielten. Im Gegensatz dazu gehen die ON-/OFF-Modelle aus den vorangegangenen Abschnitten von periodisch wiederkehrenden maximalen Bursts aus.

Nachfolgend wird ein messbasiertes Verfahren untersucht, um zu ermitteln, wieviel besser es vorhandene Netzressourcen im Vergleich zu parameterbasierten Verfahren auslasten kann.

Als Vertreter der messbasierten Zugangskontrollverfahren wurde eine Implementierung des Verfahrens von F. Kelly [Kel91], [KZZ96] verwendet [MSA]. Das Verfahren beruht wie das Lucent-Verfahren auf einer Methode der *Large Deviation Theory* – der *Many Sources Asymptotic* [CW96]. Das Verfahren ist darüber hinaus auch für Verkehre mit selbstähnlichem Verhalten [GK92], [Gib96] sowie für Paketverkehre geeignet. Eine Messreihe *j*, bestehend aus einer Sequenz von IP-Paketen, wird in Intervalle (*Epochs*) gleicher Länge *t* unterteilt. Die Bytes pro Intervall bilden eine Zufallsgröße $X_j[0,t]$, mit der das MSA-Tool die Effektive Bitrate α_j nach Kelly berechnen kann.

$$\alpha_j(s,t) = \frac{1}{s \cdot t} \cdot \log E\left[e^{s \cdot X_j[0,t]} \right]; \qquad 0 < s,t < \infty$$

Gleichung 5-30: Effektive Bitrate nach Kelly

Die Parameter *s* und *t* hängen von den Systemparametern *C* und *B* sowie vom Verkehrsgemisch auf dem Link ab und repräsentieren den momentanen Zustand des Netzes. Der Parameter *s* wird in der Einheit kbit · *s*, der Parameter *t* in ms angegeben. Während der Parameter s von dem Verhältnis der Mittleren Raten zu den Spitzenbitraten des Verkehrsgemisches abhängt und somit ein Maß für den statistischen Multiplexgewinn ist, steht der Parameter *t* für ein Zeitintervall, das die wahrscheinliche Dauer bis zum Pufferüberlauf angibt. Das Verhalten der Pufferüberlaufwahrscheinlichkeit wird dabei in Zusammenhang mit der Kapazität des Links und der Anzahl der überlagerten Quellen gesetzt. In den hier betrachteten Fällen betrug der Parameter *s* bei den Audioquellen zwischen 0.24 und 0.36, bei den Videoquellen zwi-

schen 0.06 und 0.11. Erhöht man die Bedienrate oder die Größe des Puffers, sinkt der Wert von *s*. Werden die Effektiven Bitraten anhand einer solchen asymptotischen Analyse abgeleitet, basiert das Verfahren auf der *Many Source Asymptotic*. Für eine Verlustwahrscheinlichkeit von $\varepsilon \leq e^{-\gamma}$ kann die maximale Anzahl zulässiger Verbindungen berechnet werden mit:

$$\sum_{j=1}^{J} N_j \cdot \alpha_j(s,t) \;\leq\; C + \frac{1}{t} \cdot \left(B - \frac{\gamma}{s} \right) = C^*$$

Gleichung 5-31: Zugangskontrolle nach Kelly

Für die geeignete Wahl der Intervall-Länge *t* und des Parameters γ für die untersuchten Quellenverkehre sei an dieser Stelle auf die Diplomarbeit von M. Czermin verwiesen [Cze00]. Die oben angegebenen Ergebnisse wurden mit den Parametern $\gamma = 15$ für alle Sprachverbindungen und $\gamma = 6$ für alle Videoverbindungen berechnet.

5.5.2.5 Simulationen

Um die untersuchten Zugangskontrollverfahren hinsichtlich der Einhaltung ihrer QoS-Garantie zu überprüfen, werden Simulationen durchgeführt (siehe Abbildung 5-3, Abschnitt 5.2).

Dazu wird ein Testnetz mit konfigurierten Dienstklassen simuliert und die maximalen Auslastungswerte für verschiedene Verkehrsgemische ermittelt. Dabei wird das Knotenmodell aus Abschnitt 3.2.3 nachgebildet und als Quellen die gemessenen Verkehre (Paketsequenzen) verwendet. Weil bei der Simulation keine Verkehrsmodelle, sondern die originalen Messreihen verwendet werden, stellen die in der Simulation ermittelten Auslastungswerte eine obere Grenze für die Zugangskontrollverfahren dar.

Um diese Auslastungsgrenze zu bestimmen, werden während der Simulation die Verzögerungen und Verluste der Pakete gemessen. Es werden zwei Szenarien unterschieden:

- A: Homogene Verkehrsaggregation
- B: Heterogene Verkehrsaggregation

Ziel der Simulationen im Fall A ist es, die maximale Anzahl *N* an Quellenverkehren zu ermitteln, die auf einem Link überlagert werden können, ohne die geforderte QoS-Garantie zu verletzen.

Ziel der Simulationen im Fall B ist es, den minimalen Kapazitätsbedarf $C_{sim,aggr}$ exemplarisch für ein paar ausgewählte Verkehrsgemische zu bestimmen.

Das Vorgehen bei der Simulation wird in Abbildung 5-33 vereinfachend für einen einzelnen Knoten und eine einzelne Dienstklasse dargestellt.

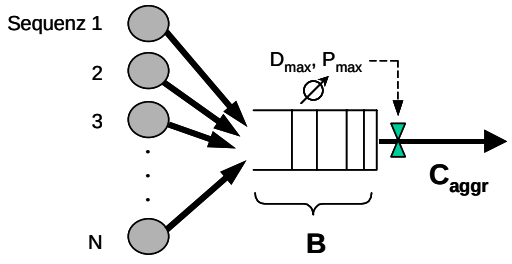


Abbildung 5-33: Ermittlung der maximalen Auslastung einer Dienstklasse (Knoten)

Netzknoten
Bei einer Aggregation von N Sequenzen an einem Knoten i wird von einem idealisierten Netzknoten ausgegangen, der aus N Eingängen und einem Ausgang besteht. Trifft ein Paket an irgendeinem Eingang ein, kann es ohne Verzögerung an den Ausgang vermittelt werden. Der Ausgang wird durch einen FIFO-Puffer modelliert, aus dem mit einer konstanten Bedienrate C_i Pakete ausgelesen werden. Pakete, die an mehreren Eingängen zeitgleich eintreffen, werden in zufälliger Reihenfolge in den Ausgangspuffer geschrieben. Die Paketverzögerungen werden vom Zeitpunkt des Einlesens eines Paketes in den Puffer bis zum Zeitpunkt des vollständigen Verlassens gemessen. Auf diese Weise fließen in die gemessenen Verzögerungswerte sowohl das Queuing-Delay als auch das worst case Scheduling-Delay eines realen Netzknotens mit einem WFQ-Scheduler ein.

Verkehrsquellen
Bei der Zusammensetzung eines Verkehrsgemisches werden im Folgenden zwei Arten der Aggregation unterschieden. Bei der Bildung eines homogenen Aggregats werden ausschließlich Sequenzen einer bestimmten Applikation und Einstellung überlagert. Bei der Bildung eines heterogenen Aggregats werden mehrere Messungen verschiedener Applikationen und Einstellungen zu einem einzigen Paketstrom zusammengefügt.

Für die Bildung eines Verkehrsgemisches benötigt man sehr viele voneinander unabhängige Quellenströme. Um möglichst viele unabhängige Quellen für die Simulation zu bekommen, werden alle Messreihen verwendet. Folglich werden nicht nur diejenigen Messungen, die ein worst case Verhalten der Quelle enthielten, sondern auch Messungen mit ganz normalem Teilnehmerverhalten und Beleuchtungsverhältnissen für die Simulation herangezogen. Die einzelnen Pakete der Messung werden gemäß ihrem Zeitstempel sequenziell von vorne beginnend aus der jeweiligen Datei ausgelesen und in die Simulation eingespeist. Ist die Anzahl von unabhängigen Quellen nicht ausreichend, werden längere Messungen in mehrere Sequenzen von ca. 5 Minuten Dauer aufgespalten. Dadurch konnten pro Applikation und Einstellung ca. 20 verschiedene 5 Minuten lange Sequenzen erzeugt werden. Darüber hinaus werden pro Sequenz mehrere Verkehrsquellen generiert. Dabei wird an unterschiedlichen Stellen der Sequenz mit dem Auslesen der Pakete begonnen. Die *Offsets* werden so gewählt, dass sie gleichmäßig über die Messdauer verteilt sind. Ist das Ende der Sequenz erreicht, wird wieder an den Anfang der Sequenz gesprungen.

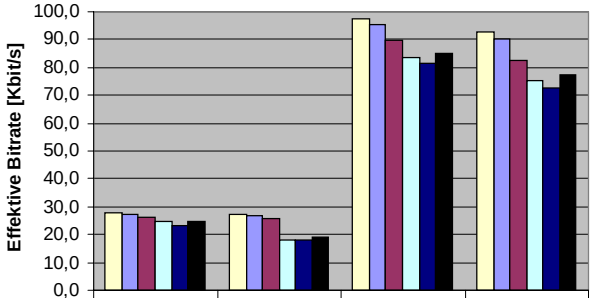
Als Simulationsumgebung wird der *Network Simulator* NS von Berkley [NS01] verwendet. Die Simulationsszenarien und Ergebnisse sind zusammen mit den berechneten Werten der Zugangskontrollverfahren im nachfolgenden Abschnitt dargestellt.

5.5.2.6 Vergleich der statistischen Verfahren

Es wird im Folgenden ein einzelner Link betrachtet, auf dem eine Dienstklasse für Echtzeitverkehre mit einer maximalen Verzögerung von $D_{max} = 10ms$ und einer Paketverlustwahrscheinlichkeit von 0.1% realisiert werden soll. Es wird von einer reservierten Bandbreite ausgegangen, die von einem statistischen Zugangskontrollverfahren verwaltet wird.

Zunächst werden Sprach- und Videoverbindungen getrennt voneinander untersucht (homogene Aggregation). Bei den Sprachverbindungen wird von einer reservierten Bandbreite von C = 15 Mbit/s, bei den Videoverbindungen aufgrund der höheren Senderaten von einer reservierten Bandbreite von C = 150 Mbit/s ausgegangen. Anhand der in den vorangegangen Abschnitten eingeführten Zugangskontrollverfahren wird für jeden Verkehrstyp ermittelt, wie viele Verbindungen N_{max} von diesem Typ maximal zugelassen werden können. Daraus lässt sich für jedes Verfahren eine Effektive Bitrate berechnen ($c = C/N_{max}$). Neben den Berech-

nungen werden parallel Simulationen durchgeführt. Die Größe des Puffers *B* bei der Simulation kann aus der Beziehung: $B = D \cdot C$ berechnet werden. Die Effektiven Bitraten der Sprachverkehre sind zusammen mit den Simulationsergebnissen in dem nachfolgenden Balkendiagramm dargestellt.



	Vcon G.728	NM G.723	NM G.711	LL G.711
Lucent-stat	28,0	27,6	97,2	92,6
NEC-stat	27,3	27,1	95,2	90,3
NEC-Kombi	26,1	26,0	89,5	82,4
Kelly	24,5	18,2	83,6	75,5
Simulation I	23,0	17,9	81,3	72,8
Simulation II	24,8	19,3	84,8	77,2

Abbildung 5-34: Homogene Aggregation (Sprache), Linkgeschwindigkeit 15 Mbit/s

Bei der Simulation mit Sprachsequenzen hat die Synchronität der Sprachquellen einen großen Einfluss auf das Ergebnis. Wie Kapitel 4 gezeigt hat, liefern konstantbitratige Sprachcodecs feste Paketabstände *T*. Treffen an einem Netzknoten von mehreren unabhängigen Quellen desselben Typs regelmäßig Pakete gleichzeitig ein, arbeiten die Quellen aus Sicht des Puffers synchron. Dies wiederum hat erhebliche Auswirkungen auf das Pufferverhalten. Die Folgen sind höhere Verluste und damit eine niedrigere mittlere Auslastung des Links. Arbeiten alle Quellen synchron, liefert die Simulation Ergebnisse, die nahe an den Ratenwerten der verlustlosen Verfahren liegen. Arbeiten die Quellen völlig asynchron, d.h. treffen die Pakete von verschiedenen Quellen *N* innerhalb eines Frameabstandes *T* so ein, dass die Paketabstände eine Gauss-Verteilung mit einem Mittelwert um *T/N* ergeben, liefert die Simulation Ratenwerte, die leicht über dem Mittelwert des Verkehrgemisches liegen.

Aus diesem Grund wurden zwei verschiedene Simulationen durchgeführt. Bei Simulation I werden die Pakete eines Traces mit einem Offset von *T/N* ausgelesen, bei Simulation II hingegen werden ca. 5% der Quellen synchron gestartet. Bei den restlichen 95% der Quellen wird wie bei Simulation I verfahren. Die Simulationsergebnisse der Variante II enthalten somit eine gewisse Sicherheitsreserve. Für die Zugangskontrollverfahren stellen die Simulationsergebnisse aus Variante I das Minimum der Effektiven Bitraten dar, das keines der Verfahren unterschreiten sollte.

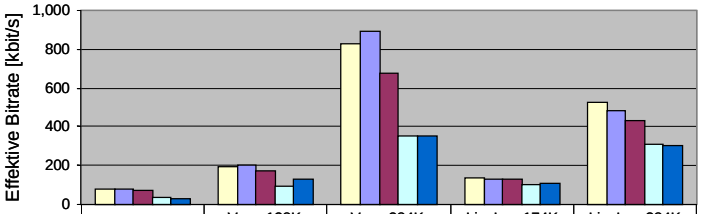
Das Ergebnis zeigt, dass Simulation I aufgrund fehlender Modellierung des Verkehrs und des Pufferverhaltens die niedrigsten Werte liefert. Die Abweichungen der Ergebnisse aus Simulation I, II oder Kelly von denen der prädiktiven Verfahren hängen stark davon ab, wie sehr die bei der Simulation verwendeten Quellen (Messsequenzen) den bei der Verkehrscharakterisierung gesteckten Rahmen tatsächlich ausnutzen. In den meisten Fällen verhält sich eine Verkehrsquelle gutmütiger als es die worst case Charakterisierung erlauben würde. Der Unterschied wird besonders bei Netmeeting G.723.1 mit Sprachpausenunterdrückung deutlich. Zudem ist anzumerken, dass das in Kapitel-4 festgestellte langzeitabhängige Verhalten dieser Quelle zu keiner Unterschätzung der Paketverluste führt. Das liegt daran, dass die maximale

Dauer eines Bursts begrenzt und die Charakterisierung nach dem worst case Verhalten der Quelle ausgerichtet wurde.

Die Ergebnisse des messbasierten Verfahrens von Kelly liegen geringfügig über den Simulationsergebnissen I. Das Kelly Verfahren wird direkt auf die Messwerte angewendet. Die Messdaten entsprechen dabei demselben Verkehrsgemisch wie bei der Simulation I. Die Abweichungen liegen somit allein in der Modellierung der Warteschlangenlänge und der daraus abgeleiteten Abschätzung der Verlustwahrscheinlichkeit begründet.

Die Verfahren NEC_{stat} und $Lucent_{stat}$ basieren auf einer worst case Beschreibung der Quellen und einem daraus abgeleiteten worst case Quellenmodell. Aufgrund des pessimistischen Modells und der damit verbundenen statistischen Reserven liefern diese Verfahren deutlich höhere Ratenwerte als beispielsweise Simulation I, II oder das Verfahren nach Kelly. Interessant dabei ist, dass das einfachere und damit schnellere NEC-Verfahren in den meisten Fällen niedrigere Effektive Bitraten liefert als das Lucent-Verfahren. Diese Verfahren ermöglichen bei der hier untersuchten Aggregationsstufe von 15 Mbit/s eine maximale Linkauslastung zwischen ca. 80% bei LiveLan G.711 und ca. 89% bei Netmeeting G.723.1. Bei der Bestimmung der Auslastung wird nicht die mittlere Rate als Basis verwendet, sondern die Token-Füllrate.

Das Verfahren NEC_{Kombi} liefert von allen prädiktiven Verfahren die höchsten Auslastungswerte. Die maximale Linkauslastung liegt zwar aufgrund der Verkehrsmodellierung immer noch ca. 5% unter den Werten aus Simulation II, aber 5-9% höher als die Werte von $Lucent_{stat}$ oder NEC_{stat}. Zudem kann man aus Abbildung 5-34 erkennen, dass sich das NEC_{Kombi}-Verfahren für große N, z.B. bei Vcon Audio G.728, an die Werte der anderen statistischen Verfahren annähert.



	Vcon 64K Video	Vcon 128K Video	Vcon 384K Video	LiveLan 174K Video	LiveLan 384K Video
Lucent stat	78.481	196.563	830.551	138.153	521.639
NEC stat	80.117	201.456	890.450	132.195	480.690
NEC Kombi	73.322	173.933	674.365	128.235	433.113
Kelly	39.100	91.100	355.500	103.000	307.500
Simulation stat	26.044	131.342	349.705	107.327	303.053

Abbildung 5-35: Homogene Aggregation (Video), Linkgeschwindigkeit 150 Mbit/s

Ähnliche Schlussfolgerungen können auch bei der Untersuchung mit Videoverkehren gezogen werden. Die Effektiven Bitraten der Videoverkehre sind zusammen mit den Simulationsergebnissen in Abbildung 5-35 dargestellt. In Tabelle 5-7 sind die Anzahl *N* der mit den jeweiligen Verfahren zugelassenen Verbindungen sowie die erzielte prozentuale Auslastung ρ des 150 Mbit/s Links aufgelistet.

$N=C/c_{eff}$ $\rho = N \cdot r_{TB}/C$	LL-384		LL-174		Vcon-384		Vcon-128	
	N	ρ	N	ρ	N	ρ	N	ρ
NEC-stat	312	72.8%	1134	90.7%	168	44.8%	744	59.5%
Lucent-stat	287	67.0%	1085	86.8%	180	48.0%	763	61.0%
NEC-Kombi	346	80.7%	1169	93.5%	222	59.2%	862	69.0%
Simulation ($\rho = N \cdot R_{mean}/C$)	494	97.0%	1397	97.4%	428	96.0%	1578	96.3%

Tabelle 5-8: Erzielbare Linkauslastung

Vergleicht man die Ergebnisse aus Abbildung 5-35 mit denen der Sprachverkehre, so fällt auf, dass die Unterschiede zwischen den verschiedenen Berechnungsmethoden und der Simulation bei den Videoverkehren viel größer sind. Ferner liefert das $Lucent_{stat}$ Verfahren bei Video ähnliche Ergebnisse wie das NEC_{stat} Verfahren und bei Vcon Video 384 kbit/s sogar einen niedrigeren Ratenwert als das NEC_{stat} Verfahren. Nach wie vor liegen die Werte des NEC_{Kombi} Verfahrens bei allen Verkehrstypen unterhalb denen der rein statistischen Verfahren. Bei hochbitratigen Verbindungen ist der Unterschied größer als bei niederbitratigen.

Die Unterschiede zwischen den verschiedenen Zugangskontrollverfahren hängen von der Aggregationsstufe, d.h. der Anzahl *N* der überlagerten Verbindungen und den Verkehrseigenschaften der Verkehre ab. Die Verfahren versuchen dabei den statistischen Multiplexgewinn auszunutzen. Dieser ist umso größer, je mehr Verbindungen überlagert werden und je variabler die Senderaten der Quellen sind. Wendet man diese Verfahren auf Verbindungen mit großem Unterschied zwischen mittlerer und Spitzenbitrate sowie großen Burst-Längen an, kann man die Effizienz der jeweiligen Verfahren am deutlichsten erkennen.

Aus Tabelle 5-8 wird ersichtlich, dass für große *N* der Unterschied zwischen den prädiktiven Verfahren und der Simulation kleiner wird. Betrachtet man hingegen die Sprachverbindungen in Abbildung 5-34, so ist diese Konvergenz nicht erkennbar. Konvergenz tritt nur mit den Verkehren auf, mit denen ein nennenswerter statistischer Multiplexgewinn erzielbar ist, d.h. mit Verkehren mit hohen Spitzenbitraten und großen Burst-Llängen.

Der Verlauf der Effektiven Bitraten der prädiktiven Verfahren ist in Abbildung 5-36 exemplarisch für ein bursthaftes Verkehrsgemisch dargestellt.

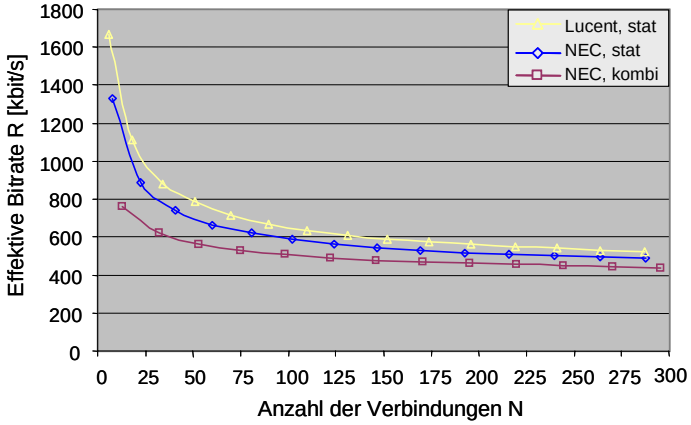


Abbildung 5-36: LiveLan 384 kbit/s Video R(N)

Die Linkkapazität wurde schrittweise von 10 Mbit/s auf 150 Mbit/s erhöht und dabei jeweils die maximale Anzahl von zulässigen Verbindungen *N* bestimmt. Man kann den Verläufen *R(N)* entnehmen, dass die statistischen Verfahren den Multiplexgewinn für große *N* besser abschätzen können. Ferner nehmen die relativen Unterschiede zwischen den einzelnen Ver-

fahren für große *N* ab. Steigt die Linkkapazität weiter auf 300 Mbit/s an, liefert das $Lucent_{stat}$ Verfahren niedrigere Effektive Bitraten als das NEC_{stat} Verfahren.

Zum Abschluss werden noch zwei heterogene Szenarien betrachtet, in denen Sprach- und Videoverbindungen überlagert werden. In Tabelle 5-9 wird die Zusammensetzung der Verkehrsgemische in den Szenarien I und II gezeigt.

Quellentyp		Vcon			LiveLAN		
		Video 128K	Video 384K	Audio G.728	Video 174K	Video 384K	Audio G.711
Anzahl	I	20	10	30	20	10	30
	II	100	80	700	100	80	700

Tabelle 5-9: Heterogene Verkehrsgemische

Für jedes dieser Verkehrsgemische wurde der Ressourcenbedarf sowohl berechnet als auch über Simulationen bestimmt. Die Ergebnisse sind in der Abbildung 5-37 und Abbildung 5-38 einander gegenübergestellt.

Die Ergebnisse zeigen, dass die Zugangskontrollverfahren auch bei einem heterogenen Verkehrsgemisch die Paketverluste nicht unterschätzen. Wie im homogenen Fall ermittelt das NEC_{Kombi} −Verfahren den geringsten Ressourcenbedarf. Zudem ist seine Komplexität im Gegensatz zum Lucent-Verfahren problemlos beherrschbar. Für das Lucent-Verfahren hingegen werden spezielle Optimierungsprogramme benötigt. Bei den Berechnungen hier wurde auf die Vereinfachung aus Gleichung 5-28 zurückgegriffen.

Die genaue Kenntnis vom Verhalten der Quellen und der Zugangskontrollverfahren erlaubt die Bildung von heterogenen Verkehrsgemischen ohne eine Aufweichung der QoS-Garantie. Dadurch können die Dienstklassen auf Netzebene allein auf der Grundlage der Dienstgütespezifikationen unabhängig vom Verkehrsgemisch gebildet werden. Die Anzahl der erforderlichen Dienstklassen lässt sich somit auf ein Minimum reduzieren.

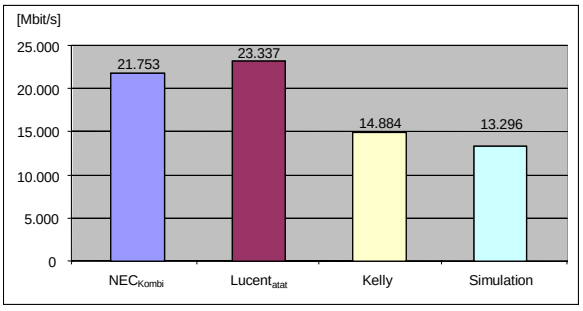


Abbildung 5-37: Ressourcenbedarf für Szenario I

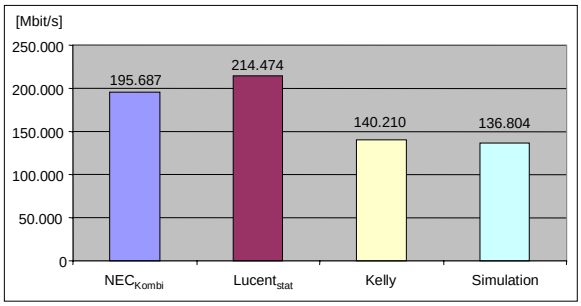


Abbildung 5-38: Ressourcenbedarf für Szenario II

5.6 Zugangskontrollverfahren für das RM-System

Das Ressourcenmanagement wurde eingeführt, um Übertragungsdienste für Echtzeitverkehre in IP-Netzen zu realisieren. Erreicht wird dies durch entsprechende Dienstklassen auf IP-Ebene und ein Zugangskontrollverfahren auf Ressourcenmanagement-Ebene. In den vorangegangenen Abschnitten wurden mehrere Verfahren untersucht, die nun hinsichtlich ihrer Eignung für das Ressourcenmanagement-System bewertet werden.

Für den Einsatz eines Verfahrens im RM-System ist Folgendes zu beachten:

- Die Komplexität der Verfahren sollte so gering wie möglich sein, da sie einen maßgeblichen Einfluss auf die Skalierbarkeit der RM-Architektur besitzt.
- Das Verfahren für harte QoS-Garantie darf die notwendigen Ressourcen keinesfalls unterschätzen, sodass das zugrunde liegende QoS-Kriterium unter keinen Umständen verletzt wird.
- Das Verfahren für weiche QoS-Garantie sollte die vorhandenen Netzressourcen möglichst gut auszunutzen und darüber hinaus keinerlei Einschränkungen in Bezug auf das Verkehrsgemisch besitzen.

Die untersuchten Verfahren werden nun hinsichtlich ihrer Komplexität, ihrer Zuverlässigkeit in der Einhaltung der Dienstgüte und ihrer erzielbaren Auslastung bewertet. In Tabelle 5-8 sind die Resultate zusammengefasst. Als Bewertungsschema werden drei qualitative Noten vergeben: gut geeignet (+), geeignet (o), weniger geeignet (-).

Zugangskontrollverfahren		Algorithmische Komplexität	Einhaltung der Dienstgüte	Erzielbare Auslastung	Vorschlag für RM
Harte QoS	IntServ CTB	+	+	o	x
	IntServ STB	+	+	-	
	$Lucent_{vl}$	+	-	+	(x)
	NEC_{vl}	o	-	+	
Weiche QoS	$Lucent_{stat}$	-	+	- (o)*	
	NEC_{stat}	o	+	o (o)*	
	NEC_{Kombi}	o	+	+ (+)*	x

* hohe Aggregationsstufe

Tabelle 5-10: Bewertung der Zugangskontrollverfahren

Bei den Verfahren mit **harter QoS-Garantie** kommt nur das IntServ CTB-Verfahren in Frage. Neben dem IntServ STB-Verfahren ist es das einzige, das wirklich eine harte QoS-Garantie ermöglicht. Für eine vergleichbare Komplexität liefert es höhere Auslastungswerte als das STB-Verfahren. Das Lucent-Verfahren hingegen liefert bei ähnlicher Komplexität zwar eine nochmals um bis zu 25% höhere maximale Auslastung der Ressourcen, berücksichtigt jedoch nur das Queuing-Delay und unterschätzt dadurch den eigentlichen Ressourcenbedarf im worst case. Das Lucent-Verfahren kann eine Alternative zum IntServ CTB-Verfahren in den Fällen sein, in denen ein Netzbetreiber die hohen Kosten für eine harte QoS-Garantie in Form des großen Kapazitätsbedarfes nicht aufbringen will. Werden auf allen Teilstrecken nicht nur echtzeitkritische Verkehre übertragen und ist der Anteil echtzeitkritischer Verkehre im Vergleich zu nicht echtzeitkritischen Verkehren kleiner 75%, ist bei Priorisierung des echtzeitkritische Verkehrs eine Aufweichung der QoS-Garantie vertretbar. Ähnliches gilt auch für das NEC-Verfahren. Es besitzt jedoch eine höhere Komplexität, da es im Gegensatz zu allen anderen Verfahren keine Effektive Bitrate pro Verbindung sondern eine Gesamtrate für das Aggregat berechnet. Dazu werden die Verkehrsparameter aller aktiven Verbindungen auf dem Link benötigt.

Die Verfahren mit **weicher QoS-Garantie** besitzen eine größere Komplexität. Für das Ressourcenmanagement ist jedoch gerade die Komplexität des Algorithmuss ein entscheidender Faktor. Sie hat einen starken Einfluss auf die Reaktionszeit des RM-Systems , die Dauer des Verbindungsaufbaus und damit auf die Skalierbarkeit der RM-Architektur.

Das Lucent-Verfahren scheidet wegen seiner Komplexität insbesondere bei einem heterogenen Verkehrsgemisch aus. Es beinhaltet die Suche nach einem Maximum einer mehrdimensionalen Funktion. Für das RM-System wird dieses Verfahren nur anwendbar, wenn man die Effektiven Bitraten für den homogenen Aggreagationsfall vorab berechnet, in einer Tabelle im Ressourcen-Manager ablegt und diese auch im heterogenen Fall anwendet.
Das NEC_{stat}-Verfahren kommt ohne eine solche Optimierung aus, ist deutlich einfacher und darüber hinaus auch für ein heterogenes Verkehrsgemisch anwendbar. Für die meisten der hier betrachteten Quellen lieferte es bessere Auslastungswerte als das Lucent-Verfahren. Bei sehr bursthaften Quellen mit hoher Spitzenbitrate ist das Lucent-Verfahren etwas effektiver in der Abschätzung des statistischen Multiplexgewinns. Keines der beiden Verfahren kam jedoch an die Auslastungswerte von NEC_{Kombi} heran.

Festzuhalten ist, dass alle statistischen Verfahren ihre Dienstgüte eingehalten und im Vergleich zu den Simulationsergebnissen noch ausreichend Kapazitätsreserven gezeigt haben.

Aufgrund der wesentlich geringeren Komplexität der NEC-Verfahren gegenüber den Lucent-Verfahren und der hohen Effizienz des NEC_{Komb}-Verfahrens wird das NEC_{Komb}-Verfahren für das RM-System vorgeschlagen und implementiert.

6. Die Ressourcenmanagement Architektur

Nachdem in Kapitel 3 die Merkmale und Grundzüge der Ressourcenmanagement-Architektur im Überblick vorgestellt wurden, ist die Zielsetzung dieses Kapitels, eine detaillierte Beschreibung der Komponenten, Mechanismen und Protokolle zu geben. Zudem wird gezeigt, wie die Ergebnisse aus den Kapiteln 4 und 5 in eine Realisierung der RM-Architektur integriert werden können.

6.1 Anforderungen

Das Ziel der RM-Architektur ist es, die zur Verfügung stehenden Ressourcen eines IP-Netzes so zu verwalten, dass echtzeitkritische Dienste wie Telefonie, Videotelefonie oder Videokonferenzen abgewickelt werden können.

Die Anforderungen an die Systemarchitektur lassen sich direkt aus der in Abschnitt 3.1 beschriebenen Zielsetzung ableiten. Das RM-System soll folgende Anforderungen erfüllen:

- **Dienstgütegarantie / Granularität**
  - Dienstklasse für harte QoS-Garantie
  - Dienstklasse für weiche QoS-Garantie
- **Leichte Einführbarkeit in eine bestehende Systemumgebung**
  - Vermeidung einer direkten Teilnehmerschnittstelle
  - Unabhängigkeit von der Netztechnologie
  - Autokonfigurationsmechanismen
- **Skalierbarkeit für große Netze**
  - Trennung von Reservierung und Paketverarbeitung
  - Domänenkonzept
- **Minimierung der Rufaufbauzeiten**
  - Optimiertes Domänendesign

Der Aspekt der Minimierung der Rufaufbauzeiten wurde bislang nicht betrachtet und wird daher kurz erläutert. Rufaufbauzeiten setzen sich aus der jeweiligen Signalisierungsdauer von Dienststeuerung und Ressourcenreservierung zusammen. In der Regel wird die Dienstsignalisierung solange unterbrochen, bis die Ressourcenreservierung abgeschlossen ist. Eine Minimierung der Rufaufbauzeiten ist somit im betrachteten Kontext mit einer Minimierung der Reaktionszeit des RM-Systems gleichzusetzen. Die Reaktionszeit des RM-Systems kann durch leistungsfähige Server sowie durch die Minimierung der Anzahl der am Rufaufbau beteiligten Instanzen verringert werden. Kennt man die Aktivität der Teilnehmer, kann dies durch eine geschickte Wahl der Domänengrenzen erreicht werden.

Im Folgenden wird zunächst die Dienstgüte der RM-Architektur geeignet modelliert. Im Anschluss daran werden die einzelnen Komponenten der Architektur definiert, die Protokolle spezifiziert und der innere Aufbau der Komponenten gezeigt. Abschließend werden verschiedene Anwendungsszenarien der RM-Architektur vorgestellt und ein Einblick in die prototypische Implementierung gegeben.

6.2 Dienstgütespezifikation

In Abschnitt 1.3 wurde festgestellt, dass für die Garantie einer gewissen Dienstqualität die signalverarbeitenden Prozesse im Terminal und die Datenübertragung im IP-Netz koordiniert werden müssen. Dabei wurde festgelegt, dass das Ressourcenmanagement den Terminals eine bestimmte Übertragungsqualität garantiert. In diesem Abschnitt wird die Dienstgüte zunächst abstrakt aus der Sicht des Teilnehmers modelliert und dann schrittweise in applikations- und netzspezifischen Parametern konkretisiert. Dabei wird die Rolle der Dienststeuerung bzw. des Ressourcenmanagements definiert.

Die Dienstgüte in Kommunikationssystemen wird meist in verschiedenen Abstraktionsebenen beschrieben. Ein Teilnehmer hat keinerlei Kenntnisse über den Ressourcenbedarf, sondern formuliert seine Anforderungen an die Dienstqualität gegenüber der Dienststeuerung in abstrakter Form. Die Dienststeuerung setzt diese dann in konkrete applikations- und systemspezifische Parameter für das Teilnehmergerät um und leitet daraus die netzspezifischen Performance-Parameter für das RM-System ab. Dieser Abbildungsvorgang soll nun genauer betrachtet werden.

Der Teilnehmer verwendet ein abstraktes Dienstgütemodell, welches die Qualität eines Dienstes nach subjektiven Kriterien beschreibt. Er kann die Dienstqualität des Kommunikationssystems nur subjektiv, d.h. mit seinem Auge oder Ohr, wahrnehmen und beurteilen. Das im Zusammenhang des RM-Systems verwendete Modell eines Teilnehmers sieht vor, dass die Qualität eines Echtzeitdienstes von dem initiierenden Teilnehmer beim Aufruf vorgegeben wird. Dazu wurden für die Spezifikation der subjektiven Qualität auf der Teilnehmerseite von ETSI TIPHON [Tip00] Qualitätsklassen eingeführt. Der Qualitätsbegriff von TIPHON schließt neben der Nutzsignalverarbeitung auch die Signalisierungsdauer eines Verbindungsaufbaus mit ein. Beispiele für solche Qualitätsklassen sind „beste Qualität“, „hohe Qualität“, „akzeptable Qualität“ oder „geringe Qualität“. Die subjektiven Qualitätsbezeichnungen des Teilnehmers beziehen sich immer auf einen bestimmten Diensttyp (z.B. IP-Telephonie). Bei der Spezifikation eines Dienstes werden verschiedene Dienstklassen unterschieden. Neben dem interaktiven Echtzeitdienst (*Realtime*) gibt es einen nicht-interaktiven Echtzeitdienst (*Streaming*), einen interaktiven Datendienst (*Interaktive*) und einen normalen Datendienst (*Best Effort*). Die Wahl der Qualitätsklasse stellt zusammen mit der Dienstklasse die Anforderungen an das Kommunikationssystem dar.

Der **Teilnehmer** bestimmt mit seinem Aufruf eines Dienstes eine Dienstklasse, welche vom Dienstentwickler definiert wurde. Dazu wählt er beim Aufruf eine Qualitätsklasse, mit welcher der Dienst abgewickelt werden soll. Die Spezifikation einer Dienstklasse enthält für jede Qualitätsklasse bereits konkrete Angaben hinsichtlich der Dienstgütegarantie wie der maximal zulässigen Ende-zu-Ende Verzögerung oder der Verluste bei der Signalverarbeitung und Übertragung.
Das **Kommunikationssystem,** bestehend aus mehreren Endgeräten und einem Transportnetz, muss nun dafür sorgen, dass die vom Teilnehmer geforderte Dienstqualität auch tatsächlich erreicht wird. Dazu müssen die dienst- und qualitätsspezifischen Ende-zu-Ende Anforderungen auf das Kommunikationssystem abgebildet werden. Die Ende-zu-Ende

Anforderungen werden auf technisch fassbare und überprüfbare Parameter der beteiligten Systemkomponenten heruntergebrochen. Diese Abbildung nimmt die **Dienststeuerung** des Endgerätes vor, indem sie zunächst das zu verwendende Kodierverfahren sowie die Priorität festlegt, mit welcher der Anwendungsprozess im Vergleich zu anderen Systemprozessen behandelt wird. Danach kann die Dienststeuerung die Verzögerungen bei der Signalverarbeitung innerhalb des Endgerätes abschätzen und zusammen mit der Ende-zu-Ende Dienst- und Qualitätsklassendefinition die Anforderungen an das Transportnetz stellen. Die netzspezifischen QoS-Parameter umfassen Angaben über die maximal zulässigen Ende-zu-Ende Verzögerungen und Verluste der Nutzdatenpakete bei der Übertragung zwischen den Endgeräten.

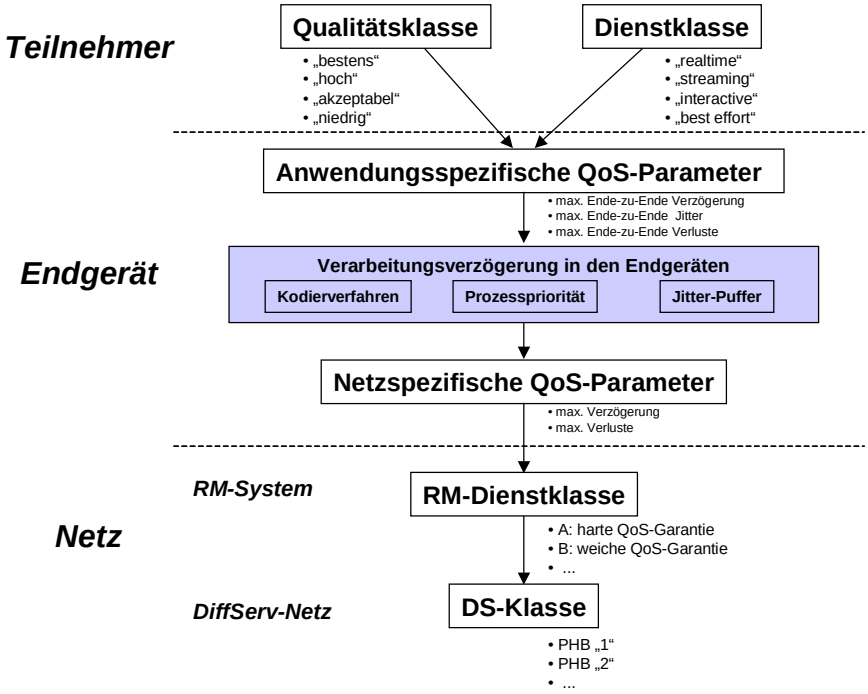


Abbildung 6-1: Ablauf der Dienstgütekonkretisierung

Auf der Netzseite werden die zwischen den Endgeräten Ende-zu-Ende gültigen netzspezifischen QoS-Parameter auf die zur Verfügung stehenden Ressourcen des Netzes abgebildet. Dies erfolgt für echtzeitkritische Verkehre auf zwei Ebenen, der Signalisierungsebene (RM) und der Netzebene (Netzknoten).

Zunächst wird im **RM-System** während der Dienstsignalisierung eine Zugangskontrolle durchgeführt. Dabei werden die netzspezifischen QoS-Parameter auf eine RM-Dienstklasse abgebildet. Jeder RM-Dienstklasse auf RM-Ebene ist eine DS-Klasse auf Netzebene zugeordnet. Wird eine Verbindung zugelassen, können anschließend von den Endgeräten Nutzdaten gesendet werden. In den Endgeräten oder im Zugangsnetz werden die IP-Pakete der Verbindung anhand von IP-Adressen und Portnummern einer DS-Klasse zugewiesen.

Die Schritte bei der Umsetzung der abstrakten Dienstgütebeschreibung eines Teilnehmers bis hin zur Auswahl einer geeigneten Dienstklasse auf Netzebene wird in Abbildung 6-1 dargestellt.

Abschließend soll als **Beispiel** für eine Dienstklassenspezifikation soll der IP-Telephoniedienst TIPHON dienen. Dieser zeigt eine Zuordnung von subjektiven Qualitätsurteilen, Kodierverfahren und netzspezifischen QoS-Parameterwerten. Die subjektiv

empfundene Dienstqualität wird mit Hilfe des E-Modells bestimmt. Das E-Modell der ITU [G.107] dient bei einer Sprachverbindung zur Bestimmung der Auswirkungen von mehreren Übertragungsparametern auf die Konversation. Es liefert einen qualitativen Bewertungsfaktor R (50 < R < 100), um die Auswirkungen verschiedener Übertragungsbedingungen auf die „Mund-zu-Ohr"-Übertragungsqualität miteinander vergleichen zu können. Die einzelnen Qualitätsklassen werden in nachfolgender Tabelle beschrieben.

Klasse	Dienstbeschreibung, Zufriedenheitsgrad	R-Wert	Kodier-verfahren	Verbindungs-aufbau	Ver-zögerung	Verluste	Jitter
BEST	besser als PSTN, alle Teilnehmer sehr zufrieden	90 ≤ R ≤ 100	besser als G.711	< 1,5s–4s	< 150 ms	0	0
HIGH	ähnlich zum PSTN, alle Teilnehmer zufrieden	80 ≤ R ≤ 90	besser als G.726	< 4s–13s	< 250 ms	3%	75ms
MEDIUM	ähnlich zum GSM, einige Teilnehmer unzufrieden	70 ≤ R ≤ 80	besser als GSM - FR	< 7s–25s	< 350 ms	15%	125ms
LOW	ähnlich zum klassischen Internet, viele Teilnehmer unzufrieden	60 ≤ R ≤ 70	-	< 7s–25s	< 450 ms	25%	225ms

Tabelle 6-1: Qualitätsklassen bei TIPHON

Die erlaubte Verzögerung beim Verbindungsaufbau kann schwanken, je nachdem, ob ein Adressauflösungsprozess notwendig ist und, ob die Verbindung über ein *Gateway* geführt wird oder nicht. Neben der Spezifikation der Dienstgüte werden zusätzlich drei Klassen von Terminals mit unterschiedlichen Anschlussraten definiert:

Klasse	Beschreibung	Erreichbare Qualitätsklasse	Verzögerung im Terminal
A	große Bandbreite, z.B. LAN	BEST-HIGH	10-20ms
B	mittlere Bandbreite, z.B. ISDN (< 64 kbit/s)	MEDIUM-HIGH	40-60ms
C	geringe Bandbreite, z.B. Modem (<25 kbit/s)	LOW-MEDIUM	60-100ms

Tabelle 6-2: Terminalklassen bei TIPHON

Aus den Ende-zu-Ende-Anforderungen der Dienstklassen und den Anforderungen der Terminals lassen sich die netzspezifischen QoS-Parameter ableiten.

6.3 Komponenten

Nachfolgend werden die Komponenten der RM-Architektur definiert. Es handelt sich dabei um Endgeräte, Dienststeuereinheiten, Ressourcen-Manager und Topologie-Manager. Endgeräte und Dienststeuereinheiten werden im Rahmen von Dienstarchitekturen standardisiert und an die RM-Architektur angebunden. Als konkrete Beispiele für solche Dienstarchitekturen werden im Folgenden der ITU-Standard H.323 [H.323] und der IETF-Standard SIP [HSS99] verwendet.

6.3.1 Terminal-Client

Der Terminal-Client stellt einen Anwendungsprozess auf einem Terminal dar. Er repräsentiert die Dienststeuerung des Teilnehmers und die Nutzdatenquelle/-Senke einer Echtzeitanwendung.

Die Dienststeuerung koordiniert die Ausführung der Dienste, indem sie die erforderlichen Kommunikationsbeziehungen herstellt, den Ablauf der Sitzung überwacht und auf Ereignisse reagiert. Ereignisse können z.B. Teilnehmerinteraktionen oder das Ablaufen von Timern sein.

Für die Steuerung von Echtzeitanwendungen in IP-Netzen wurden in den vergangenen Jahren zwei endgerätebasierte Dienstarchitekturen entworfen: Der H.323-Standard der ITU und der SIP-Standard der IETF. Der H.323-Standard wurde speziell zur Steuerung von Telefonie-, Videotelefonie- und Multimediakonferenzen entwickelt, während der SIP-Standard allgemeiner gefasst ist. SIP dient der Steuerung von Multimediadiensten im Internet, unabhängig von einem bestimmten Medium. Vergleiche der beiden Dienstarchitekturen sind in [HJ98], [Gli00] und [GKM01] zu finden.

In beiden Dienstarchitekturen stößt der Terminal-Client den Auf- und Abbau oder die Modifikation einer Sitzung zwischen Teilnehmern über seine Dienststeuerung an. Er besitzt hierfür eine standardisierte Schnittstelle zu einer Dienststeuereinheit DS. Er signalisiert der DS beim Aufbau einer Sitzung, welche Nutzdatenverbindungen er aufbauen will und gibt dabei die Verbindungsparameter (Adresse, Kodierverfahren, Verkehrscharakteristik, QoS-Parameter) vor. Die Signalisierung des Verbindungsaufbaus erfolgt je nach der gewählten Dienststeuerarchitektur bzw. der verfügbaren Varianten in einer (SIP, H.323) oder in zwei Stufen (H.323).

Im Fall von SIP sendet der SIP-Client beim Aufbau einer SIP-Session eine *INVITE*-Nachricht über seinen SIP-Proxy an das Terminal des gerufenen Teilnehmers. Der SIP-Proxy unterstützt eine dynamische Adressverwaltung und übernimmt die Weiterleitung der Nachricht zum Ort des Zielteilnehmers. Die *INVITE*-Nachricht enthält alle Verbindungsparameter, die der Teilnehmer mit Hilfe des SDP-Protokolls (*Session Description Protocol*) [HJ98] spezifiziert hat. Das SDP sieht nur applikationsspezifische Parameter der Nutzdatenverbindungen vor (Codec, etc.), so dass diese im RM auf entsprechende TB-Verkehrsparameter umgesetzt werden müssen. Die Antwort des gerufenen Teilnehmers wird auf demselben Weg zurück zum Initiator geleitet. Alle Funktionen des SIP-Proxy Servers erfolgen transparent für den Teilnehmer. Da der Aufbau einer Sitzung einstufig erfolgt, muss die Ressourcenreservierung an den Aufbau einer Session gekoppelt werden. Die Ressourcenreservierung muss stattfinden, bevor der gerufene Teilnehmer über die Sitzung informiert wird.

In den meisten Varianten von H.323 erfolgt der Aufbau eines H.323-Rufes in zwei Stufen. Zunächst sendet der H.323-Client eine H.225.0 *ARQ*-Nachricht (*Admission Request*) [H.225] an seinen H.323-Gatekeeper und wartet auf die Erlaubnis, mit dem Verbindungsaufbau fortzufahren. Dabei werden u.a. Alias-Adressen in IP-Adressen aufgelöst und Zugangsberechtigungen überprüft. Die Ressourcenreservierung kann an diesen Zugangskontrollprozess angekoppelt werden. Die Parameter der *ARQ*-Nachricht wurden nachträglich von der ITU im Annex.N [AN.323] des H.323-Standards um den Parameter *BandWidthDetails* erweitert. Er enthält die Verbindungsparameter: Dienstklassenbezeichnung (*GuaranteedServicesClass*, *ControlledServicesClass*, *unspecified ServicesClass*), Verkehrsparameter (*BitrateClass*) und QoS-Parameter (*DelayErrorClass*). Erst nach dem Empfang einer *ACF*-Nachricht (*Admission Confirm*) von Seiten des Gatekeepers darf der H.323-Client den gerufenen Teilnehmer kontaktieren und eine *SETUP*-Nachricht senden. Dieser kann dann den Verbindungswunsch annehmen oder ablehnen.

Erst nachdem der Aufbau des Rufes bzw. der Session und damit alle H.245-Verbindungen zwischen den Endgeräten [H.245] von dem gerufenen Teilnehmer bestätigt wurden, darf mit der Übertragung der Nutzdaten begonnen werden. Die Nutzdaten werden in der Regel direkt an den Terminal-Client und nicht über eine DS gesendet.

6.3.2 Dienststeuereinheit DS

Die Dienststeuereinheit DS ist eine zentralisierte Dienststeuerungsinstanz im Netz. Sie stellt das netzseitige Gegenstück zur endgerätebasierten Dienststeuerung der Terminal-Clients dar. Die DS dient primär der Teilnehmerregistrierung und Adressauflösung, bietet ein dynamisches Adressmanagement und unterstützt dadurch die Teilnehmermobilität. Beim Aufruf eines Dienstes setzt sie teilnehmerfreundliche Alias-Adressen (Domain-Names, E.164 Adressen, Email-Adressen) in IP-Adressen um. Die DS hat darüber hinaus die Funktion eines Proxy-Servers, der stellvertretend für einen z.B. ausgeschalteten oder abgestürzten Terminal-Client reagieren kann. Von ihr können Signalisierungsnachrichten zum Auf- und Abbau oder zur Modifikation einer Sitzung bearbeitet und weitergeleitet werden. Nutzdaten werden jedoch nicht an eine DS adressiert.

Eine DS stellt eine Signalisierungsinstanz im Netz dar, die von einem Dienstbetreiber auch für weitergehende Aufgaben herangezogen werden kann. Zu diesen Aufgaben gehören Managementdienste wie z.B. Dienstzugangskontrolle, Netzbetreiberauswahl, Teilnehmerverwaltung (*User Profile Management*), Überwachung des Endgerätezustandes und der Verbindungsdauern (*Status Monitoring*) sowie die Tarifierung des Dienstes. Darüber hinaus kann eine DS den Teilnehmer bei der Lokalisierung von speziellen Servern im Netz und deren Einbindung in eine laufende oder zu etablierende Multimediasitzung unterstützen. Es wird im Folgenden davon ausgegangen, dass die DS den Verbindungs- und Gerätezustand des Terminal-Client überwacht und im Fehlerfall (Rechnerabsturz, Zeitüberschreitung, ect.) die Freigabe der reservierten Ressourcen des Terminal-Clients anstößt. Ein Beispielmechanismus wäre das IRQ/IRR-Verfahren *(Information Request/Response*), wie es im H.323 vorgesehen ist.

Spezielle Server sind z.B. *Gateways* oder *Border-Nodes*. Gateways stellen Netzübergänge in andere Netze wie z.B. GSM, ISDN oder PSTN dar. Border-Nodes hingegen sind Übergangsknoten zu anderen IP-Netzbetreibern. Sie werden wie Terminal-Clients behandelt und terminieren somit Signalisierungs- und unter Umständen auch Nutzdatenströme. Daneben gibt es auch noch Server für die Steuerung von Mehrteilnehmerkonferenzen MCU (*Multipoint Control Unit*) oder für die Steuerung von Telephoniediensten (*Supplementary Services*).

Die Dienststeuereinheit wird bei der Dienststeuerarchitektur H.323 mit dem Namen *Gatekeeper* bezeichnet. Bei SIP gibt es entsprechend der IETF-Philosophie für unterschiedliche Funktionen verschiedene Server. Für die Teilnehmerregistrierung wurde der *SIP Registrar* und für die Adressauflösung der *SIP Proxy Server* und der *SIP Redirect Server* eingeführt. Für beide hier näher betrachteten Dienststeuerarchitekturen wird im Rahmen dieser Arbeit die Dienststeuereinheit mit einer Schnittstelle zum RM-System versehen. Die Dienststeuereinheiten stellen somit das Bindeglied zwischen der Dienstarchitektur und der RM-Architektur dar. Die Reservierung der Netzressourcen sollte zu einem möglichst frühen Zeitpunkt der Dienstablaufsteuerung angestoßen werden. Die Ressourcen sollten bereits reserviert sein, wenn einer der gerufenen Teilnehmer über den Verbindungswunsch informiert wird (*Ringing*). Der frühestmögliche Zeitpunkt während des Dienstablaufs ist der Moment, bei dem einer DS die Verbindungsparameter für alle Medienströme der Sitzung vorliegen. Benötigt werden z.B. die IP-Adressen der Teilnehmer, die Medienströme, d.h. die verwendeten Kodierverfahren oder die Verkehrsparameter sowie die erforderlichen netzspezifischen QoS-Parameter. Die Dienstablaufsteuerung wird solange unterbrochen, bis das RM-System entweder eine erfolgreiche Ressourcenreservierung Ende-zu-Ende bestätigt und die Fortführung der Dienstsignalisierung erlaubt oder aufgrund Ressourcenmangels eine sofortige Beendigung des Dienstes durch die Dienststeuerung erzwingt.

6.3.3 Ressourcen-Manager RM

Der RM stellt eine zentralisierte Signalisierungsinstanz im Netz dar, die einer Dienststeuerung einen Ressourcenverwaltungsdienst anbietet. Dazu stellt der RM der Dienststeuerung für die Übertragung von echtzeitkritischen Medienströmen mehrere RM-Dienstklassen bereit. Jede RM-Dienstklasse besitzt eine Dienstgütespezifikation für die Ende-zu-Ende Übertragung von IP-Paketen. Die Dienstgütespezifikation besteht aus einer harten oder weichen Garantie hinsichtlich der Ende-zu-Ende Verzögerung und der Verluste.
Die Aufgabe des RMs ist es, den Teilnehmerzugang zu den Ressourcen zu kontrollieren. Für jede RM-Dienstklasse wurden bei der Konfiguration des Netzes explizit Ressourcen (Link-Pufferkapazitäten) reserviert. Der RM besorgt sich von einem Topologie-Manager ein Abbild der Netztopologie und der reservierten Ressourcen in Form von Datenobjekten. Die Aufgabe der Zugangskontrolle erfüllt er anhand der Datenobjekte stellvertretend für die Netzknoten, indem er deren Lastzustand überwacht und eine Zugangskontrollfunktion durchführt. Der RM greift dabei nicht auf die Netzknoten direkt, sondern auf Datenobjekte zu. Die Datenobjekte liegen in seiner lokalen Datenbank und repräsentieren die Endgeräte sowie die Knoten des Netzes.

Auf eine Anfrage der Dienststeuerung hin versucht er Ressourcen zu reservieren, die für die Umsetzung eines Dienstes gemäß den Anforderungen aus der Dienststeuerung erforderlich sind. Zunächst muss der RM die von der Dienststeuerung des Endgerätes ermittelten netzspezifischen QoS-Parameter auf eine RM-Dienstklasse abbilden. Danach muss er überprüfen, ob die für diese RM-Dienstklasse noch zur Verfügung stehenden Ressourcen ausreichen, die neuen Medienströme gemäß der Dienstgütespezifikation aufzunehmen. Dazu verwendet er die verbindungs- und netzspezifischen QoS-Parameter der DS sowie die Pfad- und Auslastungsinformationen und trifft die Annahmeentscheidung anhand eines der in Kapitel 5 untersuchten Zugangskontrollverfahren. Können alle Verbindungen angenommen werden, tätigt er eine Reservierung, indem er den Auslastungszustand aller beteiligten Netzelemente aktualisiert. Daraufhin meldet er der DS den Erfolg zurück. Sind die Ressourcen auf mindestens einem beteiligten Netzelement nicht ausreichend, bricht der RM den Reservierungsprozess ab und meldet der DS den Misserfolg. Das Ergebnis des Reservierungsprozesses (Erfolg/Misserfolg) veranlasst die Dienststeuerung, den Verbindungsaufbau fortzusetzen oder für einen unmittelbaren Abbruch zu sorgen.

In größeren Netzen kann der Ressourcenverwaltungsdienst auch verteilt realisiert werden. Die Ressourcen eines Netzes werden dann in mehrere Bereiche (*RM-Domänen*) aufgeteilt und von jeweils einer RM-Instanz verwaltet. Um eine Reservierung über mehrere RM-Instanzen aufbauen zu können, wird zwischen den RM-Instanzen ein asynchrones Transaktionsprotokoll verwendet. Eine Transaktion wird von einem RM angestoßen und besteht aus einer Anfrage und einer Antwort. Eine Transaktion kann der Aufbau, die Modifikation oder der Abbau einer Reservierung sein.

6.3.4 Topologie-Manager TM

Der TM hat die Aufgabe, die in seiner TM-Domäne befindlichen RM-Instanzen mit aktuellen Topologie-, Pfad- und Konfigurationsdaten der Netzknoten zu versorgen. Dazu besitzt er einen Topologieerkennungs- und Überwachungsdienst. Die RM-Instanzen können sich beim TM registrieren und danach jederzeit Topologiedaten anfordern. Zudem werden sie vom TM über Änderungen im Netz informiert.
Aufgabe des Topologieerkennungsdienstes ist es, ein Abbild der Netztopologie in Form eines Datenmodells anzulegen und in einer Datenbank zu speichern. Dazu greift er direkt auf die Netzelemente zu. Die Zugriffsrechte sowie eine Angabe über die Größe des zu erkundenden

Netzes werden ihm von Seiten der Administration vorgegeben. Der Prozess der Topologieerkennung wird bei der Initialisierung eines TM und danach in regelmäßigen Abständen sowie asynchron durch externe Ereignisse wie z.B. Topologieänderungen angestoßen.
Es ist die Aufgabe des Überwachungsdienstes, dass die RM-Instanzen sofort von Änderungen im Netz zu informieren und ihnen möglichst frühzeitig wieder aktualisierte Topologiedaten zur Verfügung zu stellen. Der TM wertet dazu Nachrichten des Netzmanagements aus und stößt gegebenenfalls den Topologieerkennungsprozess an. Ist die Topologieerkennung abgeschlossen, teilt er den RM-Instanzen mit, dass neue Topologiedaten vorliegen.

6.4 Konfiguration

6.4.1 Netz und Endgeräte

Es wurde bereits erwähnt, dass das RM-System auf einem DiffServ-Netz basiert und für eine Dienststeuerung einen Echtzeitübertragungsdienst anbietet. In diesem Abschnitt wird anhand eines Beispiels gezeigt, wie mit einem RM-System die von TIPHON spezifizierten Dienstklassen aus Abschnitt 6.2 realisiert werden können.

Das Beispiel soll zeigen, wie die Anforderungen der verschiedenen Applikations- und Verkehrstypen auf die Möglichkeiten und Fähigkeiten des Netzes abgebildet werden können (siehe Abbildung 6-1). Letztere werden durch die verwendeten Zugangskontrollverfahren des RM-Systems und die Konfiguration des IP-Netzes (DS-Klassen) bestimmt. Tabelle 6-3 zeigt ein Beispiel für eine mögliche Netzkonfiguration.
Zunächst müssen die konkreten Anforderungen der Applikationen an das Kommunikationssystem ermittelt werden. Dazu werden die im Netz vorkommenden Applikationstypen einer TIPHON-Klasse zugewiesen. Aus der Dienstklassenspezifikation der TIPHON-Klassen werden die netzspezifischen QoS-Parameter abgeleitet. Im Folgenden wird davon ausgegangen, dass für die Realisierung der TIPHON-Klassen ein DiffServ-basiertes Netz mit fünf DS-Klassen zur Verfügung steht. Um die drei höherwertigen TIPHON-Klassen „BEST“, „HIGH“ und „MEDIUM“ über diese DiffServ-Klassen abwickeln zu können, wird ein RM-System installiert, das für die DS-Klassen „1“, „2“ und „3“ eine Zugangskontrolle durchführt.

Jede TIPHON-Klasse muss einer DS-Klasse zugewiesen werden. Dabei ist zu beachten, dass das PHB (*Per Hop Behaviour*) der DS-Klassen für die TIPHON-Klassen „BEST“, „HIGH“ und „MEDIUM“ eine minimale Bedienrate auf jedem Link garantiert. Danach sind für die jeweiligen DS-Klassen geeignete Zugangskontrollverfahren zu definieren. Zudem müssen die Ende-zu-Ende gültigen QoS-Parameter der TIPHON-Klassen auf die Netztopologie übertragen werden. Für jede RM-Domäne ist ein PDB (*Per Domain Behaviour*), d.h. eine Dienstgütespezifikation, zu definieren. In diesem Beispiel wird vereinfachend von einer einzigen RM-Domäne ausgegangen.

Medium/ Appli-kation	ATM-Verkehrs-klasse	TIPHON-Klasse Applikation	RM-Dienstklasse Gültigkeit: Netz, Ende-zu-Ende		DS-Klasse (IP/MAC) Gültigkeit: Netz, pro Knoten	
			Name	Beschreibung	DS codepoint	Konfiguration (Scheduler, Puffer)
Audio (Echtzeit)	CBR	BEST	"A"	Harte Garantie Verzögerung: < 130 ms Verluste: 0 %	„1"	minimale Bedienrate, fester Puffer
Audio/Video (Echtzeit)	CBR / VBR	HIGH	"B"	Weiche Garantie der Verzögerung: < 200 ms Verluste: 3 %	„2"	minimale Bedienrate, maximaler Puffer
Audio/Video/ Daten (Echtzeit)	CBR / VBR	MEDIUM	"C"	Weiche Garantie der Verzögerung: < 270 ms Verluste: 15 %	"3"	minimale Bedienrate, maximaler Puffer
Daten	VBR	LOW	-	-	„4"	Prio 1
Daten	ABR	-	-	-	„5"	Prio 2

Tabelle 6-3: Beispiel für eine Netzkonfiguration

Für die TIPHON-Klasse „LOW" kann der Netzbetreiber nur die DS-Klasse „4" verwenden. Das PHB dieser Klasse wird durch einen Priority-Scheduler bestimmt. Ein Management der DS-Klasse „4" durch das RM-System wird nicht vorgenommen. Wird für die Realisierung der PHB das in Abschnitt 3.2.3 definierte Knotenmodell verwendet, steht für die DS-Klasse „4" auf jedem Link mindestens eine Bedienrate *C'* zur Verfügung, die sich aus der Linkkapazität *C* abzüglich der virtuellen Linkrate *C** ergibt (= Summe der minimalen Bedienraten der DS-Klassen 1-3). Der Netzbetreiber vertraut an dieser Stelle auf sein ausreichend dimensioniertes Netz und nimmt in Hochlastzeiten ein kurzzeitiges Verdrängen des Verkehrs der DS-Klasse „5" durch den Verkehr der DS-Klasse „4" in Kauf.

Nachdem die Konfiguration des Netzes festgelegt wurde, muss diese auf die Netzelemente übertragen und dort umgesetzt werden. Dabei sind zwei Vorgänge von besonderer Bedeutung: die Filterung und Klassifizierung (Markierung) der IP-Flows.

Klassifizierung

Die Zuweisung einzelner IP-Flows zu einer DS-Klasse kann statisch oder dynamisch erfolgen.

Eine statische Zuweisung kann vom Konfigurationsmanagement des Netzes vorgenommen werden. Dabei müssen das RM-System und alle Netzelemente (Endgeräte oder Zugangsknoten), welche die Markierung der IP-Pakete vornehmen, einheitlich konfiguriert werden. Von Seiten der Administration werden netzweit gültige Klassifizierungsregeln spezifiziert. Die Regeln können entweder manuell oder automatisiert im Netz verteilt werden. Im automatisierten Fall werden sie in Form von *Policies* spezifiziert und in einem zentralen *Policy-Server* abgelegt. Von dort werden sie über Protokolle wie z.B. COPS oder SNMP auf alle Netzelemente verteilt (*Policy-based Networks*).

Eine dynamische Zuweisung kann durch das RM-System pro Reservierung erfolgen. Das RM-System trifft eine Dienstgütevereinbarung mit dem Teilnehmer und bestimmt, welcher DS- bzw. RM-Klasse die IP-Pakete der Verbindung zugeordnet werden. Das RM-System kann mittels COPS oder SNMP entsprechende *Classifying-Filter* in den Zugangsknoten mit Verbindungsparametern (Port-Nummern, IP-Adressen) und einer DS-Klassenkennung (DSCP: *DiffServ Codepoint*) konfigurieren.

Filterung

In Intranets vertraut das RM-System darauf, dass es das Verhalten aller Applikationen kennt (siehe Messung und Charakterisierung der Quellenverkehre in Kapitel 4 und 5). In einer offenen Umgebung ist es jedoch sinnvoll, wenn die Medienströme überwacht werden. Die Überwachung der Medienströme erfordert die Konfiguration von Filtern mit den zuvor von der Dienststeuerung ausgehandelten Verkehrsparametern der Quelle (TB-Parametern). Die Filterung kann vom RM-System gesteuert werden, indem es Policing-Einheiten im Endgerät und/oder im Zugangsknoten konfiguriert.

Beispiel

In Abbildung 6.4 wird das Zusammenspiel zwischen der Applikationssteuerung, der Dienststeuerung und dem Konfigurationsmanagement (*Policy Control*) am Beispiel eines Endgerätes näher erläutert. Dabei erfolgt die Konfiguration der *Classifying-Filter* statisch und die der *Policy-Filter* dynamisch.

Die Applikationssteuerung koordiniert den zeitlichen Ablauf der Prozesse im Endgerät. Die Dienststeuerung handelt die Verbindungsparameter mit den anderen Dienstteilnehmern aus und stößt dabei die Ressourcenreservierung im Netz an. Die Mediensteuerung ist für die Signalverarbeitung und Paketierung der Sprach- und Videosignale verantwortlich. Das Konfigurationsmanagement ist für das Verteilen und Anwenden der netzspezifischen Konfigurationsregeln (*Policies*) zuständig.

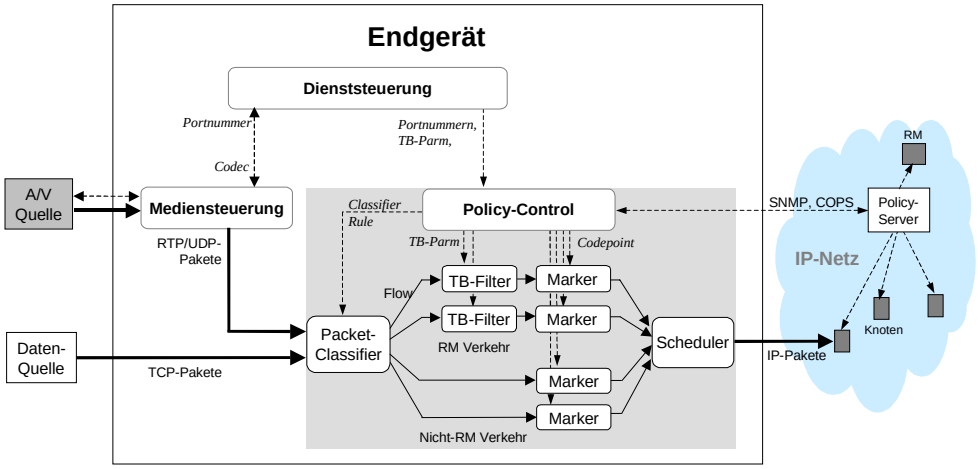


Abbildung 6-2: Ressourcensteuerung im Endgerät

Es wird in diesem Beispiel davon ausgegangen, dass die Mediensteuerung der Applikation die TB-Parameter der Quelle kennt und diese der Dienststeuerung beim Verbindungsaufbau mitteilt (z.B. H.323 Annex.N). Wird eine Verbindung zugelassen, teilt die Applikationssteuerung dem lokalen Konfigurationsmanagement (*Policy Control*) des Endgerätes die Portnummern und Verkehrsparameter mit.

Das Konfigurationsmanagement kann die IP-Pakete anhand der Portnummern einzelnen Transportverbindungen zuweisen. Die Pakete einer vom RM-System zugelassen Echtzeitverbindung, werden anschließend gemäß der Verkehrsparameter gefiltert und danach im IP-Paketkopf durch das Setzen bestimmter Bits (*DSCP*) als zu einer DS-Klasse zugehörig gekennzeichnet (Markierung). Das Setzen der DSCP findet nach einer statischen, netzweit einheitlichen Abbildungsvorschrift statt. Diese wurde dem lokalen Konfigurationsmanagement vom *Policy-Server* mitgeteilt.

6.4.2 RM-Domänen

Ein RM verwaltet die Ressourcen eines zusammenhängenden Netzbereiches (RM-Domäne). Eine RM-Domäne umfasst Netzknoten und Terminals, die zu einem bestimmten Adressbereich (z.B. ein oder mehrere IP-Subnetze) gehören, sowie alle zugehörigen Links. Der Adressbereich wird bei der Initialisierung eines RM durch das Netzmanagement fest vorgegeben. Zusammenhängend ist ein solcher Netzbereich nur dann, wenn alle möglichen Netzpfade zwischen den verschiedenen Terminals und Netzknoten über Links gehen, die ebenfalls zu demselben Netzbereich gehören. Diese Anforderung an die Topologie einer RM-Domäne wird im Folgenden auch als „Routing-kompakt“ bezeichnet.

Die Aufteilung größerer Netze in mehrere RM-Domänen ist durch die Beschränkungen eines RM-Servers hinsichtlich seiner Performance bedingt. Ein einzelner RM ist in den Signalisierungsablauf einer Dienststeuerung eingebunden und bearbeitet sequenziell die Reservierungsanfragen der Teilnehmer. Während die Anfrage eines Teilnehmers bearbeitet wird, ist der Prozess des Verbindungsaufbaus solange unterbrochen, bis der Teilnehmer vom RM die Erlaubnis zum Fortfahren bekommt. Die Zeitdauer eines Verbindungsaufbaus ist aus Sicht des Systems so gering wie möglich zu halten.

Die Größe einer RM-Domäne ist somit durch die Reaktionszeit eines RM begrenzt, welche aus Gründen der Dienstgütedefinition einen gewissen Maximalwert $T_{R,max}$ nicht überschreiten darf. Die Reaktionszeit eines RM auf eine Reservierungsanfrage hängt von der Anzahl der gleichzeitig ankommenden Reservierungsanfragen, der Komplexität der durchzuführenden Operationen pro Anfrage und der Performance des RM ab. Sowohl die Anzahl der an eine RM-Domäne angeschlossenen Teilnehmer als auch die Komplexität der Entscheidungsfindung wächst linear mit der Anzahl der Netzknoten N. Geht man davon aus, dass die Performance eines RM begrenzt ist, gibt es ein N_{max}, ab der $T_{R,max}$ für $N > N_{max}$ nicht mehr eingehalten werden kann. Die Aufteilung eines Netzes in mehrere RM-Domänen ermöglicht eine Limitierung der vorzuhaltenden Topologiedatenmenge in einem einzelnen RM und für das Gesamtsystem das parallele Bearbeiten von Reservierungen durch verschiedene RM-Instanzen.

In [RG02] wird gezeigt, wie man ein gegebenes Netz geeignet in RM-Domänen aufteilen und darin die zentralen RM-Instanzen platzieren kann, so dass die Reaktionszeit des RM-Systems minimiert wird. Das Optimierungsproblem besteht aus zwei Teilaufgaben:

1. Aufspaltung des Netzes in einzelne Domänen, basierend auf einer maximalen Ankunftsrate von Reservierungsanfragen pro RM-Instanz und einem maximalen *Hop-Count* der Pfade innerhalb einer RM-Domäne. Für jeden Netzknoten wird eine Ankunftsrate von Reservierungsanfragen gegeben. Ausgehend von einer RM-Domäne, wird jeder Netzknoten dieser Domäne zugewiesen, solange die maximal zulässige Ankunftsrate der zentralen RM-Instanz oder der maximale *Hop-Count* nicht überschritten werden. Wird jedoch eine der Grenzen überschritten, muss die Domäne aufgeteilt werden. Bei der Wahl der Domänengrenze wird nach dem minimalen Nachrichtenaustausch zwischen den RM-Instanzen optimiert. Jede RM-Domäne muss für sich wiederum „Routing-kompakt“ sein.
2. Platzierung der RM-Instanzen basierend auf einer gegebenen Domänentopologie. Optimiert wird nach den minimalen Signallaufzeiten der Reservierungsnachrichten.

In [RG02] wird das Optimierungsproblem als ein *Mixed-Integer* Programm (MIP) mathematisch formuliert und auf verschiedene Netze angewendet.

6.4.3 TM-Domänen

Eine TM-Domäne stellt ebenso wie die RM-Domäne einen Teilbereich eines größeren Netzes dar. Jeder Knoten eines Netzes wird genau einer TM-Domäne zugewiesen und von dem jeweiligen TM hinsichtlich seiner Konfiguration abgefragt. Die Aufteilung größerer Netze in mehrere TM-Domänen ist im Regelfall nicht durch Performance-Anforderungen an einen Server bedingt. Ein TM greift während der Topologieerkundung auf Konfigurationsdateien der Netzknoten zu. Ein solcher Zugriff erlaubt dem Zugreifenden einen tiefen Einblick in das Innerste des Netzes und gibt ihm unter Umständen sogar die Möglichkeit, die Konfiguration des Netzes zu verändern und dem Netzbetreiber großen Schaden zuzufügen. Problematisch ist, dass ein Zugriff auf die Netzknoten nicht ohne Weiteres von einem feindlichen Angriff

auf das Netz unterschieden werden kann. Daher trifft das Netzmanagement entsprechende Sicherheitsvorkehrungen und macht einen Lese-Zugriff normalerweise nur unter Verwendung von Passwörtern bzw. *Security Strings* (SNMP) möglich. Die Größe einer TM-Domäne ist daher durch die Vergabe von Zugriffsberechtigungen von Seiten der Netzadministration begrenzt. In der Regel besitzt ein Netzadministrator nur innerhalb seines Zuständigkeitsbereiches die notwenigen Zugriffsrechte. Aus diesem Grund sind die Grenzen einer TM-Domäne häufig die Grenzen eines administrativen Bereiches eines Netzbetreibers.

6.4.4 Systeminitialisierung

Für die Systeminitialisierung sind drei *Discovery*-Prozeduren vorgesehen. Das erste Verfahren betrifft die Zuordnung der RM- und TM-Instanzen. Mehrere RM können sich bei einem TM unter Angabe ihres Zuständigkeitsbereiches (Satz von IP-Subnetzadressen) registrieren und erhalten selektiv die entsprechenden Topologiedaten.

Das zweite *Discovery*-Verfahren dient der dynamischen Zuordnung von DS und RM. Es kann sowohl von den DS als auch von den RM angestoßen werden. Eine DS benötigt für alle IP-Subnetze, in denen sich registrierte Teilnehmer befinden, die Adresse des zugehörigen RM. Wird ein RM in einer bestehenden Systemumgebung hochgefahren, lädt er alle DS unter Angabe seines Zuständigkeitsbereiches ein, sich bei ihm zu registrieren. Wird eine DS hochgefahren, kann sie sich ebenfalls im Netz bei den bereits aktiven RM bekannt machen und damit eine Registrierungsprozedur initiieren. Bei der Registrierung muss sich die DS gegenüber dem RM mit den Parametern *Operator-IP*, *Sender Type* und *Sender Address* identifizieren. Dabei gibt die DS auch die Version des verwendeten Dienststeuerprotokolls an.

Das dritte *Discovery*-Verfahren läuft zwischen den einzelnen RM ab und dient dem Auffinden benachbarter RM. Zwei benachbarte Domänen teilen sich einen gemeinsamen Link, durch den die Domänengrenze verläuft. Endet eine Reservierung innerhalb einer RM-Domäne an einem solchen Grenz-Link, muss der RM wissen, welcher Nachbar-RM jenseits des Links für die Weiterführung der Reservierung zuständig ist. Nur so ist eine Reservierung Ende-zu-Ende über mehrere RM-Domänen hinweg möglich. Ein RM macht sich somit unter Angabe seiner Grenz-Links bei seinen Nachbar-RM bekannt.

Alle hier betrachteten *Discovery*-Verfahren können von beiden Seiten angestoßen werden. Sie basieren auf einer Registrierungsanfrage und einer Registrierungsbestätigung/-ablehnung. Für die Discovery-Prozesse werden innerhalb einer Administrativen Domäne *Multicast*-Adressen eingerichtet. Dadurch kann die Zuordnung zu den jeweiligen Instanzen dynamisch erfolgen.

Die Konfiguration der RM-Instanzen erfolgt über eine Konfigurationsdatei. Analog zur Tabelle 6-3 enthält sie die Zuordnung der RM-Klassen zu:

- Dienstklassenspezifikation, Zugangskontrollverfahren, DS-Klassen (DSCP)
- Applikationstypen (Portnummernbereiche) und Medienströme (Codecs)

Darüber hinaus enthält sie eine Zuordnung der Applikationstypen und Medienströme zu den gemessenen und charakterisierten TB-Parametern.

6.5 Kommunikationsbeziehungen

Nachdem im vorangegangenen Abschnitt die einzelnen Komponenten der RM-Architektur eingeführt wurden, wird in diesem Abschnitt betrachtet, in welcher Beziehung diese Komponenten zueinander stehen. In Abbildung 6-3 sind die Kommunikationsbeziehungen zwischen den Architekturkomponenten dargestellt.

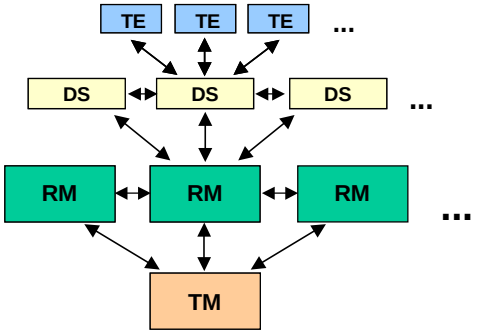


Abbildung 6-3: Kommunikationsbeziehungen zwischen den Komponenten

Jede Teilnehmerapplikation (TE) ist fest an eine Dienststeuereinheit (DS) im Netz angebunden. Der Teilnehmer registriert sich bei der Initialisierung seines Systems bei einer DS und richtet alle künftigen Verbindungswünsche zunächst an diese DS. In heutigen Systemen können sich in der Regel mehrere 100 Teilnehmer bei einer DS registrieren.
Einer RM-Instanz wiederum sind eine oder mehrere Dienststeuereinheiten fest zugeordnet. Eine DS registriert sich dazu bei einem RM. Wie viele DS sich bei einem einzelnen RM registrieren dürfen, ist durch die maximal zulässige Ankunftsrate eines RM begrenzt (siehe Abschnitt 3.5.5: Domänenbildung). Diese schätzt der RM aus der Anzahl der bei einer DS registrierten Teilnehmer sowie einem Erfahrungswert bezüglich der Aktivität der Teilnehmer (z.B. BHCA: *Busy Hour Call Attempts*) ab. Umgekehrt ist eine DS im Allgemeinen nur einem RM zugeordnet. Unterstützt die Dienststeuerung eine Teilnehmermobilität, kann in Ausnahmefällen eine Kommunikationsbeziehung zu weiteren RM-Instanzen erforderlich sein. Jeder Verbindungswunsch eines Teilnehmers löst bei einer DS eine Reservierungsanfrage an den zuständigen RM aus.
Liegt der Zielteilnehmer nicht im Verwaltungsbereich dieses RM, leitet er die Anfrage an einen seiner Nachbarn weiter. Somit tauschen benachbarte RM-Instanzen gegenseitig Reservierungsnachrichten aus.
Eine RM-Instanz wird dynamisch einer TM-Instanz zugeordnet (z.B. Systeminitialisierung). Umgekehrt können einer TM-Instanz mehrere RM-Instanzen zugeordnet sein. Die Anzahl der RM, die sich bei einem TM registrieren, hängt von der Größe der TM-Domäne ab. Ein RM registriert sich bei dem TM, der Zugriff auf die Ressourcen seiner RM-Domäne hat. Auf Anfrage erhält ein RM vom TM die ihn betreffenden Topologiedaten des Netzes, die er dann lokal speichert. Nachrichten müssen somit nur bei der Systeminitialisierung oder bei Änderungen der Netztopologie ausgetauscht werden.

6.5.1 Schnittstellen

Die RM-Architektur besitzt sowohl interne wie externe Systemschnittstellen. Diese sind in Abbildung 6-4 dargestellt. Bei den Komponenten des RM-Systems gibt es eine Schnittstelle zwischen einer RM- und einer TM-Instanz (I2) sowie eine zweite zwischen zwei benachbarten RM-Instanzen (I1). Nach außen bietet das RM-System eine Schnittstelle zu den Dienststeuereinheiten (E1) sowie eine weitere zum Netzmanagement (E2).

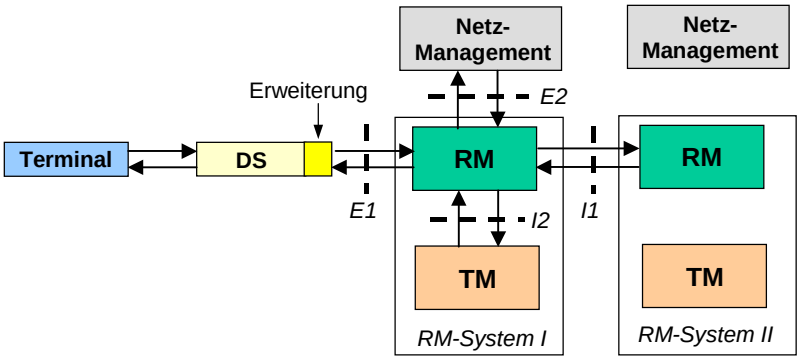


Abbildung 6-4: Schnittstellen der RM-Architektur

Von außen muss das RM-System für Reservierungsanfragen zugänglich gemacht werden. Eine direkte Schnittstelle zum Teilnehmer ist prinzipiell möglich. Es wurde jedoch aus Gründen der leichteren Einführbarkeit des RM-Systems in eine bestehende Systemumgebung auf eine solche verzichtet. Stattdessen besitzt das RM-System eine Schnittstelle zu einer oder mehreren Dienststeuereinheiten. Um die Einführbarkeit des RM-Systems weiter zu verbessern, basiert die Schnittstelle *E1* auf der Terminal-DS Schnittstelle. Daher sind die über *E1* ausgetauschten Nachrichten Teil der Dienststeuerung und nicht des Reservierungsprotokolls. Das hier vorgestellte Konzept sieht im RM eine Protokollumsetzung auf die RM-Signalisierung vor.
Das RM-System benötigt bei seiner Initialisierung Konfigurationsdaten bezüglich der Domänengrenzen oder der Dienstklassenspezifikation. Ferner hält das RM-System wichtige Daten über den Zustand des Netzes, die von außen zugänglich gemacht werden sollen. Daher besitzt das RM-System eine Schnittstelle zu einem externen Konfigurations- und Netzmanagementsystem.
Die Schnittstellen *I1* und *I2* des Systems werden in den nachfolgenden Abschnitten genauer definiert.

6.5.2 Reservierungsprotokoll

Das Reservierungsprotokoll läuft zwischen benachbarten RM-Instanzen ab und ist somit Teil der Schnittstellenspezifikation *I1* aus Abbildung 6-4. Angestoßen wird eine Reservierung durch eine standardisierte Signalisierung einer Dienststeuerungsarchitektur auf der Teilnehmer-DS Schnittstelle. Die entsprechenden Nachrichten werden von der DS an das RM-System weitergeleitet und dort in Reservierungsnachrichten des RM-Systems umgesetzt. Die Nachricht einer Dienststeuerung bezieht sich immer auf den Kontext z.B. einer Multimediasitzung. Eine Multimediasitzung kann mehrere Kommunikationsbeziehungen enthalten, falls z.B. mehrere Medienströme und/oder Teilnehmer beteiligt sind.

Aggregation

Analog zur Dienstsignalisierung bezieht sich eine Nachricht des Reservierungsprotokolls ebenso auf den Kontext einer Multimediasitzung. Jede Reservierungsnachricht kann mehrere Medienströme enthalten, für die Ressourcen reserviert werden sollen. Jeder Medienstrom, d.h. jede unidirektionale RTP-Verbindung, stellt in der RM-Instanz einen eigenen Reservierungsprozess dar. Dabei wird zunächst der Pfad durch das Netz ermittelt und dann für alle Medienströme mit derselben Quell- und Zielknotenkombination sequenziell die Zugangskontrolle durchgeführt. Erst wenn für alle Medienströme die Ressourcen reserviert werden konnten, ist der Reservierungsvorgang abgeschlossen.

6.5.2.1 Transaktionen

Das RM-Protokoll arbeitet zustandsbasiert und sieht vier Funktionen vor:

- *Systeminitialisierung*
- *Aufbau, Modifikation* und *Abbau* einer Reservierung
- *Statusüberwachung* von Reservierungszuständen
- *Transportmechanismus* für nicht näher spezifizierte Daten (Tunnelmechanismus)

Für jede dieser Funktionen ist ein gesicherter Transaktionsmechanismus definiert. Die Sicherung erfolgt über Bestätigungsmeldungen der Partnerinstanz. Die Statusüberwachung wird über *Timer* ausgelöst und dient der Freigabe von blockierten Ressourcen in Fehlerfällen. Der Transportmechanismus wird z.B. für die Benachrichtigung benachbarter RM beim Erkennen von Topologieänderungen oder in Interworking-Szenarien für das Tunneln von Reservierungsnachrichten anderer QoS-Architekturen benötigt (siehe Abschnitt 6.8.1.1). Die Systeminitialisierung wurde in Abschnitt 6.4.4 beschrieben. Im Folgenden liegt der Schwerpunkt der Betrachtungen bei den Reservierungsfunktionen.

Ein Reservierungsvorgang wird immer dann von einer Dienststeuereinheit angestoßen, wenn die Dienststeuerung eines Teilnehmers eine neue Nutzdatenverbindung aufbaut, eine bestehende modifiziert oder abbaut. Die Reservierungsanfrage der DS kann nur dann angenommen werden, wenn alle beteiligten RM die Reservierung zugelassen haben. Erfährt einer der RM eine Blockierung auf einem seiner Links, bricht er die Reservierung unmittelbar ab und teilt dies allen anderen RM mit. Empfängt ein RM eine Nachricht über einen Reservierungsabbruch von einem anderen RM, gibt er bereits belegte Ressourcen wieder frei.

Während einer laufenden Multimediasitzung sieht das RM-System die Möglichkeit vor, den Ressourcenbedarf den Wünschen der Teilnehmer und den sich ändernden Anforderungen an das Netz anzupassen. Eine *Modifikation* einer bestehenden Reservierung besteht nun darin, Verbindungsparameter einzelner bestehender RTP-Verbindungen dynamisch zu verändern. Das bedeutet, dass die bereits getätigten Reservierungen zu dieser Sitzung aufzuheben und unmittelbar im Anschluss daran neue Reservierungen vorzunehmen sind.

Beim *Abbau* wird für einen Teilnehmer die ganze Sitzung beendet und alle dazugehörigen Reservierungen werden abgebaut. Dabei bezieht sich die Freigabe der Ressourcen immer nur auf alle Verbindungen von und zum Sender einer entsprechenden Signalisierungs-Nachricht. Im Falle einer Mehrteilnehmerkonferenz bleiben somit alle Verbindungen zwischen den anderen Teilnehmern unberührt. Um überflüssige Nachrichten innerhalb des RM-Systems zu vermeiden, werden auch beim Reservierungsabbau Redundanzen erkannt.

6.5.2.2 Routing

Aufbauend auf der Konfiguration der RM- und TM-Domänen sowie der Systeminitialisierung aus Abschnitt 6.4.4 wird im Folgenden kurz erklärt, wie anhand eines gegebenen Routings im Netz das Routing der Reservierungsnachrichten erfolgen kann. Der Sachverhalt wird anhand von Abbildung 6-5 näher erläutert. Dazu ist ein Beispielnetz, bestehend aus drei RM-Domänen, dargestellt.

Zur Veranschaulichung werden in drei Ebenen getrennt voneinander dargestellt: die Topologie des physikalischen Netzes und das virtuelle Abbild der Topologie im RM-System mit dem Reservierungspfad und dem Signalisierungspfad. Die **Netzebene** enthält das physikalische Netz und die durch das Routing-Protokoll eingestellten Nutzdatenpfade. Die **RM-Ebene** wird durch die RM repräsentiert. Sie enthält die Sichtweisen der jeweiligen RM auf ihre zu verwaltenden Ressourcen. Sowohl die Ressourcen als auch der Reservierungspfad existieren nur virtuell in Form eines Datenmodells. Auf der Grundlage dieses Datenmodells

ermitteln die RM den Pfad der Nutzdaten, auf denen die Reservierung vorzunehmen ist. Die **Signalisierungsebene** zeigt nun den Pfad der Signalisierungsnachrichten, der sich vom Pfad der Nutzdaten unterscheiden kann, da der Signalisierungspfad über bestimmte Zwischenknoten laufen muss, an denen die RM angeschlossen sind.

Ein Verbindungsaufbauwunsch eines Teilnehmers wird zunächst an die zugehörige DS gerichtet. Die DS kann dann anhand der IP-Quelladresse des Teilnehmers den zuständigen RM ermitteln und eine Reservierungsanfrage an ihn richten. Liegen alle Teilnehmer innerhalb seiner Domäne, kann er die Annahmeentscheidung alleine treffen.
Liegt jedoch einer der Zielteilnehmer außerhalb seiner Domäne, muss er die Reservierungsanfrage zu einem Nachbar-RM weiterleiten. Hat ein RM mehrere angrenzende RM-Domänen, kann die Nachbar-Domäne anhand des Nutzdatenpfades ermittelt werden. Der Nutzdatenpfad beginnt bei der Quelle und kann vom RM bis zur Domänen-Grenze verfolgt werden. Zunächst werden die erforderlichen Ressourcen innerhalb der Domäne bis zur Domänen-Grenze reserviert. Nach erfolgreicher Reservierung kann dann anhand des ermittelten Grenz-Links zur Nachbar-Domäne der Nachbar-RM bestimmt und die Reservierungsanfrage an ihn weitergeleitet werden. Dieser Vorgang wird solange wiederholt, bis die RM-Domäne des gerufenen Teilnehmers erreicht ist und die notwendigen Ressourcen auf allen Links des Datenpfades zwischen den Terminals reserviert sind. Der RM der Domäne des Zielteilnehmers bestätigt die erfolgreiche Reservierung. Diese läuft über alle an der Reservierung beteiligten RM bis zum RM der Domäne des Initiators zurück.

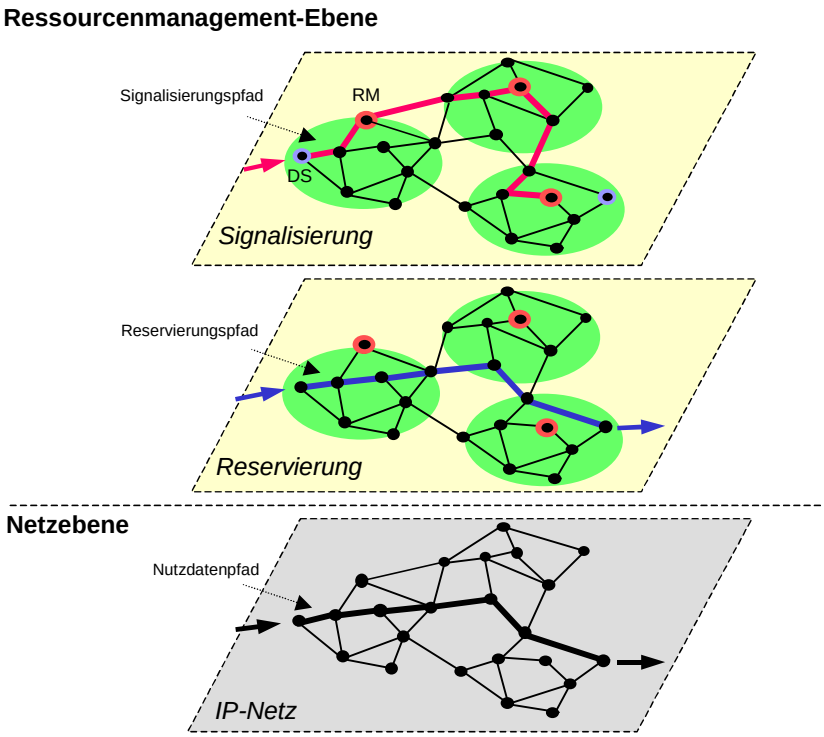


Abbildung 6-5: Schichtenmodell

Der soeben beschriebene Routing-Vorgang bezieht sich auf uni- und bidirektionale Reservierungen gleichermaßen, solange symmetrische Routen vorliegen. Im Fall von asymmetrischen Routen kann es während der Bestätigungsphase zu einer Gabelung des Signalisierungspfades kommen. Der RM der Ziel-Domäne (und danach alle vorangegangenen RM) überprüft z.B. bei einer bidirektionalen Verbindung, ob der Datenpfad vom Zielteilnehmer zum Initiator auch tatsächlich denselben Weg durch das Netz geht, oder nicht. Falls für den Rück-Pfad ein anderer Grenz-Link zu einer anderen RM-Domäne erkannt wird,

wird vom RM der betreffenden Domäne eine neue unidirektionale Ende-zu-Ende Reservierung in Richtung des Initiators angestoßen. Gleichzeitig bestätigt der RM auf dem Hin-Pfad nur die Reservierungen in Hin-Richtung und veranlasst die Freigabe der bereits getätigten Reservierungen in Rück-Richtung.

6.5.2.3 Nachrichtenformat und Parameter

Nachfolgende Tabelle enthält alle Nachrichten des Reservierungsprotokolls und eine kurze Funktionsbeschreibung.

Methode	Bezeichnung	Beschreibung
RA	Reservierungsanfrage	veranlasst eine RM-Instanz, eine Reservierung vorzunehmen.
RB	Bestätigung einer Reservierungsanfrage	positive Bestätigung einer Reservierungsanfrage durch eine RM-Instanz.
RZ	Zurückweisung einer Reservierungsanfrage	die Blockierung einer Reservierungsanfrage veranlasst eine RM-Instanz, die bereits getätigte Reservierung wieder zurückzunehmen.
MA	Modifizierungsanfrage	veranlasst eine RM-Instanz, parallel zu einer bereits bestehenden Reservierung eine neue Reservierung durchzuführen.
MB	Bestätigung einer Modifikation	positive Bestätigung einer Modifizierungsanfrage durch eine RM-Instanz. Sie veranlasst eine RM-Instanz, die alte Reservierung zurückzunehmen und die neue Reservierung zu bestätigen.
MZ	Zurückweisung einer Modifizierungsanfrage	die Zurückweisung einer Modifizierungsanfrage veranlasst eine RM-Instanz, die alte Reservierung zu bestätigen und die neue Reservierung zurückzunehmen.
FA	Freigabeaufforderung	veranlasst eine RM-Instanz, eine Reservierung zurückzunehmen und die reservierten Netzressourcen wieder freizugeben.
FB	Freigabebestätigung	positive Bestätigung einer Freigabeaufforderung durch eine RM-Instanz.
INFO	Informationstransport	dient der gesicherten Übertragung von Reservierungsnachrichten anderer Architekturen z.B. RSVP (Tunnelung). INFO-Nachrichten stoßen keine Zugangskontrollfunktion an.
ACK	Empfangsbestätigung	bestätigt den Erhalt einer INFO-Nachricht.
SA	Statusabfrage	dient der Überwachung der Reservierungszustände.
SB	Statusbestätigung	liefert Zustandsinformationen über eine ganze Sitzung oder über eine einzelne Verbindung zurück.

Tabelle 4: Nachrichten des Reservierungsprotokolls

Mit Hilfe der Nachrichten werden für die Reservierung erforderliche Parameter zwischen den beteiligten Instanzen transportiert. Die Parameter sind in nachfolgender Tabelle aufgelistet.

Parameter	Bedeutung	Inhalt
Res-ID	Global eindeutiger Bezeichner für eine Sitzung. Wird beim Aufbau einer Reservierung festgelegt und ab dann für alle weiteren Meldungen der Sitzung verwendet. Er wird von einer DS gesetzt und kann z.B. aus einem *CallIdentifier* (H.323) abgeleitet werden.	SessionIdentifier
Req-ID	Bezeichner für zusammengehörige Meldungen einer Reservierungsoperation. Er ist abschnittsweise gültig.	RequestIdentifier
Res-Obj	Satz an Verbindungsobjekten. Jedes Verbindungsobjekt steht für eine RTP-Verbindung: Res-Obj ::= SEQUENCE {CONNECTION-Object, ...} Eine Reservierung wird immer auf alle Verbindungsobjekte angewendet. Entweder können Netzressourcen für alle oder für keines der Verbindungsobjekte reserviert werden.	SEQUENCE of CONNECTION-Objects
Activity	Gibt den Beginn und das Ende einer Reservierung an. Wird für Reservierungen benötigt, die von einem Teilnehmer im Voraus getätigt und vom RM-System zu einem späteren Zeitpunkt aktiviert werden.	ACTIVITY-Object
Operator-ID	Entspricht einer Signatur des RM-Betreibers, von der die Nachricht gesendet wurde.	OperatorIdentifier
Sender-Type	Gibt den Typ einer Signalisierungsinstanz an, von der die Nachricht gesendet wurde. Der Typ ist entweder eine DS- oder RM-Instanz. Type ::= CHOICE {DS, RM}	CHOICE of SenderType
Path-Info	Gibt den Pfad der Signalisierungsnachrichten an. Beinhaltet einen Satz an RM-Adressen. Jeder RM, der eine Reservierungsnachricht an einen Nachbar-RM weiterleitet, ergänzt seine eigene Adresse.	SEQUENCE of TransportAddresses
IR-Adr	Ingress Router Address. Gibt die IP Adresse eines Interfaces zwischen zwei benachbarten RM-Domänen an. Der initiierende DS setzt dafür z.B. das Default-Gateway. Ein RM setzt den bei der Pfadsuche zur Zieldomäne ermittelten Port eines Knotens an der Grenze zur Nachbardomäne.	IP-Adress
Protocol-Type	Gibt den Typ und die Version des verwendeten Dienststeuerprotokolls an. (z.B. H.323 V. 3, SIP V. 1.0)	ProtocolIdentifier
Content-Type	Gibt den Inhalt des Nutzdatenfeldes an. Folgende Typen sind vorgesehen: Reason, AsymmetricRouteInd	CHOICE of ContentType
Data-Field	Wird von der Ablaufsteuerung eines RM nicht interpretiert. z.B. Reason: Grund einer Zurückweisung, z.B. Blockierung aufgrund von Überlast oder Konfigurationsänderungen des Netzes.	Text

Tabelle 3: Definition der Protokollparameter

Nachfolgend wird der Inhalt der ACTIVITY- und der CONNECTION-Objekte genauer spezifiziert.

ACTIVITY-Object

Begin	Gibt den Zeitpunkt des Beginns einer Sitzung an. Schedule :: = SEQUENCE {Date, Time}	Schedule
End	Gibt den Zeitpunkt des Endes einer Sitzung an. Schedule :: = SEQUENCE {Date, Time}	Schedule

CONNECTION-Objekt

Con-ID	Bezeichner für eine RTP-Verbindung. Muss zusammen mit dem Res-ID global eindeutig sein.	Connection Identifier
Source-Addr	Satz von Parametern, der die Nutzdatenquelle identifiziert: TransportAddress :: = SEQUENCE {IP-Adress, Port-Number}	TransportAddress
Dest-Addr	Satz von Parametern, der die Nutzdatensenke identifiziert: TransportAddress :: = SEQUENCE {IP-Adress, Port-Number}	TransportAddress
Traffic-Spec	Verkehrscharakteristik der RTP-Quelle. Besteht aus einem Satz an Complex Token-Bucket Parametern und der maximalen Paketlänge: TrafficSpecification ::= SEQUENCE {Tokenfillrate, BucketSize, Peakrate, PacketLength}	TrafficDescriptor
QoS-Spec	Ein Satz von Dienstgüteparametern. Besteht aus einem Bezeichner für die Dienstklasse des RM-Systems, einem Satz von Ende-zu-Ende netzspezifischen QoS-Parametern und einer Garantietypbezeichnung: QoSSpecification ::= SEQUENCE {ServiceClassIdentifier, TRANSPORT-QoS}	QoSDescriptor
QoS-Status	Ein Satz von Dienstgüteparametern. Damit teilt ein RM seinem Nachbar-RM den Dienstgütestatus einer Reservierung von der Quelle bis zur Domäne des Nachbar-RM mit. Die Parameter enthalten die Summe aller bis dahin entlang des Reservierungspfades aufgelaufenen Verzögerungen und Verluste: QoS-Status ::= SEQUENCE {Delay-Status, Loss-Status} Anhand dieser Status-Parameter weiß ein RM, wie groß die Verzögerung und die Verluste in seiner Domäne sein dürfen, damit die vom Teilnehmer geforderte Ende-zu-Ende Transport-QoS nicht verletzt wird.	QoS-Status

Das CONNECTION-Objekt wiederum enthält zwei Datenstrukturen TRANSPORT-QoS und QoS-STATUS, deren Aufbau in den nachfolgenden Tabellen spezifiziert wird.

TRANSPORT-QoS

QoS-Type	Gibt an, ob der Teilnehmer eine harte oder weiche QoS-Garantie fordert.	BOOLEAN
Delay-Spec	maximale Ende-zu-Ende Verzögerung in ms. Dieser Parameter dient einem RM zur Auswahl einer Dienstklasse.	INTEGER
Loss-Spec	maximale Ende-zu-Ende Verlustwahrscheinlichkeit von RTP-Paketen in $x \cdot 10^{-4}$.	INTEGER

QoS-STATUS

Delay-State	Enthält die akkumulierte Verzögerung entlang des Reservierungspfades in ms. Bei einer Reservierung über mehrere RM-Instanzen berechnet jede RM-Instanz die in ihrer Domäne anfallende Verzögerung und addiert diesen Wert zu dem alten Wert hinzu.	INTEGER
Loss-State	Enthält die akkumulierte Verlustwahrscheinlichkeit entlang des Reservierungspfades in $x \cdot 10^{-4}$.	INTEGER

Tabelle 5 zeigt den Aufbau der Reservierungsnachrichten und gibt Aufschluss darüber, welche Parameter in den Nachrichten enthalten sein müssen (*mandatory* „m“) und welche Parameter optional (*optional* „o“) sind. Das Symbol „-“ in der Tabelle bedeutet, dass der Parameter an dieser Stelle nicht vorgesehen ist.

Parameter	RA	RB	RZ	MA	MB	MZ	FA	FB	INFO	ACK	SA	SB
Res-ID	m	m	m	m	m	m	m	m	m	m	m	m
Req-ID	m	m	m	m	m	m	m	m	m	m	m	m
Res-Obj	m	m	m	m	m	m	m	m	-	-	o	o
Activity	o	o	o	-	-	-	-	-	o	o	o	o
Operator-ID	m	o	o	m	o	o	m	o	m	m	o	o
Sender-Type	o	o	-	o	o	o	o	o	-	-	o	o
Path-Info	o	o	o	o	o	o	o	o	-	-	o	o
IR-Adr	m	m	-	o	o	o	o	o	m	m	o	o
Protocol-Type	m	-	-	-	-	-	-	-	-	-	-	-
Content-Type	-	-	o	-	-	o	-	-	m	o	-	-
Data-Field	-	-	o	-	-	o	-	-	m	o	o	o

Tabelle 5: Nachrichtenaufbau

Das hier spezifizierte Protokoll koordiniert und überwacht die Verteilung der Zustandsinformationen im Netz. Das Protokoll transportiert die komplette Verbindungsstruktur einer Multimediasitzung, d.h. die Verbindungs- und QoS-Parameter aller RTP-Verbindungen. Es unterstützt die unmittelbare Reservierung von uni- oder bidirektionalen Verbindungen ebenso wie Vorabreservierungen. Darüber hinaus ermöglicht es Inter-Domain Reservierungen und unterstützt das *Interworking* mit anderen QoS-Architekturen (z.B. DiffServ-BB, IntServ). Letzteres wird durch die Wahl der Parameter (z.B. Verkehrsparameter: TB-Parameter) und durch einen Transport-Mechanismus für Nachrichten anderer Reservierungsprotokolle erreicht.

Ist wie im Konfigurationsbeispiels aus Tabelle 6-3 (Abschnitt 6.4.1) keine eigene Verkehrsklasse für Signalisierungsverkehr vorgesehen, werden alle Reservierungsmeldungen des RM-Systems für nicht-echtzeitkritische Daten mit dem DSCP „4“ markiert.

6.5.2.4 Inter-Domain Reservierungen

Der erste RM des Reservierungspfades ermittelt zunächst die Ende-zu-Ende gültigen, netzspezifischen QoS-Parameter (Transport-QoS) und bestimmt dann die RM-Dienstklasse. Jede RM-Dienstklasse besitzt eine vorkonfigurierte Dienstklassenspezifikation. Diese enthält ein **QoS-Budget**, welches das Ende-zu-Ende Verhalten einer RM-Domäne (PDB: *Per Domain Behavior*) definiert. Bei einer Inter-Domain Reservierung garantiert jeder RM des Reservierungspfades ein QoS-Budget. Das Verhalten mehrerer RM-Domänen kann durch Summenbildung der QoS-Budgets ermittelt werden. Beim Aufbau einer Inter-Domain Reservierung ordnet jeder RM die Verbindungen einer Reservierung einer RM-Dienstklasse nach eigenen Regeln zu, addiert sein QoS-Budget zu den aktuellen Werten der QoS-Status Parameter und vergleicht diese mit den Transport-QoS Parametern. Werden die Transport-QoS Parameter klar überschritten, wird die Reservierung blockiert. Im anderen Fall wird die Zugangskontrolle durchgeführt.

6.5.3 Anbindung an eine Dienststeuerung

In diesem Abschnitt wird die Anbindung des RM-Systems an eine Dienststeuerung näher untersucht. Um den RM mit möglichst geringem Aufwand in eine bestehende Systemumgebung einbinden zu können, sind in der Ablaufsteuerung der DS nur geringfügige Änderungen erlaubt. Die Steuerung einer DS muss jedoch um zwei Funktionen erweitert werden, welche die Systeminitialisierung und die Dienstablaufsteuerung betreffen:

- Initialisierung: während der System-Initialisierung muss die DS den ihr zugehörigen RM suchen und sich bei ihm registrieren.

- Dienstablaufsteuerung: beim Aufruf des Dienstes durch einen Teilnehmer muss die Ablaufsteuerung an geeigneter Stelle unterbrochen und mit einer Reservierungsanfrage an das RM-System versehen werden.

Die Realisierung sieht vor, dass die Signalisierung des Teilnehmers zum Verbindungsaufbau auf der DS-RM Schnittstelle für die Reservierung verwendet wird. Dadurch kann die DS die Nachrichten des Teilnehmers einfach an den RM weiterleiten. Die Protokollumsetzung findet dann im RM statt. In den nächsten Abschnitten wird für die Dienstarchitekturen H.323 und SIP jeweils ein konkreter Lösungsvorschlag für die Einbindung des Reservierungsverfahrens in die Ablaufsteuerung gemacht.

6.5.3.1 H.323

Der Ablauf der Signalisierung wird anhand eines Beispiels erläutert (siehe Abbildung 6-6). Jede Reservierung wird z.B. durch den Empfang einer ARQ-Nachricht eines Gatekeepers ausgelöst. Sie beruht zu diesem Zeitpunkt allein auf den Annahmen des Initiators über die zu etablierende Sitzung bezüglich der Medienströme (Anzahl, Codec) in Sende- und Empfangsrichtung. Die Reservierung in Phase 1 wird Ende-zu-Ende vorgenommen und von allen beteiligten Instanzen bestätigt.
Da die Signalisierung symmetrisch ist, wird sie von allen beteiligten Terminals angestoßen. Demzufolge trifft nicht nur senderseitig sondern auch empfängerseitig (pro Teilnehmer) eine ARQ-Nachricht beim RM-System ein. Auf Seite des Empfängers sind jedoch keine zusätzlichen Informationen für das RM-System im ARQ enthalten. Dies gilt zumindest bei einer Sitzung mit nur zwei Teilnehmern ohne MCU. Es werden daher redundante ARQ-Nachrichten im RM-System erkannt und verworfen. Sie werden lediglich in den Fällen benötigt, in denen die Senderseite keinen RM besitzt, wohl aber die Empfängerseite.
Die getroffenen Annahmen zu Beginn von Phase P_1 können nach der Signalisierung zwischen den Terminals überholt sein, so dass die bereits getätigte Reservierung an die neuen Verhältnisse angepasst werden muss.

Nach Phase P_1 findet eine Signalisierung zwischen den Endgeräten statt. Wird als Dienststeuerung eine flexible Multimedia-Signalisierung verwendet, bei der die einzelnen Medienströme dynamisch zwischen den Endgeräten verhandelt werden können (Wahl des Codecs), kann sich der Ressourcenbedarf noch einmal ändern. Ansonsten können Teilnehmer auch zu einem späteren Zeitpunkt d.h. irgendwann während einer Sitzung neue Medienströme hinzufügen oder bereits existierende abbauen (z.B. BRQ).

Phase P_3 „*Modifikation*“ besteht darin, den Ressourcenbedarf den Anforderungen der Terminals und den Wünschen der Teilnehmer anzupassen. Während die Anforderungen der Terminals spätestens mit dem Aushandeln der Verbindungsparameter (z.B. *Capability Exchange*) bekannt sind, können sich die Wünsche der Teilnehmer während der Sitzung jederzeit ändern. Das bedeutet, dass die bereits getätigten Reservierungen zu dieser Session möglichst schnell aktualisiert werden müssen. Dazu sind einzelne Reservierungen (Verbindungen) aufzuheben und unmittelbar im Anschluss daran Neue vorzunehmen.

In Phase P_4 „Abbau“ wird für einen Teilnehmer die Sitzung beendet und alle zugehörigen Reservierungen abgebaut. Dabei bezieht sich die Freigabe der Ressourcen immer auf alle Verbindungen von und zu dem Sender einer entsprechenden Nachricht (z.B. einer DRQ-Nachricht). Im Falle einer Mehrteilnehmerkonferenz bleiben somit alle Verbindungen zwischen den anderen Teilnehmern davon unberührt. Um überflüssige Nachrichten innerhalb

des RM-Systems zu vermeiden, werden auch beim Reservierungsabbau Redundanzen erkannt und die entsprechenden Nachrichten verworfen.

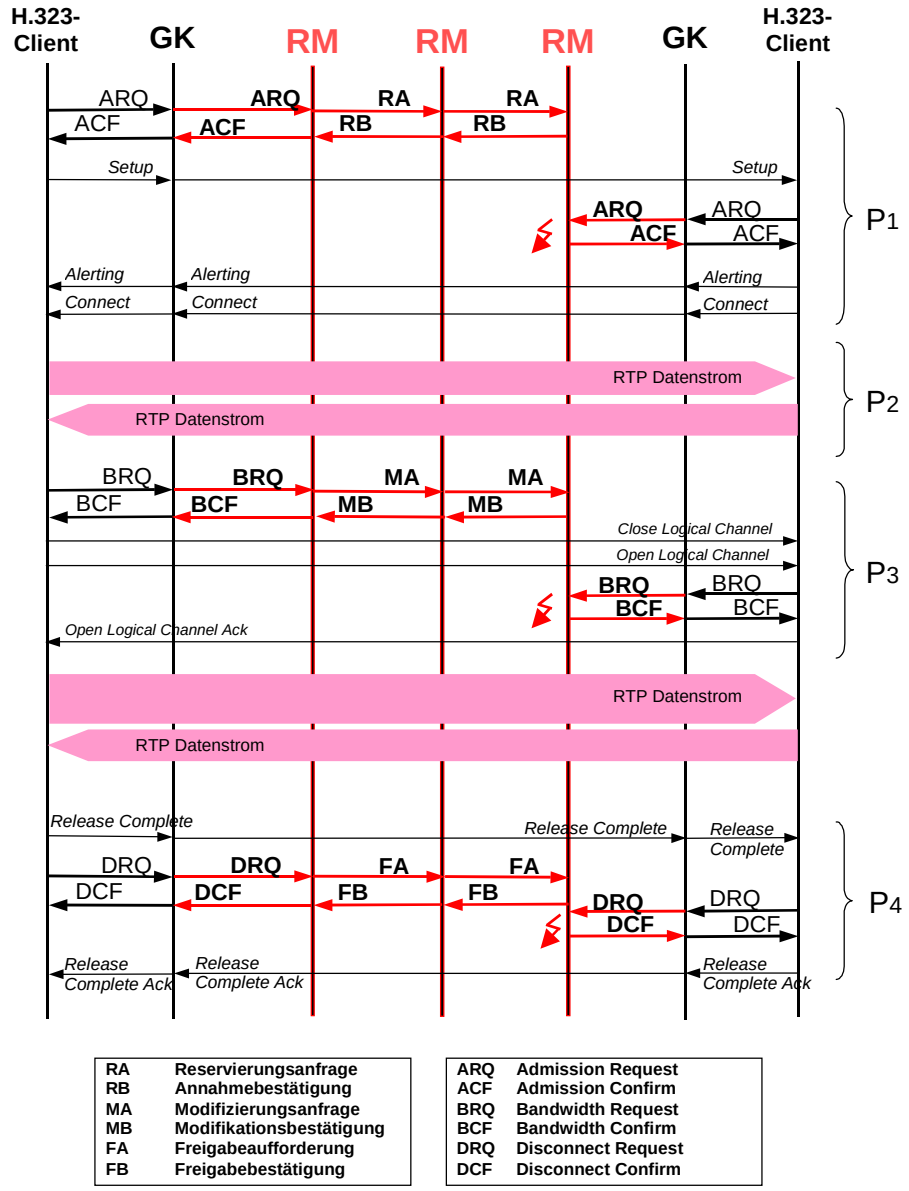


Abbildung 6-6: Signalablaufdiagramm für H.323 (*Variante: "GK-Routed", "Fast Connect"*)

6.5.3.2 SIP

Das im vorangegangenen Abschnitt vorgestellte Reservierungsverfahren ist prinzipiell auch auf andere Dienstarchitekturen wie z.B. SIP übertragbar. Abbildung 6-7 zeigt ein Signalisierungsbeispiel für eine mögliche Integration des RM-Systems in eine SIP-Umgebung.

Der *SIP Client*$_A$ sendet eine INVITE-Nachricht zu seinem *Proxy*$_1$. Sie enthält eine *Alias*-Adresse von *Client*$_B$, die schrittweise von *Proxy*$_1$, *Proxy*$_2$ und *Proxy*$_3$ in die zugehörige IP-Adresse aufgelöst wird. Die Signalisierung von *SIP Client*$_A$ wird in diesem Beispiel immer über die *SIP Proxys* zu *Client*$_B$ geführt. Der *SIP Proxy*$_3$ besitzt mit Erhalt der INVITE-Nachricht alle Verbindungsdaten und kann die Reservierung anstoßen. Das Weiterleiten der INVITE-Nachricht an den *Client*$_B$ wird verzögert, bis das RM-System die erfolgreiche Ressourcenreservierung bestätigt. Zudem muss der *SIP Proxy*$_3$ dahingehend erweitert werden,

dass er sich Zustandsinformationen der Reservierung merken kann. Ansonsten kann er einen Verbindungsabbau nicht mehr einer Reservierung zuordnen. In der Signalisierungsvariante *Record Route* werden alle Nachrichten einer Transaktion über die Proxy-Server gesendet. Dadurch ist *SIP Proxy*$_3$ über den Status der Sitzung informiert und kann den Dienst überwachen. Er kann im Fehlerfall anstelle von *Client*$_A$ oder *Client*$_B$ agieren und den Dienst tarifieren.

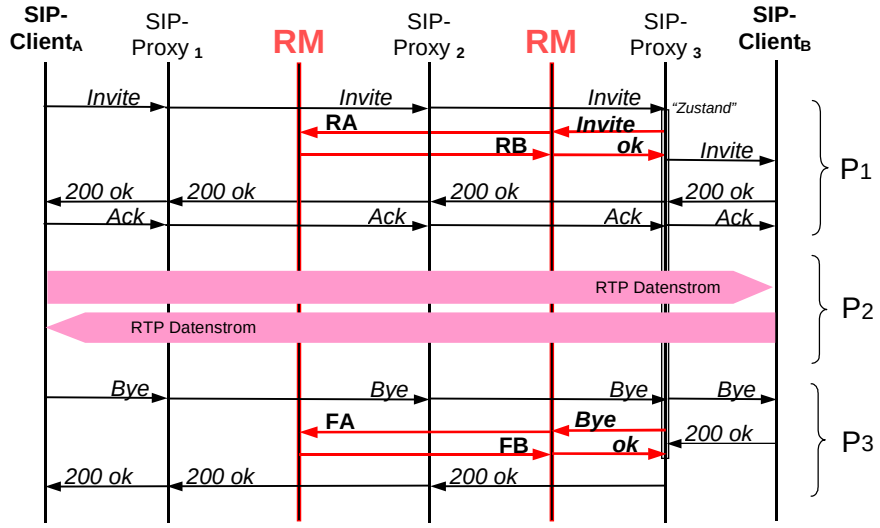


Abbildung 6-7: Signalablaufdiagramm für SIP (*Variante: „Record Route“*)

Das hier gezeigte Beispiel stellt eine Alternative zum IETF-Draft [Cam02] dar. In diesem ist jedes Endgerät (Quelle) für die Ressourcenreservierung verantwortlich. Die jeweiligen Reservierungen erfolgen dabei in Senderichtung und werden über ein erweitertes SIP-Protokoll koordiniert.

6.5.4 Anbindung an den TM

Für die Implementierung der RM-TM Schnittstelle wurde eine Client-Server Architektur gewählt (siehe Abbildung 6-12, Abschnitt 6.6.5). Der TM-Server übernimmt die Befragung und Überwachung der Netzknoten. Er legt eine Topologiedatenbank an, verwaltet diese und bietet den RM über ein Client-Server Protokoll selektiven Zugriff auf die Topologiedaten an.

Der Nachrichtenaustausch zwischen RM und TM erfolgt asynchron zur Ressourcenreservierung. Als Transportmechanismus wird UDP verwendet. Jede Anfrage des RM wird daher vom TM bestätigt und durch Timer überwacht.

Der RM versucht sich während seiner Initialisierung bei einem TM-Server zu registrieren. Dabei gibt er seine Domänengrenzen (Liste von IP-Subnetzen) und einen UDP-Port an, auf dem er auf Meldungen vom TM-Server hört. Liegen alle IP-Subnetze der RM-Domäne innerhalb der Grenzen der TM-Domäne, bestätigt der TM-Server die Registrierung und teilt dem TM-Client mit, wie er auf die Topologiedaten der Subnetze zugreifen kann (Liste von Subnetz-IDs). War die Registrierung erfolgreich, fordert der TM-Client umgehend die Topologiedaten zu allen Subnetzen seiner Domäne an. Für jedes Subnetz stellt er eine eigene Anfrage (*Topologie_Anfrage*) mit der jeweiligen Subnetz-ID. Diese wird vom TM-Server bestätigt (*Topologie_Bestätigung*). Danach baut der TM-Server eine TCP-Verbindung zum TM-Client auf, überträgt sequenziell alle Datenobjekte des Subnetzes und baut anschließend die TCP-Verbindung wieder ab.

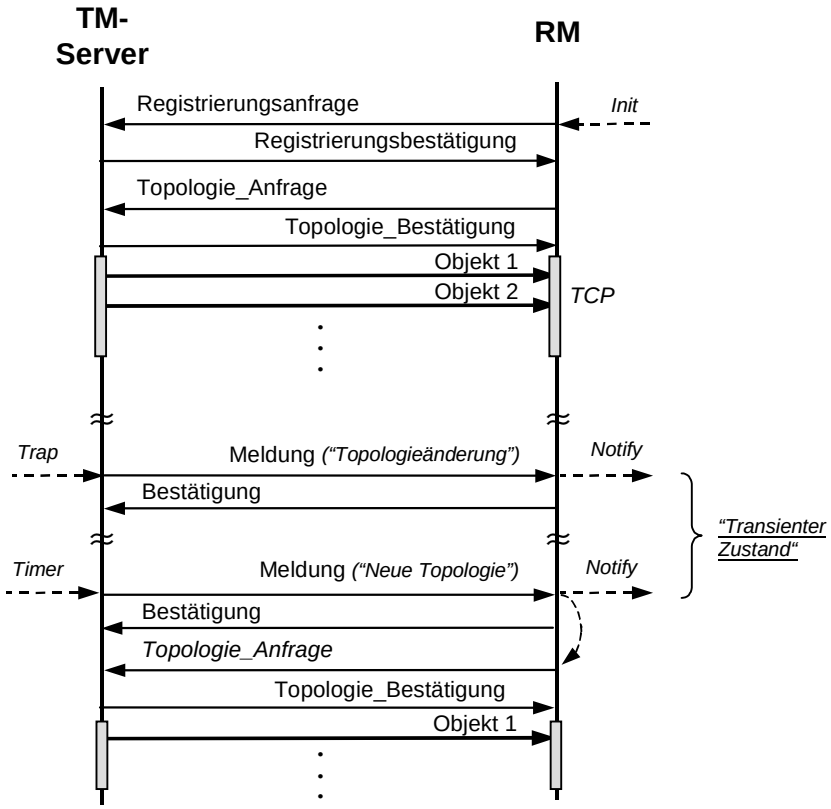


Abbildung 6-8: Signalablaufdiagramm TM-RM

Für die Ausgabe der Objektinhalte besitzt jedes Datenobjekt eine Ausgabemethode, mit der alle Konfigurationsdaten des Objektes seriell in einem Datenstrom ausgelesen werden können. Besteht die RM-Domäne aus mehreren Subnetzen, wird obige Prozedur entsprechend oft wiederholt.

Kommt es im Netz spontan zu Veränderungen, teilt dies der TM dem RM mit. Der RM wird ebenfalls vom TM informiert, wenn das Netz wieder einen stabilen Zustand erreicht hat und aktuelle Topologiedaten vorliegen. Die entsprechende Nachricht veranlasst den RM, die neuen Topologiedaten abzurufen.

6.6 Aufbau des Ressourcen-Managers

In diesem Abschnitt wird der schematische Aufbau eines RM gezeigt. Abbildung 6-9 gibt einen Einblick in die innere Struktur. Dabei sind die einzelnen Funktionsblöcke und deren Interaktionen dargestellt. Bei den Interaktionen werden drei Vorgänge unterschieden: der Initialisierungsvorgang des Systems (dünne Pfeile), der Reservierungsvorgang (dicke Pfeile) und der Re-Konfigurationsvorgang bei Änderungen der Topologie (gestrichelte Pfeile).

Die Initialisierung eines RM wird von mehreren Seiten vorgenommen. Über eine Konfigurationsschnittstelle erhält der RM die erforderlichen Daten hinsichtlich seiner Domänengrenzen und der Spezifikationen der RM-Dienstklassen des RM-Systems (siehe Absatz 6.4, Tabelle 6-3). Die Konfigurationsdaten der Verkehrsklassen (IP, MAC) bezieht der RM zusammen mit den anderen Topologiedaten vom TM.

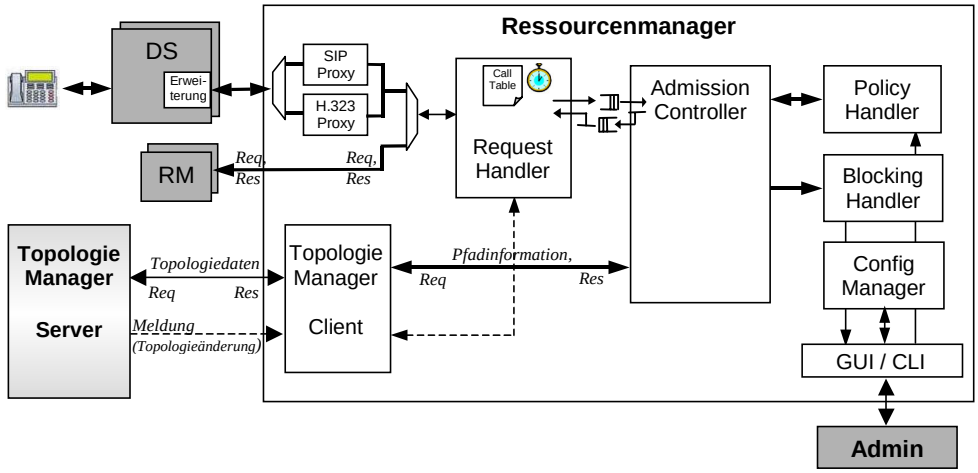


Abbildung 6-9: Aufbau eines Ressourcen-Managers

Sobald das RM-System initialisiert ist, kann es Reservierungsanfragen entgegennehmen und bearbeiten. Die Teilnehmer sind über ihre DS an das RM-System angebunden. Die DS werden daher mit einer Schnittstelle zum RM-System versehen. Die Reservierungsanfragen einer DS werden von einem *Request Handler* entgegen genommen. Dieser arbeitet sequenziell alle Reservierungsanfragen ab und koordiniert alle zur Zeit in Bearbeitung befindlichen Reservierungen des Systems. Soll beispielsweise eine neue Reservierung etabliert werden, stößt der *Request Handler* eine Zugangskontrollprozedur im *Admission Controller* an und wartet auf eine Reaktion. Der *Admission Controller* holt sich vom *Topologie-Manager Client* alle relevanten Informationen bezüglich des Datenpfades der zu tätigenden Reservierung, führt auf der Basis dieser Pfadinformationen und der Verbindungsparameter aus der Reservierungsanfrage eine Zugangskontrolle durch und teilt dem *Request Handler* das Ergebnis der Zugangskontrolle mit.

Während des laufenden Betriebes eines RM-Systems kann es vorkommen, dass sich die Topologie des Netzes durch Knoten- oder Linkausfälle spontan verändert. Ändern sich die Routen im Netz, muss das RM-System sofort informiert werden. Diese Funktion übernimmt der TM-Server, der dem TM-Client eine entsprechende Meldung sendet (siehe Abschnitt 6.5.4) und den RM dadurch in einen Ausnahmezustand versetzt. Der TM-Client informiert den *Request Handler*, der in dieser Ausnahmesituation den Betriebszustand ändert und alle Gegenmaßnahmen des RM steuert.

Im Folgenden werden die einzelnen Funktionsblöcke aus Abbildung 6-9 genauer beschrieben.

6.6.1 DS-Proxy

Der DS-Proxy stellt die Schnittstelle zwischen der Dienststeuerung und dem RM-System dar.

Beim Dienstaufruf durch den Teilnehmer muss die DS das RM-System über den neuen Verbindungswunsch informieren und eine Reservierung anstoßen. Um die Modifikation der DS so gering wie möglich zu halten, sieht das RM-Konzept vor, dass die DS die Nachricht des Teilnehmers an das RM-System weiterleitet und auf eine Antwort des RM-Systems wartet. Dabei wird vom RM die Signalisierung der Dienststeuerung auf die Signalisierung des Ressourcenmanagements umgesetzt. Die Nachricht der DS wird dazu an einen DS-Proxy weitergeleitet. Im RM-System sind je nach der vorhandenen Systemumgebung verschiedene DS-Proxies vorgesehen: z.B. H.323-Proxy, SIP-Proxy. Dadurch wird eine Unabhängigkeit des RM-Systems von einer bestimmten Dienstarchitektur erreicht. Ein RM kann um einen

neuen DS-Proxy erweitert und dadurch flexibel an andere Dienststeuerungssysteme angebunden werden.

Im DS-Proxy findet die notwendige **Protokollumsetzung** statt. Zu den allgemeinen Aufgaben des DS-Proxy gehören, dass er für alle Verbindungen einer Multimediasitzung ein CONNECTION-Objekt anlegt (siehe Absatz 6.5.2). Dabei definiert er den TRANSPORT-QoS Parameter, d.h. die von der Dienststeuerung geforderten Ende-zu-Ende netzspezifischen QoS-Parameter sowie die Art der Dienstgütegarantie.

Die Umsetzung erfordert zunächst eine Bestimmung der RM-Dienstklasse. Bei SIP und den H.323 Versionen 1-4 erfolgt diese anhand der signalisierten Portnummern und der Angaben zum Codec (*Payload-Type*) erfolgt. Dazu ermittelt der DS-Proxy die Dienstklassenkennung sowie die netzspezifischen QoS-Parameter (QoS-Spec) der Dienstklassenspezifikation (siehe Tabelle 6-3). Zusätzlich ist eine Abbildung von applikationsspezifischen Verbindungsparametern auf die Verkehrsbeschreibung (TB-Parameter) erforderlich. Wenn die TB-Parameter nicht vom Teilnehmer signalisiert werden, muss der RM die TB-Parameter der jeweiligen Applikationen des Intranets kennen. Dazu müssen die Verkehre gemessen, charakterisiert und deren Verkehrsprofile z.B. in einer Tabelle abgelegt werden (Profil-Tabelle). Ein *Offline*-Verfahren wurde in Abschnitt 4.4 und 5.4 beschrieben. Prinzipiell ist auch denkbar, dass im Zugangsbereich des Netzes Messsysteme installiert sind, die *online* messen, charakterisieren und automatisch für die Aktualisierung der Profil-Tabelle in den RM sorgen.

Im speziellen Fall eines H.323-Systems, welches noch nicht auf den Erweiterungen des Annex.N basiert, ergänzt der RM-Proxy im GK die grobe Verkehrsbeschreibung der RAS-Signalisierung. Dabei wird der Parameter *Bandwidth Aggregated* über eine vorkonfigurierte Umsetzungstabelle durch eine Sequenz von TB-Parametern ersetzt.

In zukünftigen H.323-Versionen, welche den Annex.N unterstützen, wird schon in der RAS-Signalisierung eine H.323-QoS Dienstklasse spezifiziert. Daneben wird eine genaue Verkehrsbeschreibung *BitRateClass* und eine QoS-Beschreibung *delayErrorClass* angegeben. Aus diesen Parametern können alle Werte des CONNECTION-Objekts bestimmt werden.

Damit ein Teilnehmer Vorabreservierungen, d.h. Reservierungen tätigen kann, die erst zu einem späteren Zeitpunkt aktiv werden sollen, muss die Reservierungsanfrage einen Parameter über den Zeitpunkt und die Dauer der Vorabreservierung enthalten (z.B. SIP - SDP). Unterstützt eine Dienststeuerung diese Funktionalität nicht (z.B. H.323), werden diese Parameter der RM-Signalisierung vom DS-Proxy auf Initialwerte gesetzt.

6.6.2 Config Manager

Über den *Config Manager* wird der RM konfiguriert. Über ihn kann man einstellen, welche DS-Klassen im Netz vom RM verwaltet werden, welche applikationsspezifischen Dienstklassen (TIPHON, H.323-Annex.N) auf welche DS-Klassen abgebildet werden und welches *Admission Control* (AC) Verfahren dabei eingesetzt werden soll. Ferner muss dem RM mitgeteilt werden, wie groß die von ihm zu verwaltende Netzdomäne ist (Liste von Subnetzen). Diese Einstellungen können über eine Telnetverbindung oder eine Konfigurationsdatei vorgenommen werden.
Der *Config Manager* ist für die korrekte Initialisierung des RM verantwortlich. Er hat dafür zu sorgen, dass sich der RM in seinem Umfeld anmeldet. Dazu gehört, dass er sich selbst bei einem Topologie-Manager (*TM Discovery*) registriert, alle DS in seiner RM-Domäne von seiner Anwesenheit informiert (*DS Discovery*) und diese auffordert, sich bei ihm zu registrieren. Außerdem muss er andere RM im Netz suchen und sich bei seinen unmittelbaren

Domänennachbarn (*RM Neighbourhood Discovery*) anmelden. Die Anmeldung eines RM bei seinen Nachbarn beinhaltet den Aufbau von Routingtabellen auf RM-Ebene. Diese Tabellen werden benötigt, um Reservierungsanfragen über mehrere RM hinweg weiterleiten zu können. Sie tauschen dabei Informationen aus, welche IP-Zielnetze über welchen Nachbar-RM erreichbar sind. Um die Tabellen möglichst klein zu halten, ist auf eine geeignete Aggregation der Routing-Informationen zu achten (vgl. BGP).

6.6.3 Request Handler

Der *Request Handler* stellt die zentrale Koordinationsstelle im RM dar. Er ist zum einen für die Verwaltung der Zustandsinformationen einer Multimediasitzung und zum anderen für die Kommunikation des RM mit den DS und Nachbar-RM zuständig. Der *Request Handler* hat Kenntnis vom Zustand aller bereits aktiven Reservierungen im RM. Darüber hinaus besitzt er ein Zeitmanagement zum Überwachen der Zustände (Fehlerbehandlung) und Aktivieren von Vorabreservierungen.

Ein einzelner RM kann eine Reservierungsanfrage von einer DS (z.B. H.323: ARQ, BRQ, DRQ) oder einem Nachbar-RM (RA, MA, FA) empfangen. Jede Transaktion kann nur dann erfolgreich abgeschlossen werden, wenn alle beteiligten RM-Instanzen den Vorgang bestätigen. Eine Transaktion wird durch den *Request Handler* überwacht. Dazu vergibt dieser Reservierungszustände nach dem Zustandsautomaten aus Abbildung 6-10, kontrolliert die Verweildauer in den jeweiligen Zuständen durch *Timer* und sendet gegebenenfalls Nachrichten erneut aus.

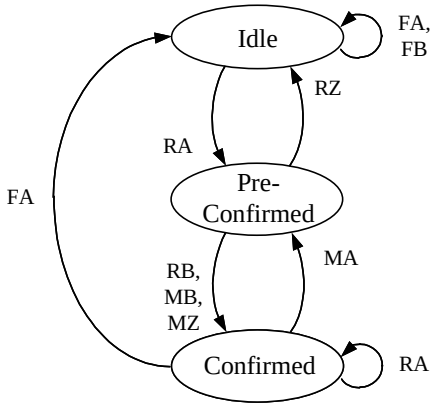


Abbildung 6-10: Zustandsautomat einer Reservierung am Beispiel des H.323

In dem Zustandsmodell gibt es drei Zustände, je nachdem ob sich eine Transaktion in Phase 1 (*pre confirmed*) oder in Phase 2 (*confirmed*) befindet. Sie werden mit Hilfe von *Timern* ständig auf ihre Gültigkeit hin überwacht. Im Fehlerfall, d.h. bei Ausfall eines Gatekeepers oder Nachbar-RM, würden Reservierungen nicht mehr freigegeben. Bei Ablauf eines Timers wird er erneut gesetzt und eine Statusabfrage gesendet. Läuft der Timer ein zweites Mal ab, gibt der RM die belegten Ressourcen frei.

Zum besseren Verständnis wird die Arbeitsweise des *Request Handler* genauer beschrieben. Der *Request Handler* liest aus einem Empfangspuffer Reservierungsnachrichten aus, die an den RM gestellt werden. Für jede neue Multimediasitzung legt er zunächst einen Eintrag in einer *Call Table* an. Andernfalls kann er eine Reservierungsnachricht einer bestehenden Reservierung zuordnen.

Ein RM ist für mehrere DS zuständig und kann mehrere Nachbar-RM besitzen. Dadurch können bei einem *Request Handler* viele Reservierungsanfragen gleichzeitig ankommen.

Manche dieser Nachrichten sind redundant (z.B. H.323: symmetrische Signalisierung) und können sofort beantwortet werden. Bei anderen handelt es sich um Bestätigungen (RB, MB, FB, SB), die vom *Request Handler* selbst bearbeitet werden können. Die übrigen Nachrichten (RA, MA) müssen an den *Admission Controller* weitergeleitet werden. Der *Request Handler* stößt dazu einen Reservierungsprozess im *Admission Controller* an. Dies geschieht normalerweise nachdem die Nachricht empfangen und ein Eintrag in der Call Table angelegt wurde. Im Fall einer Vorabreservierung kann die Reservierung auch zu einem späteren Zeitpunkt durchgeführt werden, der im *ACTIVITY-Object* angegeben ist. Ist der Reservierungsvorgang angestoßen, kann der *Request Handler* solange weitere Nachrichten bearbeiten. Liegt das Ergebnis der Zugangskontrolle vor, kann er die Reservierungsanfrage abschließend beantworten.

Um die Nebenläufigkeit dieser Vorgänge auszunutzen und die Reaktionszeit eines RM so gering wie möglich zu machen, wurden der *Request Handler* und der *Admission Controller* als zwei Prozesse realisiert. Beide Prozesse kommunizieren miteinander asynchron über zwei Warteschlangen, die nach dem FIFO-Prinzip bedient werden. Stößt ein *Request Handler* einen Zugangskontrollprozess an, schreibt er eine Meldung in eine Warteschlange und kann danach sofort wieder neu eingegangene Reservierungsnachrichten bearbeiten. Parallel dazu liest der *Admission Controller* Meldungen, führt die Zugangskontrolle durch und liefert ein Ergebnis zurück. Das Ergebnis schreibt er in eine andere Warteschlange, die vom *Request Handler* bedient wird. Das Ergebnis enthält immer eine Entscheidung (ja/nein) und optional die Kennung eines Grenz-Links. Liegt dem *Admission Controller* ein Ergebnis vor, aktualisiert der *Request Handler* den Reservierungszustand. Wird ein Grenz-Link als Rückgabeparameter mitgeliefert, ermittelt er den Nachbar-RM und sendet diesem eine Reservierungsanfrage (FA, MA) mit dem Kontext der Sitzung. Im anderen Fall beantwortet er die Reservierungsanfrage, indem er an die Instanz, von der die Reservierungsanfrage kam (DS, Nachbar-RM), eine FB-/MB-Nachricht sendet.

Topologieänderungen

Bei Topologieänderungen werden die bis zu diesem Zeitpunkt vom RM verwendeten Topologiedaten (teilweise) ungültig. Topologieänderungen werden vom TM-Server erkannt und dem RM gemeldet. Der *Request Handler* wechselt daraufhin seinen Betriebszustand solange, bis das Netz wieder einen stabilen Zustand erreicht hat, neue Topologiedaten vorliegen und alle noch aktiven Reservierungen auf die neue Topologie umgelegt wurden. Das Zustandsdiagramm ist in Abbildung 6-11 dargestellt.

Ein *Request Handler* sieht dabei zwei Strategien vor, wie der RM während des transienten Zustandes mit neuen Reservierungsanfragen der Teilnehmer umgehen kann [TGM03], [Gla02]. Im optimistischen Fall bearbeitet er alle neuen Reservierungsanfragen auf der Basis der alten Topologiedaten, im pessimistischen Fall blockiert er alle Anfragen. Die Freigabe von bereits reservierten Ressourcen wird in beiden Fällen zugelassen.

Liegen die neuen Topologiedaten vor, überträgt der *Request Handler* alle gültigen Reservierungen aus den alten Topologieobjekten in die Neuen. Er verwendet dazu die *Call Table* und stößt für jeden Eintrag eine Reservierung an.

Treten in der neuen Topologie kapazitive Engpässe auf, so kann es sein, dass nicht alle alten Reservierungen übertragen werden können. In diesem Fall beendet der *Request Handler* einzelne Sitzungen, indem er die Freigabe der Ressourcen veranlasst und die DS darüber informiert. Die DS ist dann für die Beendigung der Sitzung verantwortlich.

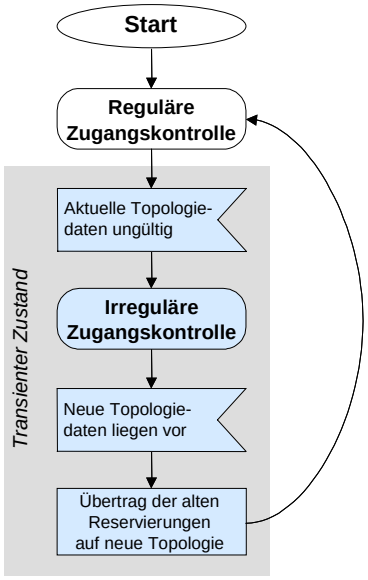


Abbildung 6-11: Zustandsdiagramm des Request Handlers

Der *Request Handler* besitzt die Möglichkeit, alle alten Sitzungen nach einem bestimmten Kriterium zu sortieren, bevor er mit der Übertragung auf die neue Topologie beginnt. Dadurch kann er die Blockierungswahrscheinlichkeiten für einzelne Gruppen von Sitzungen erhöhen oder reduzieren. Er kann dabei verschiedene Strategien verwenden:

A) FIFO-Prinzip:
Reihenfolge nach dem Eingangszeitpunkt der Reservierungsanfrage.
B) Maximale Auslastung:
Reihenfolge so, dass möglichst viele Verbindungen zugelassen werden können.
C) Priorisierung einzelner Gruppen von Sitzungen.

Beispiele, wie solche Strategien umgesetzt werden können, sind unter [TGM03] zu finden.

6.6.4 Admission Controller

Der *Request Handler* stößt eine Reservierung an, indem er dem *Admission Controller* das ein Res-Objekt übergibt.

Der *Admission Controller* arbeitet die Res-Objekte sequenziell ab. Er stößt dazu als erstes eine Zugangskontrolle im *Policy Handler* an. Der *Policy Handler* überprüft anhand der *Operator-ID*, ob die DS (z.B. Gatekeeper), über den die Anfrage an den RM gelangt ist, überhaupt registriert ist und eine Zugangsberechtigung zum RM-System besitzt. Danach stellt der *Admission Controller* für jedes CONNECTION-Objekt eine Anfrage an den TM-Client. Die Anfrage enthält die Parameter *Source-Address*, *Dest-Address* und den *DiffServ Codepoint*. Der TM-Client liefert eine Sequenz von Referenzen auf Topologieobjekte, in denen die entsprechenden Konfigurationsdaten (Kapazität des Ausgangsports, Puffergröße und reservierte Bedienrate) gespeichert sind. Mit jedem Topologieobjekt (Knotenausgang) ist ein Auslastungszustand assoziiert. Die Auslastungszustände beinhalten eine Liste von TB-Parametern all derjenigen RTP-Datenströme, die bereits zugelassen sind und die ebenfalls über diesen Ausgangsport übertragen werden.

Aus den Topologieobjekten kann der *Admission Controller* die darin enthaltenen Pfadinformationen auslesen und zusammen mit den Verkehrsparametern *Traffic-Spec* eine der Dienstklasse entsprechende AC-Funktion aufrufen. Die CAC-Funktion liefert für jeden Link des Pfades eine positive oder negative Rückmeldung. Der *Admission Controller* koordiniert

alle AC-Funktionsaufrufe, die zu einem Reservierungspfad (*CONNECTION-Object*) gehören. Nur wenn alle Rückmeldungen eines Reservierungspfades positiv waren, ist die Reservierung für ein *CONNECTION-Object* erfolgreich abgeschlossen.

Enthält das *Res-Object* einer Transaktion mehrere *CONNECTION-Objects*, wird obige Prozedur nacheinander für alle *CONNECTION-Objects* wiederholt. Der *Admission Controller* legt in diesem Fall zu Beginn die Reihenfolge der Abarbeitung fest. Da eine Transaktion nur dann erfolgreich ist, wenn für alle *CONNECTION-Objects* eine Reservierung durchgeführt werden konnte, kann die mittlere Reaktionszeit des RM-Systems reduziert werden, wenn RTP-Verbindungen mit großem Kapazitätsbearf (hoher Blockierungswahrscheinlichkeit) zuerst bearbeitet werden. Dabei werden die *Traffic-Spec* Parameter der *CONNECTION-Objects* herangezogen und die RTP-Quellen mit dem höchsten R-Wert (*Token Fill-Rate*) und dann gegebenenfalls mit dem höchsten B-Wert (*Bucket Size*) zuerst bearbeitet.

Konnte obige Prozedur für alle *CONNECTION-Objects* des *Res-Objects* erfolgreich durchgeführt werden, aktualisiert der *Admission Controller* die Lastzustände der beteiligten Topologieobjekte und teilt dem *Request Handler* das Ergebnis mit. Kommt es dagegen auch nur zu einer einzigen Blockierung auf einem beliebigen Link, wird die gesamte Reservierung abgebrochen und ein Eintrag in einem *Logfile* vorgenommen (Netzknoten, Ausgangsport, DS-Klasse, Datum, Uhrzeit, IP-Adressen der Quelle und Senke, Dienstserver-ID). Wird eine Blockierung von einem *Request Handler* festgestellt, filtert er seinen Empfangspuffer nach FA-Nachrichten zur Freigabe bereits existierender Reservierungen. FA-Nachrichten werden dann in der Folgezeit bevorzugt bearbeitet.

6.6.5 Topologie-Manager Client

Der TM-Client ist an einen TM-Server angebunden und hat auf dessen Topologiedatenbank selektiven Zugriff (siehe Abschnitt 6.5.4). Er fordert die Topologieinformationen seiner RM-Domäne bei Bedarf vom TM-Server an (*Topologie_Anfrage*). Ist er zugriffsberechtigt, sendet der TM-Server dem TM-Client eine Kopie der entsprechenden Topologiedatenobjekte zu (*Topologie_Bestätigung*). Eine einzelne Anfrage des TM-Client bezieht sich auf ein Subnetz seiner RM-Domäne. Die Ausgabe der Topologiedaten erfolgt sequenziell für alle Datenobjekte des Subnetzes. Jedes Datenobjekt besitzt eine Ausgabemethode, mit der alle Konfigurationsdaten des Objektes seriell in einem Datenstrom ausgelesen werden können. Besteht die RM-Domäne aus mehreren Subnetzen, wird obige Prozedur entsprechend oft wiederholt. Der TM-Client kopiert die Datenobjekte in seinen Arbeitspeicher und bietet dem RM ein API.

Auf die RM-intere Anfrage des *Admission Controllers* ermittelt der TM-Client alle für die Reservierung relevanten Pfadinformationen. Dazu gibt er dem *Admission Controller* eine Liste von Referenzen auf die entsprechenden Datenobjekte. Der TM-Client bietet dazu eine Suchfunktion von Knotenobjekten an. Ausgehend von einer Quelladresse ermittelt er alle Knotenobjekte, die auf dem Pfad zur Zieladresse liegen. Über die Referenzen kann der *Admission Controller* direkt auf die Datenobjekte zugreifen und alle darin gespeicherten Informationen auslesen.

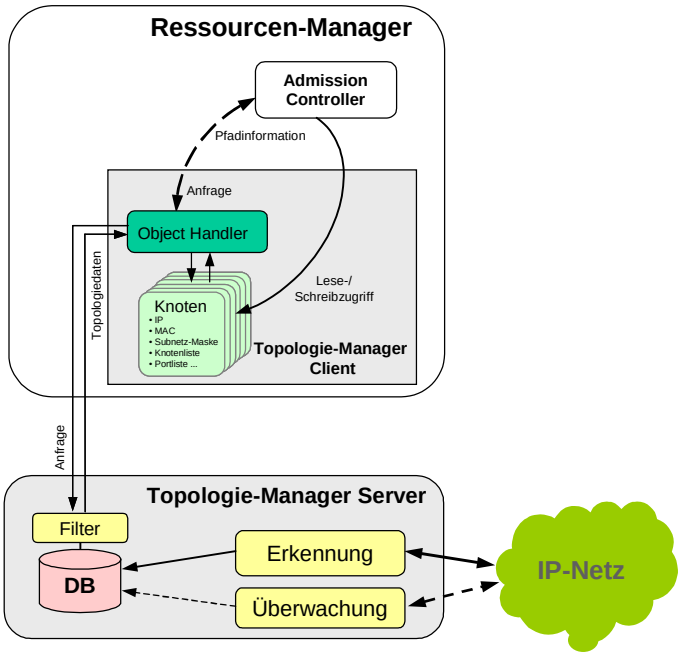


Abbildung 6-12: Client-Server Architektur

Zu den Funktionen des TM-Clients gehört, dass er anhand der MAC- und IP- Quell- bzw. Zieladressen der Endgeräte, zwischen denen die Reservierung aufgebaut werden soll, die zugehörigen Knotenobjekte des Ende-zu-Ende Pfades findet. Der TM-Client greift dazu beim Systemstart über eine externe Schnittstelle auf die Datenbank eines Topologiemanager-Servers zu und kopiert alle Topologiedaten seiner RM-Domäne.

Der Suchalgorithmus soll nun kurz erklärt werden. Er basiert auf der Tatsache, dass zwischen zwei beliebigen Knoten der hier betrachteten Netze (siehe Abschnitt 3.2.1) nur ein einziger Verbindungsweg existiert. Die OSI Schicht-2 Topologie der Subnetze ist meist sternförmig und kann als eine Baumstruktur dargestellt werden. Ähnliches gilt für die OSI Schicht-3 Topologie bei Verwendung von Verkehrslenkungsverfahren (*Single Path Routing*), welche zwischen den Knoten eindeutige Wege einrichten. Bei der Datenaufbereitung werden die Topologieobjekte vom TM-Server einander so zugeordnet, dass sie eine baumförmige Struktur bilden. Dabei werden die IP-Forwarding- und Bridging-Tabellen ausgewertet und in den Knotenobjekten auf entsprechende Referenzen umgesetzt. Für die Suche auf Schicht-3 wird ein *Common-Prefix* Baum erzeugt, der ein schnelles Auffinden der Subnetze ermöglicht. Für die Wegesuche innerhalb eines Subnetzes wird pro Subnetz eine n x n Bit-Matrix angelegt, in der die Verknüpfung der n-Knoten des Baumes angegeben ist. Für die Umsetzung der Knotenadressen in Objektkennungen werden *Hash-Tables* eingerichtet.

Der TM-Client muss die Datenstrukturen bei der Pfadsuche anwenden. Zunächst bestimmt er das Subnetz und die Objektkennung des Quellknotens. Es beginnt eine Suche innerhalb des Subnetzes entweder bis zum Zielknoten oder bis zum Router (Subnetzgrenze). Liegt der Zielknoten in einem anderen Subnetz, endet die Suche auf Schicht-2 an einem Routerobjekt. Danach wird anhand des *Common-Prefix* Baumes das nächste Subnetz auf dem Weg in Richtung des Zielsubnetzes ermittelt und die Wegesuche im nächsten Subnetz fortgesetzt. Der Prozess zwischen der abwechselnden Suche auf Schicht-3 und Schicht-2 wird solange wiederholt, bis der Zielknoten erreicht wurde. Alle bereits gefundenen Referenzen werden dem *Admission Controller* mitgeteilt. Das Auslesen der Objektinhalte sowie das Aktualisieren der Lastzustände ist Aufgabe des *Admission Controllers*.

6.7 Aufbau des Topologie-Managers

In diesem Abschnitt wird der schematische Aufbau eines TM-Servers gezeigt. Die RM-Architektur sieht vor, dass RM- und TM-Instanzen unabhängig voneinander realisiert werden. Daher können sie auf unterschiedlichen Systemen laufen und über ein Transaktionsprotokoll miteinander kommunizieren.

Der TM besitzt zwei voneinander unabhängige Aufgaben. Die eine besteht darin, ein Abbild der Netztopologie in Form eines Datenmodells zu erzeugen. Sie wird im Folgenden mit dem Begriff „Netzerkennung“ bezeichnet. Die andere Aufgabe ist das Erkennen und Überwachen von Topologieänderungen und wird im Folgenden „Netzüberwachung“ genannt.

Der Prozess der Netzerkennung wird bei der Initialisierung eines TM in regelmäßigen Abständen oder asynchron durch den Prozess der Netzüberwachung angestoßen. Die Netzüberwachung erfolgt parallel zur Netzerkennung. Bei Topologieänderungen informieren die Netzknoten den TM-Server. Die Überwachungseinheit versucht dann den Zeitpunkt zu bestimmen, zu dem sich der Netzzustand wieder stabilisiert hat. Ist dieser Zeitpunkt erreicht, stößt die Überwachungseinheit einen Netzerkennungsprozess an. Die einzelnen Aufgaben werden als unabhängige Prozesse realisiert, die von einer Ablaufsteuerung koordiniert werden. Daraus ergibt sich der schematische Aufbau eines TM-Servers, der in nachfolgender Abbildung dargestellt ist.

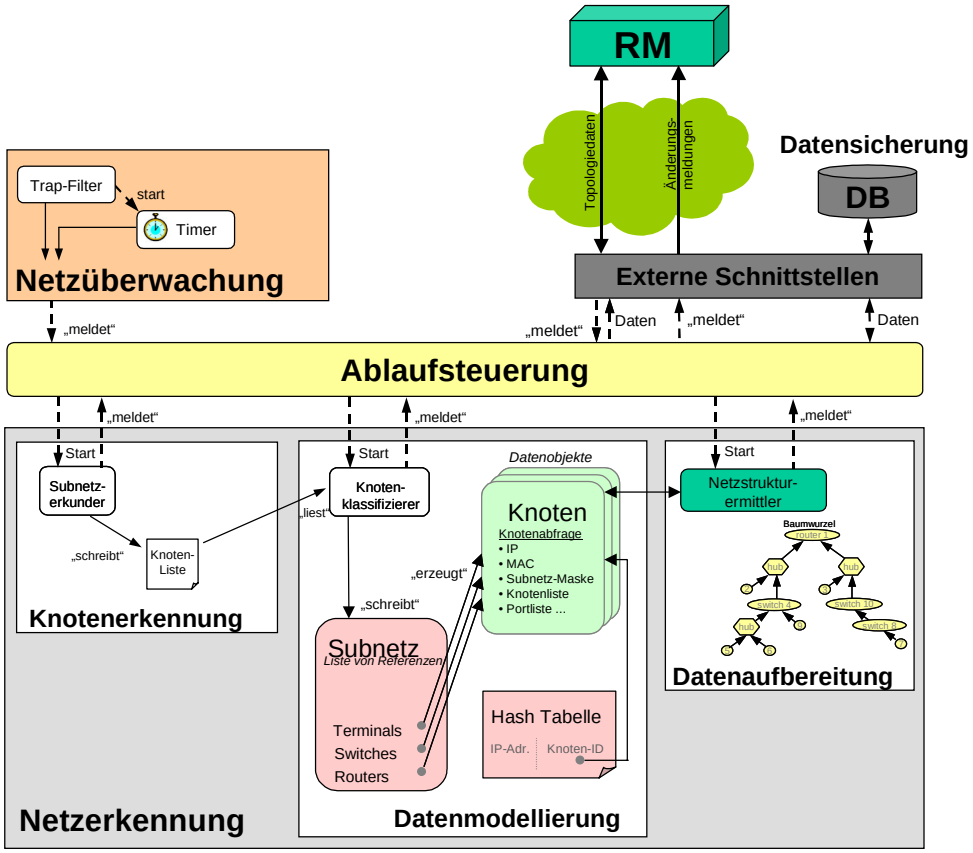


Abbildung 6-13: Schematischer Aufbau des Topologie-Managers

6.7.1 Netzerkennung

Die Größe des zu erkundenden Netzes wird dem TM von Seiten der Administration durch eine Liste von Subnetzadressen vorgegeben. Der Prozess der Netzerkennung lässt sich in mehrere Teilprozesse unterteilen:

- Erkennung aller aktiven Netzknoten (Knotenerkennung).
- Klassifizierung der Netzknoten, gezieltes Abfragen der Knotenkonfiguration und Speicherung der Daten in einem objektorientierten Datenmodell (Datenmodellierung).
- Ermittelung der Objektbeziehungen aus den einzelnen Knotenkonfigurationen und Verknüpfung der Datenobjekte (Datenaufbereitung).

Bei der Netzerkennung greift der TM direkt auf alle Netzelemente der Subnetze zu. Als Zugriffsmechanismus verwendet er das SNMP-Protokoll [LMS02]. Mit Hilfe des SNMP-Protokolls kann der TM mit den SNMP-Agenten der einzelnen Geräte kommunizieren und dabei wie ein zentraler SNMP-Manager agieren. SNMP-Agenten verwalten die lokal in einem Gerät gespeicherte Managementinformation und stellen die Schnittstelle eines managementfähigen Gerätes zu einer zentralen Netzmanagementinstanz (SNMP-Manager) dar. Die Managementinformation eines Gerätes liegt in Form eines objektorientierten Datenmodells, den sogenannten *Managed Objects* (*MO),* vor. Bei den MO handelt es sich um eindimensionale Datenfelder ohne spezielle Zugriffsmethoden. Die Menge aller MO bildet zusammen die *Management Information Base* MIB [CR90], welche den aktuellen Zustand sowie die Historie eines Gerätes beschreibt. Die Bezeichnung MIB wird zugleich auch für die semantische und syntaktische Definition der MO verwendet. Die Beschreibung der Daten erfolgt gemäß einer modifizierten Teilmenge des OSI-Standards ASN.1 (*Abstract Syntax Number One*). Das Datenformat ist in [CPS99] (SMI: *Structured Management Information*) definiert. Die MO einer MIB sind in einer Baumstruktur angeordnet. Der für das RM-System relevante Zweig dieses Baumes ist der Unterbaum der mib-2 mit der Referenz: mib-2 OBJECT IDENTIFIER ::= { iso(1) org(3) dod(6) internet(1) mgmt(2) 1 }. Alle hier verwendeten Objekte beziehen sich auf diesen Unterbaum.
Das SNMP-Protokoll erlaubt dem TM eine synchrone und eine asynchrone Kommunikation mit den SNMP-Agenten. Bei der synchronen initiiert der TM die Kommunikation mit dem Aufruf entsprechender **GetRequest**, **GetNextRequest** oder **SetRequest** Methoden. Der SNMP-Agent beantwortet jede dieser Anfragen mit einer **GetResponse** Meldung, welche die vom TM angeforderten Daten oder eine Quittung für ein erfolgreiches Setzen eines Datenfeldes enthält.

6.7.1.1 Knotenerkennung und Datenmodellierung

Die Knotenerkennung und die Datenmodellierung werden in zwei unabhängigen Prozessen realisiert, die über einen Stack von erkannten IP-Adressen koordiniert werden. Der Ablauf ist vereinfacht in Abbildung 6-14 dargestellt und besteht im Wesentlichen aus drei aufeinanderfolgenden Phasen:

- Erkundung aller aktiven Geräte im Netz.
- Klassifizierung der Geräte nach Typen.
- Abfragen der gerätespezifischen Konfigurationsdaten.

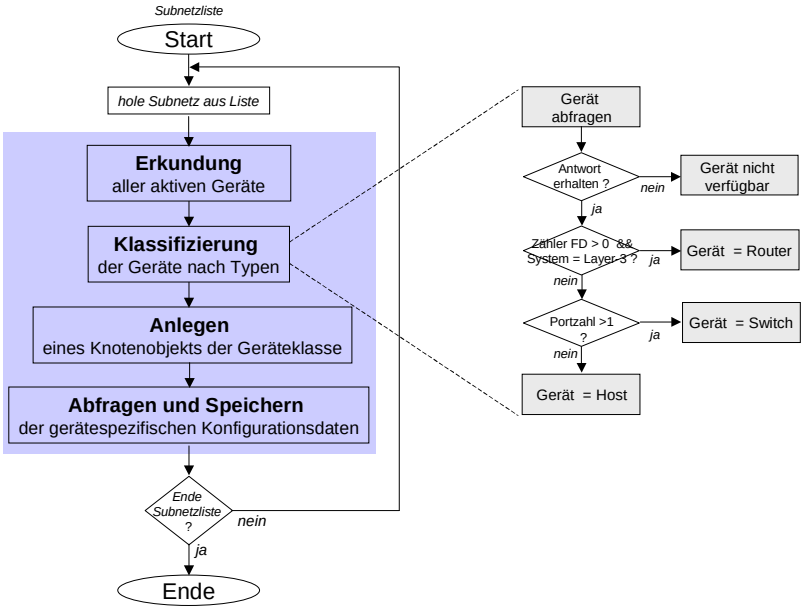


Abbildung 6-14: Topologieerkennungsprozess (Ablaufdiagramm)

Die Erkundung nach aktiven Geräten erfolgt durch Abfragen aller IP-Adressen in einem vorgegebenen Netzbereich (z.B. Subnetz). Auf die Anfrage mit einer ICMP-Nachricht (*Internet Control Management Protocol*) **echoRequest** (*ping*) antworten alle eingeschalteten Geräte innerhalb weniger Millisekunden mit einem **echoReply**. Antwortet ein Gerät, wird seine IP-Adresse in den Stack gelegt. Erhält der TM keine Antwort, kann es entweder sein, dass die IP-Adresse in dem Subnetz nicht vergeben wurde, dass das Gerät gerade abgeschaltet ist, dass es sich um ein nicht SNMP-fähiges Gerät handelt oder dass die SNMP-Nachricht (UDP-Paket) verloren gegangen ist. Alle Fälle führen zu Verzögerungen bei der Topologieerkennung. Die ersten drei Fälle können vom TM weder unterschieden noch vermieden werden. Um Verzögerungen aufgrund von Paketverlusten zu reduzieren, werden alle IP-Adressen zweimal befragt. Knoten, die bereits geantwortet haben, werden dabei übersprungen. Die Existenz von HUBs kann nachträglich vom TM rekonstruiert werden (Datenaufbereitung).

Der Prozess der Datenmodellierung arbeitet nach und nach alle Adressen des Stacks ab und versucht, jedes Gerät in Terminals, Switches und Router zu klassifizieren. Diese Unterscheidung ist für die nachfolgende Ermittlung der spezifischen Konfigurationsdaten der Geräte wichtig. Dazu greift der TM zum zweiten Mal auf die Netzknoten zu und holt sich die für die Klassifizierung notwendigen Informationen. Aufgrund der Tatsache, dass verschiedene Hersteller (Cisco, Enterasys, Extreme Networks und 3Com) die Inhalte der MO unterschiedlich setzen, müssen mehrere Parameter für eine eindeutige Klassifizierung der Geräte herangezogen werden. Mit Hilfe der MO system.*sysServices* (Layer-2, Layer-3), *ip.ipForwDatagrams* (Zähler FD) und *interfaces.ifNumber* (Portzahl) kann in allen untersuchten Fällen eine eindeutige Klassifizierung der Geräte vorgenommen werden. Das genaue Vorgehen der Klassifizierung ist in Abbildung 6-14 (rechts) dargestellt und kann in [Con03] nachgelesen werden.

Für jeden Gerätetyp wird eine Objektklasse definiert. Nach erfolgter Klassifizierung der Geräte wird für jedes identifizierte Gerät eine Objektinstanz vom Typ *Host-Class*, *Switch-Class* oder *Router-Class* angelegt. Jede Objektklasse enthält gerätespezifische Abfrageroutinen, die bei der Initialisierung aufgerufen werden. Bei der Abfrage der gerätespezifischen Konfigurationsdaten wird zum dritten Mal auf einen Netzknoten zugegriffen. Die ermittelten Konfigurationsdaten werden in dem Datenobjekt abgespeichert.

Zu den Parametern gehören Port-Adressen, Port-Konfigurationen (Linkkapazitäten, RM-Dienstklassenkonfigurationen) sowie Routing- und Bridging-Tabellen.
Die RM-Dienstklassen müssen gemäß der Anforderungen des Knotenmodells in Kapitel 3.4.3 eingerichtet sein und können mit Geräten der DiffServ-Architektur realisiert werden. Für das Management der RM-Dienstklassen wurden in der DiffServ-MIB [BCS02] entsprechende MO definiert. Leider waren zum Zeitpunkt der Implementierung des TM-Servers keine Geräte mit einer DiffServ-MIB verfügbar. Die Spezifikation der DiffServ-MIB im IETF-Standard sieht jedoch alle notwendigen Konfigurationsparameter zum Einrichten von RM-Dienstklassen im Sinne des Knotenmodells vor. Sie sind in MO enthalten, welche den Bearbeitungsprozess der an einem Eingang oder Ausgang ankommenden Pakete beschreiben. Die MOs müssen bei der Ermittlung der Konfigurationsdaten abgefragt werden, wobei insbesondere folgende MOs eine entscheidende Rolle spielen:

- diffServClassifier OBJECT IDENTIFIER ::= { diffServMIBObjects 2 }
- diffServAlgDrop OBJECT IDENTIFIER ::= { diffServMIBObjects 6 }
- diffServQueue OBJECT IDENTIFIER ::= { diffServMIBObjects 7 }
- diffServScheduler OBJECT IDENTIFIER ::= { diffServMIBObjects 8 }

Das *diffServClassifier*-Objekt gibt an, welche RM-Dienstklassen auf dem jeweiligen Port des Knoten realisiert sind und nach welchen Kriterien die ankommenden Pakete den einzelnen RM-Dienstklassen zugewiesen werden. Das *diffServAlgDrop*- und das *diffServQueue*-Objekt bezieht sich auf eine RM-Dienstklasse und repräsentiert u.a. die reservierten Ressourcen, d.h. die Puffergröße und die minimale Bedienrate. Das *diffServScheduler*-Objekt bezieht sich auf mehrere RM-Dienstklassen und enthält u.a. den verwendeten Scheduling-Algorithmus.

Testläufe des TM-Servers in realen Netzen haben ergeben, dass sich nicht alle Hersteller von SNMP-fähigen Geräten an die Vorgaben des Standards halten. Sie bieten in der Regel eine eigene Schnittstelle zu einem proprietären Managementsystem. Manche MOs sind daher überhaupt nicht in der MIB enthalten, bei anderen Objekten fehlen Parameter oder sind falsch gesetzt. Werden herstellerspezifische Eigenheiten bei der Implementierung der SNMP-MIBs festgestellt, so kann der TM-Server entsprechend angepasst werden. Dies kann geschehen, indem man die Anzahl der Objektklassen erweitert. Für jeden Gerätetyp eines bestimmten Herstellers kann eine eigene Objektklasse definiert werden, die dann herstellerspezifische Abfrageroutinen z.B. über ein CLI (*Command Line Interface*) enthält.

Nach der Datenmodellierung liegt für jedes Gerät im Netz ein entsprechendes Datenobjekt vor, in dem sich alle Konfigurationsparameter befinden, die von den Geräten entweder über die standardisierte MIB zur Verfügung gestellt wurden oder über herstellerspezifische Zugriffsmethoden durch gezielte Erweiterungen der TM-Klassenbibliothek in Erfahrung gebracht werden konnten.

6.7.1.2 Datenaufbereitung

Die nach der Datenmodellierung vorliegenden Objekte enthalten lediglich lokale Gerätedaten, die durch eine Subnetzkennung ergänzt wurden. Jedes Objekt eines Subnetzes enthält nur eine lokale Sicht eines Gerätes auf das Netz. Die einzelnen physikalischen Geräte sind jedoch durch Links miteinander verbunden und für jedes Gerätepaar gibt es einen fest eingestellten Weg durch das Netz. Die Verbindungsstruktur des Netzes spiegelt sich zu diesem Zeitpunkt noch nicht in der Objektstruktur wieder. Um die Topologie eines Netzes grafisch anzeigen zu können oder den Verbindungsweg durch das Netz für ein Knotenpaar ermitteln und darstellen zu können, müssen die lokalen Sichten der Netzknoten zu einer globalen Sicht zusammengefasst werden.
Der logische Zusammenhang der Objekte soll nun bei der Datenaufbereitung hergestellt werden. Dazu werden die einzelnen Datenobjekte gemäß der Verbindungsstruktur des Netzes

über Referenzen verbunden. Die Verbindungsstruktur wird vom TM aus den Verkehrslenkungstabellen der Geräte (OSI-Schicht 2: *Bridging-Table*, OSI-Schicht 3: *IP-Forwarding-Table*) abgeleitet.
Subnetze sind schleifenfrei aufgebaut (siehe Abschnitt 3.2.1) und können daher als ein Knotenbaum mit einem Wurzelknoten dargestellt werden. Die Blätter des Baumes sind die Endgeräte. Beginnend bei einem beliebigen Wurzelknoten werden alle Knoten des Subnetzes nacheinander in den Baum eingefügt. Dabei werden die bekannten Beziehungen zwischen den Knotenadressen, Portnummern und MAC-Listen (Adressen der Nachbarknoten) verwendet. Hängen an einem Port eines Switches mehrere Host-Objekte, wird ein HUB-Knoten zwischen dem Switchport und den Hosts eingefügt. Das genaue Verfahren kann in [CHH02] nachgelesen werden.
Nach der Datenaufbereitung sind alle Host-Objekte genau einem Switch- oder HUB-Objekt und alle HUB- oder Switch-Objekte genau einem anderen Switch- oder Router-Objekt zugeordnet.

6.7.2 Netzüberwachung

Der Netzüberwachungsprozess beobachtet das Netz hinsichtlich Fehlern und Konfigurationsänderungen. Die Netzüberwachung verfolgt zwei Ziele: Zum einen sollen Änderungen der Datenpfade möglichst frühzeitig erkannt und den angeschlossenen RM-Instanzen mitgeteilt werden. Zum anderen soll der Zeitpunkt, zu dem das Netz wieder einen stabilen Zustand erreicht, bestimmt und ein Topologieerkennungsprozess angestoßen werden.
Änderungen der Datenpfade treten in der Regel durch Fehler im Netz auf, wenn Links (z.B. Baggerschäden), Teile von Netzknoten (z.B. Ports) oder ganze Knoten ausfallen. Im Netz sind dazu Fehlermeldungs- (SNMP) und Re-Konfigurationsmechanismen (z.B. Routingprotokoll) vorgesehen.
Das SNMP-Konzept sieht vor, dass jedes Gerät sich selbst überwacht, Fehlerfälle erkennt und dem zentralen Netzmanagement meldet. Für das Beseitigen von Fehlerfällen gibt es verschiedene automatische Re-Konfigurationsmechanismen, die dezentral in den Netzknoten ablaufen. Dazu werden Zustandsnachrichten zwischen den Netzknoten ausgetauscht. Danach trifft jeder Knoten lokal eine Entscheidung und konfiguriert sich entsprechend um.

Fällt ein einzelner Link oder ein ganzer Knoten aus, befindet sich das Netz in einem transienten Zustand. Zu diesem Zeitpunkt kann es sein, dass Routen verwendet werden, die bereits ungültig sind und dadurch Pakete verloren gehen. Ziel der Re-Konfiguration ist die Wiederherstellung der Konnektivität zwischen allen noch aktiven Knoten. Dieser Prozess wird von den Routing-Protokollen unterstützt. Die Nachbar-Knoten einer Fehlerstelle erkennen den entsprechenden Ausfall als Erstes und teilen dies den anderen Knoten im Netz mit. Danach werden für die ausgefallenen Verkehrswege Alternativen gesucht. Jeder Knoten trifft für sich eine Entscheidung, ob die Wegewahl geändert werden muss. Gegebenenfalls muss der Knoten seine Verkehrslenkungstabelle aktualisieren. Haben alle Knoten des Netzes den Fehler zur Kenntnis genommen und entsprechend darauf reagiert, geht das Netz wieder in einen stabilen Zustand über. Ähnliches gilt auch im LAN (OSI-Schicht 2) für das *Spanning Tree (ST)* Protokoll. Im LAN können im Fehlerfall durch das ST-Verfahren deaktivierte Netzknoten aktiviert und *Bridging*-Tabellen entsprechend angepasst werden. Dadurch kann sich nicht nur die Verkehrslenkung sondern auch die Netzstruktur verändern.

Sowohl für die Fehlererkennung als auch für den Zeitpunkt der Fehlerbehebung sieht das SNMP-Protokoll Status-Meldungen vor, mit denen ein zentraler Manager informiert werden kann. Die Netzknoten müssen dazu so konfiguriert werden, dass sie in diesen Fällen sogenannte *Trap*-Meldungen an einen TM senden. Für eine dynamische Netzüberwachung sieht das TM-Konzept eine solche Konfiguration der Netzknoten vor.

Trifft eine *Trap*-Meldung bei einem TM ein, teilt dieser sofort allen betroffenen RM-Instanzen mit, dass sich das Netz in einem transienten Zustand befindet. Ein konkreter Rückschluss auf den Fehler und damit eine Einschränkung des transienten Zustandes auf einen kleinen Netzbereich ist leider in vielen Fällen nicht möglich. Ändert ein Knoten seine Verkehrslenkungs- oder Bridging-Tabelle, teilt er dies dem TM über eine *Trap*-Meldung mit. Ein einziger Fehler im Netz kann sowohl Konsequenzen auf die Struktur des Schicht-2 Netzes, als auch auf die Verkehrslenkung der IP-Schicht haben. Eine Vielzahl von Netzknoten kann von einem einzigen Fehler betroffen sein. Aus der sich ergebenden Flut von *Trap*-Meldungen Rückschlüsse auf die Fehlerursache zu ziehen, ist ein komplexes und nicht immer eindeutig lösbares Problem (z.B. Mehrfachfehler). Ebenso kann der Zeitpunkt über die Beendigung eines transienten Zustandes nicht vom TM vorausberechnet werden.
Im TM werden daher die eingehenden Trap-Meldungen von einem Dämon-Prozess empfangen und danach gefiltert, ob die Meldung die Topologie des Netzes betrifft. Trifft dies zu, wird ein Timer gestartet. Gleichzeitig wird veranlasst, dass der RM, in dessen Domäne der *Trap* aufgetreten ist, über die Topologieänderung informiert wird. Dabei setzt der TM einen ganzen Netzbereich (Subnetz, Kernnetz) in einen transienten Zustand. Läuft der Timer ab, startet der TM einen neuen Topologieerkundungsprozess, aktualisiert seine Datenbankeinträge und informiert die RM-Instanzen, dass sich der Netzzustand stabilisiert hat und neue Topologiedaten vorliegen.
Der Ablauf des Timers bedeutet, dass während einer gewissen Zeit Δt kein *Trap* mehr empfangen wurde. Der TM geht dann von einem stabilen Netzzustand aus. Das Intervall Δt soll so klein wie möglich sein, damit zu einem möglichst frühen Zeitpunkt die aktuellen Topologiedaten vorliegen und den RM-Instanzen angeboten werden können. Das Intervall Δt wird dabei entsprechend der Größe der TM-Domäne und den Konvergenzzeiten der Routing Protokolle gewählt.

6.8 Betreibermodelle

In diesem Abschnitt werden die Einsatzmöglichkeiten des Ressourcenmanagements in einer Multi-Provider Umgebung diskutiert. Dabei wird zwischen einer heterogenen und einer homogenen Umgebung unterschieden.

Zunächst werden zwei Beispiele für die Anwendbarkeit der RM-Architektur in einer heterogenen Umgebung geliefert. Dabei werden im einen Fall zwei IntServ-Netze und im anderen Fall zwei PSTN-Netze über ein RM-System verbunden. Im Anschluss daran wird die RM-Architektur in einer homogenen Umgebung auf verschiedene operative Anwendungsszenarien abgebildet.

6.8.1 Beispiele für eine heterogene Umgebung

Für eine heterogene Umgebung werden im Folgenden zwei Interworking-Szenarien betrachtet. Sie sollen zeigen, dass sich die RM-Architektur auch mit anderen QoS-Architekturen koppeln lässt. Zunächst werden zwei IntServ-Domänen, dann zwei PSTN-Netze über ein RM-System verbunden.

6.8.1.1 Szenario 1: IntServ – RM – IntServ

In diesem Szenario (siehe Abbildung 6-15) werden zwei IntServ-Netze über ein CoS-Netz verbunden, welches von einem RM-System verwaltet wird. Es soll eine Reservierung zwischen zwei RSVP-Clients über zwei RM-Domänen hinweg aufgebaut werden.
Das Besondere an diesem Beispiel ist, dass das RM-System an keine Dienststeuerung angekoppelt werden kann, sondern eine Umsetzung von unterschiedlichen

Reservierungsprotokollen vorgenommen werden muss. Dazu muss für das RM-System ein RSVP-Proxy definiert und in den RM-Instanzen an den Netzgrenzen installiert werden. Erhält ein RM eine *PATH*-Nachricht, kann er noch keine Reservierung vornehmen, da er die vom Empfänger gewünschte Dienstqualität noch nicht kennt. Diese Information steckt in der *RESV*-Nachricht, die zu einem späteren Zeitpunkt vom Zielteilnehmer als Antwort auf die *PATH*-Nachricht gesendet wird. Er kann folglich frühestens die Reservierung zu dem Zeitpunkt vornehmen, zu dem die *RESV*-Nachricht vorliegt.

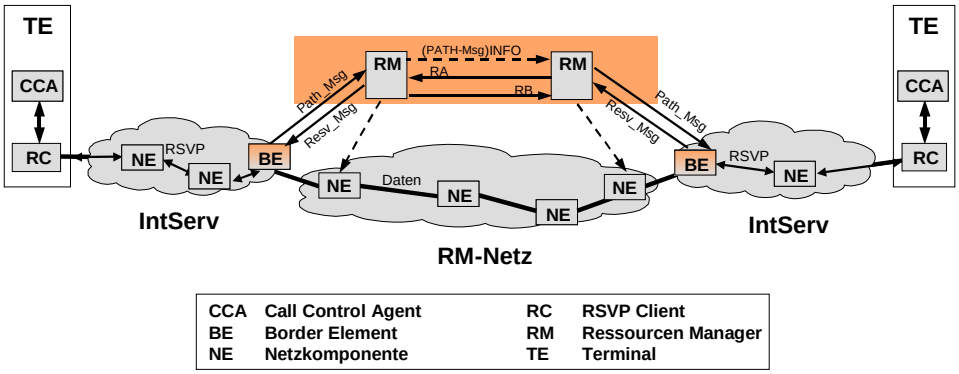


Abbildung 6-15: Signalisierungsbeispiel: IntServ - RM - IntServ

Bei diesem Szenario spielen die RSVP-Router an der Grenze zum RM-System eine Schlüsselrolle (BE). Zunächst müssen sie die ausgehenden PATH-/RESV-Nachrichten herausfiltern und an einen ihnen zugewiesenen RM weiterleiten. Antwortet der RM beispielsweise auf eine gesendete PATH-Nachricht mit einer RESV-Nachricht, konnte die Reservierung bis zum Zielteilnehmer durchgeführt werden und der BE sendet eine RESV-Nachricht in Richtung des initiierenden Teilnehmers.

Um den Austausch der Refresh-Nachrichten über das RM-Netz zu vermeiden, können die BE wie Zielteilnehmer agieren und entsprechende *Refresh*-Nachrichten generieren. Läuft ein *Refresh-Timer* im BE ab, sendet er eine *PathTear*-Nachricht an den RM.

Das RM-System erhält somit zum Aufbau einer Reservierung die initiierende PATH-Nachricht des RSVP-Quellnetzes und die RESV-Nachricht als Antwort des RSVP-Zielnetzes. Für die Koordination der RSVP-Zustände des Quell- und Zielnetzes wird der Tunnelmechanismus des RM-Protokolls verwendet. Eine Lösung könnte in diesem Fall wie folgt aussehen:

Der RM benötigt eine modifizierte Ablaufsteuerung und eine neue RSVP-Schnittstelle. Ferner muss ein RM die BE Knoten in den angrenzenden RSVP-Netzen kennen. Empfängt RM-1 eine PATH-Nachricht von BE-1, speichert er Zustandsinformationen (z.B. Res-ID, Req-ID, RSVP-Hop, Sender_TSPEC) und ermittelt den Datenpfad bis zum Border-Router (BR) seiner Domäne und damit zum Nachbar-RM (RM-2). RM-1 schätzt für seine Domäne die RSVP-Parameter C_{tot}, D_{tot} und die Laufzeit (*Minimum Path Latency*) ab, bestimmt den Hop-Count und das Minimum der noch verfügbaren Link-Bandbreiten des Pfades und aktualisiert die entsprechenden RSVP-Felder des ADSPEC-Objektes. Danach packt er die PATH-Nachricht in eine INFO-Nachricht, kennzeichnet den Inhalt (*Content-Type*) als eine RSVP-Nachricht und tunnelt sie durch die RM-Domäne zu RM-2. RM-2 verfährt wie RM-1, ermittelt BE-2 und erkennt, dass BE-2 ein RSVP-fähiger Router ist, packt die PATH-Nachricht aus und sendet diese an BE-2. Zu einem späteren Zeitpunkt antwortet BE-2 mit einer RESV-Nachricht. Nun liegen alle Verbindungsdaten vor und RM-2 kann die Reservierung anstoßen. Zunächst muss der *per-service Header* des FLOW_SPEC-Objektes ausgewertet und auf eine

RM-Dienstklasse abgebildet werden. Die TB-Quellenparameter können direkt übernommen werden. Die RESV-Nachricht wird anschließend in das *Data-Field* der RA-Nachricht eingepackt und an RM-1 geschickt. RM-1 erkennt, dass die Reservierung von BE-1 angestoßen wurde, packt die RESV-Nachricht wieder aus und sendet sie an BE-1. RM-1 sendet daraufhin RM-2 eine RB-Nachricht und bestätigt den erfolgreichen Aufbau der Reservierung.

6.8.1.2 Szenario 2: PSTN – RM – PSTN

In diesem Szenario werden zwei PSTN-Netze über ein CoS-Netz verbunden, welches von einem RM-System verwaltet wird. Für die Signalisierung wird eine Dienstarchitektur mit Megaco/H.248 verwendet, die gemeinsam von der ITU-T (H.248) und der IETF (RFC 3015) zur Steuerung Multimedialer Dienste im Internet standardisiert wurde [CGR00]. Die Architektur wurde entwickelt, um durchschaltevermittelte Telekommunikationsnetze (z.B. PSTN) mit dem Internet zu verbinden. Am Netzübergang werden *Gateways* eingesetzt, deren Funktionalität hinsichtlich der Verarbeitung von Nutzdaten (MG: *Media Gateway*) und der Steuerung (MGC: *Media Gateway Controller*) in zwei unabhängige Komponenten aufgeteilt wird. Die MGC sind an die Signalisierung der PSTN-Teilnehmer angebunden. Der MGC setzt diese auf eine H.323 Signalisierung (alternativ: SIP-Signalisierung) um. Um den MGC_2 am Ausgang des IP-Netzes adressieren zu können, ist eine Adressumsetzung (PSTN- nach IP-Adresse) notwendig. Entweder wendet sich der MGC dazu an einen GK (oder SIP-Proxy), dann lässt sich dieses Szenario direkt auf den Standardfall abbilden. Oder er verwendet einen anderen Adressumsetzungsdienst, dann müssen die MGC um eine Schnittstelle zum RM-System erweitert werden. Auf der Seite des RM-Systems sind keine Änderungen vorzunehmen.

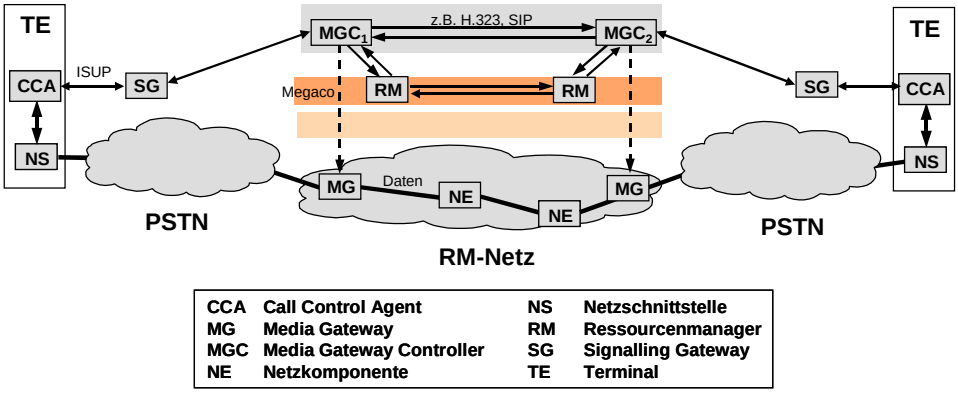


Abbildung 6-16: Signalisierungsbeispiel: PSTN - RM – PSTN

6.8.2 Beispiele für eine homogene Umgebung

Geht man von einem bestehenden IP-Netz mit entsprechend dimensionierten und konfigurierten DS-Klassen aus, so repräsentiert das Ressourcenmanagement einen Zusatzdienst, der eine bessere Übertragungsqualität bietet als das bestehende CoS-Netz. Es verwaltet die Ressourcen eines Netzbetreibers. Die Instanz, die den Zusatzdienst anbietet und betreibt, kann der Netzbetreiber selbst oder eine unabhängige Person sein, z.B. ein Dienstbetreiber. Ein Dienstbetreiber kann ein CoS-Netz mit entsprechend dimensionierten Kapazitäten von einem Netzbetreiber anmieten und dieses selbst an Endkunden vertreiben.

Sehr kleine Firmen besitzen zumeist einen einzigen Firmenstandort. Sie betreiben kein eigenes IP-Netz, sondern sind an Netze anderer Betreiber angeschlossen. Innerhalb ihres

Netzes bieten sie ihren Mitarbeitern einen Dienstzugang über eine Dienststeuereinheit. Für echtzeitkritische Verbindungen nach außen (z.B. zu Kunden) sind sie auf fremde Netzbetreiber (z.B. VoIP-Betreiber) angewiesen. Ein Netzbetreiber kann die erforderliche Übertragungsqualität mit Hilfe einer RM-Architektur oder einer IntServ-Architektur erreichen. Zwischen dem Netzbetreiber und der Firma gibt es hinsichtlich des Datenvolumens, der Dienstgüte und des Tarifes entsprechende Vereinbarungen. Das Ressourcenmanagement obliegt in diesem Fall allein dem Betreiber des QoS-Netzes (Betreibermodell-A).

In größeren Firmen tritt häufig der Fall ein, dass mehrere entfernt liegende Firmenstandorte miteinander verbunden werden müssen. Jeder Firmenstandort besitzt ein eigenes Netz, die über Netze anderer Betreiber miteinander verbunden werden. Dazu mieten die Firmen einzelne Standleitungen oder ganze virtuelle Netze zwischen ihren einzelnen Standorten fest an. So können sie im Kernnetz zwar nicht als ein Netzbetreiber jedoch als ein Dienstbetreiber auftreten, der seinen Mitarbeitern die angemieteten Netzressourcen zur Verfügung stellt und verwaltet (Betreibermodell B).

Die Signalisierungsbeziehungen zwischen dem Teilnehmer (Mitarbeiter), dem Dienstanbieter (Firma) und dem Netzbetreiber sind für diese beiden Fälle in Abbildung 6-17 und Abbildung 6-18 dargestellt. Im Betreibermodell-A kümmert sich die Firma lediglich um den Teilnehmerzugang zu einem z.B. VoIP-Netzbetreiber. Die Firma verwaltet nur die Ressourcen innerhalb ihres Netzes bis zum Zugangsknoten des Netzbetreibers und überlässt diesem dann alles Weitere. Eine Kontrolle der tatsächlich erzielten QoS kann nur Ende-zu-Ende zwischen den Terminals z.B. mittels RTCP erfolgen. Über welche Netze die Verbindung läuft und welche QoS-Mechanismen eingesetzt werden, bleibt der Firma verborgen.

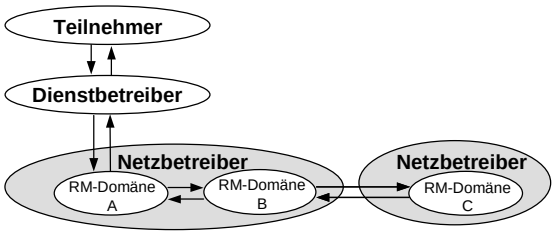


Abbildung 6-17: Betreibermodell A - RM Teil des Netzbetreibers

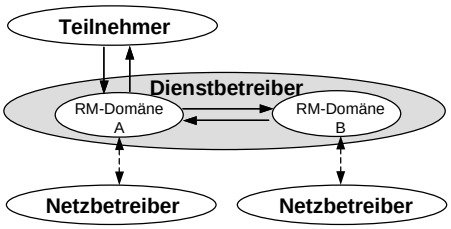


Abbildung 6-18: Betreibermodell B - RM Teil des Dienstbetreibers

Anders beim Betreibermodell-B. Hier besitzt eine Firma mehrere Standorte, die über ein virtuelles privates Netz (*Virtual Private Network* VPN) miteinander verbunden sind. Die Firma hat dazu von einem oder mehreren Netzbetreibern Ressourcen fest angemietet. Den Zugang zu diesen Ressourcen verwaltet sie selbst mit Hilfe eines RM-Systems. Sowohl die Struktur des VPN als auch die gemieteten Ressourcen werden dem RM-System entweder vom Netzbetreiber direkt über einen selektiven Zugriff auf Topologiedaten eines TM oder über das Konfigurationsmanagement des RM-Systems statisch vorgegeben. Bei größeren Standorten kann die Firma selbst als Netzbetreiber auftreten, im Kernnetz zwischen den Standorten sind es externe Betreiber.

Im Folgenden sollen die verschiedenen Rollen und Aufgaben der am Prozess der Ressourcenreservierung beteiligten Personen (Teilnehmer, Dienstbetreiber, Netzbetreiber) diskutiert werden.

Ein Teilnehmer stößt eine Reservierung an, indem er eine entsprechende Nachricht an seinen Dienstbetreiber schickt. Die Anfrage enthält einen Verbindungsaufbauwunsch zu einem

bestimmten Ziel mit einer bestimmten Übertragungsqualität. Der Dienstbetreiber stellt dazu eine Dienststeuereinheit DS zur Verfügung, welche die Anfrage des Teilnehmers entgegennimmt und bearbeitet. Ein Teil dieser Bearbeitung ist die Ressourcenreservierung, welche in dieser Architektur als ein eigenständiger Dienst realisiert ist.

Wird dieser Reservierungs-Dienst vom Netzbetreiber übernommen (Betreibermodell A), wählt die DS einen Netzbetreiber aus und signalisiert einem vom Netzbetreiber zur Verfügung gestellten RM eine Reservierungsanfrage. Die DS wartet mit der weiteren Bearbeitung der Verbindungsanfrage des Teilnehmers auf die Reaktion des Netzbetreibers. Der Dienstbetreiber kennt in diesem Fall nur den Zugangsknoten des Netzbetreibers, an den der Teilnehmer seine Nutzdaten zu schicken hat. Vom Netz des Netzbetreibers kennt er weder die Topologie, noch die tatsächlich zur Verfügung stehenden Ressourcen. Ein Tarifierungsmodell zwischen einem Dienstbetreiber und einem Netzbetreiber könnte in diesem Fall ein Pauschalvertrag über eine maximale Datenmenge sein, die der Dienstbetreiber an einem bestimmten Punkt des Netzes einspeisen kann. Alternativ dazu könnte mit dem Netzbetreiber auch eine Abrechnung pro Verbindung vereinbart werden.

Wird dieser Reservierungs-Dienst vom Dienstbetreiber übernommen (Betreibermodell B), betreibt er die RM und hat dadurch Einblick in die Topologie und die aktuellen Lastzustände des Netzes. Vom Netzbetreiber mietet er Ressourcen an und verwaltet selbst den Zugriff auf diese Ressourcen. Die Signalisierung zwischen dem Dienstbetreiber und dem Netzbetreiber findet über eine Schnittstelle zwischen den RM und TM statt. Sie beinhaltet die Konfiguration von Überwachungseinheiten (Policer) der Verkehrsquellen am Netzzugang mit den Verbindungsparametern, sowie die Bereitstellung der Topologiedaten, welche nur bei Systeminitialisierung oder bei Änderungen der Topologie stattfindet. Der Topologieerkennungs-, Bereitstellungs- und Netzkonfigurationsdienst wird jedoch nach wie vor vom Netzbetreiber übernommen. Die Sicht eines RM auf die Netztopologie kann insofern eingeschränkt sein, als sie nur die Routen und die für diesen Dienstbetreiber angemieteten Ressourcen umfasst. Diese Freiheitsgrade lässt die RM-Architektur aufgrund der Trennung von Netz- und Reservierungsebene durch ein anpassbares Protokoll zwischen den TM- und RM-Instanzen zu.

Aufgabenverteilung

Zusammenfassend erfolgt nun eine genauere Betrachtung der Aufgabenverteilung zwischen einem Teilnehmer, einem Dienst- und einem Netzbetreiber für das Betreibermodell-A. Dieses Modell ist insofern interessant, als das Ressourcenmanagement besser in den operativen Netzbetrieb eingebunden und dadurch für weitere Aufgaben herangezogen werden kann. Die Aufgabenverteilung ist in Abbildung 6-19 dargestellt.

Im Folgenden werden die Aufgaben des Netzbetreibers genauer betrachtet. Der Netzbetreiber besitzt ein ausreichend dimensioniertes Netz mit DiffServ-Technologie, welches für Echtzeitverkehre um ein RM-System erweitert wurde. In den Netzknoten hat er dazu eine oder mehrere DS-Klassen eingerichtet. Einige davon werden ausschließlich vom RM-System verwaltet (RM-Dienstklassen). Der Netzbetreiber definiert zu jeder dieser RM-Dienstklassen eine Dienstgütebeschreibung und konfiguriert die RM-Instanzen mit den entsprechenden Zugangskontrollverfahren. Der Netzbetreiber kann nun einem Dienstbetreiber einen echtzeitfähigen Übertragungsdienst anbieten. Der Teilnehmerzugang zu den RM-Dienstklassen erfolgt über die RM-Instanzen.

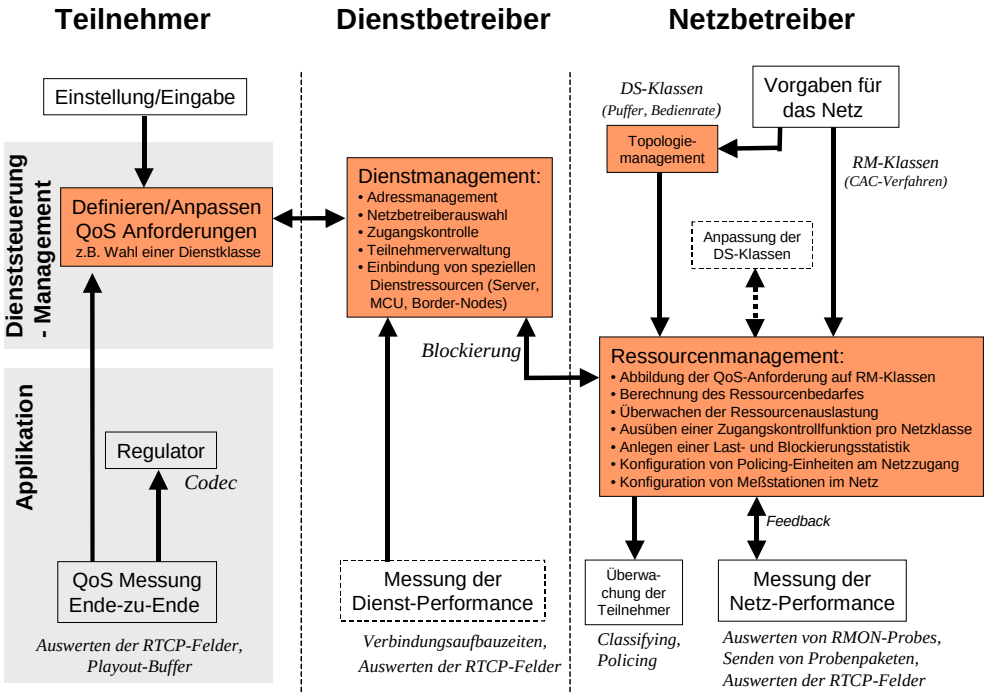


Abbildung 6-19: Funktionsverteilung für das Betreibermodell-A

Die Dienstgütebeschreibung umfasst QoS-Parameter sowie eine Aussage hinsichtlich der Einhaltung dieser Parameter. Die Einhaltung der Dienstgütegarantie wird durch das Zugangskontrollverfahren im RM sichergestellt. Da dieses Zugangskontrollverfahren nur auf einer virtuellen Ebene auf der Basis eines Datenmodells des Netzes arbeitet, kann es in Ausnahmefällen bei der Datenübertragung zur Verletzung der Dienstgütegarantie kommen, ohne dass das RM davon informiert wird. Es ist daher sinnvoll, die Arbeitsweise des Ressourcenmanagements durch Messungen im physikalischen Netz zu überprüfen (z.B. *One-way/Round-trip Delay Metric* [AKZ99], [Alm99], *Delay Variation Metric* [DC02], *One-way* Loss Metric [AlK99]). Stellt das Netzmanagementsystem entsprechende Messstationen im Netz zur Verfügung, kann das RM-System über die aktuelle Netz-Performance informiert werden. Durch eine solche messtechnische Unterstützung besteht zudem für das Ressourcenmanagement die Möglichkeit, die mit den Zugangskontrollverfahren erzielbare Netzauslastung auf die speziellen Anforderungen des Netzes hin zu optimieren.

Prinzipiell besitzt ein RM wichtige Daten für den laufenden Netzbetrieb und kann daher auch für weitergehende Aufgaben herangezogen werden. Er hat zu jedem Zeitpunkt die komplette Zustandsinformation aller der von ihm kontrollierten logischen Teilnetze des CoS-Netzes. Darüber hinaus kennt er die Blockierungswahrscheinlichkeiten auf allen Links ebenso wie die Verkehrsbeziehungen und damit die Verkehrsflüsse im Netz. Diese Informationen sind für den laufenden Netzbetrieb (z.B. Routing-Optimierung, Dimensionierung der DS-Klassen) aber auch für die Erweiterungsplanung eines Netzes von großer Bedeutung. So können diese Informationen einem Netzbetreiber zur Verfügung gestellt oder grundsätzlich auch dazu verwendet werden, bestimmte Prozesse selbst anzustoßen.
Je nachdem, wie eng das Ressourcenmanagement mit den anderen operativen Betriebs- oder nicht-operativen Planungsprozessen des Netzes verbunden wird, fällt es mehr in den Zuständigkeitsbereich des Dienstbetreibers oder in den des Netzbetreibers. Abbildung 6-19 stellt die möglichen Aufgaben und Schnittstellen des Ressourcenmanagements dar.

6.9 Prototypische Implementierung

Das hier vorgestellte Konzept des Ressourcenmanagement-Systems wurde prototypisch für eine H.323 Umgebung implementiert. Es wurden ein Prototyp des Ressourcemanagers und des Topologie-Managers erstellt und in einer H.323-Systemumgebung getestet. Abbildung 6-20 zeigt dazu den Aufbau des für den Prototypen verwendeten Testnetzes.

Die Netztechnik des Testnetzes besteht aus zwei Layer 2-Switches (*Super Stack 2200*) und einem Layer 3-Switch (*Core Builder 3500*) der Firma 3Com. Sie unterstützen auf Layer 2 den Standard IEEE 802.q und auf Layer 3 DiffServ-spezifische Funktionalitäten. In den Netzknoten können manuell Verkehrsklassen für Sprache und Video eingerichtet werden. Der von 3Com implementierte *Scheduling*-Algorithmus sieht nur einen *Simple-Priority* Mechanismus vor. Ein L3-Switch bietet zusätzlich die Möglichkeit der Durchsatzbegrenzung durch Puffermanagementmethoden (*Policer).*

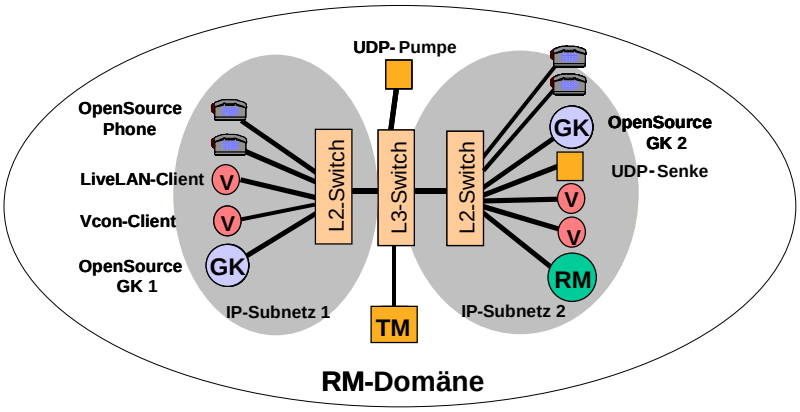


Abbildung 6-20: RM-Testnetz

Die Implementierung des TM wurde zudem in einem größeren Testnetz mit Gerätschaften unterschiedlicher Hersteller wie Cisco, Extreme Networks und Enterasys Networks getestet [Con03]. Die Gerätetypen waren im Einzelnen:

- Cisco: Catalyst 6500, 5500, 2900
- Extreme Networks: Summit 5i, Summit 48 und Alpine 3808
- Enterasys Networks: Matrix E1 und X-Pedition 8000

Mit diesen Geräten wurden verschiedene Topologien erzeugt und Fehlerfälle (Linkausfälle) simuliert. Die Geräte wurden vorher so konfiguriert, dass sie Trap-Meldungen an den TM schicken. Der TM-Prototyp hat sämtliche Topologien richtig erkannt und dabei alle Geräte korrekt klassifiziert. Linkausfälle führten unmittelbar zur Benachrichtigung der betroffenen RM. Nach Ablauf eines Timers wurde die Topologie erneut erkundet und die RM über das Vorliegen aktueller Topologiedaten informiert.

6.9.1 Ressourcen-Manager

Der in dieser Arbeit vorgestellte RM wurde in den Arbeiten [Tro01], [CHH02] prototypisch implementiert. Er wurde in C++ mit Microsoft Visual C++ Version 6 unter Windows 98 entwickelt und unter Windows 95/98/2000 getestet. Als H.323-Gatekeeper wurde ein *Open-Source Code* aus einem Open-H.323 Projekt [OpH323] eingesetzt, der um eine Schnittstelle zum RM erweitert wurde. Als H.323-Clients werden neben den *Open-H.323 Clients* vorwiegend auch kommerzielle Produkte von Vcon (Armada Escort) und PictureTel

(LiveLAN) verwendet. Alle diese H.323-Clients basieren nicht auf den im Annex.N spezifizierten Erweiterungen des H.323.

Der RM selbst wurde in der Entwicklungsumgebung des Open-H.323 Projektes unter Verwendung derselben Programm- und Klassenbibliotheken erstellt. Für die Realisierung der funktionalen Einheiten *Request-Handler* und *Admission-Controller* ist jeweils ein eigener *Thread* zuständig. Während der Request-Handler von vielen DS gleichzeitig Reservierungsnachrichten empfangen muss, arbeitet der *Admission-Controller* eine Reservierungsnachricht nach der anderen ab. Eine Aufteilung der Aufgaben in zwei Threads entkoppelt die Prozesse Nachrichtenempfang und Zustandsverwaltung von der Zugangskontrolle. Diese Lösung erhöht die Reaktivität eines RM-Managers und bietet zudem Performance-Vorteile, falls der RM auf einem Biprozessor-Rechnersystem installiert wird. Die Kommunikation zwischen den *Threads* wird über zwei FIFO Warteschlangen (*Message Queues*) entkoppelt. Jeder *Thread* schreibt in eine Warteschlange Nachrichten, die vom anderen *Thread* weiter bearbeitet werden müssen und liest aus der anderen Warteschlange Nachrichten aus. Die Blöcke *Blocking-Handler* und *Policy-Handler* werden im Prototyp lediglich durch Methodenaufrufe realisiert.

In den Prototypen wurden alle in Abschnitt 5.6 vorgeschlagenen Zugangskontrollverfahren integriert. Sie können bei der Initialisierung des RM einzelnen RM-Dienstklassen zugewiesen werden. Heutige Netzknoten unterstützen weder eine automatische Konfiguration mit *Policies* noch besitzen sie eine Implementierung der DiffServ-MIB. Daher muss die Konfiguration der DS-Klassen (*Classifier, Scheduler*) in den Netzknoten und im TM manuell erfolgen. Die Abfrage der DiffServ-MIB ist im TM vorgesehen, so dass der TM in Zukunft alle Konfigurationsdaten über das Auslesen der DiffServ-MIB erhält.

6.9.2 Topologie-Manager

Der TM wurde in den Arbeiten [CHH02], [Con03] prototypisch implementiert. Für die Implementierung wurde eine Client-Server Architektur gewählt. Der TM-Server wurde als eigenständige Applikation realisiert, während der TM-Client als Prozess in die RM-Applikation integriert wurde.

Beide Teile des TM sind modular aufgebaut, so dass einzelne Module gegen andere ausgetauscht oder neue hinzugefügt werden können, ohne bereits vorhandene Module verändern zu müssen. Es wurde sowohl für den TM-Server als auch für den TM-Client ein objektorientierter Ansatz gewählt.

Für die Zugriffe auf die Netzknoten wird die Net-SNMP Bibliothek [Net02] verwendet. Zum gefilterten Empfangen von Ethernet-Paketen wird die *lipcap*-Bibliothek [Lip02] verwendet. Sie erlaubt, auf die MAC-Adressfelder der empfangenen Pakete zuzugreifen und darin enthaltene Antworten (icmp echo reply) den zuvor gestellten Anfragen zuzuordnen (icmp echo request).

Für die Datenmodellierung des TM-Server wurde die STL-Klassenbibliothek (*Standard Template Library*) von SGI [STL02] verwendet. Jedes einzelne Gerät des Netzwerkes wird durch ein C++ Objekt repräsentiert. Die Objekte sind gemäß der realen physikalischen Struktur durch Zeiger verknüpft. Ein Objekt dient primär der Informationsspeicherung der Konfigurationsdaten des Gerätes, bietet darüber hinaus jedoch verschiedene Zugriffsmethoden. Die Zugriffsmethoden unterscheiden sich hinsichtlich der Client- und Server-Seite.

Anhand der Datenobjekte kann die Topologie des Netzes auch visualisiert werden. Die Anordnung der Netzknoten in einer zweidimensionalen Form wird durch ein externes

Programm mit dem Namen „*Neato*“ realisiert und mit Hilfe von *DotView* graphisch ausgegeben. „*Neato*“ ist Teil einer Sammlung von Programmen *Graphviz* zur Darstellung von Graphen, welche von AT&T entwickelt wurde und im Quellcode verfügbar ist [Gra02].

6.9.3 Fazit

Mit dem vorgestellten Prototypen konnte die Funktionsweise des RM-Systems im Zusammenspiel mit mehreren H.323-Client- und GK-Instanzen und damit die Realisierbarkeit des RM-Ansatzes gezeigt werden. Im Einzelnen konnten folgende Punkte nachgewiesen werden:

- Prozess der Topologieerkennung.
- dynamische Anbindung des RM an den TM.
- Arbeitsweise des Reservierungsverfahrens im Zusammenspiel mit einer H.323-Dienststeuerung für den Auf-, Abbau und die Modifikation von H.323-Rufen verifiziert.
- Etablierung einer Ende-zu-Ende Reservierung.
- Korrektheit der Modellierung der Netztopologie.
- Blockierung und Abbruch eines Rufes bei Überlastung des Netzes.

Ferner konnte der prinzipielle Unterschied zwischen einem Zugangskontrollverfahren mit harten und einem mit weichen Garantien vorgeführt werden. Die Arbeitsweise der statistischen Zugangskontrollverfahren bei hoher Verkehrsaggregation konnte in diesem Testnetz aufgrund der kleinen Anzahl zur Verfügung stehender Verkehrsquellen allerdings nicht nachgewiesen werden. Die Reaktionszeit eines RM auf eine einzelne Anfrage beträgt nur wenige Millisekunden, wenn der RM auf einem eigenen Rechner läuft, der nicht weiter belastet ist.

Der erstellte Prototyp kann als Grundlage für weitere Arbeiten dienen und schrittweise zu einem kommerziellen Produkt ausgebaut werden. Dazu ist beispielsweise der Programmcode unter Performance-Aspekten zu optimieren. Ferner sind geeignete Schutzmechanismen gegen Systemausfälle einzubauen. Zur Konfiguration von *Classifying*-Filter und *Policing*-Einheiten im Netz sind der RM und der TM-Server um eine Schnittstelle zu erweitern.

7. Zusammenfassung

In der Vergangenheit wurde eine Vielzahl von QoS-Architekturen entwickelt. Bislang konnte sich allerdings keiner dieser Ansätze durchsetzen. Ein Trend hin zur DiffServ-Technologie zeichnet sich ab; zu einem flächendeckenden Einsatz ist es jedoch bis heute nicht gekommen. Ob diese Technologie allein ausreicht, um echtzeitkritische Anwendungen wie *High-Quality* Telefonie, Multimedia *Videoconferencing* oder HDTV *(High Density Television)* in Zukunft über das Internet abzuwickeln ist fraglich. Auf der anderen Seite hat das Scheitern der IntServ-Architektur aus technischer Sicht gezeigt, dass eine gute Skalierbarkeit und ein geringer Management-Overhead für den praktischen Einsatz von QoS-Architekturen entscheidend sind.

Der Aspekt der Einführbarkeit (Migration) wird jedoch bei keiner der bisherigen QoS-Architekturen hinreichend beachtet. Alle diese Architekturen machen Veränderungen auf der Teilnehmer-Netz-Schnittstelle erforderlich. Sie erschweren somit die Einführung und erhöhen den Managementaufwand. Manche von ihnen basieren zudem auf proprietären Scheduling-Verfahren oder auf einer Manipulation des IP-Headers und können aus diesem Grund nur begrenzt eingesetzt werden.

Daher wurde eine neue, gut skalierbare QoS-Architektur entwickelt, die leicht in eine bestehende Systemumgebung eingeführt werden kann. Die vorgestellte **RM-Architektur** ermöglicht basierend auf der DiffServ-Technologie die Realisierung verschiedener, vordefinierter Dienstklassen für echtzeitkritische Verkehre mit harten und weichen QoS-Garantien.

Für das Architektur-Design wurde ein zentralisierter Ansatz mit einem Ressourcen-Manager gewählt. Um eine gute Skalierbarkeit zu erzielen, wurden die Funktionen Signalisierung und Netzzugangskontrolle von der Paketverarbeitung getrennt. Dazu wurden zentralisierte Server, die Ressourcen-Manager, eingeführt. Die Aufgaben eines RM bestehen aus der Verarbeitung der Signalisierungsnachrichten und dem Verwalten der Zustandsinformationen, während die Netzknoten für das Weiterleiten der Pakete verantwortlich sind. Durch diese Aufgabenteilung wird erreicht, dass in den Netzknoten keine Zustände und Verarbeitungsschritte pro Flow notwendig sind.

Um den Verwaltungsaufwand möglichst gering zu halten und die Einführung in eine bestehende Systemumgebung zu erleichtern, wurde auf eine Teilnehmerschnittstelle verzichtet. Stattdessen wurde der RM an bereits vorhandene Dienststeuereinheiten im Netz gekoppelt. Die Protokollanpassung wurde auf Seiten des RM vorgenommen, so dass nur minimale Eingriffe bei den Dienststeuereinheiten notwendig waren.

Um die Einführbarkeit des RM-Systems weiter zu verbessern und Unabhängigkeit von der Netztechnologie zu erreichen, wurde ein eigenständiger Topologieerkennungsdienst definiert.

Für dessen Realisierung wurde ein zentralisierter Ansatz mit einem Topologie-Manager gewählt. Der TM greift über standardisierte Zugriffsverfahren auf Konfigurationsdaten der Netzknoten zu und generiert daraus ein Datenmodell der Topologie. Die Anbindung an einen RM erfolgt dynamisch über eine Registrierungsprozedur. Ein registrierter RM kann selektiv auf die Topologiedaten des TM zugreifen. Die Ressourcenreservierung führt der RM virtuell auf der Basis der Topologiedaten durch.
Für größere Netze wurde das Konzept von Verwaltungsdomänen sowohl auf Netz- als auch auf RM-Ebene eingeführt. Die TM- und RM-Domänen können unabhängig voneinander konfiguriert werden. Es wurde darüber hinaus gezeigt, nach welchen Gesichtspunkten diese zu planen sind.

Um mehrere Dienstklassen für echtzeitkritische Verkehre zu realisieren, wurde ein **Knotenmodell** definiert. Es basiert auf DiffServ-Mechanismen, garantiert eine maximale Trennung zwischen den Verkehren unterschiedlicher Dienstklassen. Das Knotenmodell schafft die notwendigen Voraussetzungen für die Zugangskontrollverfahren, um innerhalb einer Dienstklasse die gewünschte Dienstgüte gewährleisten zu können.

Ferner wurde ein **Reservierungsverfahren** entworfen, das eine Zugangskontrolle Ende-zu-Ende, d.h. auf allen Schicht-2 und Schicht-3 Links, anstößt. Das spezifizierte Protokoll koordiniert und überwacht die Verteilung der Zustandsinformationen im Netz. Es kann den vollständigen Kontext einer Sitzung transportieren und dadurch die Reservierungen mehrerer uni- oder bidirektionaler Verbindungen bündeln. Zudem unterstützt es Vorabreservierungen, Inter-Domain Reservierungen und das *Interworking* mit anderen QoS-Architekturen. Unter Berücksichtigung anderer Reservierungsverfahren wurden die Protokollparameter definiert und darüber hinaus ein generischer Transport-Mechanismus für Nachrichten anderer Reservierungsprotokolle eingeführt. Um eine größtmögliche Unabhängigkeit von den QoS-Mechanismen der Netze zu erzielen, werden die Dienstgüteparameter der Teilnehmer auf Ende-zu-Ende gültige, netzspezifische Transport-QoS Parameter abgebildet. Um über mehrere RM-Domänen hinweg eine Dienstgütegarantie geben zu können, wurden QoS-Budgets eingeführt. Ein QoS-Budget ist ein Teil der Dienstgütespezifikation und beschreibt das Ende-zu-Ende Verhalten einer Domäne. Die QoS-Budgets werden entlang des Reservierungspfades aufsummiert und der jeweiligen Nachbar-Domäne mitgeteilt.

Für die Realisierung der Dienstklassen mit harten und weichen QoS-Garantien wurden geeignete parameterbasierte **Zugangskontrollverfahren** vorgeschlagen. Dazu wurden umfangreiche Untersuchungen unternommen. Diese waren notwendig, da zum einen von der IETF (mit Ausnahme des *Guaranteed Service* von IntServ) bislang keine Zugangskontrollverfahren standardisiert wurden und zum anderen, weil im Rahmen von DiffServ in den vergangenen Jahren zwar viele messbasierte aber keine parameterbasierten Zugangskontrollverfahren veröffentlicht wurden. Daher wurden auch Verfahren untersucht, die ursprünglich für ATM-Netze entwickelt worden waren.

Jedes parameterbasierte Zugangskontrollverfahren beruht auf ganz bestimmten Annahmen über die Charakteristik des Verkehrs. Im Gegensatz zu ATM-, gemischtem LAN- oder Internetverkehr wurden echtzeitkritische IP-Verkehre bislang wenig erforscht.

Deshalb wurden zunächst umfangreiche **Messungen** von Sprach-, Video- und Signalisierungsverkehren kommerzieller Videokonferenzapplikationen (Microsoft, PictureTel und Vcon) durchgeführt. Bei den Messungen wurden nicht nur die Eigenschaften der Applikationen bei den unterschiedlichen Einstellungen untersucht, sondern auch die Einflüsse des Teilnehmerverhaltens und des Systems, auf dem die Applikationen installiert sind. Die Messdaten wurden nach verschiedenen Methoden analysiert. Dabei wurden insbesondere die

mittleren Senderaten pro Zeitintervall, die Verteilungen und Korrelationen von Paketlängen und -abständen sowie die Langzeitabhängigkeiten ermittelt. Die Ergebnisse geben einen detaillierten Einblick in das Sendeverhalten der Quellen.

Aufbauend auf den Messungen wurde danach ein geeignetes Verkehrsmodell ausgewählt. Dabei hat sich das *Token Bucket* Modell für die RM-Architektur als besonders geeignet herausgestellt. Zum einen kann mit diesem Modell das „worst case"-Verhalten der Quellen ebenso beschrieben werden wie ihr statistisches Verhalten. Zum anderen werden die Parameter dieses Modells in einer Vielzahl von anderen Reservierungsverfahren als Verkehrsparameter verwendet, so dass dadurch eine gute Interoperabilität mit anderen QoS-Architekturen garantiert wird.

Das TB-Modell bietet einen Satz an Parametern, nach denen ein Verkehr gefiltert werden kann. Will man Verkehre mit diesen Parametern charakterisieren, so dass sie ungehindert einen entsprechend konfigurierten Filter passieren können, ergeben sich Freiheitsgrade. Für die **Charakterisierung** der gemessenen Quellen wurde eine Methodik entwickelt, nach der alle Quellenverkehre charakterisiert und daraus Parametersätze pro Applikation und gewählter Einstellung ermittelt wurden.

Für die nach dem TB-Modell charakterisierten Verkehre wurden schließlich geeignete Zugangskontrollverfahren ausgewählt. Bei Anwendung der Verfahren auf die ermittelten TB-Parametersätze konnte deren Eignung für die IP-Quellen untersucht werden. Dabei wurden Auslastungswerte einzelner Links für homogene wie heterogene Verkehrsgemische bei unterschiedlichen Aggregationsstufen ermittelt. Die Einhaltung der Dienstgütegarantie wurde durch Simulationen überprüft, bei denen die Messreihen als Verkehrsquellen verwendet wurden. Auf diese Art konnten für die Implementierung der RM-Architektur Zugangskontrollverfahren vorgeschlagen werden, die sich für die Realisierung von Dienstklassen mit harter oder weicher QoS-Garantie eignen.

Das hier vorgestellte Konzept des Ressourcemanagement-Systems wurde **prototypisch** für eine H.323 Umgebung implementiert und getestet. Der RM und der TM wurden mit C++ unter Windows 98 realisiert. Als H.323-Gatekeeper wurde ein *Open-Source Code* aus dem Open-H.323 Projekt eingesetzt, der um eine Schnittstelle zum RM erweitert wurde. Als H.323-Clients wurden neben den Open-H.323 Clients auch kommerzielle Produkte von Vcon (Armada Escort) und PictureTel (LiveLAN) verwendet.

Die Realisierung des RM sieht einen Abbildungsmechanismus von applikationsspezifischen Parametern auf netzspezifische Parameter für die Fälle vor, in denen die Dienstsignalisierung keine Verkehrs- oder Dienstgüteparameter enthält. Die Zugangskontrollverfahren wurden so in den RM integriert, dass eine flexible Zuordnung eines Verfahrens zu einer Dienstklasse im Netz möglich ist. Die Realisierung des TM kann Topologieänderungen im Netz erkennen und die registrierten RM darüber informieren. Der TM kann zudem problemlos hinsichtlich seiner Zugriffsverfahren, z.B. um herstellerspezifische Methoden, erweitert werden. Als Netztechnologie wurden moderne Switches und Router verschiedener Hersteller verwendet.

Mit dem Demonstrator wird die Realisierbarkeit des RM-Ansatzes für eine Intranet-Umgebung nachgewiesen. Mit dem Prototypen konnte die Funktionsweise des RM-Systems im Zusammenspiel mit mehreren H.323-Client- und GK-Instanzen sowie mit einer heterogenen Netzumgebung gezeigt werden.

Der Prototyp wurde von der Firma Siemens übernommen und soll zu einem kommerziellen Produkt der HiPath-Serie ausgebaut werden.

Literaturverzeichnis

[ABF98] W. Almesberger, J.-Y. Le Boudec, T. Ferrari: “SRP: a Scalable Resource Reservation for the Internet”, 6th International Workshop on Quality of Service (IWQoS'98), Mai 1998

[ACZ95] E. Amir, S. McCanne, H. Zhang: “An Application Level Video Gateway”, ACM Multimedia '95, San Francisco, 1995

[Ada97] A. Adas: “Traffic Models in Broadband Networks”, IEEE Communications Magazine, S. 82-89, Juli 1997

[AKZ99] G. Almes, S. Kalidindi, M. Zekauskas: “A One-way Delay Metric for IPPM“, IETF-Standard, RFC 2679, September 1999

[AlK99] G. Almes, S. Kalidindi, M. Zekauskas: “ A One-way Packet Loss Metric for IPPM”, IETF-Standard, RFC 2680, September 1999

[Alm99] G. Almes, S. Kalidindi, M. Zekauskas: “A Round-trip Delay Metric for IPPM”, IETF-Standard, RFC 2681, September 1999

[AM95] R.Y.Awdeh, H.T.Mouftah: “Survey of ATM-Switch Architectures“, Computer Networks and ISDN Systems, Vol. 27, 1995

[AMS82] D. Anick, D. Mitra, M.M. Sondhi: “Stochastic Theory of a Data-Handling System with Multiple Sources”, Bell Sys. Tech. J., Vol. 61 No. 8, S. 1871-1894, 1982

[AN.323] ITU – T H.323 Annex N, TD 43 STUDY GROUP 16: “End-to-End QOS and service Priority Control and Signaling in H.323 Systems”, TD-55/WP2, Genf, Schweiz, November 2000

[Aqu00] AQUILA Home Page: http://www.ist-aquila.org

[Asg02] A. Asgari, et al.: “A Scalable Real-Time Monitoring System for Supporting Traffic Engineering”, IEEE Workshop on IP Operations and Management (IPOM 2002), Dallas, USA, November 2002

[Aud91] N.C. Audsley, A. Burns, M.F. Richardson, A.J. Wellings: “Hard Real-Time Scheduling: The Deadline-Monotonic Approach”, 8th IEEE Workshop on Real-Time Operating Systems and Software, 1991

[Ba01] A. Bak, et al.: “Traffic Handling in AQUILA QoS IP Network” 2nd International Workshop on Quality of future Internet Services QofIS, Coimbra, Portugal, September 2001

[Bak01] F. Baker, C. Iturralde, F. Le Faucheur, B. Davie: „Aggregation of RSVP for IPv4 and IPv6 Reservations“, RFC 3175, September 2001

[BB01] C. Boutremans, J.-Y. Le Boudec: “Adaptive Delay Aware Error Control For Internet Telephony”, Internet Telephony Workshop 2001, New York, USA, April 2001

[BCP00] G. Bianchi, A. Capone, C. Petrioli: “Throughput Analysis of End-to-End Measurement-based Admission Control in IP”, IEEE INFOCOM 2000, Tel Aviv, Israel, März 2000

[BCS94] R. Braden, D. Clark, S. Shenker: “Integrated Services in the Internet Architecture: an Overview”, IETF Standard, RFC 1633, Juni 1994.

[BCS02] F. Baker, K. Chan, A. Smith: “Management Information Base for the Differentiated Services Architecture”, RFC 3289, May 2002

[Ber94] J. Beran: “Statistics for Long-Memory Processes“, Chapman & Hall, New York, USA, 1994

[Ber01] L. Berger, D. Gan, et al.: “RSVP Refresh Overhead Reduction Extensions“, IETF Standard, RFC 2961, April 2001

[Bla98] S. Blake, D. Black, et al.: “An Architecture for Differentiated Services”, IETF Standard, RFC 2475, Dezember 1998

[Bil86] P. Billingsley: „Probability and Measure“, 2nd ed., Wiley, New York, USA, 1986

[Bra65] P. Brady: “A Techniques for investigating ON-OFF Patterns in Speech”, Bell System Technical Journal, 44: 1-21, Januar 1965

[Bra97] R. Braden, et al.: “Resource Reservation Protocol (RSVP) Version 1 Funcional Specification”, RFC 2205, September 1997

[Bran02] C. Brandauer, et al.: “AC Algorithms in AQUILA QoS IP Networks”, 9th Polish Teletraffic Symposium, 2002”, 2nd Polish-German Teletraffic Symposium PGTS’02, Gdansk, Polen, September 2002

[Bre73] R. Brent: “Algorithms for Minimization without Derivatives”, Englewood Cliffs, Prentice-Hall, New York, USA, 1973

[Bre00] D. Bretthauer: “Modellierung von Echtzeitquellen mit fr-ARIMA Modellen“, Diplomarbeit am Lehrstuhl für Kommunikationsnetze, Technische Universität München, 2000

[Cam02] G. Camarillo, W. Marshall, J. Rosenberg: “Integration of Resource Management and SIP”, IETF Draft, draft-ietf-sip-manyfolks-resource-07.txt, April 2002

[Cam03] M. Campanella et al.: “Multidomain End to End IP QoS and SLA”, 2nd Workshop Quality of Service in Multiservice IP Networks (QoS-IP 2003), Mailand, Italy, Februar 2003

[CGR00] F. Cuervo et al.: “Megaco Protocol Version 1.0”, IETF- Standard, RFC 3015, November 2000

[Ch98] P. Chandra et al.: „Darwin: A Customizable Resource Management for Value-Added Network Services“, 6th IEEE International Conference on Network Protocols, Austin, Texas, USA, Oktober 1998

[Cha01] K. Chan, J. Seligson, et al.: “COPS Usage for Policy Provisioning (COPS-PR)”, IETF-Standard, RFC 3084, März 2001

[Cha98] P. Chandra, A. Fischer, C. Kosak, P. Steenkiste: „Network Support for Application-Oriented QoS“, 6th International Workshop on Quality of Service, Napa, California, USA, Mai 1998

[CHH02] M. Conradt, T. Hagenmaier, S. Hodes: „Konzeptionierung und Implementierung eines Tools zur Erkennung der Topologie von IP-Netzen“, Interdisziplinäres Projekt am Lehrstuhl für Kommunikationsnetze, Technische Universität München, 2002

[CJ99] S. Casner, V. Jacobson: “ Compressing IP/UDP/RTP Headers for Low-Speed Serial Links”, IETF Standard, RFC 2508, Februar 1999

[CK00] C. Cetinkaya, E. Knightly: “Egress Admission Control”, IEEE INFOCOM 2000, Tel Aviv, Israel, März 2000

[Clo03] K McCloghrie, M. Fine, et al.: “Structure of Policy Provisioning Information (SPPI)“, IETF Standard, RFC 3159, Februar 2003

[CN98] S. Chen, K. Nahrstedt: “An Overview of Quality-of-Service Routing for the Next Generation HighSpeed Networks: Problems and Solutions”, IEEE Network Magazine, Vol. 12, S. 64-79, Nov.-Dezember 1998

[Con03] M. Conradt, „Erweiterung eines Topologiemanagement-Tools zur Ressourcenverwaltung in IP-Netzen“, Diplomarbeit am Lehrstuhl für Kommunikationsnetze, Technische Universität München, 2003

[Cor03] G. Cortese, et al.: “CADENUS: Creation and Deployment of End-User Services in Premium IP Networks”, IEEE Communications Magazine, Vol. 41, No. 1, Januar 2003

[CPS99] K. McCloghrie, D. Perkins, J. Schoenwaelder: “Structure of Management Information Version 2 (SMIv2) “, IETF Standard, RFC 2578, April 1999

[CR90] K. McCloghrie, M. Rose: “Management Information Base for Network Management of TCP/IP-based Internets: MIB-II”, IETF RFC 1157, Mai 1990

[CS98] F.M. Chiussi, V. Sivaraman: “Achieving High Utilization in Guaranteed Services Networks Using Early-Deadline-First Scheduling”, 6th IEEE/IFIP IWQoS’98, Napa, California, USA, Mai 1998

[CW96] C. Courcoubetis, R. Weber, “Buffer Overflow Asymptotics for a Buffer Handling many Traffic Sources”, J. of Applied Probability, Vol. 33, 1996

[Cze00] M. Czermin, “Vergleich mehrer Verfahren zur Bestimmung Effektiver Bitraten für Echtzeitverkehre in Intranets”, Diplomarbeit am Lehrstuhl für Kommunikationsnetze, Technische Universität München, Juli 2000

[Dav02] B. Davie, A. Charny, et al.: “An Expedited Forwarding PHB (Per-Hop Behaviour)”, IETF Standard, RFC 3246, März 2002

[DC02] C. Demichelis, P. Chimento: “IP Packet Delay Variation Metric for IPPM“, IETF-Standard, RFC 3393, November 2002

[DNP99] M Degermark, B. Nordgren, S. Pink: „IP Header Compression“, IETF Standard, RFC 2507, Februar 1999

[Ebe01] J. Eberspächer: “Die Zukunft des Internet im Lichte von Konvergenz”, Internet@Future, Jahrbuch der Telekommunikation und Gesellschaft 2001, Hrsg. H. Kubicek, Hüthig Verlag, Heidelberg, S.17-27, 2001

[EDP92] J. Escobar, D. Deutsch, C. Partridge: „Flow Synchronization Protocol“, IEEE GLOBECOM’92, Orlando, Florida, USA, Juni 1992

[Ele00] V. Elek, G. Karlsson, R. Ronngren: „Admission Control based on End-to-End Measurements“, IEEE INFOCOM 2000, Tel Aviv, Israel, März 2000

[Elv95] A. Elvalid et al.: “Fundamentals Bounds and Approximations for ATM Multiplexers with Applications to Video Teleconferencing”, IEEE JSAC, Vol. 13, No. 6, S. 1004-1016, August 1995

[EM93] A. Elvalid, D. Mitra: “Effective Bandwidth of General Markovian Traffic Sources and Admission Control of High-Speed Networks“, IEEE/ACM Trans. Networking, Vol. 1, S. 329-43, 1993

[EMW95] A. Elvalid, D. Mitra, R. Wentworth: “A New Approach for Allocating Buffers and Bandwidth to Heterogeneous, Regulated Traffic in an ATM Node”, IEEE JSAC, Vol. 13, No. 6, S. 1115-1127, August 1995

[Eng03] T. Engel, et al.: “AQUILA: Adaptive Resource Control for QoS Using an IP-based Layered Architecture”, IEEE Communications Magazine, Vol. 41, No. 1, Januar 2003

[ENW96] A. Erramilli, O. Narayan, W. Willinger, “Experimental Queueing Analysis with Long-Range Dependent Packet Traffic”, IEEE/ACM Transactions on Networking, Vol.4, No.2, April 1996

[Fad03] M. MacFaden, D. Partain, J. Saperia, W. Tackabury: “Configuring Networks and Devices With SNMP”, IETF-Draft, draft-ietf-snmpconf-bcp-12.txt, Januar 2003

[Feh99] G. Feher et al.: “Boomerang – A Simple Protocol for Resource Reservation in IP Networks“, IEEE Workshop on QoS Support for Real-Time Internet Applications, Vancouver, Canada, Juni 1999

[Fer00] T. Ferrari: “End-To-End Performance Analysis with Traffic Aggregation”, TERENA Networking Conference, TNC'00, Lissabon, Portugal, Mai 2000

[FPR00] T. Ferrari, G. Pau, C. Raffaelli: “Priority Queuing Applied to Expedited Forwarding: a Measurement-Based Analysis”, 1st Int. Workshop on Quality of future Internet Services, QofIS'2000, Berlin, Deutschland, März 2000

[FJ93] S. Floyd, V. Jacobson: “Random Early Detection gateways for Congestion Avoidance”, IEEE/ACM Transactions on Networking, V.1 N.4, S. 397-413, August 1993

[FJ95] S. Floyd, V. Jacobson: “Link-sharing and Resource Management Models for Packet Networks”, IEEE/ACM Transactions on Networking, Vol. 3, No. 4, August 1995

[FKT97] V. Firoiu, J. Kurose, D. Towsley: “Efficient Admission Control for EDF Schedulers”, INFOCOM '97, Kobe, Japan, April 1997

[FM02] M. Fine, K. McCloghrie, et al.: “Differentiated Services Quality of Service Policy Information Base”, IETF-Draft, draft-ietf-diffserv-pib-09.txt, Juni 2002

[Fos99] I. Foster, C. Kesselman, C. Lee, R. Lindell, K. Nahrstedt, A. Roy: “A Distributed Resource Management Architecture that Supports Advance Reservations and Co-Allocation”, Intl. Workshop on Quality of Service, 1999

[G.107] ITU-T Recommendation G.107: “The E-Model - A Computational Model for Use in Transmission Planning”, Geneva, Switzerland, November 1998

[GAN91] R. Guerin, H. Ahmadi, M. Naghshineh: “Equivalent Capacity and its Application to Bandwidth Allocation in High-Speed Networks”, IEEE JSAC, Vol. 9, S. 968-981, September 1991

[Geo96] L. Georgiadis, R. Guerin, V. Peris, R. Rajan, “Efficient Support or Delay and Rate Guarantees in an Internet”, SIGCOMM, San Franzisco, S. 106-116, August 1996

[GGP97] L. Georgiadis, R. Guerin, A. Parekh: “Optimal Multiplexing on a Single Link: Delay and Buffer Requirements”, IEEE Trans. Infor. Theory 43, S. 1518-1535, September 1997

[Gib96] R.J. Gibbens: “Traffic characterisation and effective bandwidths for broadband network traces”, Research Report 1996-9, Stat. Lab., University of Cambridge, 1996

[Gio96] S. Giordano, et al.: “A new Call Admission Control Scheme Based on the Self Similar Nature of Multimedia Traffic“, IEEE ICC '96, Dallas, USA, Juni 1996

[GK92] R.J. Gibbens, F.P. Kelly: “Measurement-based Connection Admission Control”, 5th Int. Teletraffic Congress ITC, Teletraffic Scienece and Engineering, Elsevier Science B.V., S.879-888, 1997

[Gla00] J. Glasmann, P. Hierholzer, K. Klaghofer, H. Müller, C. Prehofer: “Abstraktion der Netztopologie für die Bandbreitenkontrolle in Paketnetzen“, Patentanmeldung, Deutsches Patentamt, Mai 2000

[Gla02] J. Glasmann, H. Müller, J. Totzke, M. Tromparent: “Verfahren zur adaptiven Bandbreitenkontrolle bei Topologieänderungen in paketorientierten Netzen“, Patentanmeldung, Deutsches Patentamt, August 2002

[GKM01] J. Glasmann, W. Kellerer, H. Müller: “Service Development and Deployment in H.323 and SIP”, 6th IEEE Symposium on Computers and Communications (ISCC) 2001, Tunisia, Juli 2001

[Gli00] R. Glitho: “Advanced Service Architectures for Internet Telephony: A Critical Overview”, IEEE Network Magazine, S. 38-44, July/August 2000

[GM02] J. Glasmann, H. Müller: “Resource Management Architecture for Realtime Traffic in Intranets”, Networks 2002, Joint IEEE International Conferences ICN and ICWLHN, Atlanta, USA, August 2002

[GME02] J. Glasmann, H. Müller, J. Eberspächer: „Ressourcen-Management Architektur für Echtzeitverkehre in Intranets“, Praxis der Informationsverarbeitung und Kommunikation PIK, Hsg. Prof. Dr. Hans W. Meuer, Saur-Verlag, München, Vol. 2, 2002

[GPS96] L. Georgiadis, R. Guerin, V. Peris, K. Sivarajan: “Efficient Networks QoS Provisioning Based on per Node Traffic Shaping”, IEEE Trans. on Networking, S. 482-501, August 1996

[Gra02] Graphviz: http://www.research.att.com/sw/tools/graphviz/

[GRC00] J. Glasmann, A. Riedl, M. Czermin: “Estimation of Token Bucket Parameters for Videoconferencing Systems in Corporate Networks”, SOFTCOM’00, Split, Oktober 2000

[Gru91] R. Grunenfelder et al.: “Characterization of Video as Autoregressive Moving Average Processes and Realted Queuing Systems Performance”, IEEE JSAC, Vol. 9, No. 3, S. 284-293, 1991

[GT02] J. Glasmann, M. Tromparent: “Topology Discovery in the Context of Resource Management in IP-Networks”, 10th International Conference on Software, Telecommunications and Computer Networks, SoftCOM‘02, Split, Oktober 2002

[GW94] M. Garrett, W. Willinger: “Analysis, Modeling and Generation of Self-Similar VBR Video Traffic“, ACM SIGCOM’94, London, UK, 1994

[H.225] ITU-T Recommendations H.225.0: “Call Signalling Protocols and Media Stream Packetization for Packet-Based Multimedia Communication Systems”, Geneva, Switzerland, Februar 1998

[H.245] ITU-T Recommendations H.450: “Generic Functional Protocol for the Support of Supplementary Services in H.323”, Geneva, Switzerland, 1998

[H.323] ITU-T Recommendation H.323: “Packet-Based Multimedia Communications Systems”, Geneva, Switzerland, 1998

[Han03] R. Hancock et al.: “Next Steps in Signaling: Framework”, NSIS Working Group, draft-ietf-nsis-fw-02.txt, März 2003

[Hei99] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski: “Assured Forwarding PHB Group”, IETF Standard, RFC 2597, Juni 1999

[HJ98] M. Handley, V. Jacobson: “SDP: Session Description Protocol”, IETF RFC 2327, April 1998

[HL86] H. Heffes, D. Lucantoni: “A Markov modulated Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance”, IEEE JSAC, Vol. 4, S. 856-868, 1986

[HR89] J. Haslett, A. Raftery: ”Space-time Modeling with Long-Memory Dependence: Assessing Ireland’s wind power Resources” Journal of the Royal Statistical Society, Applied Statistics, Vol. 38, S. 1-50, 1989

[HSS99] M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg: “SIP: Session Initiation Protocol”, IETF RFC 2543, März 1999

[HTL91] D. Heyman, A. Tabatabai, T. Lakshman: “Statistical Analysis and Simulation Study of Video Teleconferencing Traffic in ATM Networks, IEEE GLOBECOM’91, S. 21-277, 1991

[Hui90] J. Y. Hui: “Switching and Traffic Theory for Integrated Broadband Networks“, Kluwer, Boston, Massachusetts, USA, 1990

[I802.3] IEEE 802.3: “CSMA/CD (Ethernet)”, IEEE Standard, Last Update 2002

[Jam95] S. Jamin, et al.: “Comparison of Measurement-based Admission Control Algorithms for Controlled-Load Service”, IEEE INFOCOM '97, April 1997

[JS00] W. Jiang, H. Schulzrinne: “Analysis of On-Off Patterns in VoIP and Their Effect on Voice Traffic Aggregation”, 9th IEEE International Conference on Computer Communication Networks, 2000

[Kan96] H. Kaneko, J. Stankovic, S. Sen, K. Ramamritham: “Integrated Scheduling of Multimedia and Hard Real-Time Tasks”, 17th IEEE Real-Time Systems Symposium (RTSS '96), Washington D.C., USA, Dezember 1996

[Kel91] F. Kelly: “Effective bandwidths at multi-class queues”, Queuing Systems, Vol. 9, S. 5-16, 1991

[Kel96] F. Kelly, S. Zachary and I.B. Ziedins: “Notes on effective bandwidths”, Stochastic Networks: Theory and Applications, Royal Statistical Society Lecture Notes Series, Vol. 4, S. 141-168, Oxford University Press, 1996

[Kli98] O. Klinger: “Modellierung von Echtzeitverkehrsquellen im LAN“, Diplomarbeit am Lehrstuhl für Kommunikationsnetze, Technische Universität München, 1998

[Kni01] P. Yuan, J. Schlembach, A. Skoe, E. Knightly: „Design and Implementation of Scalable Edge-Based Admission Control“, International Workshop on QoS in Multiservice IP Networks (MQoS '01), Rome, Italy, Januar 2001

[KS99] E. Knightly, N. Shroff: “Admission Control for Statistical QoS: Theory and Practice”, IEEE Networks Magazine, S. 20-29, März/April 1999

[KZZ96] F.P. Kelly, S. Zachary and I.B. Ziedins: “Notes on effective bandwidths”, In "Stochastic Networks: Theory and Applications", Royal Statistical Society Lecture Notes Series, 4, Oxford University Press, S. 141-168, 1996

[Lee91] D. Lee, B. Melamed, A. Reibman, B. Sengupta: “Analysis of a Video Multiplexer using TES as a Modeling Methodology”, IEEE GLOBECOM’91, S. 16-19, 1991

[Lel94] W.E. Leland, M. Taqqu, W. Willinger, D. Wilson: “On the Self-Similar Nature of Ethernet Traffic“, IEEE/ACM Trans. on Networking, vol. 2, no. 1, Februar 1994

[LG90] T. Little, A. Ghafoor: “Synchronization and Storage Models for Multimedia Objects”, IEEE Journal of Selected Areas in Communications, Vol. 8, No. 3, S. 413-427, April 1990

[Lin94] K. Lindberger: “Dimensioning and Design Methods for Integrated ATM Networks”, The Fundametal Role of Teletraffic in the Evolution of Telecommunications Networks, ITC-13, Hrsg. A. Jensen, S. 807-813, 1988

[Lip02] lipcap: http//www.tcpdump.org/; wincap: http.//netgroup-serv.polito.it/wincap/

[LL96] L. Lamont, L. Li, et al.: “Synchronization of Multimedia Data for a Multimedia News-on-Demand Application”, IEEE Journal of Selected Areas in Communications, 1996

[LM97] D. Linn, R. Morris: “Dynamics of Random Early Detection”, ACM SIGCOMM, S. 127-137, Sophia Antipolis, Frankreich, September 1997

[LMS02] D. Levi, P. Meyer, B. Stewart: “A Simple Network Management Protocol (SNMP)”, IETF, RFC 3413, Dezember 2002

[LW94] J. Liebeherr, D. Wrege: „Design and Analysis of a High-Performance Packet Multiplexer for Multiservice Networks with Delay Guarantees“, Technical Report CS-94-29, University of Virginia, 1994

[MSA] C. Courcoubetis, V.A. Siris. ”Measurement and analysis of real network traffic”, 7th Hellenic Conference on Informatics (HCI'99), Ioannina, Greece, August 1999, http://www.ics.forth.gr/netgroup/msa/

[Net02] Net-SNMP: http://www.net-snmp.org/

[Nic02] K. Nichols, S. Blake, F. Baker, D. Black: “Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers”, IETF Standard, RFC 2474, Dezember 1998

[NJZ99] K. Nichols, V. Jacobson, L. Zhang: “A Two-bit Differentiated Services Architecture for the Internet“, IETF RFC 2638, Juli 1999

[Nor94] I. Norros: “A Storage Model with Self-Similar Input”, Queuing Systems, Vol. 16, S. 387-396, 1994

[NS01] Network Simulator Berkeley, NS-2, http://www.isi.edu/nsnam/ns/

[OpH323] Open H.323 Project, http://www.openh323.org

[Oht94] N. Ohta: “Packet Video, Modeling and Signal Processing”, Artech House Publisher, Boston, USA, 1994

[PE96] H. Perros, K. Elsayed: “Call Admission Control Schemes: A Review”, IEEE Communications Magazine, S. 82-91, November 1996

[PGC97] K. Park, G. Kim, M. Crovella: “On the Effect of Traffic Self-similarity on Network Performance”, SPIE-The International Society for Optical Engineering, San Diego, California, USA, 1997

[PGM00] C. Prehofer, J. Glasmann, H. Müller: “Scalable Resource Management Architecture for VoIP”, 5th International Conference on Protocols for Multimedia Systems, PROMS 2000, Krakau, Oktober 2000

[PHS00] P. Pan, E. Hahne, H. Schulzrinne: “BGRP: Sink-Tress-Based Aggregation for Inter-Domain Reservations“, KICS, Journal of Communications and Networks, Vol. 2, No. 2, Juni 2000

[PS98] P. Pan, H. Schulzrinne: “YESSIR: A Simple Reservation Mechanism for the Internet”, International Workshop on Network and Operating System Support for Digital Audio and Video NOSSDAV, Cambridge, England, S. 141-151, Juli 1998

[QB99] B. Teitelbaum, et al.: “Internet2 Qbone: Building a Testbed for Differentiated Services”, IEEE Network, Vol. 13, No. 5, S. 8-16, September/Oktober 1999

[REH99] R. Rejaie, D. Estrin, M. Handley: “Quality Adaptation for Congestion Controlled Video Playback over the Internet”, SIGCOMM, 1999

[RF95] A. Romanow, S. Floyd: “Dynamics of TCP Traffic over ATM Networks”, IEEE JSAC, V. 13 N. 4, S. 633-641, Mai 1995

[RG02] A. Riedl, J. Glasmann: “On The Design of Resource Management Domains”, International Symposium on Performance Evaluation of Computer and Telecommunication Systems, SPECTS 2002, San Diego, USA, Juli 2002

[RH95] K. Rothermel, T. Helbig: “An Adaptive Stream Synchronization Protocol”, Network and Operating System Support for Digital Audio and Video, 1995

[Rie02] A. Riedl: “Hybrid Genetic Algorithm for Routing Optimization in IP Networks Utilizing Bandwidth and Delay Metrics “, IEEE Workshop on IP Operations and Management (IPOM’02), Dallas, USA, Oktober 2002

[RK96] A. Rueda, W. Kinsner: “A Survey of Traffic Characterization Techniques in Telecommunication Networks“, IEEE Canadian Conference on Electrical and Computer Engineering, Calgary, Canada, Vol. 2, S. 830-833, Mai 1996

[RR97] G. Ramamurthy, Q. Reng: “Multiclass Connection Admission Control Policy for High Speed ATM Switches”, IEEE INFOCOM’97, S. 964-974, Kobe, Japan, März 1997

[RS94] K. Ramamritham, J. Stankovic: “Scheduling Algorithms and Operating Systems Support for Real-Time Systems”, Proceedings of the IEEE, Vol. 82, No. 1, Januar 1994

[Sch96] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson: „RTP: A Transport Protocol for Real-Time Applications”, IETF RFC 1889, Januar 1996

[Sc96] H. Schulzrinne: „RP Profile for Audio and Video Conferences with Minimal Control“, IETF Standard, RFC 1890, Januar 1996

[ScP98] O. Schelén, S. Pink: “Resource Reservation Agents in the Internet”, 8th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'98), Cambridge, UK, Juli 1998

[SP98] O. Schelén, S. Pink: “Resource Sharing in Advance Reservation Agents”, Journal of High Speed Networks, Special Issue on Multimedia Networking, Vol. 7, No. 3-4, 1998

[SPG97] S. Shenker, C. Partridge, R. Guerin: “Specification of Guaranteed Quality of Service”, IETF Standard, RFC 2212, September 1997

[SPL01] SPLUS, Version 4.5, MathSoft Inc., Seattle, Washington, USA, 2001

[SRG03] S. Sharafeddine, A. Riedl, J. Glasmann, J. Totzke: “On Traffic Characteristics and Bandwidth Requirements of Voice over IP Applications”, 8th IEEE Symposium on Computers and Communications (ISCC) 2003, Antalya, Turkey, Juni 2003

[SS98] S. Schäffer, W. Schilder: „Verkehrsmessung und Modellierung“, Interdisziplinäres Projekt am Lehrstuhl für Kommunikationsnetze, Technische Universität München, 1998

[SSZ98] I. Stoica, S. Shenker, H. Zhang: „Core-stateless Fair Queuing: A Scalable Architecture to Approximate Fair Bandwidth Allocations in High-Speed Networks”, ACM SIGCOM, Vancouver, Canada, August 1998

[Ste90] R. Steinmetz: “Synchronization Properties in Multimedia Systems”, ”, IEEE Journal of Selected Areas in Communications, Vol. 8, No. 3, S. 401-412, April 1990

[STL02] SGI STL: http://www.sgi.com/tech/stl/

[Str01] J. Strassner, E. Ellesson, B. Moore, A. Westerinen: “Policy Core Information Model -- Version 1 Specification", IETF-Standard, RFC 3060, Februar 2001

[SW97] S. Shenker, J. Wroclawski: “General Characterization Parameters for Integrated Service Network Elements”, IETF Standard, RCF 2215, September 1997

[SZ95] C. Schmidt, M. Zitterbart: “Towards Integrated QoS Management“, 5th IEEE Workshop on Future Trends of Distributed Computing Systems, Chenju, Korea, August, 1995

[SZ99] I. Stoica, H. Zhang: „Providing Guaranteed Services Without Per Flow Management“, ACM SIGCOM, Cambridge, Massachusetts, September 1999

[SZE97] I. Stoica, H. Zhang, T. Eugene: “A Hierarchical Fair Service Curve Algorithm for Link-Sharing, Real-Time and Priority Service“, SIGCOMM’97, Symposium on Communications Architectures and Protocols, S. 249-262, Cannes, September 1997

[Tan02] A. Tanenbaum: “Computer Networks“, 4. Auflage, Prentice Hall, New York, USA, August 2002

[TEQ98] TEQUILA-Project: IST-1999-11253, http://www.ist-tequila.org

[Tip00] ETSI Technical Specification ETSI TS 101 329-2 v. 2.2.2: Telecommunications and Internet Protocol Harmonization over Networks (TIPHON): “End-to-End Quality of Service in TIPHON Systems”, Part 2, ETSI, Juli 2000

[TGM03] M. Tromparent, J. Glasmann, H. Müller, J. Totzke: „Admission Control Strategies in Transient Network States”, 10th International Conference on Telecommunications, ICT '2003, Papeete, Tahiti, Februar 2003

[Tri01] P. Trimintzios et al.: “A Management and Control Architecture for Providing IP Differentiated Services in MPLS-based Networks”, IEEE Communications Magazine, Vol. 39, No. 5, Mai 2001

[Tro01] M. Tromparent: „Prototypische Implementierung eines Ressourcen-Managers“, Diplomarbeit am Lehrstuhl für Kommunikationsnetze, Technische Universität München, 1/2001

[TZ99] W. Tan, A. Zakhor: “Real-Time Internet Video Using Error Resilient Scalable Compression and TCP-Friendly Transport Protocol”, IEEE Transactions on Multimedia, 1999

[TZ01] U. Thürmann, M. Zitterbart: “IP-Telefonie über Differentiated Services“, Praxis der Informationsverarbeitung und Kommunikation PIK, Hsg. Prof. Dr. Hans W. Meuer, Saur-Verlag, München, Vol. 1, 2001

[Ven97] B. Venkateshwara: “Performance of Finite-Buffer Queues under Traffic with Long-Range Dependence”, IEEE GLOBECOM’96, Vol. 1, S. 607-611, November 1996

[Wal00] M. Wallner: “Bestimmung effektiver Bitraten für Echtzeitverkehre in IP-Netzen“, Diplomarbeit am Lehrstuhl für Kommunikationsnetze, Technische Universität München, 2000

[WC96] Z. Wang, J. Crowcroft: “Quality of Service Routing for Supporting Multimedia Applications”, IEEE Journal of Selected Areas in Communications, Vol. 14, Issue 07, September 1996

[WC00] D. Wu, H.J. Chao: “Efficient bandwidth allocation and call admission control for VBR service using UPC parameters”, Int. J. Com. Systems, Vol. 13, no. 1, S. 29-50, John Wiley, Februar 2000

[Wil97] W. Willinger, M. Taqqu, R. Sherman, D. Wilson: “Self-Similarity through High-Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level”, IEEE Trans. on Networking, Vol. 5, USA, S. 71 – 86, 1997

[WL96] D.E. Wrege, J. Liebeherr: "Video Traffic Characterization for Multimedia Networks with a Deterministic Service", IEEE INFOCOM '96, San Francisco, USA, S. 537-544, März 1996

[Wr97] J. Wroclawsk: “The Use of RSVP with IETF Integrated Services”, IETF Standard, RFC 2210, September 1997

[Wro97] J. Wroclawsk: “Specification of the Controlled-Load Network Element Service”, IETF Standard, RFC 2211, September 1997

[WSH01] S. Waldbusser, J. Saperia, T. Hongal: ”Policy Based Management MIB”, IETF-Draft, draft-ietf-snmpconf-pm-12.txt, Februar, 2003

[ZF94] H. Zhang, D. Ferrari: ”Rate-Controlled Service Disciplines”, J. of High Speed Networks, S. 389-412, 1994

[Zh95] H. Zhang: “Providing End-to-End Performance Guarantees Using Non-Work-Conserving Disciplines”, Computer Communications: Special Issue on System Support for Multimedia Computing, 18(10), Oktober 1995

[Zha95] H. Zhang: “Service Disciplines for Guaranteed Performance in Packet-Switching Networks”, Proceedings of the IEEE, Vol. 83, No. 10, S. 1373-1396, Oktober 1995