Ein System zur Übertragung multimedialer Echtzeitdatenströme über Internet-Zugangsnetze

Vom Fachbereich Elektrotechnik und Informationstechnik

der Universität Hannover

zur Erlangung des akademischen Grades

Doktor-Ingenieur

genehmigte

Dissertation

von

Dipl.-Ing. Lutz Grüneberg

geboren am 25. Oktober 1963 in Hannover

1998

1. Referent:	Prof. Dr.-Ing. Helmut Pralle
2. Referent:	Prof. Dr.-Ing. Klaus Jobmann
Tag der Promotion:	27. Mai 1998

Vorwort

Die vorliegende Arbeit entstand während meiner Assistententätigkeit am Lehrgebiet Rechnernetze und Verteilte Systeme der Universität Hannover.

Mein besonderer Dank gilt Herrn Prof. Dr.-Ing. Helmut Pralle für die Anregung zum Thema und die wertvollen Diskussionen. Die hervorragenden Arbeitsmöglichkeiten am Insititut haben wesentlich zum Gelingen beigetragen.

Herrn Prof. Dr.-Ing. Klaus Jobmann danke ich für die spontane Übernahme des Koreferats und für sein Interesse an meiner Arbeit.

Mein Dank gilt ebenfalls allen Kollegen am Institut und den Mitarbeitern der Gruppe Kommunikationssysteme am Regionalen Rechenzentrum der Universität Hannover, die auf die eine oder andere Weise einen Beitrag zum Entstehen und Gelingen der Arbeit geleistet haben. Hervorheben möchte ich hier die Herren Dipl.-Ing. Michael Fromme, Dr.-Ing. Fritz Hüsemann und Dr. rer. nat. Bernd-Uwe Pagel (FernUniversität Hagen), die mir mit ihrer kritischen Durchsicht des Manuskripts und den fruchtbaren Diskussionen bei der Erstellung dieser Arbeit sehr geholfen haben.

Hannover, im Mai 1998

Lutz Grüneberg

Kurzfassung

In zunehmenden Maße wird das Internet für die Übertragung multimedialer Datenströme genutzt. Diese Anwendungen erzeugen Datenströme mit grundlegend anderer Charakteristik als klassische Anwendungen in Rechnernetzen wie Datei-Übertragung oder entfernter Rechnerzugriff. Sie stellen damit veränderte Anforderungen an Infrastruktur und Transportprotokolle.

In besonderem Maße trifft dies für Datenströme zu, die der interaktiven interpersonellen Kommunikation dienen. Hierbei ist die Begrenzung der Paketlaufzeit von übergeordneter Bedeutung. Besonders im Bereich von Internet-Zugangsnetzen mit geringen Datenraten sowie entsprechender Zugangsdienste wird diese Forderung nur selten erfüllt.

In der vorliegenden Arbeit wird ein neues Konzept zur Übertragung multimedialer Echtzeitdatenströme über Internet-Zugangsdienste geringer Datenrate vorgestellt. Es ermöglicht Nutzern an Heimarbeitsplätzen, die über einen ISDN-B-Kanal mit dem Internet verbunden sind, die Teilnahme an multimedialen Multipoint-Videokonferenzen im Internet.

Der Ansatz hebt sich von vergleichbaren Systemen dadurch ab, daß der Aufbau von Stauungen in vermittelnden Komponenten des Netzpfades zwischen Internet und Heimarbeitsplatz in Überlastsituationen vermieden wird. Dies begrenzt die Paketlaufzeiten und Paketlaufzeitschwankungen nachhaltig.

Den Schwerpunkt der Arbeit bilden der Entwurf und die Implementierung des *MBone Access Gateway* (MAGW), eines verteilten Gateways auf Anwendungsebene zur Realisierung dieses Konzepts. Das MAGW unterstützt die Übertragung von Datenströmen im Format des *Real-Time Transport Protocol* (RTP) und des *User Datagram Protocol* (UDP).

Das Gateway besteht aus zwei Diensten, von denen sich einer auf einem Rechner im regulären Internet mit direktem Zugriff auf den Multicast-Backbone im Internet (MBone) befindet. Seine Partnerinstanz ist auf dem Rechner des Nutzers am Heimarbeitsplatz oder auf einem Rechnersystem in einem privaten Internet installiert. Die Komponenten des Systems sind durch zwei Datenverbindungen verknüpft. Eine Verbindung dient der Authentifizierung und der Anforderung der zu übertragenden Echtzeitdatenströme. Die zweite Verbindung wird zur Übertragung der Nutzdaten sowie zum Austausch von Informationen zum aktuellen Lastzustand des Netzpfades genutzt. Letztgenannte wird als Tunnel bezeichnet.

Das in der Arbeit entwickelte Regelungsverfahren basiert auf der Abschätzung der Verweilzeiten von Paketen in den Warteschlangen vermittelnder Komponenten durch den Nachrichten-Empfänger. Dabei wird die von der Paketgröße abhängige Serialisierungszeit der Pakete berücksichtigt. Die so gewonnenen Informationen über den aktuellen Lastzustand des Netzpfades

dienen als Grundlage für eine dynamische Datenratenregelung der zu übertragenden Echtzeitdatenströme beim Nachrichten-Sender.

Zur effizienten Nutzung des Tunnels zwischen den verteilten Komponenten des Systems werden die Pakete der Echtzeitdatenströme einer Header-Kompression sowie, abhängig vom Charakter des Datenstroms, optional einer verlustbehafteten oder verlustfreien Kompression unterzogen. Dadurch wird die Datendurchsatzrate des Tunnels erhöht, die Verzögerung der Pakete durch das MAGW verringert und die aus der Kapselung der Nutzdatenpakete in UDP-Pakete resultierenden, nachteiligen Folgen bezüglich Durchsatzrate und Verzögerung mehr als ausgeglichen.

Für die Vermittlung von Datenströmen aus dem Internet in private Internets unterstützt das MAGW die Umsetzung von Adressen des privaten Internets in Adressen des regulären Internets. Ergänzend ist der Betrieb des Systems über Firewalls vorgesehen.

Das System gestattet dem Nutzer am Heimarbeitsplatz die etablierten Medien-Werkzeuge im Kontext des MBone ohne Modifikation zu nutzen. Die zusätzlich erforderliche Kommunikation mit dem MAGW wird von Hilfsprogrammen übernommen, die anstelle der originären Medienwerkzeuge gestartet werden. Ihre ausschließliche Aufgabe besteht in der Anforderung der erforderlichen Datenströme aus dem Internet und dem Start des entsprechenden Werkzeuge.

Konzeptionell unterstützt das MAGW die vertrauliche Kommunikation durch Verschlüsselung der Datenströme mit einem abgestuften Verfahren. Die vorliegende Implementierung des MAGW realisiert allerdings noch nicht alle Varianten dieses Verfahrens.

Obwohl das Augenmerk bei der Entwicklung des MAGW bei der Nutzung von ISDN-basierten Internet-Zugangsdiensten lag, ist das System auch über andere Zugangsnetze nutzbar. Wesentlichen Anteil an der damit verbundenen Flexibilität hat die Beschränkung auf die Nutzung der Internet-Transportprotokolle TCP und UDP.

Wenngleich der Schwerpunkt auf der Nutzung des MBone und der MBone-Werkzeuge liegt, lassen sich die Konzepte weitgehend auf H.323-basierte Systeme übertragen, da auch hier das in RFC 1889 spezifizierte Real-Time-Transport-Protocol (RTP) Verwendung findet.

Schlagworte: Internet-Protokolle, MBone, Multimediale Konferenzen

Abstract

The Internet is increasingly used for the transport of multimedia data streams showing characteristics different from streams induced by conventional internet applications like file transfer and remote host access. Additional features are required for the infrastructure and transport protocols. This especially holds for data streams caused by interactive interpersonal communication for which the limitation of packet transmission time is of central interest. Current Internet access networks and services rarely fulfil these requirement.

This thesis presents a new concept for the transmission of real-time multimedia data streams over low-speed serial links supporting interactive interpersonal communication. It enables a homeworker who is linked to the Internet via a single ISDN-B-Channel to participate in multimedia multipoint conferences on the Internet. Compared to other approaches the variation of packet transmission time is reduced by avoiding queue congestion in routers along the network path between the regular Internet and the homeworker's computer.

The main contribution of this work is the design and implementation of a distributed application-layer gateway, called *MBone Access Gateway* (MAGW). MAGW supports the transmission of data streams using the IETF *Real-time Transport Protocol* (RTP) as well as the *User Datagram Protocol* (UDP). The gateway consists of two services. The first of which is running on a computer system directly connected to the regular Internet and the Multicast-Backbone on the Internet (MBone), the second is installed on the homeworker's computer or on a system in a private Internet. Both components are interconnected by a control connection used for authentication and requesting multimedia data streams as well as a datagram data link responsible for the transmission of real-time multimedia streams and the exchange of information concerning the current load of the network path between the MAGW components. The latter is called tunnel.

The control method to avoid the congestion is based on a queuing delay estimation for packets passing the tunnel. The delay estimation considers the delay caused by queues of active network components as well as the delay resulting from serialization and transmission. The retrieved information about the current load situation serves as a decision base for a dynamic data rate control mechanism.

Protocol header and adaptive payload compression are used to increase the throughput of the tunnel and to minimize the overall transmission delay. Further compression over-compensates the negative impact of UDP encapsulation of the transmitted data on the system performance.

The forwarding of data streams from the Internet to private Internets and vice versa is supported by address mapping. This allows forwarding over Firewalls as well.

The user communicates with the MAGW via so-called wrapper applications which encapsu-

late the original media tools (MBone tools). Hence, it is possible for the homeworker to use unmodified versions of the desired media tools.

The design of MAGW allows private communication by encryption. However, the current MAGW implementation provides only restricted support for private communication.

Although the emphasis of the MAGW design lies on ISDN based internet access networks, the concept is also applicable to access networks with higher data rates. The flexibility of the solution is achieved by the consequent use of the IP transport protocols UDP and TCP which make no assumptions about the lower level protocols.

Moreover, MAGW is not limited to the MBone and the MBone tools. The concept can easily be transferred to ITU H.323 based systems since they deploy RTP for media stream transmission too.

Keywords: Internet protocols, MBone, multimedia conferencing

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Verzeichnis der Abkürzungen

ACFC Address and Control Field Compression

ADPCM Adaptive Delta Pulse Code Modulation

ADSL Asymmetrical Digital Subscriber Line

AOL America Online, privater Online- und Internet-Zugangsdienst

ALF Application Layer Framing

ASCII American Standard Code for Information Interchange

ATM Asynchronous Transfer Mode

B-WiN Deutsches Breitband-Wissenschaftsnetz

bps Bits pro Sekunde

CBC Cypher Block Chaining Mode

CC Contributing Source Count

CHAP Challenge Handshake Authentication Protocol

CNAME Canonical-Name

CSRC Contributing Source Identifiers

DECT Digital Enhanced Cordless Telecommunications

DES Data Encryption Standard

DLPI Data Link Provider Interface

DVI Digital Video Interface, Intel Real-Time Video

DVMRP Distance Vector Multicast Routing Protocol

FIFO First In First Out

FTP File Transfer Protocol

GPS Global Positioning System

GSM Group Speciale Mobile

HDLC High-Level Data Link Control

HDSL High bitrate Digital Subscriber Line

HTTP Hypertext Transfer Protocol

ICMP Internet Control Message Protocol

IETF Internet Engineering Task Force

IGMP Internet Group Management Protocol

ILP Integrated Layer Processing

IP Internet Protocol

IPCP PPP Internet Protocol Control Protocol

IPv6 Internet Protocol Version 6

IPX Internetwork Packet Exchange Protocol

ISDN Integrated Services Digital Network

ISP Internet Service Provider

ITU International Telecommunication Union

kbps Kilobits pro Sekunde

L2TP Layer Two Tunneling Protocol

LAN Local Area Network

LCP Link Control Protocol

LPC Linear Predictive Codec

MAG MBone Access Gate

MAGW MBone Access Gateway

MAS MBone Access Server

MASCP MBone Access Server Control Protocol

MBone Multicast Backbone im Internet

MCU Multipoint Control Unit

Modem Modulator-Demodulator, Gerät zur Übertragung digitaler Signale über analoge Verbindungen mit Bandbegrenzung

MSB Most Significant Bit

MSN Microsoft-Network, privater Online- und Internet-Zugangsdienst

MTU Maximum Transmission Unit

NCP Network-Layer Control Protocol

NTP Network Time Protocol

PAP Password Authentication Protocol

PCM Pulse Code Modulation

PDU Protocol Data Unit

PFC Protocol Field Compression

PIM Protocol Independent Multicast

PoP Point of Presence

POTS Plain Old Telephone System

PPP Point to Point Protocol

RADIUS Remote Authentication Dial In User Service

rat Robust Audio Tool, MBone Medienwerkzeug

RFC Request for Comments

RRZN Regionales Rechenzentrum für Niedersachsen, Universität Hannover

RSVP Resource Reservation Protocol

RTP Real-Time Transport Protocol

RTCP Real-Time Transport Control Protocol

RTT Round Trip Time

SAP Session Announcement Protocol

SDP Session Description Protocol

SLIP Serial Line Internet Protocol

SSL Secure Socket Layer

SSRC Synchronistation Source Identifier

SVC Switched Virtual Circuit

T-Online Online- und Internet Zugangsdienst der Deutschen Telekom AG

TACACS Terminal Access Controller Access Control System

TCP Transmission Control Protocol

TTL Time to Live

UDP User Datagram Protocol

VADSL Very high bitrate Asymmetrical Digital Subscriber Line

vat Visual Audio Tool, MBone Medienwerkzeug

VHDSL Very High bitrate Digital Subscriber Line

WAN Wide Area Network

wb Whiteboard, MBone Medienwerkzeug

WFQ Weighted Fair Queuing

WWW World Wide Web

Kapitel 1

Einführung und Motivation

Im digitalen Leben ist es nicht wichtig, zu einer bestimmten Zeit an einem bestimmten Ort zu sein, da eine Übertragung der Orte möglich sein wird.

Nicholas Negroponte, Total Digital [Neg95].

Begriffe wie Tele-Working und Heimarbeitsplätze tauchen heute immer häufiger in der öffentlichen Diskussion auf. Große Unternehmen haben längst begonnen, die mit der Einführung von Heimarbeitsplätzen verbundenen Vorteile für sich in geeigneten Arbeitsumfeldern zu nutzen.

Eine Voraussetzung für die Einführung dieser Methoden ist eine ausgereifte und belastbare kommunikationstechnische Infrastruktur. Zudem müssen ausgereifte Systeme zur Informationsverarbeitung mit standardisierten Schnittstellen zur Verfügung stehen. Die Erfüllung dieser generellen Forderungen kann dem jeweiligen Unternehmen heute Wettbewerbsvorteile verschaffen und wird daher mit großem Engagement verfolgt. Der Einsatz bewährter Internet-Technologien erscheint als tragfähiger Ansatz und begründet den aktuellen Trend zur Einführung von Intranets. Die Verfügbarkeit dieser Technologien schafft optimale Voraussetzungen für die Ortsunabhängigkeit von Arbeitsplätzen in vielen Bereichen des Dienstleistungssektors.

Während bisher in der Regel vielreisende Mitarbeiter, wie Vertreter und Verkäufer, von den Vorteilen des Tele-Working profitieren, ist davon auszugehen, daß zukünftig immer mehr Heimarbeitsplätze eingerichtet werden. Während heute nur Arbeitsbereiche ausgelagert werden können, in denen die Arbeiten im wesentlichen von einer Person ausgeführt werden und wenig Kommunikation mit anderen erfordern, werden zukünftig Teledienste auch die Auslagerung von Bereichen erlauben, die mehr interaktive interpersonelle Kommunikation beinhalten. Aufgrund der breiten Verfügbarkeit und Robustheit werden Internet-Protokoll-basierte Techniken in diesem Umfeld eine besondere Relevanz besitzen.

Eine Rahmenbedingung bei der Einführung von Telearbeit ist das Arbeitsrecht. Es ist heute durch zeitorientierte Arbeitsverhältnisse und örtlich feste Arbeitsplätze geprägt. Die Telearbeit ermöglicht hingegen außerordentlich flexible Arbeitszeiten, auftragsorientierte Arbeitsverhältnisse und die weitgehende Unabhängigkeit vom Ort. Bevor diese Punkte politisch und administrativ nicht aufgearbeitet sind, ist eine breite Einführung der Telearbeit nicht zu erwarten. Eine rücklehnende und abwartende Haltung ist unter dem Aspekt zunehmender Globalisierung und damit einhergehendem Wettbewerb jedoch nicht akzeptabel.

Dem Bildungsbereich kommt bei dieser Entwicklung eine wichtige Rolle zu. Die auf den Arbeitsmarkt strömenden Schüler und Hochschulabsolventen sollten in der Anwendung telematischer Techniken erfahren sein und dem Umgang mit Systemen zur Informationsverarbeitung aufgeschlossen gegenüber stehen. So fordert es beispielsweise die Europäische Union im Bericht der Task-Force "Multimedia und Lernprogramme" [Eur96].

Vorreiter können und müssen die Universitäten sein. Auch wenn die Begriffe Heimarbeitsplatz und Telearbeit nicht explizit auftauchen, ist dies die Umgebung par excellence für telematische Anwendungen. Diese Feststellung begründet sich auf den aktuellen Entwicklungen hinsichtlich Distant-Education, virtueller Universitäten und lebenslangem Lernen auf der einen und den in diesem Sinne idealen Rahmenbedingungen auf der anderen Seite:

- Studierende verbringen einen großen Teil ihrer universitären Ausbildung am heimischen Arbeitsplatz, sind also in der Heimarbeit – auch ohne Datenkommunikation – erfahren.
- Der Verbreitungsgrad von Rechnern und Online-Zugängen ist in dieser Bevölkerungsgruppe besonders hoch.
- Das Breitband-Wissenschaftsnetz, welches die deutschen Universitäten verbindet, basiert auf der Internet-Technologie und bietet genügend Reserven zur intensiveren Nutzung von Anwendungen, die einen höheren Bedarf an Datenraten haben als konventionelle Internet-Anwendungen wie File-Transfer und entfernter Rechnerzugang.
- Die meisten Universitäten bieten ihren Studierenden und den Mitarbeitern kostenfreie Zugänge über das öffentliche Telefonnetz auf der Basis von V.34-Modems und ISDN an.
- Vielerorts werden studentische Arbeitssäle und Studentenwohnheime direkt mit in LANs üblichen Datenraten an das universitäre Datennetz angeschlossen.
- Die Universitäten verfügen über umfassende Erfahrungen mit Internet-Technologien: Das Internet wurde maßgeblich durch Universitäten entwickelt und nahezu alle universitären Informationssysteme basieren seit langer Zeit darauf. Daher sind auch die Studierenden und Mitarbeiter qualifizierte Nutzer dieser Technologien.

Besonders der letzte Punkt besitzt große Wichtigkeit. Der Gebrauch der Internet-Werkzeuge ist inzwischen fest in der Arbeitstechnik der Studenten verankert. An der Praxis orientierte Lehrangebote zum Umgang mit dem Internet erlauben den einfachen Einstieg und alltägliche Aufgaben, wie das Ausleihen von Büchern in Bibliotheken, werden durch den Gebrauch von Internet-Anwendungen vereinfacht.

Dies gilt in gleicher Weise auch für die wissenschaftlichen Mitarbeiter der Universitäten. Verstärkt wird der Bedarf in dieser Nutzergruppe durch die zunehmende Spezialisierung in der Wissenschaft. Das Internet ist in diesem Bereich das essentielle Kommunikationsmedium zum Informationsaustausch mit Kollegen in aller Welt. Es ist vielfach wichtiger als das Telefon. Die Motivation für den Zugriff auf das Medium Internet vom Heimarbeitsplatz aus ist jedoch geringer, da am Arbeitsplatz in der Universität in der Regel ein permanenter Internet-Zugang vorhanden ist.

In der aktuellen universitären Bildungslandschaft konzentriert sich die rechnergestützte Kommunikation der Studierenden von ihren Heimarbeitsplätzen auf asynchrone Anwendungen wie Electronic-Mail, Net-News, World-Wide-Web, File-Transfer und entfernten Rechnerzugang. Kommunikationspartner aus technischer Sicht sind dabei vielfach Rechnersysteme an der Universität. Die zeitlich synchrone interpersonelle Kommunikation mit Partnern in Lerngruppen und Seminaren erfolgt in aller Regel bei persönlichen Zusammentreffen in der Universität. Aufgrund der örtlichen Nähe ist dies einfach möglich.

Wenn – und daran besteht wenig Zweifel – neuere Modelle, wie das der virtuellen Universität, realisiert werden, ist die örtliche Nähe nicht mehr gegeben.[1] Daraus folgt verstärkter Bedarf für den Einsatz telematischer Techniken zur zeitlich synchronen interpersonellen Kommunikation. Wenn auf den ersten Blick das Telefon hier als beste Wahl erscheinen mag, verblaßt es jedoch bei näherer Betrachtung der Anforderungen: Bei der diskursiven Erarbeitung von Lehrstoff ist das Medium Sprache nur eines von mehreren. Genau so wichtig ist die Tafel oder das Whiteboard. Zudem ist festzuhalten, daß es um Kommunikation in einer räumlich verteilten Gruppe mit in der Regel mehr als zwei Mitgliedern geht. Telematische Anwendungen, die diese Anforderungen erfüllen, werden unter dem Schlagwort *multimediale Multipoint-Konferenzen* zusammengefaßt. In dieser Arbeit fungieren Studierende und wissenschaftliche Mitarbeiter an Universitäten als exemplarische Nutzergruppe für multimediale Multipoint-Konferenzen.

Das zentrale Anliegen dieser Arbeit ist die Bereitstellung eines Systems zur Übertragung multimedialer Echtzeitdatenströme über Zugangsnetze, das den Nutzern an Heimarbeitsplätzen die Teilnahme an multimedialen Multipoint-Konferenzen ermöglicht.

Prinzipiell gibt es zur Zeit drei technische Ansätze für die Durchführung multimedialer Multipoint-Konferenzen:

1. Konferenz-Systeme nach der ITU-Empfehlung H.320 [Int97a], die durch Systeme wie etwa Intels Pro-Share realisiert werden. Sie basieren auf der Nutzung von 2 ISDN-B-Kanälen und erfordern im Spezialfall der Multipoint-Konferenz im Public-Network eine oder mehrere *Multipoint Control Units* (MCU).

2. Konferenz-Systeme nach der ITU-Empfehlung H.323. Sie beschreibt “Visual Telephone Systems and Equipment for Local Area Networks which provide a non-guaranteed Quality of Service” [Int96c]. Diese Empfehlung ist seit November 1996 in Kraft. Erste Implementierungen von H.323-konformen Anwendungen liegen vor.

3. Konferenz-Systeme für den Einsatz im *Multicast-Backbone im Internet* (MBone). Sie werden im allgemeinen als MBone-Werkzeuge bezeichnet. Ihre Domäne sind Multipoint-Konferenzen auf der Basis von IP-Multicast. Es handelt sich dabei nicht um monolithische Werkzeuge, sondern um kleine Anwendungen mit einfachen Protokollen, die für den Einsatz in unterschiedlichen Szenarien unkompliziert zusammengestellt werden können.

[1] Als Beispiel für eine solche Universität mag auf den ersten Blick die 1974 gegründete Fernuniversität Hagen erscheinen. Im Wintersemester 1996/1997 waren über 50% der Studierenden Teilzeithörer und nur gut 15% Vollzeithörer [Fera]. Tatsächlich bieten jedoch weit verteilte Studienzentren den Raum für persönliche Zusammentreffen. Es handelt sich folglich nur in Ansätzen um eine virtuelle Universität. Den Weg zur vollständig virtuellen Universität soll das Projekt *Virtuelle Universität/FernUniversität Online* [Ferb] ebnen.

Der MBone und die MBone-Werkzeuge sind seit 1992 im Einsatz und werden kontinuierlich verbessert. Sie sind insbesondere für die Kommunikation in Gruppen mit vielen Mitgliedern geeignet und finden breite Anwendung im Internet.

Konferenz-Systeme nach der ITU-Empfehlung H.320 sind im hier betrachteten Umfeld nur in Einzelfällen sinnvoll einsetzbar. Ideal geeignet sind diese Systeme für die audiovisuelle Kommunikation von zwei Partnern, die über das ISDN-Netz verbunden sind. Tatsächlich verfügt heute nur eine kleine Zahl von Studierenden über ISDN-Anschlüsse. Auch sind an den Arbeitsplätzen der wissenschaftlichen Mitarbeiter und Professoren an den Universitäten üblicherweise keine ISDN-Anschlüsse vorhanden. Problematisch für die Kommunikation in Gruppen sind zudem die von der Hardware her aufwendigen MCUs. Weiterführende Informationen hierzu enthält [CWHC96].

Konferenz-Systeme nach der ITU-Empfehlung H.323 fokussieren primär auf eng gekoppelte audiovisuelle Kommunikation mit zwei oder mehr Partnern. Bemerkenswert an H.323 ist, daß hiermit über Gateways auch die Integration von Konferenz-Systemen nach den ITU-Empfehlungen H.320, H.321 [Int96a], H.322 [Int96b] und H.324 [Int96d] möglich ist. Es steht zu erwarten, daß diese Empfehlung in Zukunft einen hohen Stellenwert für eng gekoppelte, audiovisuelle Kommunikation erlangen wird.

Die MBone-Werkzeuge eignen sich insbesondere für die Durchführung von Multipoint-Konferenzen. Wesentlich ist, daß die zum Einsatz kommenden Protokolle einfach und robust sind. Diese Merkmale wurden seit langem unter Beweis gestellt. Hervorzuheben ist zudem die Konzentration auf lose gekoppelte Konferenzen. Sie ermöglicht die flexible Zusammenstellung von Werkzeug-Gruppen für spezifische Einsatzszenarien. Ebenfalls vorteilhaft ist die freie Verfügbarkeit der meisten Werkzeuge. Nachteilig ist die lose Kopplung der zum Einsatz kommenden Werkzeuge untereinander. Sie erhöht die Nutzungskomplexität gegenüber monolithischen Anwendungen.

Der wesentliche Beitrag dieser Arbeit ist die Entwicklung eines Konzepts für die Vermittlung von Echtzeitdatenströmen über Netzzugänge mit geringen Datenübertragungsraten. Es ermöglicht die Durchführung multimedialer Multipoint-Konferenzen zwischen Nutzern an Heimarbeitsplätzen und Partnern im regulären Internet. Als Basis dient die im Internet entwickelte MBone-Technologie mit ihren Anwendungen, den MBone-Werkzeugen.

Das hier vorgestellte System hebt sich von vergleichbaren Ansätzen ab, indem der Aufbau von Stauungen[2] in vermittelnden Komponenten des Netzpfades in Hochlastsituationen vermieden wird. Dies begrenzt die Paketlaufzeiten und Paketlaufzeitschwankungen nachhaltig. Grundlage hierfür ist die Abschätzung des Queuing-Delays der Pakete durch den Nachrichten-Empfänger unter Berücksichtigung der von der Paketgröße abhängigen Serialisierungszeit der Pakete. Diese Daten über den aktuellen Lastzustand des Netzwerkpfades dienen dem Nachrichten-Sender als Grundlage für eine dynamische Datenratenregelung.

Wenngleich der Schwerpunkt auf der Nutzung des MBone und der MBone-Werkzeuge liegt, lassen sich die Konzepte weitgehend auf H.323-basierte Systeme übertragen, da auch hier das

[2]Unter einer *Stauung* wird hier der nicht leere Zustand einer Warteschlange an der Netzwerkschnittstelle einer aktiven Netzwerkkomponente verstanden. Zu übertragende Pakete werden nicht unmittelbar auf das angrenzende Netzwerk ausgegeben, sondern in der Warteschlange zwischengespeichert oder, im Fall einer überfüllten Warteschlange, verworfen.

in RFC 1889 spezifizierte *Real-Time Transport Protocol* Verwendung findet [SCFJ96].

Bevor auf die konkrete Anbindung von Heimarbeitsplätzen von Studenten und wissenschaftlichen Mitarbeitern eingegangen wird, wird die bestehende Infrastruktur dargestellt und auf die grundlegenden Probleme der Übertragung von Echtzeit-Datenströmen zu Heimarbeitsplätzen eingegangen. Dabei wird ein Überblick über verfügbare Systeme zur Ankopplung von Heimarbeitsplätzen gegeben. Die Analyse liefert die Anforderungen an ein Software-System zur Anbindung von Heimarbeitsplätzen, welches im Entwurfsabschnitt entwickelt wird. Das System wurde exemplarisch implementiert. Das entsprechende Kapitel beschreibt die Realisierung und enthält Messungen zum Verhalten des Systems. Die Meßergebnisse zeigen, daß das im Rahmen dieser Arbeit entwickelte System hinsichtlich der bei interaktiver interpersoneller Kommunikation kritischen Verzögerung multimedialer Echtzeitdatenströme unter normalen Betriebsbedingungen den Anforderungen gemäß der ITU-Empfehlung G.114 [Int93] gerecht wird. Die Arbeit schließt mit einer Zusammenfassung und gibt einen Ausblick auf die Entwicklungsmöglichkeiten vor dem Hintergrund der Einführung des *Resource Reservation Protocol* (RSVP) [Bra97] und der Etablierung der nächsten Generation des Internet-Protokolls (IPv6) [DH95].

Kapitel 2

Analyse

2.1 Private Infrastruktur aus heutiger Sicht

Bevor auf die Frage der Anbindung von Heimarbeitsplätzen an Datennetze auf Basis des Internet-Protokolls eingegangen wird, wird zunächst erklärt, was im Rahmen dieser Arbeit unter einem Heimarbeitsplatz verstanden wird und welche Ausprägungen solcher Arbeitsplätze heute im universitären Umfeld üblich sind.

2.1.1 Heimarbeitsplätze

Traditionell werden Arbeitsplätze innerhalb der Wohnung eines Arbeitnehmers als Heimarbeitsplatz bezeichnet. Seit vielen Jahren gibt es Heimarbeitsplätze, vor allem im Umfeld von Fertigungsprozessen, die viel manuelle Arbeit und wenig Maschinen erfordern. Auch Vertreter und Verkäufer nutzen seit langer Zeit die private Wohnung für ihre Arbeit. Ebenso nutzen Studenten, wie bereits dargestellt, die private Wohnung für Arbeiten zu ihrem Studium.

In dieser Arbeit wird unter einem Heimarbeitsplatz ein Teleworking-Arbeitsplatz verstanden. Durch Anwendung telematischer Methoden wird der Versuch unternommen, die Entfernung zu den Kollegen, Kommilitonen oder Lehrern zu überbrücken. Die Überbrückung der Distanz erfolgt auf der Basis von Netzen, hier speziell Datennetzen, die zum Datentransport die Internet-Protokolle benutzen.

Die Schnittstelle zwischen Mensch und Netz bildet der Rechner. Heute sind dies zumeist Personal-Computer unterschiedlicher Ausprägung. Während von Studierenden der Ingenieurwissenschaften und Informatik vielfach UNIX-basierte Systeme eingesetzt werden, bedienen sich andere Studentengruppen gerne der Macintosh-Rechner von Apple. Der Großteil benutzt jedoch PC-kompatible Systeme, die unter dem Betriebssystem Microsoft-Windows betrieben werden.

	Installierte Basis	Neue Geräte
CD-ROM-Laufwerk	33%	63%
Grafikkarte	49%	58%
Modem	22%	40%
Soundkarte	29%	33%
Scanner	10%	31%
Videokarte	4%	8%

Tabelle 2.1: PC-Ausstattung mit Multimedia-Komponenten

2.1.2 Ausstattung von Heimrechnern

Die wichtigste Anwendung auf Heimrechnern ist die Textverarbeitung, insbesondere bei Studierenden. Daneben werden aber häufig auch andere *Home-Office*-Anwendungen wie Tabellenkalkulation und Datenbanken eingesetzt. Ein weiterer wichtiger Bereich aus dem Blickwinkel multimedialer Anwendungen bilden Computer-Spiele. Sie erfordern häufig Erweiterungen für die Ausgabe von Audio-Daten und fördern somit die Verbreitung der dafür notwendigen Hardware.

Für den Einsatz von Heimrechnern in multimedialen Online-Konferenzen ist die Fähigkeit zur Ein- und Ausgabe von Audio-Daten aber nur eine Voraussetzung. Ergänzend müssen Video-Datenströme in Echtzeit dekodiert und dargestellt werden können. Dies erfordert entweder spezialisierte Hardware-Codecs oder hinreichende Rechenleistung zur Decodierung des Datenstroms in Echtzeit. Zusätzlich muß das Darstellungssystem hinreichende Farbtiefe für die Darstellung bieten und die Datenpfade im Rechner müssen die Übertragung der Video-Bilder erlauben.

Zudem sollte die Möglichkeit zur Generierung von Video-Datenströmen gegeben sein. Hierfür sind zusätzlich eine Video-Kamera und eine entsprechende Frame-Grabber-Karte erforderlich. Aufgrund des bisher hohen Preises in Relation zu den Kosten eines Personal-Computers und der geringen Zahl von Anwendungen für die Nutzung solcher Erweiterungen, ist der Anteil entsprechend ausgerüsteter Rechner im Vergleich zur Gesamtzahl klein.

Quantifiziert wird die Multimedia-Ausstattung von PCs in der vom Bundesministerium für Wirtschaft herausgegebenen Studie “Die Informationsgesellschaft: Fakten, Analysen, Trends” [Bun95]. Danach verfügten 1995 19% der Deutschen über einen PC. Die Ausstattung mit für Multimedia-Anwendungen wichtigen Komponenten ist Tabelle 2.1 zu entnehmen. Wenngleich die zitierte Studie nicht mehr ganz aktuell ist, wird deutlich, daß ein großer Teil der Systeme inzwischen mit Sound-Karten ausgestattet ist. Videokarten und Frame-Grabber-Karten sind nur in wenigen Fällen vorhanden.

Hohe Zuwachsraten sind für Modems und ISDN-Adapter zu erkennen. Sie dienen dem Tele-Banking, dem Informationsaustausch per Electronic-Mail oder der Informationsbeschaffung mittels World-Wide-Web im Internet. Der Zugang zum Internet erfolgt über das öffentliche Telefon- oder ISDN-Netz. Nach Angaben des Regionalen Rechenzentrums für Niedersachsen (RRZN) nutzten im Juni 1997 von etwa 32.000 Studierenden an der Universität Hannover alleine 7.200 einen Internet-Zugang über den Zugangsdienst der Universität.[1] Hinzu kommen Netz-

[1] Die Zahl der studentischen Nutzer für den Zugangsdienst der Universität wurde auf Anfrage vom zuständigen

zugänge über andere Dienste wie T-Online, AOL und Compuserve.

2.1.3 Private Internets

Neben dieser großen Zahl von Rechnersystemen, die unter Nutzung des öffentlichen Telefonnetzes auf das Internet zugreifen können, sonst aber stand-alone, d.h. ohne Netzzugang betrieben werden, gibt es eine bisher noch kleine Gruppe von Heimarbeitsplätzen, die nach einem anderen Modell betrieben werden: Mehrere Rechner in einer Wohnung oder einem Haus werden mittels preiswerter und robuster Netzwerktechnik – zumeist 10 Mbps Ethernet – zu einem *privaten Internet* zusammengeschaltet. Ein einzelner Rechner im Verbund steht als Server zur Verfügung und übernimmt den Datenaustausch mit Rechnern im Wissenschaftsnetz und auch dem weltweiten Internet[2]. Zusätzlich kann dieser Rechner die Aufgaben eines Telefonanrufbeantworters und eines Fax-Geräts wahrnehmen.

Die bisher geringe Verbreitung solcher Systeme ist nur bedingt eine Frage der Kosten für eine derartige Installation. Insbesondere private Anwender sehen heute noch einen zu geringen Nutzen oder kennen die Möglichkeiten solcher Installationen nicht. Allerdings wächst das Interesse der Gesellschaft an der Informationstechnologie. Die Einführung preiswerter und gleichzeitig leistungsfähiger Network-Computer läßt die Annahme zu, daß die Zahl privater Internets in Zukunft steigen wird[3].

Der Zugriff auf Dienste im Internet erfolgt unter Nutzung des TCP/IP-Protokolls. Zur Nutzung dieser Dienste sollten auch private Internets unter Einsatz des TCP/IP-Protokolls betrieben werden. Ein Problem des TCP/IP-Einsatzes in diesem Umfeld ist, daß ein Mangel an freien Adreßbereichen für das heute in Einsatz befindliche Internet-Protokoll in der Version 4 besteht (vgl. [Pos81a], [CCC+91], [WC92], [Dix93] und [Ull93]). Daher werden private Internets zumeist unter Nutzung der im RFC 1918 reservierten Adreßbereiche für private Internets betrieben [RMK+96].

2.2 Internet-Zugangsnetze im Überblick

In diesem Abschnitt wird die Struktur von Internet-Zugangsnetzen dargestellt. Die zugrundeliegende Fragestellung lautet, wie einzelne Rechner oder Netze an das Internet gekoppelt werden.

Mitarbeiter des RRZN mitgeteilt. Die Zahl der Studierenden wurde dem Vorlesungsverzeichnis der Universität Hannover für das Sommersemester 1997 entnommen.

[2] In der öffentlichen Diskussion werden private Internets heute üblicherweise als *Intranets* bezeichnet. Gemäß [RMK+96] lautet die korrekte Bezeichnung jedoch *Private Internet*.

[3] Network-Computer heutiger Ausprägung besitzen keinen Massenspeicher. Für den Betrieb ist im Hintergrund ein Server erforderlich, der den benötigten Speicherplatz für Betriebssystem, Programme und Daten bereitstellt. Als Kommunikationsprotokoll wird TCP/IP eingesetzt. Wenn akzeptable Interaktionsraten erreicht werden sollen, muß die Kommunikation zwischen Network-Computer und Server mit in LANs üblichen Datenraten von 10 Mbps/s und mehr erfolgen.

2.2.1 Datenübertragungsraten

Unabhängig davon, ob es sich um ein privates Internet oder einen Standalone-PC handelt, erfolgt die Anbindung an die Hochschulnetze in aller Regel über das öffentliche Telefonnetz. Dazu kommen Modems oder ISDN-Adapter zum Einsatz. Üblich sind heute Modems nach den ITU-Empfehlungen V.32 und V.34 mit Datenraten von 14.400 bps bzw. 28.800 bps. ISDN-Adapter für im Heimbereich übliche Basisanschlüsse bieten mit 64 kbps unter Nutzung eines B-Kanals[4] eine deutlich höhere Übertragungsrate, die durch Bündelung der beiden B-Kanäle auf 128 kbps erhöht werden kann. Ein weiterer Vorteil der Nutzung des ISDN-Netzes ergibt sich aus der schnelleren digitalen Signalisierung über den D-Kanal. Hier kann die Verbindung zu einem Partner innerhalb von zwei Sekunden aufgebaut werden[5]. Bei der Nutzung von Modems auf analogen Leitungen lassen sich solche Zeiten, selbst bei Nutzung des schnelleren Mehrfrequenz-Wahlverfahrens, nicht erreichen. Wenngleich der schnellere Verbindungsaufbau bei der Nutzung von ISDN keine Vorteile hinsichtlich des in dieser Arbeit entwickelten Systems mit sich bringt, ist dies ein für den Endnutzer wichtiges Komfort-Merkmal.

Von besonderem Interesse für die Nutzung von Multimedia-Anwendungen mit Echtzeitdatenströmen ist die Datenrate. Modems nach der V.32-Empfehlung erscheinen für kombinierte Audio/Video-Konferenzen ungeeignet. Bei der Verwendung üblicher Audio-Codecs entstehen Netto-Datenraten von 8 kbps beim *Linear Predictive Codec* (LPC) bzw. 13 kbps bei *Group Speciale Mobile* (GSM). Die verbleibende Datenrate ist für die zusätzliche Übertragung eines Video-Datenstroms und verteilte Anwendungen nicht ausreichend. Auch bei V.34-Modems ist die Kapazität nicht ausreichend. Anzumerken bleibt, daß modernere Codecs für Audio und Video zunehmend geringere Datenraten erfordern und damit die Einsatzmöglichkeiten für Modems verbessern. Zudem drängen verstärkt Modems mit Übertragungsraten von 56 kbps auf den Markt, so daß die mit Modems erzielbaren Datenübertragungsraten in den Bereich von ISDN gelangen. Eine Kanal-Bündelung ist jedoch nicht möglich.

2.2.2 Serialisierung von Paketen

Eine wichtige Besonderheit bei der Übertragung von Echtzeitdatenströmen über Kanäle mit niedrigen Datenraten ist die Verzögerung durch die Serialisierung der Datenpakete. Die betrachteten Protokolle der Transport- und Netzwerkschicht aus der TCP/IP-Protokollfamilie arbeiten nach dem Konzept der Paketvermittlung. Die Bitübertragungsschicht arbeitet hingegen bitseriell. Eine der Transportschicht von der Anwendung übergebene *Protocol Data Unit* (PDU) muß nebst in darunter liegenden Schichten ergänzten Protokoll-Daten bitseriell über das Medium transportiert werden. Der Empfänger kann die empfangenen Daten erst dann an die höheren Schichten weitergeben, nachdem das letzte Bit eines Pakets eingetroffen ist. Abbildung 2.1 illustriert diesen Effekt.

Unter Vernachlässigung der Verzögerungen während der Verarbeitung der Daten im Sender und Empfänger ergibt sich die Paketübertragungszeit in erster Näherung als Summe aus der Übertragungszeit für ein Bit (Δt_0) und der zeitlichen Ausdehnung eines Pakets auf dem jeweiligen

[4] Für die Erläuterung der Merkmale von ISDN-Kanälen siehe Tabelle 2.3, Seite 13.

[5] Diese Zeitspanne umfaßt neben der Wahlverzögerungszeit auch die Zeit für die Parametrisierung der Verbindung auf der Ebene des Internet-Protokolls unter Nutzung des PPP-Protokolls.

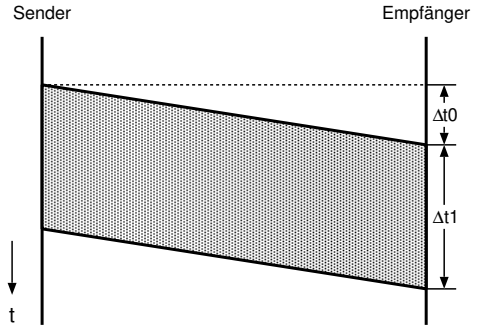

Abbildung 2.1: Verzögerungen bei der Datenübertragung

Medium (Δt_1). Die Übertragungszeit für ein Bit Δt_0 ist im wesentlichen von der Distanz zwischen Sender und Empfänger, der medienspezifischen Signalausbreitungsgeschwindigkeit sowie den Totzeitsystemen innerhalb des Verbindungsnetzwerkes abhängig. Die Übertragungszeit berechnet sich bei der Nutzung von Kupferkabeln zu

$$\Delta t_0 = \frac{d}{c_{Cu}} + K(P)$$

Dabei bedeuten:

d	Entfernung in Metern
c_{Cu}	Signalausbreitungsgeschwindigkeit in Kupfer ($c_{Cu} = 0.7\ c_{Licht}$)
$K(P)$	Totzeiten des Übertragungssystems in Abhängigkeit des gewählten Pfades P

Die Verzögerung Δt_0 wächst proportional mit der Entfernung d.

Die zeitliche Ausdehnung des Pakets auf dem Medium berechnet sich zu

$$\Delta t_1 = \frac{8(\frac{Bit}{Byte})\ Paketgröße(Byte)}{Übertragungsrate(\frac{Bit}{s})}$$

Sie ist proportional zur Paketgröße und umgekehrt proportional zur Übertragungsrate. Diese Zusammenhänge sind allgemein gültig und treffen auch für die Übertragung von Daten in lokalen Netzen zu. Die Besonderheit in dem hier betrachteten Anwendungsfall sind die geringen Übertragungsraten. Dies wird bei der Betrachtung der in Tabelle 2.2 dargestellten Paket-Serialisierungszeiten für unterschiedliche Übertragungsmedien deutlich[6].

[6]Die Paketgröße eines Ethernet-Pakets bestimmt sich aus UDP-Payload, 8 Byte UDP-Header, 20 Byte IP-Header, 18 Byte Ethernet-Protokoll-Daten und 9 Byte Präambel und Pause. Die Paketgröße eines Paketes des *Point to Point Protocols* (PPP) auf dem Medium bestimmt sich aus UDP-Payload, 8 Byte UDP-Header, 20 Byte IP-Header und 7 Byte PPP/HDLC-Protokolldaten. Siehe hierzu [Sim93]. Die Nutzung von *Protocol Field Compression* (PFC) und *Address and Control Field Compression* (ACFC) wurden vernachlässigt, da sie optional sind. Gleiches gilt für die *Van Jacobson Header Compression* für IP und TCP. Im Fall asynchroner Modem-Strecken muß mit 10 Bits pro Byte gerechnet werden, da Start- und Stop-Bits hinzuzuzählen sind.

Medium	Maximale Übertragungsrate	Größe UDP PDU	Paketgröße	Δt_1
10 Mbps Ethernet	10 Mbps	100 Byte	155 Byte	0.124 ms
64 kbps ISDN sync. PPP	64 kbps	100 Byte	135 Byte	16.86 ms
V.32 Modem mit PPP	14.4 kbps	100 Byte	135 Byte	93.75 ms

Tabelle 2.2: Paket-Serialisierungszeiten für verschiedene Übertragungsmedien [7]

Zum Vergleich beträgt die Übertragungszeit einer 100 Byte ICMP-Message[8] von der Universität Hannover zum Router im *Point of Presence* (PoP) des DFN-Vereins in Perryman/USA ca. 55 ms inklusive der Behandlung im Protokoll-Stack[9]. Dies ist größenordnungsmäßig die Hälfte der Zeit, die für die Serialisierung einer 100 Byte UDP[10] PDU für eine V.32-Modem-Verbindung erforderlich ist.

2.2.3 B-WiN Zugangsdienste

Im Fokus dieser Arbeit steht das deutsche *Breitband-Wissenschaftsnetz* (B-WiN), das die Netze der deutschen Hochschulen und Forschungseinrichtungen verbindet und Übergänge zu anderen Netzen des Internets bereitstellt. Für den direkten Zugang von Heimarbeitsplätzen zu den Hochschulnetzen und dem B-WiN gibt es zwei Alternativen:

- Der Verein zur Förderung eines Deutschen Forschungsnetzes (DFN-Verein) bietet einen allgemeinen B-WiN Zugangsdienst für Heimarbeitsplätze, das WiN-Shuttle, an [DFN].
- Die meisten Universitäten bieten einen eigenen Zugangsdienst zum Hochschulnetz für Studierende und Mitarbeiter der jeweiligen Universität an.

Der wesentliche Unterschied zwischen den Diensten aus technischer Sicht ist, daß die WiN-Shuttle-Zugangspunkte bis auf wenige Ausnahmen über 2 Mbps Verbindungen direkt an das B-WiN gekoppelt sind, wohingegen die Zugangspunkte der Hochschulen in der Regel über 10 Mbps Ethernet-Verbindungen an das jeweilige Hochschulnetz gekoppelt sind. Die netztopologische Entfernung zwischen Rechnern im jeweiligen Hochschulnetz und dem Heimarbeitsplatz ist vergleichbar. Ein Vorteil der universitären Zugangsdienste ist, daß sie den Hochschulangehörigen für eine geringere Gebühr und häufig sogar kostenlos angeboten werden. Daher konzentriert sich diese Arbeit auf die durch die Universitäten angebotenen Zugangsdienste. Die entwickelten Konzepte lassen sich auf den WiN-Shuttle-Zugangsdienst übertragen.

[7] In dieser Arbeit werden Dezimalzahlen durchgängig in der amerikanischen Schreibweise notiert, d.h. als Dezimaltrennzeichen dient ein Punkt.

[8] ICMP steht für *Internet Control Message Protocol* [Pos81b].

[9] Dieser Wert wurde unter Nutzung des *ping*-Programms vom Rechner *lanai.rvs.uni-hannover.de* zum Router *IR-Perryman1.WiN-IP.DFN.DE* am 21. Juni 1997 ermittelt. Die sich ergebende minimale *Round Trip Time* (RTT) von 109 ms wurde halbiert.

[10] UDP steht für *User Datagram Protocol* [Pos80].

Kanal	Merkmale
A	4 kHz analoger Telefonkanal
B	64 kbps digitaler Kanal für PCM-kodierte Sprache gemäß G.711 oder digitale Daten
C	8 oder 16 kbps digitaler Kanal
D	16 oder 64 kbps digitaler Kanal für Außerbandsignalisierung
E	64 kbps digitaler Kanal für interne ISDN-Signale
H	384, 1536 oder 1920 kbps digitaler Kanal

Tabelle 2.3: Merkmale von Kanälen im Telefonnetz

2.2.4 Zugangsnetze

Sowohl beim Zugangsdienst des DFN-Vereins als auch bei den universitären Zugangsdiensten wird das Festnetz der Deutschen Telekom AG für den Zugang vom Heimarbeitsplatz zum IP-Netz[11] des Dienstanbieters genutzt. Auf dieses inzwischen weitestgehend digitalisierte Netz kann der Kunde gemäß [KB94] über die in Tabelle 2.3 beschriebenen Schnittstellen zugreifen.

Diese Kanäle werden in folgenden Kombinationen als Produkt angeboten:

Basisanschluß	$2B + 1D_{16}$
Primärmultiplexanschluß	$30B + 1D_{64}$
(Hybridanschluß	$1A + 1C$)

Weit verbreitet sind zusätzlich die konventionellen 3.1 kHz bandbegrenzten analogen Telefonanschlüsse, die häufig als *Plain Old Telephone System* (POTS) bezeichnet werden.

Während im Privatbereich POTS- und ISDN-Basisanschlüsse vorherrschen, nutzen Dienstanbieter in der Regel Gruppen von POTS-Anschlüssen, die über Sammelrufnummern erreichbar sind, gebündelte Basisanschlüsse sowie Primärmultiplexanschlüsse, wobei ein Trend zur verstärkten Nutzung von Primärmultiplexanschlüssen festzustellen ist.

Der Anschluß der Zugangspunkte in den Universitäten erfolgt in vielen Fällen nicht direkt am Netz der Telekom, sondern über die Telekommunikationsanlage (TK-Anlage) der jeweiligen Einrichtung. Vorteilhaft hierbei ist, daß an das Telefonnetz der Hochschule angeschlossene Einrichtungen ohne direkten Zugang zum Datennetz diesen Zugang ohne zusätzliche Gebühren über das hochschuleigene Telefonnetz realisieren können.

Es steht zu erwarten, daß auf längere Sicht die Nutzung des Festnetzes der Deutschen Telekom AG als Zugangsnetz zum universitären Datennetz der übliche Weg bleiben wird. Vor dem Hintergrund der Liberalisierung des Telekommunikationsmarkts entstehen jedoch weitere Zugangsmodelle, die es den neu auf den Markt strebenden Konkurrenten der Telekom AG erleichtern, das Last-Mile-Problem[12] zu überwinden.

[11] IP steht für *Internet Protocol* [Pos81a].

[12] Die neuen Mitbewerber der Telekom AG können verhältnismäßig einfach leistungsfähige Netze aufbauen, die ihre Endpunkte im Ortsbereich haben. Problematisch ist die Überbrückung der *Last Mile* zum Kunden, d.h. die Verbindung des Kunden zur Ortsvermittlung. Dieses Netz gehört der Telekom. Die zweite Komponente des Last-Mile-Problems ist, daß die installierten Leitungen derzeit nur relativ geringe Übertragungsraten zulassen, die auch die Telekom z.Z. beim Angebot zusätzlicher Dienstleistungen behindert.

Ein möglicher Weg ist die Mitnutzung von Breitband-Verteilnetzen für die Daten- und Sprachkommunikation. In [Vog96] wird diese Thematik eingehend behandelt. Wenn sich die Mitnutzung der Breitband-Verteilnetze für die Datenkommunikation durchsetzt, werden voraussichtlich die entsprechenden Netzbetreiber als Internet-Provider auftreten. Damit entstünde eine mit heutigen Netzzugängen über Online-Dienste wie T-Online, AOL und MSN vergleichbare Situation. Hier wird den Nutzern in der Regel zwar eine IP-Schnittstelle bereitgestellt, innovative Anwendungen, wie beispielsweise Multipoint-Videokonferenzen auf dem MBone, können allerdings nicht genutzt werden. Techniken zur Überwindung dieser Probleme, wie die Nutzung von *IP Encapsulation within IP* [Per96], werden aller Erfahrung nach für Echtzeitdatenströme kaum einsetzbar sein: die sich ergebenden Paketlaufzeiten und die Varianzen der Paketlaufzeiten sind zu groß.[13]

Bereits heute verfügbar sind Angebote der Telekom, die ISDN-basiert Zugang zum T-Internet-Dienst bieten. Zur Zeit sind die Preise für diese Anschlußform noch so hoch, daß eine private Nutzung i.a. nicht wirtschaftlich ist. Beispiele aus den USA zeigen jedoch, daß derartige Angebote zu günstigen Preisen möglich sind und eine Alternative zu dem heute üblichen Internet-Zugang über Wählverbindungen werden können.[14]

Mittelfristig ist die Nutzung von Datenraten im Mbps-Bereich über die Anschlußleitung möglich. Beispiele hierfür sind die xDSL-Techniken, wie *High bitrate Digital Subscriber Line* (HDSL), *Asymmetrical Digital Subscriber Line* (ADSL), *Very High bitrate Digital Subscriber Line* (VHDSL) und *Very high bitrate Asymmetrical Digital Subscriber Line* (VADSL). Zu beachten ist, daß die für viele Anwendungen besonders interessanten asymmetrischen Techniken wie ADSL und VADSL für die hier betrachteten Szenarien schlecht einsetzbar sind, da der Kanal vom Subscriber zur Vermittlungsstelle einer starken Bitratenbegrenzung unterliegt.

Letztlich bleibt abzuwarten, welche IP-Dienste in den Angeboten enthalten sein werden. Die weiteren Betrachtungen in diesem Kapitel basieren auf der heute realisierten Struktur von Netzzugängen, d.h. der Nutzung des Festnetzes der Deutschen Telekom AG über POTS oder ISDN. Weiter wird unterstellt, daß dem Nutzer das volle Spektrum der Internet-Dienste angeboten wird und als durchgängiges Protokoll der Netzwerkschicht das Internet-Protokoll zum Einsatz kommt.

2.2.5 Internet-Zugänge und Zugangs-Protokolle

Die folgenden Betrachtungen zur Architektur von Netzwerkzugängen konzentrieren sich auf sogenannte Wählzugänge. Ihr wesentliches Merkmal ist, daß jeder Verbindungsaufbau vom Heim-

[13] Als negatives Beispiel kann hier T-Online genannt werden. Die Vermittlung der Daten zwischen T-Online-Kunden und Nutzern im B-WiN erfolgte lange Zeit über die USA. Von der Universität Hannover gelangten die Pakete erst nach der Vermittlung durch 14 IP-Router in das Netz der Deutschen Telekom AG. Der WWW-Server `www.t-online.de` wurde erst nach der Vermittlung der Daten durch 19 IP-Router erreicht.

[14] Praktisch alle Carrier in den USA bieten inzwischen Internet-Zugangsdienste an. Pacific-Bell bietet beispielsweise seinen Kunden einen privaten Internet-Zugang über 33 kbps Modems für $19.95 an. Soll für den Internet-Zugang ein dedizierter Anschluß genutzt werden, entsteht zusätzliche Anschlußgebühr von $11.25 (für *Flat Rate Service*). Verbindungsgebühren entfallen bei Anschlüssen mit Flat-Rate-Service. Ähnliche Angebote gibt es für ISDN-Zugänge mit Kanal-Bündelung. Siehe hierzu http://www.pacbell.com/products/residential/ (Stand August 1997).

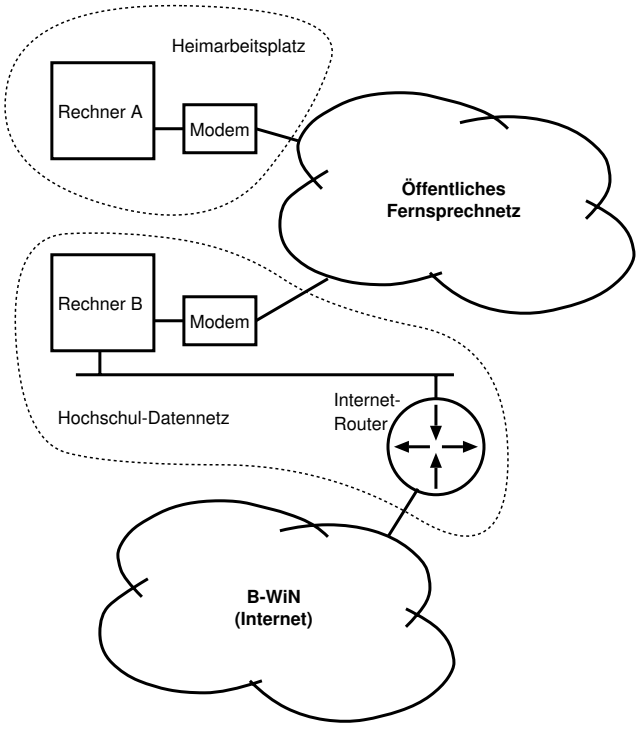

Abbildung 2.2: Struktur eines klassischen Wählzugangs

arbeitsplatz ausgeht. Abbildung 2.2 zeigt einen solchen Zugang in seiner klassischen Ausprägung.

Der Rechner am Heimarbeitsplatz ist über ein Modem an das öffentliche Telefonnetz gekoppelt. Auf der Seite des Zugangsdienstes befindet sich ein Rechner, der auf der einen Seite direkt mit dem Datennetz verbunden ist. Zudem sind über serielle Schnittstellen Modems angeschlossen, die wiederum mit dem Telefonnetz verbunden sind. In klassischen Anwendungen behandelt der unter UNIX betriebene Rechner die serielle Schnittstelle wie eine gewöhnliche Terminal-Schnittstelle. Wird vom Heimarbeitsplatz eine Verbindung aufgebaut, reagiert der UNIX-Host darauf mit der üblichen Login-Prozedur. Nachdem sich der entfernte Benutzer angemeldet hat, wird anstelle eines Kommandointerpreters ein Programm zum Umschalten der sogenannte Line-Discipline gestartet. Danach ist der Austausch von IP-Datagrammen zwischen den Rechnern A und B unter Nutzung des *Serial Line Internet Protocol* (SLIP) [Rom88] möglich.

SLIP definiert eine einfache Kapselung für IP-Datagramme. Das eigentliche IP-Datagramm wird mit einer Ende-Marke (0xc0) versehen und über die asynchrone serielle Leitung übertragen. Um Seiteneffekte durch Störungen auf der Leitung zu vermeiden, senden übliche Implementierungen zudem eine Ende-Markierung direkt vor dem Beginn des Datagramms. Zudem existiert ein Mechanismus, um im Datenstrom enthaltene Bytes, die der Ende-Markierung entsprechen, durch Escape-Zeichen zu markieren. Einen guten Überblick liefert hierzu [Ste94].

Das dargestellte System für den Netzzugang auf der Basis von Modems und SLIP weist eine Reihe von Nachteilen auf:

- Die Nutzung eines UNIX-Hosts für den Anschluß von Modems ist nur für eine kleine Zahl

von Modems handhabbar, da die Zahl der seriellen Schnittstellen in der Regel begrenzt ist.

- Für jedes Modem ist ein einzelner Telefonanschluß erforderlich. Bei einer größeren Zahl von Zugängen erhöht sich die Komplexität des Systems. Damit steigt auch die Zahl potentieller Fehlerquellen.

- SLIP basiert auf der Vergabe fester IP-Adressen. Üblicherweise wird jedem Nutzer exklusiv eine IP-Adresse zugeordnet. Dies führt bei größeren Nutzerzahlen zur raschen Erschöpfung des Adreßraumes. Verfahren, bei denen die IP-Adressen dynamisch zugeordnet werden, erfordern die Übermittlung der Adresse vor der Änderung der Line-Discipline auf SLIP. Da hierfür keine standardisierten Verfahren etabliert sind, vergrößert sich der Aufwand für die Einrichtung des Netzzugangs beim Endnutzer unnötig.

- Das SLIP-Protokoll bietet keine Möglichkeit zur Markierung eines bestimmten Payload-Formats. Über eine SLIP-Verbindung können ausschließlich IP-Datagramme übertragen werden. Protokolle wie *Internetwork Packet Exchange Protocol* (IPX) und AppleTalk sind nicht nutzbar.

Der erste Kritikpunkt wurde durch die Einführung sogenannter Terminal-Server überwunden. Dies sind dedizierte Geräte, die auf der LAN-Seite eine Ethernet-Schnittstelle haben und auf der anderen Seite eine feste Anzahl von V.24-Schnittstellen bieten. An die Schnittstellen können Modems oder Terminals angeschlossen werden. Die erforderlichen Komponenten zur Nutzer-Authentifikation werden auf einen zentralen UNIX-Host ausgelagert, mit dem der Terminal-Server über das TACACS- oder RADIUS-Protokoll kommuniziert (vgl. [Fin93], [Rig97]). So können mehrere Terminal-Server auf einen Nutzer-Datenbestand zugreifen.

Die zunehmende Digitalisierung des Telefonnetzes brachte eine Abwandlung der Terminal-Server hervor, die sogenannten *Access-Router*. Dies sind ebenfalls dedizierte Geräte, die auf der LAN-Seite in der Regel einen Ethernet-Port bieten und S0-Schnittstellen zum WAN-Anschluß haben. Geräte für den hier betrachteten Einsatzzweck bieten als WAN-Schnittstelle üblicherweise zwei S_{2m}-Schnittstellen, wodurch der gleichzeitige Zugang von 60 Nutzern möglich ist. Geräte dieser Familie ermöglichen zudem den Einbau von Erweiterungsmodulen, so daß ein Teil oder alle B-Kanäle über nachgeschaltete Modems verfügen. So können auch Modem-Nutzer diese Zugangsrouter benutzen.

Die Schwächen des SLIP-Protokolls werden durch Einsatz des *Point to Point Protocol* (PPP) überwunden [Sim94]. Es beinhaltet als zentrales Element ein *Link Control Protocol* (LCP) für das Management der Verbindung. Nach dem Verbindungsaufbau wird der Nutzer optional authentifiziert. Dazu dienen das *Password Authentication Protocol* (PAP) [LS92] sowie das *Challenge Handshake Authentication Protocol* (CHAP) [Sim96]. PPP ermöglicht die Nutzung unterschiedlicher Network-Layer-Protokolle wie das Internet-Protocol, IPX und Apple-Talk. Nach der Authentifizierungsphase beginnt die *Network-Layer Protocol Phase*. In dieser Phase wird jedes *Network-Layer Control Protocol* (NCP) konfiguriert.

Das NCP für IP trägt den Namen *PPP Internet Protocol Control Protocol* (IPCP) und ist in [McG92] spezifiziert. Es erlaubt die Abstimmung der zu verwendenden IP-Adressen sowie weiterer Parameter wie die Nutzung der *Van Jacobson Header Compression* für TCP/IP. Die Header-Compression vermeidet die volle Übertragung der Header für IP und das *Transmission Control*

Protocol (TCP) [Pos81c] (40 Byte) auf der Basis einer Inter-Datagramm-Codierung. Das Verfahren ist nur für TCP-Verbindungen anwendbar und somit vorerst nur von untergeordnetem Interesse. Im Entwurf wird detaillierter auf die Header-Compression eingegangen (Abschnitt 3.2.5, Seite 97).

Eine wichtige Eigenschaft von PPP ist, daß nicht nur zeichenorientierte asynchrone Verbindungen, sondern auch bitorientierte synchrone Verbindungen unterstützt werden. Dies ist für die Vermittlung von IP-Datagrammen über digitale ISDN-B-Kanäle wichtig.

Über eine konfigurierte PPP-Verbindung werden IP-Datagramme in PPP-Frames übertragen. Dabei werden der PDU 5 Bytes PPP-Protokolldaten vorgestellt sowie eine CRC in zwei Bytes und ggf. eine Ende-Markierung mit einem Byte nachgestellt. Die Ende-Markierung entfällt auf bitorientierten Verbindungen. Der Anfang des Rahmens wird, wie im Ethernet, durch eine feste Bitfolge markiert. Diese Folge wird im Rahmen unter Einsatz des *Bit-Stuffing* umgangen. Somit ist ein PPP-Frame unter Vernachlässigung von Bit-Stuffing-Effekten 7 oder 8 Byte größer als das entsprechende IP-Datagramm.

Eine weitere wichtige Änderung mit der Einführung von PPP ist die Veränderung der *Maximum Transmission Unit* (MTU). Während auf SLIP-Verbindungen MTUs von 296 Byte üblich waren, wird nun verstärkt mit einer MTU von 1524 Byte gearbeitet. Wenngleich die Interaktivität dadurch negativ beeinflußt wird, ist der Vorteil evident: Die in den weit verbreiteten Ethernet-LANs übliche MTU beträgt 1500 Byte. IP-Datagramme dieser Größe können ohne Fragmentierung über den PPP-Kanal übertragen werden. Die sonst entstehende Verzögerung durch Fragmentierung und Defragmentierung wird vermieden.

Bei der Nutzung digitaler ISDN-B-Kanäle wird üblicherweise *High-Level Data Link Control* (HDLC) als Protokoll der Sicherungsschicht benutzt. Es entsteht jedoch kein weiterer Protokoll-Overhead, da sich PPP nahtlos in HDLC eingliedert.

Zusammenfassend bleibt festzuhalten, daß die Nutzung von Access-Routern die Komplexität der Komponenten beim Dienstanbieter deutlich reduziert und mit dem TACACS- oder RADIUS-Protokoll ein einheitliches und entkoppeltes System zur Verwaltung von Nutzern zur Verfügung steht. Die Nutzung von PPP als Link-Protokoll löst alle mit SLIP verbundenen Probleme und ist heute als Protokoll zur Anbindung von Heimarbeitsplätzen zu favorisieren. Damit ergibt sich die in Abbildung 2.3 dargestellte Struktur für einen modernen Wählzugang.

2.2.6 Internet-Zugang an der Universität Hannover

Das Regionale Rechenzentrum für Niedersachsen an der Universität Hannover betreibt Netzzugangsdienste für Studierende und Mitarbeiter der Universität. Abbildung 2.4 zeigt die entsprechende Konfiguration im Juni 1997. Ergänzend dargestellt ist ein dem Test dienendes, privates Internet sowie ein Ausschnitt des Institutsnetzes des Lehrgebiets Rechnernetze und Verteilte Systeme. Zudem wurde der zentrale DVMRP-MBone-Router[15] der Universität eingetragen.

Maximal 152 Nutzer können gleichzeitig den Zugangsdienst der Universität nutzen. Studierenden stehen 92 parallele Zugänge zur Verfügung von denen 60 als ISDN-Zugänge nutzbar sind.

[15] DVMRP steht für *Distance Vector Multicast Routing Protocol*. Siehe hierzu [Dee91] und [Pus97].

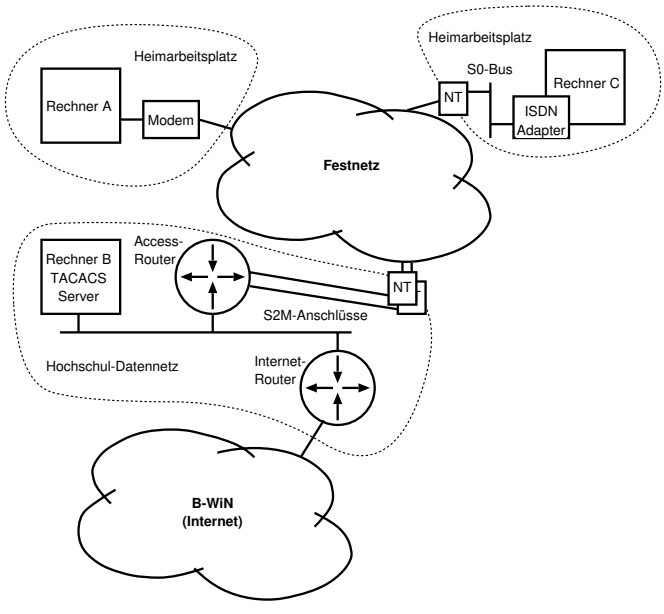

Abbildung 2.3: Struktur eines modernen Wählzugangs

36 dieser Anschlüsse sind auch mit V.34 Modems nutzbar.[16] Die restlichen Zugänge teilen sich auf zwei Terminal-Server auf (16 mal V.34-Modems und 16 mal V.24-Modems). Im Juni 1997 waren etwa 7200 Studierende für die Nutzung dieser Zugänge zugelassen. Auslastungsstatistiken des RRZN zeigen, daß die Zugänge täglich zwischen 21:00 und und etwa 01:00 vollständig ausgelastet sind.

Der Zugangsdienst für Mitarbeiter ist geringer dimensioniert. In diesem Bereich können maximal 60 Nutzer gleichzeitig über ISDN B-Kanäle zugreifen. Davon können 32 Kanäle alternativ mit V.34-Modems benutzt werden. Im Juni 1997 waren ca. 2400 Mitarbeiter der Universität für den Zugangsdienst zugelassen. Die Auslastungsstatistiken zeigen, daß die bereitstehende Kapazität in der Regel nicht vollständig ausgelastet ist.

Auch wenn die Nutzung des SLIP-Protokolls noch möglich ist, wird überwiegend das PPP-Protokoll als Zugangsprotokoll verwendet. Die Zuordnung der IP-Adressen erfolgt, bis auf wenige Ausnahmen, dynamisch aus einem festen Pool für die Dauer einer Verbindung. Obwohl fest zugeordnete Adressen aus Sicht der Anwendung Vorteile haben, kann dies wegen des begrenzten Vorrats an IP-Adressen nicht realisiert werden. Die Bündelung von B-Kanälen wird gestattet, aber nicht offiziell unterstützt.

2.3 Eigenschaften von ISDN-Internet-Zugängen

Im Kontext multimedialer Anwendungen mit der Forderung nach Übertragung isochroner Datenströme ist die Betrachtung von ISDN B-Kanälen für die Datenübertragung von und zum Heimarbeitsplatz sinnvoll. Ursächlich für diese Entscheidung sind die höhere Übertragungsrate

[16] Es können auch Modems nach V.32 oder andere Modems mit niedrigeren Bitübertragungsraten eingesetzt werden.

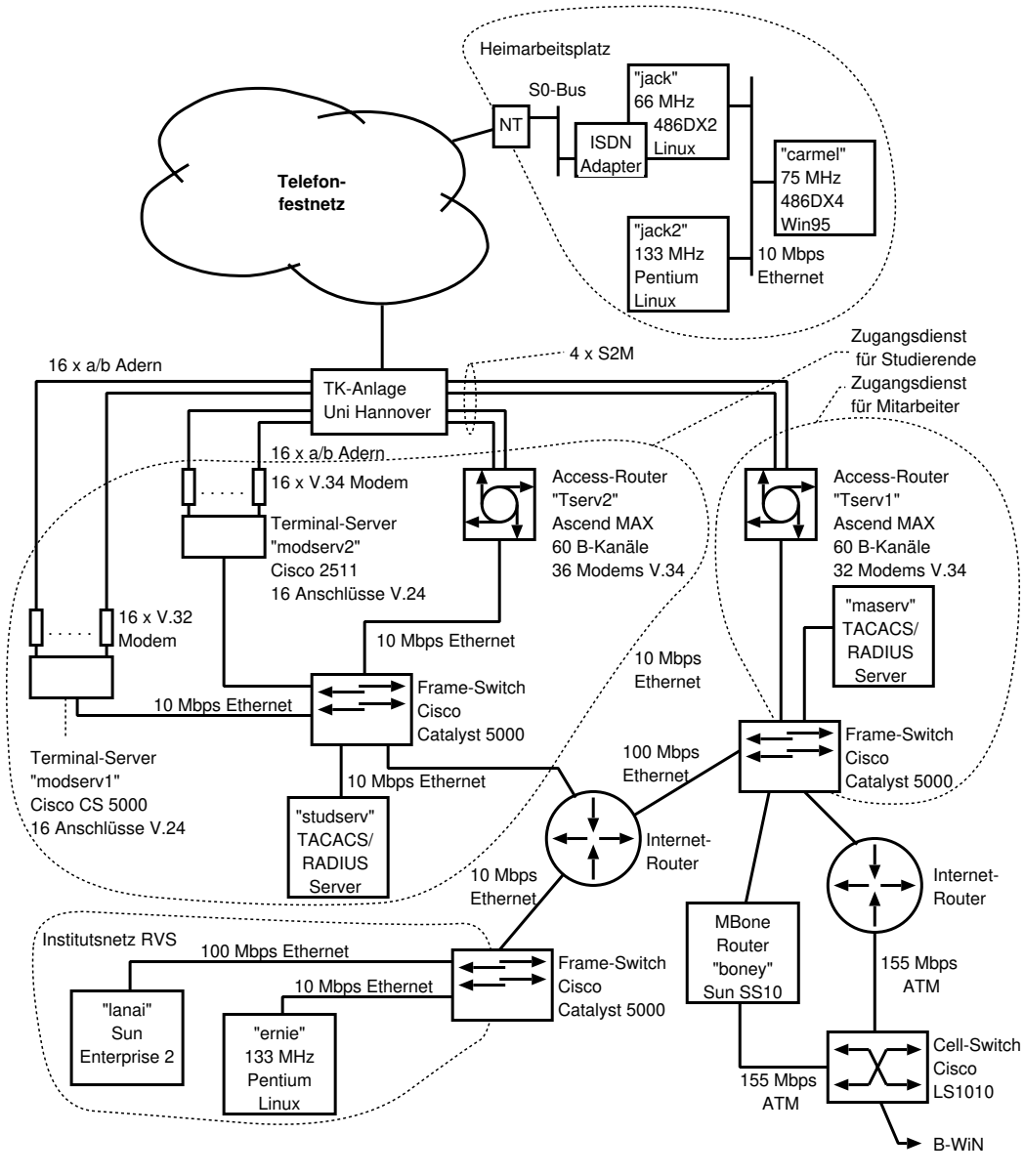

Abbildung 2.4: Zugangsdienste an der Universität Hannover

gegenüber Modems mit bereits einem B-Kanal sowie die Möglichkeit zur gebündelten Nutzung von zwei B-Kanälen an einem Basisanschluß. Damit stehen nominell 64 oder 128 kbps Übertragungskapazität zur Verfügung.

Für den Entwurf eines Systems zur Übertragung multimedialer Echtzeitdatenströme über Internet-Zugangsnetze ist eine möglichst genaue Kenntnis der Übertragungseigenschaften eines ISDN-Zugangskanals zum Austausch von IP-Datagrammen erforderlich. In diesem Abschnitt werden einige wichtige Parameter dieses Kanals vorgestellt. Die Parameter wurden auf der Basis von Messungen in unterschiedlichen Szenarien in dem in Abbildung 2.4 dargestellten Netzwerk ermittelt.

Für die Nutzung des Kanals zur Übertragung multimedialer Datenströme mit isochronem Charakter sind folgende Parameter von besonderem Interesse:

- Die Verzögerung von Datagrammen in Abhängigkeit von ihrer Größe im Falle des nicht belasteten Kanals und der Vergleich dieser Werte mit den theoretisch möglichen, berechneten Werten.
- Die maximale Durchsatzleistung unter Nutzung UDP-basierter Transportprotokolle, in der Regel RTP.
- Der Paketverlust, die Paketlaufzeit und die Veränderung der Paketlaufzeit über die Zeit in Abhängigkeit der beaufschlagten Datenrate.
- Das Verhalten des Systems bei Beaufschlagung mit Datenströmen unterschiedlicher Elastizität, insbesondere bei der Überlagerung von UDP- und TCP-Datenströmen.

2.3.1 Paketlaufzeiten

Die Ermittlung der Verzögerung von Datagrammen entlang eines Pfades läßt sich auf zwei Weisen ermitteln:

- Unter der Annahme, daß die Uhren auf den Systemen an den Pfadenden synchronisiert sind, wird eine PDU von Host A zu Host B geschickt. Diese PDU enthält einen Zeitstempel, der die Uhrzeit des Rechners A beim Versenden angibt ($C_S(t_1)$). Beim Eintreffen der PDU beim Host B wird die dortige Uhrzeit notiert($C_R(t_2)$). Die Übertragungsdauer D ergibt sich dann als $\Delta t = C_R(t_2) - C_S(t_1)$. Vernachlässigt wird hierbei die Zeit zur Behandlung der PDU in den Protokoll-Stacks der Rechner A und B. Dieser Fehler kann in Grenzen durch Berücksichtigung der durchschnittlichen Verzögerung auf dem lokalen System minimiert werden [Ste96]. Letztlich tritt diese Verzögerung aber auch bei der Übertragung von Nutzdaten auf und sollte daher nicht unberücksichtigt bleiben.

  Das eigentliche Problem bei diesem Verfahren resultiert aus der Frage, wie die Uhren auf den Systemen hinreichend genau, d.h. mit Gangunterschieden im Bereich weniger μs, zu synchronisieren sind. In [AK97] wird hierzu die Synchronisierung unter Nutzung des *Global Positioning System* (GPS)[17] oder einer Kombination von GPS und dem *Network*

[17] Gemäß [AK97] ist hierbei mit einem Gangunterschied von mehreren $10\mu s$ zu rechnen.

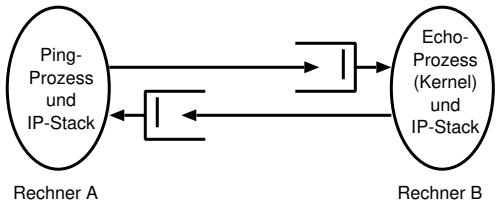

Abbildung 2.5: Warteschlangenmodell des Übertragungskanals

Time Protocol (NTP) [Mil92] vorgeschlagen. Nachteil des ersten Vorschlags ist, daß nur in Ausnahmefällen GPS-Empfänger direkt zur Verfügung stehen. Die zweite Lösung muß verworfen werden, da der Austausch von NTP-Paketen über den Kanal unter Test die Messungen verfälscht.

- Das zweite Verfahren basiert auf der Ermittlung des *Round Trip Delays* für eine PDU. Die sich ergebende Zeit wird als *Round Trip Time* (RTT) bezeichnet. Hier wird in der Regel ein ICMP-Echo-Request von Rechner A zu Rechner B geschickt. Von dort wird die PDU zurück zum Sender geschickt. Rechner A registriert die Zeit des Absendens und des Empfangs der Antwort und kann daraus die RTT bestimmen. Üblicherweise steht auf jedem System mit TCP/IP-Stack das *ping*-Programm zur Verfügung, welches dieses Verfahren implementiert. Eine weitergehende Beschreibung ist in [Ste94, Seite 85ff] zu finden. Beispiele für entsprechende Messungen sind [Ste96, Seite 291ff] zu entnehmen. Einschränkend ist zu vermerken, daß bei der Ermittlung der Übertragungszeit auf Basis der RTT mit einem maximalen Fehler von $\frac{RTT}{2}$ zu rechnen ist.[18]

Zur Ermittlung der Übertragungszeit von Datagrammen ist die Messung des RTT der unkomplizierteste Weg, wobei potentielle Fehler weitgehend ausgeschlossen werden sollten. Daraus ergibt sich die Frage nach möglichen Fehlerquellen.

Das Übertragungssystem besteht aus Leitungen (hierbei soll angenommen werden, daß sich die bitseriellen ISDN-Kanäle im Telefon-Festnetz wie Leitungen verhalten), Protokoll-Stacks und Prozessen. Auf den Leitungen entstehen deterministische Verzögerungen durch die Serialisierung der Pakete sowie durch die Übertragungszeit der Bits. Im Fall von ISDN-Verbindungen handelt es sich um einen Duplexkanal, so daß keine Verzögerungen durch die Umschaltung von Halbduplexkanälen entstehen. Auch die Protokoll-Stacks und die Prozesse arbeiten bei geringer Systemlast weitgehend deterministisch. Trotzdem weisen die RTTs Varianzen auf.

Ursache für die Varianzen sind die Warteschlangen der beteiligten Systeme. Das vereinfachte Modell in Abbildung 2.5 berücksichtigt diesen Fehler durch die Integration von Warteschlangen in den Übertragungskanal.[19]

[18] Hintergrund hierfür ist, daß dem Nachrichtensender verborgen bleibt, ob das Paket auf dem Weg vom Sender zum Empfänger und zurück gleichmäßig verzögert wurde. Im Extremfall (Datenrate des Mediums $\rightarrow \infty$, Verzögerung durch Bearbeitung $\rightarrow 0$) wird bei unsymmetrischen Queue-Belastungen die RTT ausschließlich durch die Verzögerung des Pakets in einer Richtung enstehen. Dieser Fehler beträgt maximal $\frac{RTT}{2}$, da die korrekte Übertragungszeit in der einen Richtung RTT und in der anderen Richtung $0ms$ betragen würde.

[19] Dabei unberücksichtigt bleiben die Puffer vor und in den Koppelfeldern des ISDN-Systems.

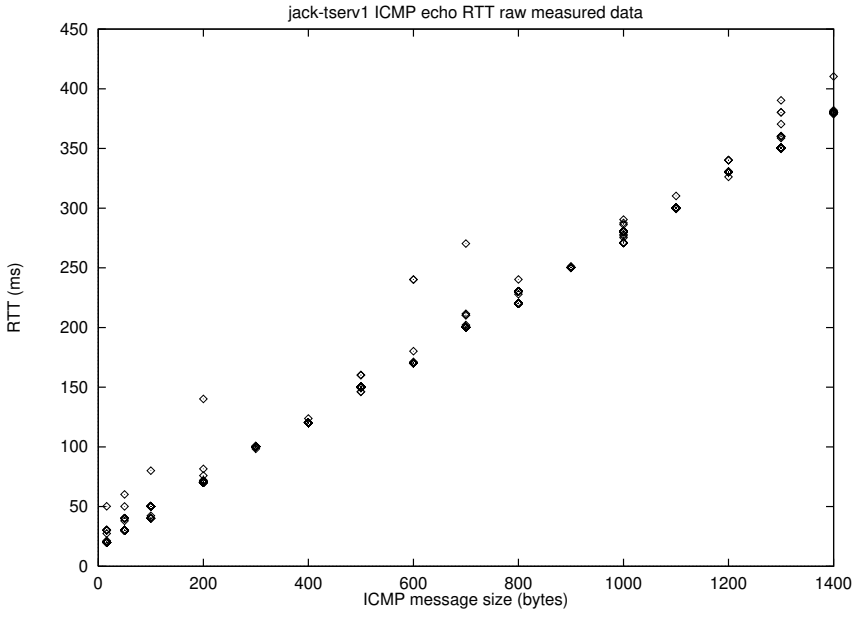

Abbildung 2.6: RTT für ICMP-Echo zwischen *jack* und *tserv1*

Die Verweilzeit der Daten in den Warteschlangen addiert sich zu den deterministischen Zeiten für die eigentliche Übertragung und die Datenbehandlung durch Protokoll-Stacks und Prozesse. Um zu zuverlässigen Aussagen hinsichtlich des Übertragungssystems zu kommen, müssen relativ viele Proben genommen werden und die minimalen Werte den weiteren Überlegungen zugrunde gelegt werden. Minimale Werte entstehen, wenn die Queues beim Eintreffen der Daten leer sind. Auf diesem Ansatz basiert das Werkzeug *pathchar* von Van Jacobson [Jac97]. Es dient der Abschätzung der Charakteristik eines Internet-Pfades und errechnet neben dem Weg, den die Datagramme durch das Internet nehmen, auch die maximalen Übertragungsraten der Verbindungsnetzwerke.

Entsprechend diesem Modell wurden Messungen der RTTs für verschiedene Paketgrößen auf der Basis von ICMP-Echo-Requests zwischen den Rechnern *jack* und *Tserv1* gemäß Abbildung 2.4 durchgeführt. Dabei wurde die Paketgröße zwischen 16 Byte und 1400 Byte in Schritten variiert. Es wurden jeweils 50 Proben für jede ICMP-Paketgröße genommen. Abbildung 2.6 zeigt die Ergebnisse einer solchen Messung.

Ausgehend von der Annahme, daß leere Warteschlangen zu minimalen RTT-Werten führen, werden im folgenden nur noch diese minimalen RTT-Werte berücksichtigt. Während die Verzögerung auf dem Übertragungskanal in Messungen mit dem direkt über einen IP-Hop erreichbaren Access-Router ermittelt werden konnte, gilt dies nicht für die zusätzlich erforderlichen Messungen bzgl. des Bulk-Transfers. Hier müssen frei programmierbare UNIX-Hosts im Institutsnetz angesprochen werden. Daher wurden auch RTT-Messungen zu den Systemen *lanai.rvs.uni-hannover.de* und *ernie.rvs.uni-hannover.de* in gleicher Weise durchgeführt. Der netztopologische Aufstellungsort der Rechner läßt sich wiederum aus Abbildung 2.4 entnehmen. Als Referenz wurde zudem eine entsprechende Messung auf dem ISDN-Interface des Klienten *jack* durchgeführt. Die Ergebnisse wurden mit in die Ergebnistabelle 2.4 aufgenommen.

ICMP Paketgröße (Payload+ICMP) (Bytes)	RTT-Anteil Serialisierung[20] (ms)	min. RTT tserv1 (ms)	min. RTT lanai (ms)	min. RTT ernie (ms)	min. RTT jack/ippp0[21] (ms)
16	10.75	20.0	26.6	27.9	0.8
50	19.25	29.6	34.2	40.0	0.8
100	31.75	40.1	50.1	50.1	0.8
200	56.75	70.1	70.1	74.2	0.8
300	81.75	94.0	100.1	100.1	0.9
400	106.75	120.1	125.7	130.0	0.8
500	131.75	146.1	149.7	150.1	1.0
600	156.75	170.1	180.1	179.7	1.0
700	181.75	200.1	200.2	210.1	1.0
800	206.75	219.7	230.1	229.7	1.0
900	231.75	250.1	260.1	260.2	1.0
1000	256.75	270.2	280.2	290.2	1.1
1100	281.75	299.4	310.1	310.2	1.0
1200	306.75	322.7	330.1	340.2	1.1
1300	331.75	348.0	359.9	360.2	1.1
1400	356.75	378.7	390.3	390.1	1.1

Tabelle 2.4: Minimale ICMP-RTTs von *jack* zu anderen Rechnern in Abhängigkeit von der Paketgröße

Bei Betrachtung der Ergebnisse wird deutlich, daß die Serialisierungszeit der Daten zur Übertragung über die ISDN-Strecke ab ca. 50 Byte Paketgröße den größten Anteil der RTT ausmacht. Die Ergebnisse der Berechnungen und Messungen sind in Abbildung 2.7 noch einmal grafisch dargestellt. Hier wird dieser Effekt noch klarer erkennbar.

2.3.2 Übertragungskapazität

Von besonderem Interesse für die hier betrachteten Szenarien ist die Übertragungskapazität der ISDN-Strecke unter Nutzung des UDP-Protokolls. In den folgenden Absätzen werden die Ergebnisse bei der Ermittlung dieser Kapazität in Abhängigkeit von der Nachrichtengröße dargestellt.

Die Messung basiert auf der Flutung des Kanals mit UDP-Datagrammen, d.h. es wurde die Übertragungsleistung bei UDP-Bulk-Transfer ermittelt. Im idealen Fall müßte die Messung zwischen den Systemen *jack* und *tserv1* erfolgen. Dies ist jedoch nicht möglich, weil auf dem Access-Router keine Programme installiert und genutzt werden können. Stattdessen wurden die Messungen zwischen Rechner *jack* und einem Rechner im Institutsnetz des Lehrgebiets Rechner-

[20] Der RTT-Anteil durch Serialisierung ergibt sich gemäß Abschnitt 2.2.2, Seite 10, zu $2 \cdot 8 \cdot (\frac{Bit}{Byte})\ (Paketgröße + 27)(Byte) / Übertragungsrate(\frac{Bit}{s})$. Der Faktor 2 ergibt sich aus der zweimaligen Serialisierung der Pakete (beim Sender und Empfänger). Der konstante Summand von 27 Byte ergibt sich aus dem Protokoll-Overhead von 7 Byte für den PPP-Header und 20 Byte für den IP-Header.

[21] *ippp0* bezeichnet das lokale PPP-ISDN-Interface des Rechners *jack*. Die Werte dieser Messungen verstehen sich als Abschätzung für die Verzögerung, die durch die Behandlung der Pakete im TCP/IP-Stack des Rechners *jack* entsteht.

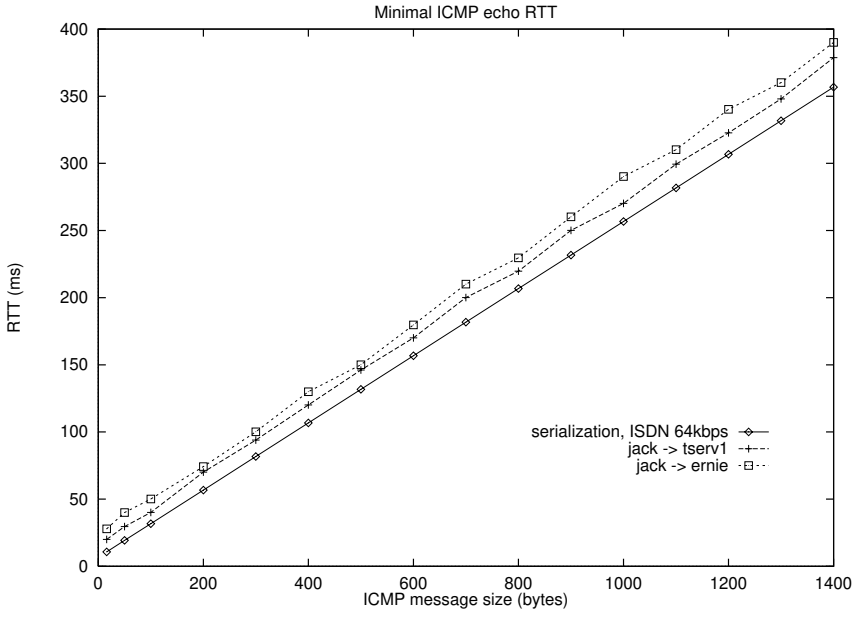

Abbildung 2.7: Minimale RTTs für ICMP-Echo zwischen *jack*, *tserv1* und *ernie*

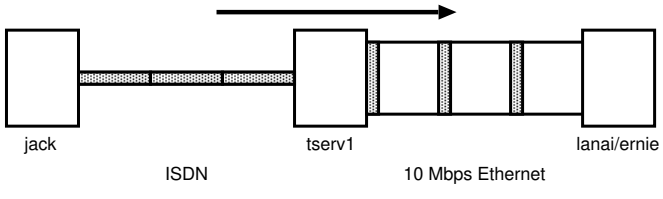

Abbildung 2.8: UDP-Bulk-Transfer zwischen *jack* und *ernie*

netze und Verteilte Systeme, *ernie*, durchgeführt. Anders als TCP verfügt UDP über keinerlei Flußkontrolle. So wird bei der Übertragung größerer Datenmengen über UDP ohne weitere Flußkontrolle das direkt anschließende Netzwerk mit UDP-Datagrammen geflutet. Fungiert der mit dem ISDN-Netz verbundene Rechner *jack* als Sender, wird der ISDN-Kanal geflutet und, soweit hinreichende Vermittlungkapazität im Access-Router und allen folgenden Komponenten des Netzpfades zwischen *jack* und *ernie* vorhanden sind, ohne Fehler und Paketverluste übertragen. Abbildung 2.8 illustriert dieses Szenario.

Paketverluste entstehen, wenn es zu Überläufen in den Queues des sendenden Rechners, der Vermittler entlang des Pfades oder des empfangenden Rechners kommt.

Problematisch sind Messungen mit *ernie* als Sender. Dieser Rechner ist direkt an ein 10 Mbps Ethernet angeschlossen, das bei der Messung geflutet wird. Damit wird auch der Access-Router überflutet, da er die Datagramme nicht so schnell weiterleiten kann, wie er sie erhält. Der etwaige Empfang von ICMP *Source Quench* Nachrichten wird die Situation nicht verbessern, da das Meßprogramm diese nicht auswertet[22]. Abbildung 2.9 stellt diese Problematik dar. Es ist of-

[22]ICMP *Source Quench* Nachrichten können von einem Router oder Host erzeugt werden, wenn IP-Datagramme mit einer höheren Datenrate eintreffen als sie verarbeitet werden können (vgl. [BP87, Sektion 2.2.3], [Bak95, Sektion 4.3.3.3]). Zum Nutzen von ICMP Source-Quench Nachrichten siehe Abschnitt 3.2.2, Seite 70ff.

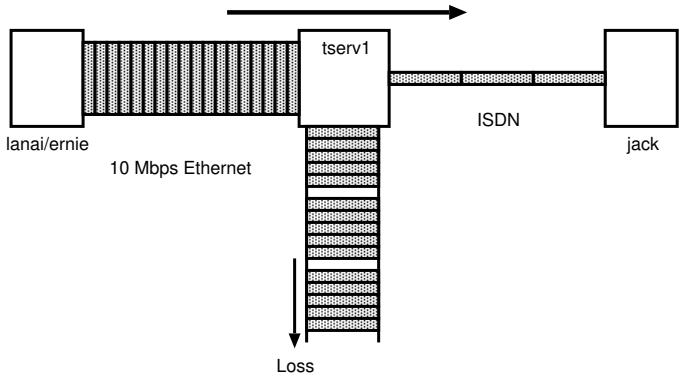

Abbildung 2.9: UDP-Bulk-Transfer zwischen *lanai* oder *ernie* und *jack*

fensichtlich, daß die Durchführung einer solchen Messung keinen relevanten Erkenntnisgewinn mit sich bringt und die Flutung des Netzes am Access-Router lediglich andere Nutzer behindern würde.

Zur Ermittlung der Bulk-Transferleistung wurden im ersten Schritt die beiden im Internet frei verfügbaren und häufig zur Ermittlung von Übertragungsraten eingesetzten Programmpakete *TTCP* und *netperf* [Hew96] evaluiert. Für die hier betrachteten Szenarien ist insbesondere der Durchsatz der ISDN-Strecke unter Nutzung des UDP-Protokolls von Interesse. Daher mußte das auszuwählende Programm die Ermittlung von Übertragungsraten über UDP gestatten. Beide Programmpakete bieten einen entsprechenden Modus in unterschiedlicher Ausführung. Ein nähere Analyse zeigte jedoch schnell, daß *TTCP* ungeeignet ist, da im UDP-Modus weder Anfang noch Ende der Messung zwischen Sender und Empfänger signalisiert werden. Die Ergebnisse der Messungen sind damit nur zur Ermittlung von Größenordnungen der Übertragungsleistung geeignet.

netperf ermöglicht die Ermittlung der *bulk data transfer performance* und der *request/response performance* über verschiedene Transportschichten. Dazu können neben TCP und UDP auch Unix-Domain-Sockets, das *Data Link Provider Interface* (DLPI) sowie das *Fore ATM API* benutzt werden. *netperf* ist als Client-Server-System realisiert. Client und Server stehen über eine TCP-Verbindung in Kontakt. Sie wird zur Konfiguration der Messung und zum Austausch der Ergebnisse benutzt.

netperf schien das geeignete Werkzeug zur Ermittlung der Übertragungsleistung über UDP zu sein und wurde daher in der Version 2.1pl1 auf den Linux-Systemen *jack* und *ernie* problemlos installiert. Auf den Sun-Systemen des RVS war das Programm bereits vorhanden.

Erste Testmessungen mit *netperf* zeigten allerdings, daß die ermittelte Übertragungsrate in einigen Fällen die theoretisch mögliche Übertragungsrate[23] überstieg. Vergleichende Messungen, bei denen die beim Empfänger eingehenden Nachrichten mit dem Paket-Monitor *tcpdump* protokolliert und später analysiert wurden, lieferten den Beweis dafür, daß die von *netperf* ermittelten Ergebnisse falsch waren. Die Analyse des Programm-Quelltextes von *netperf* bestätigte dies. Daraufhin wurde das *netperf*-Programm so überarbeitet, daß korrekte Werte ermittelt werden.

[23]Die theoretisch mögliche Übertragungsrate ergibt sich zu $\frac{(mesg.size)}{mesg.size+35} \cdot 64kbps$. Die Konstante im Nenner ergibt sich aus dem Protokoll-Overhead von 7 Byte für PPP, 20 Byte IP- und 8 Byte UDP-Header.

Nachrichtengröße (Byte)	Dauer send/recv (s)	Gesendete Nachrichten	Empfangene Nachrichten[24]	Theor. Datenrate[25] (kbps)	Sende-Datenrate (kbps)	Empfangs-Datenrate (kbps)
8	120.00/120.49	473384	20959	11.91	252.47	11.13
50	120.00/121.11	483220	11009	37.65	2169.90	36.36
100	120.00/121.79	480979	7049	47.41	3206.55	46.30
200	120.00/122.35	4103	4102	54.47	54.71	53.64
300	120.00/122.25	2894	2893	57.31	57.88	56.80
400	120.00/122.44	2237	2235	58.85	59.65	58.41
500	120.00/122.97	1829	1827	59.81	60.97	59.43
600	120.00/123.08	1544	1543	60.47	61.76	60.18
700	120.00/122.00	1323	1322	60.95	61.74	60.68
800	120.00/122.80	1173	1172	61.32	62.56	61.08
900	120.00/123.76	1056	1055	61.60	63.36	61.38
1000	120.00/122.95	948	947	61.84	63.20	61.62
1100	120.00/123.83	871	870	62.03	63.87	61.83
1200	120.00/123.72	800	799	62.19	64.00	62.00
1300	120.00/124.18	743	742	62.32	64.39	62.14
1400	120.00/123.39	687	686	62.44	64.12	62.27

Tabelle 2.5: Bulk-Transfer-Messung zwischen *jack* und *ernie* in Abhängigkeit von der Paketgröße

Die Beschreibung der Fehlerursachen sowie der durchgeführten Korrekturmaßnahmen sind in Anhang B, Seite 175, beschrieben.

Mit der korrigierten *netperf*-Version wurde die Übertragungskapazität in der bereits beschriebenen Testkonfiguration überprüft. Dazu wurden Messungen vom Rechner *jack* zum Rechner *ernie* durchgeführt. Jede Messung bestand aus dem Versenden von UDP-Datagrammen, die Nutzdaten zwischen 8 und 1400 Byte enthielten. Es wurde ermittelt, wieviele Datagramme der Sender versendet hat und wieviel Datagramme den Empfänger erreichten. Die Ergebnisse einer solchen Messung sind in Tabelle 2.5 und grafisch in Abbildung 2.10 dargestellt.

Bei Betrachtung der Ergebnisse ist auffällig, daß für die Messungen mit den Nachrichtengrößen 8, 50 und 100 Byte große Verlustraten ausgewiesen werden. Sie entstehen durch Überflutung der Queues des sendenden Rechners im Bereich des ISDN-Kartentreibers. Dies konnte durch Fehlermeldungen des Kartentreibers belegt werden. Für die weiteren Messungen sind die Verluste im wesentlichen ebenfalls diesen Queue-Problemen zuzuordnen. Die über der theoretisch möglichen Datenrate liegende Sendedatenrate ist ein sicheres Indiz hierfür. Allerdings treten die Überflutungen hier seltener auf.

Die weitere Interpretation stützt sich auf die Meßergebnisse des empfangenden Rechners. Der Berechnung der Empfangsdatenrate liegt die Zahl der empfangenen Nachrichten zugrunde. Tatsächlich wurde jeweils eine PDU mehr empfangen, die jedoch zur Festlegung des Startzeitpunktes der Messung benutzt wurde. In Abbildung 2.10 sind die theoretisch möglichen Datenraten nebst den korrigierten Empfangsdatenraten aufgetragen.

[24] Die Zahl der empfangenen Pakete ist stets um eins kleiner als die Zahl der gesendeten Paktete. Ursache hierfür ist, daß der Empfang des ersten Paket zum Anstoß des Meßintervalls diente. Siehe hierzu auch Anhang B, Seite 175ff..

[25] Die theoretische Übertragungsdatenrate ergibt sich aus dem Term $\frac{(mesg.size)}{mesg.size+35} \cdot 64kbps$.

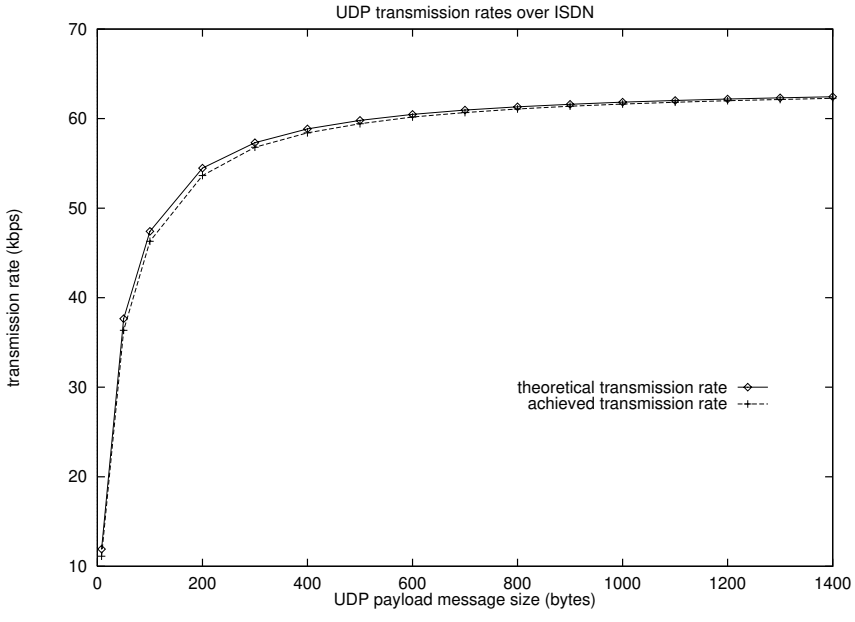

Abbildung 2.10: UDP-Übertragungsrate über ISDN in Abhängigkeit von der UDP-Nachrichtengröße

Die Grafik zeigt, daß die erzielten Datenraten nahe an den theoretisch möglichen Datenraten liegen. Die absolute Abweichung liegt zwischen 1.3 kbps bei kleinen PDUs und 0.17 kbps bei großen PDUs. Die erreichte Kanalauslastung ist bei kleinen PDUs aufgrund des Protokoll-Overheads gering, bei großen PDUs werden günstigere Werte erreicht. Hier zeigt sich der Trade-off für die betrachteten Szenarien: Kurze PDUs weisen geringe Verzögerungen auf, wohingegen große PDUs eine bessere Kanalauslastung erlauben.

Letztlich zeigt die Betrachtung der obigen Kanalmodelle, daß die Paketverluste nur dann in akzeptablen Grenzen gehalten werden können, wenn eine Begrenzung des Datenflusses beim Sender realisiert wird. Anders als bei den klassischen Anwendungen der Datenkommunikation mit Burst-Charakter ist der Datenfluß bei Audio- und Video-Datenströmen kontinuierlich, so daß die Zwischenspeicherung von Paketen in Queues zur Überwindung kurzfristiger Überlastsituationen bei dieser Anwendung eher schädlich als nützlich ist.

Hinzu kommt, daß verspätet eintreffende Audio- und Video-Daten wertlos sind, da interaktive interpersonelle Kommunikation isochrone Datenströme erfordert. Dabei sollte eine maximale Ende-zu-Ende-Verzögerung von maximal 150 ms nicht überschritten werden, wenngleich die ITU Ende-zu-Ende-Verzögerungen von bis zu 400 ms als akzeptabel ansieht[26]. Für die Übertragung über ISDN bedeutet dies, daß ein Audio-Datagramm, welches am Access-Router angelangt, während gerade mit der Serialisierung einer 1200 Byte großen PDU begonnen wurde, theoretisch nicht mehr übertragen zu werden bräuchte, da der Empfänger den Inhalt des Pakets aufgrund der erhöhten Verzögerung nicht mehr ausspielen kann. Die laufende Serialisierung kann noch bis zu 150 ms in Anspruch nehmen und erst dann kann mit der Serialisierung des

[26]Gemäß [Int93] ist eine Ende-zu-Ende-Verzögerung von 150 ms für die meisten Anwendungen akzeptabel. Eine Verzögerung unterhalb von 400 ms wird für internationale Verbindungen, z.B. unter Verwendung von Satelliten-Hops, als akzeptabel angesehen.

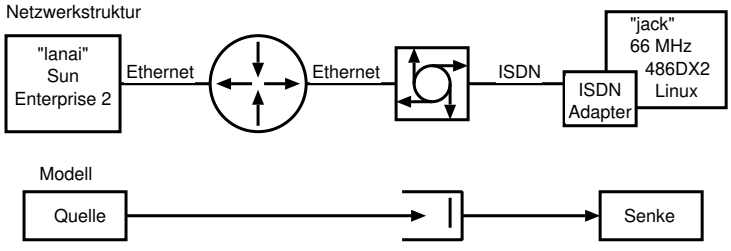

Abbildung 2.11: Versuchsaufbau zur Ermittlung der Paketverluste im *Down-Stream*

Audio-Datagramms begonnen werden.

2.3.3 Paketverlustraten

Hohe Verlustraten sind gerade für Audio-Datenströme nicht tragbar, da hierdurch die Kommunikation erheblich gestört wird. Die Begrenzung der Datenraten hat nur dann Erfolg, wenn gleichzeitig Sorge getragen wird, daß die Delays in den Randsystemen zur ISDN-Strecke nicht zu groß werden. Dies kann nur dann gewährleistet werden, wenn eintreffende Audio-Daten bevorzugt behandelt werden.

Aus diesen Überlegungen entstehen zwei Fragen:

1. Wie entwickelt sich der Paketverlust auf der Übertragungsstrecke in Abhängigkeit von der beaufschlagten Datenrate?

2. Welche Verzögerungen erfahren die Pakete in Abhängigkeit von der beaufschlagten Datenrate?

Zur Beantwortung der ersten Frage wurden exemplarische Messungen zwischen den Rechnern *lanai* und *jack* durchgeführt. Bei konstanter Paketgröße wurde die Datenrate im Grenzbereich der Übertragungskapazität variiert. Ein Problem war hierbei, daß die Erzeugung von Datenströmen fester Datenrate mit den üblichen Meßwerkzeugen nicht unterstützt wird. Daher entstand im Rahmen dieser Arbeit ein Meßwerkzeug mit dem Namen *rtest*.

rtest gestattet die Emission eines Datenstroms fester Paketrate und liefert, sofern RTTs ermittelt werden sollen, Daten über Verzögerung, Varianz der Verzögerung und weitere statistische Daten. Eine eingehendere Beschreibung von *rtest* befindet sich in Anhang A, Seite 159.

Zur Ermittlung der Paketverluste wurde die in Abbildung 2.11 dargestellte Netzwerkstrecke im *Down-Stream*[27] durch *rtest* mit einem Datenfluß beaufschlagt. An den Endsystemen *lanai* und *jack* wurde der Datenstrom mit dem *tcpdump* protokolliert. Die Messung bestand in der Übertragung von jeweils 1000 UDP-PDUs mit einer Payload von 300 Byte. Die Datenübertragungsrate wurde zwischen 40 kbps und 100 kbps variiert. Dabei ergaben sich die in Tabelle 2.6 dargestellten Ergebnisse.

[27] Unter dem *Down-Stream* wird die Datenübertragung aus dem Internet an den Arbeitsplatz des Nutzers verstanden.

Sende-datenrate (kbps)	Gesendete PDUs	Empfangene PDUs	Paket-verlust (%)	Sende-dauer (s)	Empfangs-dauer (s)	Empfangs-datenrate (kbps)
40	1000	1000	0	60	60	40
50	1000	1000	0	48	48	50
55	1000	1000	0	43	44	54
60	1000	1000	0	40	42	57
65	1000	1000	0	37	42	57
70	1000	907	9	34	38	57
75	1000	882	12	32	38	56
80	1000	773	23	30	32	57
85	1000	726	27	28	31	56
90	1000	684	32	26	29	57
100	1000	619	38	24	26	57

Tabelle 2.6: Paketverluste bei unterschiedlichen Datenraten

Interessant an diesem Meßergebnis ist, daß obwohl die theoretische Übertragungsrate bei 300 Byte/UDP-PDU einen Wert von 57.31 kbps hat, Paketverluste erst bei einer Übertragungsrate von 70 kbps auftreten. Die Ursache für diesen Effekt wird unmittelbar aus dem Ersatzschaltbild für die dieser Messung zugrunde liegenden Netzwerkstruktur deutlich: Die Sun-Workstation *lanai* emittiert die PDUs mit einer festen Paketrate. Da der daraus entstehende Datenfluß für das sich anschließende Ethernet-Netzwerk und den IP-Router marginal ist, wird er nahezu ohne Verzögerung übertragen. So gelangt ein fast isochroner Datenstrom an den Access-Router. Da die Übertragungsrate auf der anschließenden ISDN-Verbindung geringer ist, werden die Pakete solange in der Queue des ausgehenden Interfaces zwischengespeichert, bis diese überfließt.

Tatsächlich ist nicht der Paketverlust an dieser Stelle das eigentliche Problem, sondern die Verzögerung der gestauten Pakete. Sie erzeugen einen großen Jitter[28]. Dies wird aus dem Vergleich zwischen Sende- und Empfangsdauer deutlich. Wird unterstellt, daß es sich bei dem Datenstrom um einen Audio-Datenstrom handelt, liegt die Sendedauer im Bereich eines typischen *Talkspurt*[29]. Die maximale Verzögerung zwischen dem Empfang des ersten und des letzten Datenpakets des Talkspurt von ca. 5 Sekunden ist nicht akzeptabel; das Audio-Tool beim Empfänger wird PDUs mit größerer Verzögerung verwerfen. Durch ihre Übertragung wird der Kanal unnötig belegt. Diesem Gesichtspunkt wird im Entwurf, Abschnitt 3.2.2, Seite 69, Rechnung getragen.

2.3.4 Überlagerung von UDP- und TCP-Datenströmen

Ebenso interessant ist in diesem Zusammenhang die Fragestellung, wie sich das Übertragungssystem verhält, wenn der mit einem unidirektionalen UDP-Datenstrom belastete Übertragungskanal zusätzlich mit einem TCP-Datenstrom beaufschlagt wird. Die zentrale Fragestellung dabei ist, wie sich die Datenströme zueinander verhalten und welcher Datenstrom den anderen verdrängt. Dazu wurde im gleichen Szenario der Netzwerkpfad mit einem konstanten Datenstrom

[28] Gemäß [Ste93] bezeichnet der *Jitter* die maximale zeitliche Varianz beim Eintreffen der Daten am Bestimmungsort.

[29] Unter einem *Talkspurt* wird ein Audio-Datenstrom verstanden, der in einer Konferenz durch den Beitrag eines Teilnehmers ohne zwischenzeitliche Pausen entsteht.

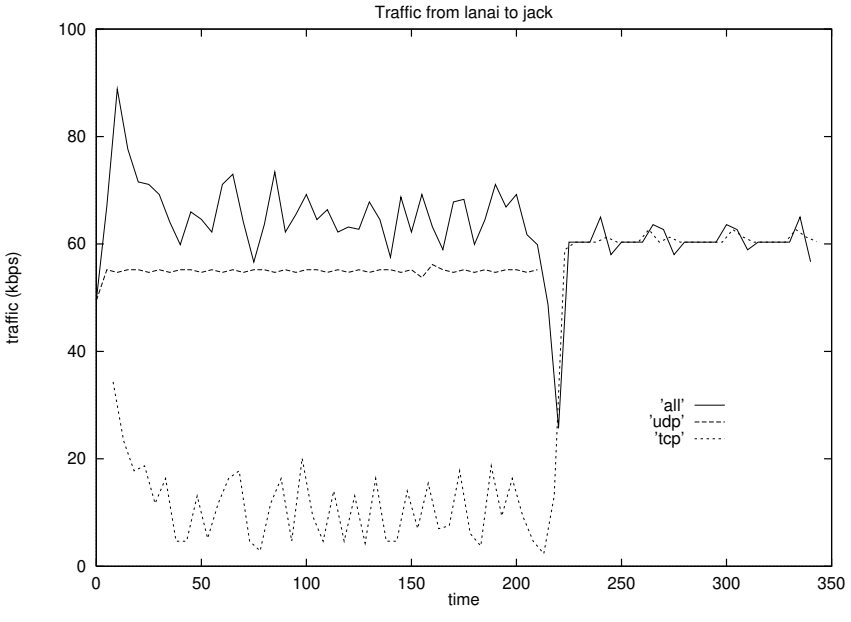

Abbildung 2.12: Überlagerung eines UDP-Flusses mit einem TCP-Fluß, gemessen am Sender *lanai*

von 55 kbps belastet und nach kurzer Zeit die Übertragung einer 650 KB großen Datei unter Nutzung des *File Transfer Protocol* (FTP) initiiert. Die sich ergebenden Datenströme wurden erneut mittels *tcpdump* auf den Rechnern *lanai* und *jack* protokolliert; die Ergebnisse zeigen die Abbildungen 2.12 und 2.13.

Bei Betrachtung der Abbildung 2.12, die die Datenflüsse beim Sender, dem Sun-Rechner *lanai* darstellt, fällt während der ersten 200 Sekunden ein nahezu konstanter UDP-Datenstrom von 55 kbps auf. Der Anstieg am linken Rand entsteht durch Randeffekte. Einige Sekunden nach dem Start der Messung wird der UDP-Datenstrom um einen TCP-Datenstrom, die Datei-Übertragung, ergänzt. Der Gesamtdatenstrom steigt deutlich über die Grenze des Übertragbaren hinaus. Paketverluste am Access-Router sind die Folge und TCP reduziert entsprechend dem Slow-Start-Algorithmus die Fenstergröße.[30] Während der emittierte UDP-Datenstrom konstant bleibt, reagiert TCP elastisch und begrenzt die Übertragungsrate beim Sender. Gesteuert wird das Verhalten durch das Ausbleiben von Quittungen des Empfängers, die auf Paketverluste hinweisen. Das Ende des UDP-Datenstrom nach ca. 200 Sekunden Meßdauer erkennt die TCP-Verbindung anhand der geringeren Verlustrate und füllt den Kanal umgehend mit großen PDUs aus, bis die Dateiübertragung beendet ist.

Das Bild rundet sich mit der Betrachtung des in Abbildung 2.13 dargestellten Meßergebnisses beim Empfänger, dem Rechner *jack*, ab: Deutlich zu erkennen ist der zu Anfang normale UDP-Datenfluß, der dann zeitgleich mit dem Anstieg des TCP-Datenstroms einbricht. Ursächlich hierfür ist das Fluten der PPP-Interface-Queue am Access-Router. Während zu Anfang die Queue mit Füllgeschwindigkeit geleert werden kann, überfüllt der TCP-Datenstrom die Queue beim initialen Slow-Start. Die Rückkopplung, daß Pakete verloren gegangen sind, erhält der Sender erst nach relativ langer Zeit. Er reagiert darauf mit der deutlichen Reduktion des emittierten

[30] Eine eingehende Erläuterung des Slow-Start findet sich z.B. in [Ste94], Seite 285 ff.

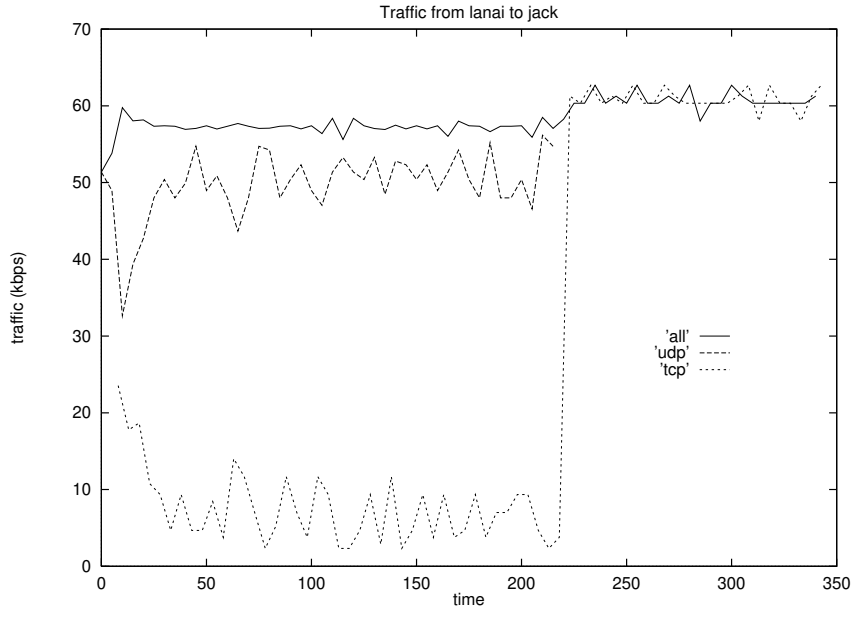

Abbildung 2.13: Überlagerung eines UDP-Flusses mit einem TCP-Fluß, gemessen am Empfänger *jack*

Datenstroms, wodurch sich der UDP-Datenstrom erholt und auf die ursprüngliche Größenordnung ansteigt. TCP versucht jedoch immer wieder die Fenstergrößen zu erhöhen und drängt in Intervallen den UDP-Datenstrom zurück. Letztlich endet der UDP-Datenstrom nach ca. 200 Sekunden und der gesamte Kanal wird vom TCP-Datenstrom ausgefüllt. Dies ist in Abbildung 2.13 an der ansteigenden Flanke des TCP-Datenstroms bei ca. 220 Sekunden zu erkennen.

Aus dieser Messung werden folgende Punkte deutlich:

- Der TCP-Datenstrom reagiert auf die Stauungen elastisch, indem die Übertragungsrate reduziert wird. Der UDP-Datenstrom wird mit konstanter Datenrate emittiert, die Paketverluste durch Stauungen sind beim Empfänger jedoch klar erkennbar. Der UDP-Datenstrom ist unelastisch gegenüber dem TCP-Datenstrom.

- Während ein einzelner konstanter UDP-Datenstrom auf einer Übertragungsstrecke geringer Übertragungskapazität auch als konstanter Datenstrom empfangen wird, schwingt die Datenrate des UDP-Datenstroms im Fall der Überlagerung mit dem TCP-Datenstrom mit kleiner Frequenz. Diese Frequenz wird durch TCP und von der Größe der Queue am Access-Router bestimmt.

- Die Steuerung der Datenrate des TCP-Datenstroms basiert auf Paketverlusten und dem Ablauf von Timern beim Sender. Aufgrund der großen Kapazität der Queue beim Access-Router sowie der toleranten Timer ist dies eine Regelschleife mit großer Reaktionszeit. Sie ist zu groß für PDUs in Echtzeitdatenströmen, für die eine konstante und kurze Verzögerung auf dem Übertragungsweg wichtig ist.

Zusammenfassend ist festzuhalten, daß bei der Überlagerung von UDP-Datenströmen durch TCP-Datenströme erstere ohne weitere Regelungsmechanismen zwar unelastisch bleiben, der

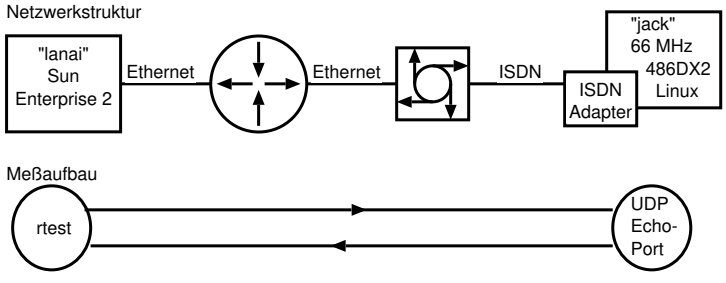

Abbildung 2.14: Messung zur Ermittlung der Paketlaufzeitentwicklung

TCP-Slow-Start sie jedoch hinsichtlich Paketverlusten und Paketlaufzeit verzerrt. Die Verzerrung stört multimediale UDP-Datenströme nachhaltig. Diesem Problem kann nur durch Ressourcen-Zuweisung und -Reservierung auf der Übertragungsstrecke Rechnung getragen werden. Die entsprechenden Mechanismen entstehen zur Zeit. Das entscheidende Element ist hierbei das *Resource Reservation Protocol* (RSVP) der *Internet Engineering Task Force* (IETF) [Bra97], welches zur Zeit in Access-Routern noch nicht angeboten wird.

2.3.5 Paketlaufzeitentwicklung bei unterschiedlichen Datenraten

Letztlich bleibt die Frage offen, wie sich Paketlaufzeiten in Talkspurts über die ISDN-Strecke entwickeln, wenn diese an der Grenze ihrer Übertragungskapazität betrieben wird. Qualitativ ist davon auszugehen, daß die ersten PDUs des Talkspurt eine kleine Verzögerung erfahren, wohingegen die letzten Pakete stark verzögert werden. Offen ist die Frage, wie sich die Funktionen der Paketlaufzeit über die Zeit und der Paketverlust quantitativ entwickeln. Dazu wurde die in Abbildung 2.14 illustrierte Messung durchgeführt.

Auf dem Rechner *lanai* wurde mit dem Meßwerkzeug *rtest* ein konstanter Paketdatenstrom erzeugt, der zum UDP-Echo-Port des Rechner *jack* gesendet wurde. *rtest* bewahrt für jede emittierte PDU einen Zeitstempel mit dem Sendezeitpunkt auf. Der UDP-Echo-Port wird auf UNIX-Systemen vom Internet-Daemon (inetd) direkt bedient und erfordert beim Ansprechen keinen Programmstart. Eine auf diesem Port erhaltene PDU wird unmittelbar und unverändert an den Sender zurück gesendet. Beim Sender wird die PDU vom *rtest*-Programm entgegen genommen und ein Zeitstempel mit dem Empfangszeitpunkt gespeichert. Nach Abschluß der Messung werden die sich ergebenden Round-Trip Paketlaufzeiten ermittelt und ausgegeben. Daneben werden auch verlorengegangene Pakete registriert.

Im Rahmen dieser Messung wurde die Strecke mit Datenraten von 40, 55 und 70 kbps beaufschlagt. Die Größe der UDP-Payload betrug 300 Byte. Es wurden jeweils 1000 PDUs gesendet, von denen aus Gründen der Übersichtlichkeit nur die ersten 500 PDUs in die in Abbildung 2.15 dargestellte Ergebnisgrafik einflossen.

Das Ergebnis zeigt deutlich, was bereits aus den Messungen zur Bestimmung der Paketverlustraten anzunehmen war: Die RTTs für Datenraten unterhalb der Übertragungskapazität der ISDN-Strecke sind nahezu gleich und konstant. Wird die Strecke mit einem größeren Datenstrom beaufschlagt, steigt die Paketlaufzeit linear auf sehr hohe Werte bis schließlich Paketverluste auftreten. Sie wurden in Abbildung 2.15 als RTTs mit dem Wert Null markiert. Ursache

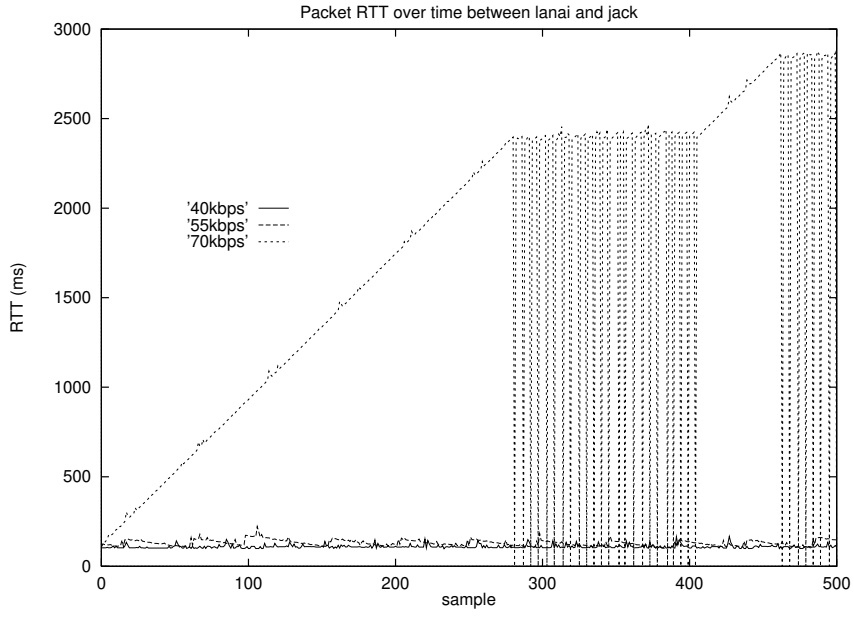

Abbildung 2.15: Meßergebnisse zur Paketlaufzeitentwicklung

für die Paketverluste ist im wesentlichen die Warteschlange im Access-Router. Wenn mehr Pakete eintreffen, als über die ISDN-Strecke übertragen werden können, werden die Pakete in der Warteschlange zwischengespeichert. Erst wenn der für die Warteschlange verfügbare Speicher belegt ist, werden eintreffende Pakete verworfen.

Aus dem Meßergebnis folgt, daß die Regelung des UDP-Datenstroms nicht auf Paketverlusten, sondern auf der Varianz der Verzögerung basieren sollte. Nur so kann erreicht werden, daß Paketverlustraten und Verzögerung möglichst klein bleiben.

Einige der aufgezeigten Probleme würden sich schon durch die Verfügbarkeit eines, bezogen auf den jeweiligen Datenstrom faireren Queuing-Verfahrens in den Routern deutlich verringern. Die Diskussion um Queuing-Verfahren in Routern wird im Entwurfskapitel (Siehe Abschnitt 3.2.1, Seite 65) aufgegriffen und vertieft.

2.4 Eigenschaften von Datenströmen auf dem MBone

In diesem Abschnitt werden die Eigenschaften von Datenströmen auf dem MBone untersucht, die durch multimediale Online-Konferenzen entstehen. Ausgehend vom Referenzmodell für multimediale Konferenzen auf dem MBone werden Anwendungs- bzw. Medienklassen und die damit verbundenen Datenströme vorgestellt. Weiter werden das *Real Time Transport Protocol* (RTP) und seine Besonderheiten kurz dargestellt. Der Abschnitt schließt mit der Präsentation von Meßergebnissen einiger typischer MBone-Konferenzen. Der Schwerpunkt der Betrachtungen liegt stets bei den durch die Konferenzen entstehenden Datenströme.

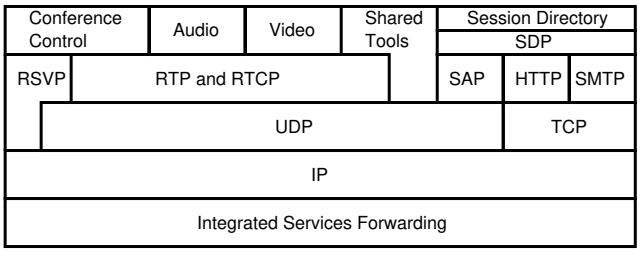

Abbildung 2.16: *Internet Conferencing Protocol Stack* nach [CWHC96]

2.4.1 Der Internet Conferencing Protocol Stack

Multimediale Online-Konferenzen auf dem MBone basieren auf dem *Internet Conferencing Protocol Stack*, der in Abbildung 2.16 dargestellt ist. Ein wesentliches Merkmal von Konferenzen entsprechend diesem Modell ist die weitgehende Unabhängigkeit der Datenströme einzelner Anwendungsklassen voneinander. Nicht alle Teilnehmer einer Konferenz müssen alle Datenströme empfangen oder senden. Hervorzuheben ist ferner die Flexibilität des Modells, wodurch die Realisierung unterschiedlicher Konferenz-Formen möglich wird. Das Spektrum reicht von Rundfunk ähnelnden Veranstaltungsübertragungen über verteilte Seminare bis hin zu klassischen Video-Konferenzen mit zwei und mehr Partnern. Multicast ist das generelle Adressierungsverfahren, wobei Unicast als Spezialfall angesehen wird. Die Kommunikation in nicht öffentlichen Gruppen wird durch Verschlüsselung der Datenströme erreicht. Nur die Nutzer, die im Besitz eines oder mehrerer Konferenz-Schlüssel sind, können partizipieren. Zudem ist die Begrenzung des Empfangsbereichs einer Konferenz über die Belegung des *Time to Live*-Feldes (TTL-Feld) im IP-Datagramm möglich.

Aus dem Internet-Conferencing-Protocol-Stack lassen sich die folgenden Anwendungsklassen und die damit verbundenen Datenströme klassifizieren:

Audio: Audio-Datenströme sind kontinuierliche Datenströme. Im MBone-Kontext nutzen sie das *Real-Time Transport Protocol* (RTP) als Transportprotokoll (vgl. Abschnitt 2.4.2). Der Datenstrom erfordert gemäß [Ste93] einen synchronen Übertragungsmodus, wobei die maximal zulässige Ende-zu-Ende-Verzögerung durch den Zwischenspeicher beim Empfänger sowie durch die tolerierbare Verzögerung für die interaktive interpersonelle Kommunikation[31] begrenzt wird. Für die Codierung der Sprachsignale können unterschiedliche Codecs eingesetzt werden. Die heute im MBone üblichen Codierungsverfahren sind im AVP-Profile [Sch96b] festgelegt. Häufig benutzt werden PCM μ-Law nach ITU-G.711[32], Intel´s DVI ADPCM[33], GSM[34], lineare 16 Bit PCM-Codierung und eine experimentelle lineare vorhersagende Codierung vom Xerox/PARC (LPC). Die zugehörigen Codecs generieren Datenströme fester Bitrate. Die Bitströme werden bei den Sendern gesammelt

[31] In öffentlichen Telefonnetzen gilt als Obergrenze ein Wert von 150 ms unter normalen Randbedingungen (terrestrische Übertragung), bzw. 400 ms in besonderen Fällen (Kommunikation über Satelliten) [Int93]. Im Internet können diese Werte nicht immer eingehalten werden.

[32] PCM steht für *Pulse Code Modulation*.

[33] DVI steht für *Digital Video Interface*, ADPCM steht für *Adaptive Delta Pulse Code Modulation*.

[34] GSM steht für *Group Speciale Mobile*.

und in einstellbaren Intervallen in RTP-PDUs transportiert. In einer Konferenz ist üblicherweise immer nur ein Sprecher aktiv, so daß im Mittel der Datenstrom eines Senders zu transportieren ist. Unterschiedliche Sender können unterschiedliche Codecs einsetzen. Die Empfänger können in der Regel fast alle Codierungen decodieren und wiedergeben. Audio-Datenströme sind empfindlich gegenüber Paketverlusten. Jeder Paketverlust macht sich beim Empfänger auditiv bemerkbar. Neue, noch in Entwicklung befindliche Übertragungsformate stellen sich diesem Problem durch redundante Übertragung der Audio-Daten [Per97].

Video: Video-Datenströme sind ebenfalls kontinuierliche Datenströme. Auch zu ihrem Transport dient RTP. Die Anzahl der Datenströme je Konferenz variiert je nach Konferenztyp stark. Das Spektrum reicht von einem Datenstrom je Konferenz bei Broadcasts bis hin zu einem Datenstrom je Teilnehmer in klassischen Video-Konferenzen. Wie bei Audio-Datenströmen ist bei Video-Datenströmen ein synchroner Übertragungsmodus mit gleichen Randbedingungen erforderlich, d.h. die Ende-zu-Ende-Verzögerung sollte nicht mehr 150 ms betragen, als Obergrenze gelten 400 ms [Int93]. Generell sind Video-Datenströme bezüglich des zu transportierenden Datenvolumens größer. Die Datenrate kann durch den Sender in großen Bereichen variiert werden; typischerweise zwischen 64 kbps und 1 Mbps. Zudem sind verhältnismäßig große Paketverluste akzeptabel. Trotz der damit verbundenen Qualitätsverluste wird die interaktive interpersonelle Kommunikation kaum eingeschränkt.

Ein initialer Satz von Codierungsverfahren wurde im AVP-Profile [Sch96b] festgeschrieben. Verfeinerungen und Erweiterungen enthalten die RFCs 2029, 2032, 2035 und 2038 (vgl. [SH96], [TH96], [BFFM96] und [HFG96]). Die wichtigsten im Einsatz befindlichen Video-Codierungen sind H.261, M-JPEG, Cell-B und NV. Mit Interesse wird auf Implementierungen gewartet, die MPEG-1 und H.263 unterstützen. Die verfügbaren Implementierungen codieren den Video-Datenstrom in Software und/oder nutzen Hardware-Codecs auf den Frame-Grabber-Einheiten. Die Decodierung erfolgt in der Regel durch Software-Codecs. Die Architektur eines solchen Systems ist in [MJ95] dargestellt.

Shared Tools: Wenngleich der Bereich der Shared-Tools ein wesentliches Argument für die Nutzung von Video-Konferenzen ist, bleibt die Vielfalt der im MBone-Kontext verfügbaren Werkzeuge hinter den Erwartungen zurück. Verfügbar sind Whiteboards, verteilte Textverarbeitung und Werkzeuge zur Abstimmung in Konferenzen. Die aus dem H.320-Bereich bekannten Werkzeuge zum Application-Sharing fehlen. Anders als bei den Werkzeugen für kontinuierliche Medienströme entstehen Datenströme, die hinsichtlich Paketverlusten und nicht geordnetem Empfang von Paketen abzusichern sind. Für diese Aufgaben werden zur Zeit sichere Multicast-Transportprotokolle entwickelt. Bestehende Implementierungen nutzen unterschiedliche Transportprotokolle nach dem Konzept des *Application Layer Framing* [CT90]. Eine klassifizierende Übersicht hierzu enthält [Fro96, Seite 36 ff.]. [FJM+95] diskutiert die im MBone-Werkzeug *Witeboard* (*wb*) realisierte Ausprägung eines solchen Protokolls. Die entstehenden Datenströme haben Burst-Charakter, der durch die Protokolle gemindert wird. Während die Paketlaufzeiten in den Hintergrund treten, ist besonders auf die Vermeidung von Paketverlusten zu achten. Zur Zeit ist kein einheitliches Protokoll oberhalb der UDP-Schicht zu erkennen, so daß weitere Betrachtungen auf der UDP-Schicht erfolgen.

Conference Control: Die in klassischen Video-Konferenzen wichtige Conference-Control spielt bei MBone-Videokonferenzen eine untergeordnete Rolle. Ursächlich hierfür sind die unterschiedlichen Paradigmen sowie die unterschiedliche Architektur gegenüber Konferenzen nach den ITU-Standards H.323 und H.320. MBone-Videokonferenzen sind lose gekoppelte Konferenzen. Die Konferenz-Kontrolle erfolgt per Medienstrom und der Beitritt und das Verlassen laufender Konferenzen geht zwanglos vonstatten. Die Privatsphäre wird durch Verschlüsselung der Medien-Datenströme erzielt. Die Publikation laufender Konferenzen erfolgt über das im folgenden Absatz dargestellte Session-Directory. Dennoch entstehen im Moment Werkzeuge für den Einsatz im MBone, die der Klasse der Conference-Control-Tools zuzurechnen sind. Dies sind sogenannte Floor-Controller für die Verwaltung von Wortmeldungen, sowie komfortablere Werkzeuge zur Abhaltung spontaner Konferenzen in geschlossenen Gruppen wie Confman (vgl. [BFGP96],[Fri96]) und H.323-basierte Konferenz-Systeme. Letztere sind allerdings nur bedingt in den Internet-Conferencing-Protocol-Stack zu integrieren. Die wesentliche Gemeinsamkeit ist, daß für den Transport der Medien-Datenströme RTP benutzt wird.

Session Directory: Das Session-Directory entstand aus der Notwendigkeit, auch ungeübten Benutzern den Zugriff auf MBone-Sitzungen zu erlauben. Das Directory soll dem Nutzer zeigen, welche Sitzungen aktuell angekündigt sind und in welchem Status sie sich auf der Basis der vom Veranstalter vorgegebenen zeitlichen Rahmenbedingungen befinden. Damit ähnelt das Session-Directory einem Fernsehprogramm. Aus technischer Sicht ebenso wichtig ist seine Funktion zur Ressourcen-Planung. Die in einer Sitzung zum Einsatz kommenden Werkzeuge und die zu verwendenden Adressen werden zusammen mit anderen Metadaten der Konferenz in einer SDP-PDU[35] aggregiert. Diese PDU kann über verschiedene Transportsysteme verbreitet werden. Der gebräuchlichste Weg ist die Distribution über den MBone. Dazu werden die SDP-PDUs in SAP-PDUs[36] eingebettet und in von der aktuellen Verkehrslast sowie der Reichweite einer Sitzung abhängigen Frequenz regelmäßig angekündigt. Potentielle Veranstalter von Konferenzen können anhand dieser Daten erkennen, welche Adressen in ihrem Einzugsbereich zu welchem Zeitpunkt belegt sind und auf Basis der zu erwartenden Verkehrslast entscheiden, ob die Übertragung einer weiteren konkurrierenden Sitzung zu einem bestimmten Zeitpunkt sinnvoll erscheint. Die Anzeige sowie die Ankündigung von Sitzungen wird von Session-Directory-Programmen übernommen. Neben dem seit langem etablierten Programm *sdr* (vgl. [Han96b] und [Han97]) sind in der letzten Zeit weitere Implementierungen entstanden, auch kommerzielle Versionen sind erhältlich.[37]

Das Session-Directory verursacht einen dauerhaften Datenstrom im MBone. Das SAP-Protokoll schreibt eine starke Datenratenbegrenzung sowie Methoden zum Erzielen eines gleichförmigen Datenstroms vor. Dennoch ist dieser Datenstrom in weitere Betrachtungen einzubeziehen.

[35] SDP steht für Session-Description-Protocol, siehe [HJ97].
[36] SAP steht für Session-Announcement-Protocol, siehe [Han96a].
[37] Für eine aktuelle Übersicht siehe hierzu `http://nic.merit.edu/~ mbone/index/titles.html`.

2.4.2 Das Real-Time Transport Protocol

Der in Hinblick auf das Datenvolumen wesentliche Anteil der Datenströme in MBone-Videokonferenzen entsteht aus der Übertragung der Audio- und Video-Datenströme. Das dazu genutzte *Real-Time Transport Protocol* (RTP) [SCFJ96] wird daher kurz vorgestellt.

Das Real-Time-Transport-Protocol (RTP) hat zwei wesentliche Aufgaben:

- Es dient dem Transport von Daten, die einen synchronen Übertragungsmodus erfordern. Die wesentliche Aufgabe besteht darin, für solche Übertragungen generell erforderliche Parameter medienunabhängig zu codieren. Dies sind neben der Markierung eines Payload-Typs Quellenbezeichner, Sequenz-Nummern und Zeitstempel.
- Ein begleitendes Kontroll-Protokoll, RTCP, hat die Aufgabe, Sendern eine Rückkopplung über die Qualität der Datenverteilung zu geben, Quellenbezeichner mit Namen zu verbinden und minimale Informationen zur Sitzungskontrolle bereitzustellen.

Konventionelle, Unicast-basierte Transportprotokolle wie TCP basieren auf dem Modell, daß vom Sender ausgelieferte Pakete vom Empfänger bestätigt werden. Die damit verbundene Flußsteuerung ist letztlich transaktionsorientiert. Nutzdaten- und Kontrolldatenfluß werden gemischt über ein Paar Transportadressen[38] abgewickelt.

Im Fall der Multicast-Kommunikation sendet ein Sender an eine Gruppe von Empfängern. Eine eng gekoppelte Flußsteuerung wie bei Unicast-Protokollen ist nicht möglich, da positive oder negative Empfangsbestätigungen bei größeren Empfängergruppen das Netzwerk fluten und damit den Nutzdatenverkehr behindern würden. RTP benutzt daher für die Übertragung des Nutzdatenstroms eine Transportadresse, an die ohne direkte Rückkopplung durch die Empfänger Daten übertragen werden. Eine weitere Transportadresse wird für das Kontrollprotokoll, RTCP, benutzt.

Ein weiteres wichtiges Merkmal von RTP ist, daß die Medienströme wie Audio- und Video nicht in einem Datenfluß gemischt werden und keine implizite Synchronisierung zwischen den Datenströmen erfolgt. Daher können die Medienströme auf dem Übertragungsweg unabhängig voneinander behandelt werden. Nachteilig ist, daß keine Lippensynchronisierung gewährleistet wird.

Für das anfallende Datenvolumen ausschlaggebend ist der Medienstrom, d.h. der RTP-Datenstrom. Durch die vereinheitlichte Übertragung entsteht je Nutzdateneinheit ein Overhead von zumindest 12 Byte. Ein 20 ms dauerndes Fragment eines PCM-Datenstroms mit 64 kbps entspricht einer zu transportierenden Payload von 160 Byte. Wird dieses Fragment über RTP transportiert, entsteht ein Protokoll-Overhead von 12 Byte: 6% des resultierenden Datenpakets ist RTP-Protokoll-Overhead. Wenngleich eine Partitionierung des Audio-Stroms in 20 ms Fragmente günstig für geringe Verzögerungen ist und daher in [Sch96b] empfohlen wird, ist der Protokoll-Overhead recht groß. Üblich sind daher auch Partitionierungen des Audio-Datenstroms in 40 ms und 80 ms Fragmente, wobei letztere insbesondere für Vorträge zu empfehlen sind. Bei Vorträgen steht weniger die schnelle, sondern die möglichst wenig Übertra-

[38] Eine Transportadresse ist üblicherweise ein Tupel aus Internet-Adresse und Port, das einen Transportendpunkt eindeutig identifiziert.

gungskapazität erfordernde Übermittlung der Datenpakete im Vordergrund. Durch längere Fragmente reduziert sich der Protokoll-Overhead und damit die beanspruchte Übertragungskapazität.

Die Übertragung von Video-Datenströmen bringt das Problem mit sich, daß einzelne Video-Frames größer als ein Paket sein können. Daher ist in der Regel die Verteilung des Video-Frames auf mehrere Pakete erforderlich. Gemäß [TH96] wird bei der häufig eingesetzten H.261-Codierung wie folgt vorgegangen:

- Die Übertragung orientiert sich an den 16x16 Pixel großen Makro-Blöcken, und nicht an den aus diesen zusammengesetzten Frames.
- Jede RTP-PDU enthält einen oder mehrere Makro-Blöcke, damit die PDUs nicht zu klein werden. Zusätzlich werden in jeder RTP-PDU zur Decodierung erforderliche Informationen der jeweiligen Group-of-Blocks übertragen.
- Alle RTP-PDUs eines Frames erhalten den gleichen Zeitstempel und individuelle Sequenznummern. Das Ende des Video-Frames wird durch ein gesetztes Marker-Bit im RTP-Header signalisiert.

In der Praxis verteilt sich ein Frame auf drei und mehr RTP-PDUs. Die PDU-Größe auf UDP-Ebene liegt in der Regel zwischen 900 und 1000 Byte[39]. Hervorzuheben ist, daß auch bei Verlust einzelner PDUs des Frames die empfangenen Daten zur Aktualisierung des Video-Bildes genutzt werden können.

Wie bereits dargestellt, wird zum Austausch von Nachrichten des Real-Time-Control-Protocols (RTCP) ein zweite Transportadresse benutzt. RTCP kennt fünf Nachrichtentypen:

Sender-Report: Diese PDUs enthalten Sende- und Empfangsstatistiken aktiver Sender sowie einen *Synchronisation Source Identifier* (SSRC). Letzterer ist eine 32 Bit Zahl, die nach einem in [SCFJ96, Anhang 6] beschriebenen Verfahren berechnet wird. Jeder Teilnehmer einer Konferenz verfügt über einen eindeutigen SSRC, der für alle Medienströme gleich bleibt. Der SSRC eines Teilnehmers kann sich im Zeitverlauf ändern, z.B. als Reaktion auf eine Kollision der gewählten SSRC mit der eines anderen Konferenzteilnehmers.

Receiver-Report: PDUs diesen Typs werden von passiven Mitgliedern einer Sitzung versendet und enthalten Statistiken über die empfangenen Daten. Zudem enthalten auch diese Nachrichten einen SSRC.

Source-Description: Nachrichten dieses Typs haben drei wesentliche Aufgaben:

- Den im Zeitverlauf möglicherweise veränderlichen SSRC des Senders mit einer unveränderlichen Beschreibung zu verbinden. Diese wird als *Canonical-Name* (CNAME) bezeichnet. Aktuelle Implementierungen der MBone-Medienwerkzeuge generieren den CNAME aus der Nutzerkennung des Teilnehmers sowie der IP-Adresse des Teilnehmer-Rechners, getrennt durch ein `@`-Zeichen (z.B. `gruen@130.75.5.239`).

[39] In Abschnitt 2.4.3 wird genauer auf die statistischen Analysen einzelner Sitzungen eingegangen.

- Den SSRC und den CNAME des Teilnehmers mit dem realen Namen des Nutzers sowie den wichtigsten Kontaktadressen des Teilnehmers zu verbinden (Mail-Adresse und Telefonnummer). Zudem ist die Übertragung von Informationen zur geographischen Position des Teilnehmers möglich.
- Informationen über das eingesetzte Medienwerkzeug bereitszustellen. Ein entsprechender Datensatz enthält zumindest den Namen des Werkzeugs sowie seine Versionsnummer. Diese Daten können für die Fehlersuche von Wert sein.

Darüber hinaus können weitere, spezialisierte Informationen als Source-Description übertragen werden (vgl. [SCFJ96, Abschnitt 6.4]). In einer Source-Description-Nachricht ist die gemeinsame Übertragung mehrere Komponenten möglich. Anzumerken bleibt, daß ausschließlich die Übertragung des CNAME gefordert wird. Alle anderen Komponenten sind optional.

BYE: Diese PDU wird von Gruppenmitgliedern versendet, bevor sie die Sitzung verlassen. Ihr wesentliches Element ist der SSRC des Teilnehmers.

Application-Specific: PDUs diesen Typs bieten die Möglichkeit zur Übermittlung anwendungsspezifischer Daten.

Jeweils mehrere dieser Nachrichten können in einem RTCP-Paket übermittelt werden. Generell werden die RTCP-Pakete in variablen Intervallen wiederholend übermittelt. Um den RTCP-Datenfluß in einer Sitzung in Grenzen zu halten, werden die Intervalle in Abhängigkeit von der Gruppengröße variiert, so daß das anfallende Datenvolumen auf 5% des Nutzdatenstroms begrenzt bleibt. Details hierzu sind [SCFJ96], Abschnitt 6.2 zu entnehmen.

Die Rückkopplung über die Qualität der Datenverteilung von den Empfängern an den Sender mittels Sender- und Receiver-Reports läßt grundsätzlich eine dynamische Anpassung der Datenrate für den Nutzdatenstrom zu, dennoch wird ein solches Verhalten von aktuellen Implementierungen nicht unterstützt.

2.4.3 Ermittlung von Datenströmen in MBone-Konferenzen

Für die weiteren Betrachtungen sind die bei MBone-Konferenzen anfallenden Datenvolumina von großem Interesse. Das besondere Augenmerk richtet sich auf die Audio- und Video-Datenströme sowie die Sitzungsankündigung mittels SAP/SDP. Während sich die durch Audio-Datenströme entstehenden Volumina gut berechnen lassen, lassen sich die Daten für den Kontrolldatenfluß und Video-Übertragungen am besten experimentell ermitteln. Wenngleich der RTP-Standard eine Datenratenbegrenzung für den Kontrollfluß vorschreibt, muß überprüft werden, ob diese eingehalten wird. Für die durch Video-Übertragungen anfallenden Datenströme lassen sich durch die Vielzahl der die Datengenerierung beeinflussenden Parameter ohnehin nur Erfahrungswerte sammeln.

Für die Übertragung von Audio-Datenströmen werden heute im MBone zwei Werkzeuge besonders häufig eingesetzt. Dies sind das am Lawrence Berkeley National Laboratory entwickelte *Visual Audio Tool* (*vat*) [JM] sowie das am University College London entwickelte *Robust Audio*

Kodierung	Sample-Time	UDP-Payload	Paketrate	UDP-Datenrate
8-bit mu-law encoded 8kHz PCM	20 ms	172 Byte	50 pps	68.8 kbps
8-bit mu-law encoded 8kHz PCM	40 ms	332 Byte	25 pps	66.4 kbps
8-bit mu-law encoded 8kHz PCM	80 ms	652 Byte	12.5 pps	65.2 kbps
Intel DVI ADPCM	20 ms	96 Byte	50 pps	38.4 kbps
Intel DVI ADPCM	40 ms	176 Byte	25 pps	35.2 kbps
Intel DVI ADPCM	80 ms	336 Byte	12.5 pps	33.6 kbps
GSM	80 ms	144 Byte	12.5 pps	14.4 kbps
Linear Predictive Coder	80 ms	68 Byte	12.5 pps	6.8 kbps

Tabelle 2.7: Merkmale von RTP-Audio-Datenströmen bei unterschiedlichen Codecs

Tool (*rat*) [SHK+]. Während *vat* die Audio-Codierungen PCM nach G.711, Intel's DVI ADPCM, GSM und LPC anbietet, ermöglicht *rat* zusätzlich noch die Nutzung einer linearen 16 Bit Codierung. Die Sampling-Frequenz beträgt für alle Codierungen 8 kHz. Als weitere Besonderheit bietet *rat* die redundante Übertragung von Audio-Daten. Hierbei wird die primäre, qualitativ hochwertigere Codierung um eine zweite ergänzt. Jedes RTP-Paket enthält zwei Payload-Teile, wobei die Sampling-Time des Inhalts der sekundären Kodierung sich von der der primären unterscheidet. Der Empfänger kann somit auf Strecken mit Paketverlusten trotzdem ein nutzbares Audio-Signal erzeugen. Das Verfahren befindet sich noch in Entwicklung und wird daher im folgenden nur am Rande betrachtet. Das Übertragungsformat ist in [Per97] dokumentiert.

Die durch die Nutzung dieser Werkzeuge auf auf der UDP-Schicht entstehenden Datenströme sind in Tabelle 2.7 zusammengetragen. Die ermittelten Werte wurden durch Messungen bestätigt. Alle Angaben unterstellen die Nutzung des in [SCFJ96] spezifizierten RTPv2 als Transportprotokoll.

Zur Ermittlung der durch das RTCP-Protokoll verursachten Datenströme sowie zur Beurteilung der Video-Datenströme wurden MBone-Konferenzen mitgeschnitten und nachträglich analysiert. Der Mitschnitt erfolgte, indem auf einem leistungsfähigen Institutsrechner, einer Sun Ultra-Enterprise 2 mit zwei Prozessoren, die Datenströme der Sitzungen empfangen wurden, ohne selbst als aktiver oder passiver Teilnehmer aufzutauchen.[40] Parallel dazu wurde mittels *tcpdump* der empfangene Datenverkehr protokolliert.

Konkret wurden die Datenströme folgender Sitzungen analysiert:

FAU-TV

FAU-TV ist ein Testkanal, auf dem, soweit möglich, kontinuierlich ein Fernsehprogramm von einem Satellitenreceiver eingespeist wird. Die Einspeisung erfolgt an der Universität Erlangen. FAU-TV wurde am 01. Juli 1997 von 19:42 bis 20:03 MET DST protokolliert. Die Meßdauer betrug 1248 Sekunden. Die Messung hatte das in Tabelle 2.8 dargestellte Ergebnis.

Während der protokollierten 20 Minuten wurden in dieser Sitzung 85 MB Daten übertragen. Dies entspricht einer Datenrate von 572 kbps. Dieser für MBone-Sitzungen große Wert wurde vor allem durch den Video-Datenstrom verursacht. Die hier ermittelten 503 kbps liegen deutlich über den sonst im MBone üblichen Werten. Der Audio-

[40] Das hierfür eingesetzte Programm gehört zum *rtest*-Werkzeug. Eine weitergehende Beschreibung befindet sich im Anhang.

	Empfangene Pakete	Empfangene Bytes	Paketrate (pps)	Datenrate (kbps)	Durchschn. Paketgröße (Bytes)	Anzahl Sender (transport)	Anzahl Sender (host)
Sitzung (Audio, Video, Whiteboard)	120085	89356444	96.22	572.80	744.11	23	11
Transmitter Audio/RTP	31171	10348772	24.98	66.34	332.00	1	1
Transmitter Audio/RTCP	258	29276	0.21	0.19	113.47	1	1
Alle Audio/RTCP	2060	216012	1.65	1.38	104.86	9	9
Transmitter Video/RTP	84586	78581503	67.78	503.73	929.01	1	1
Transmitter Video/RTCP	252	28604	0.20	0.18	113.51	1	1
Alle Video/RTCP	2016	188072	1.62	1.21	93.29	10	10
Alle Network-Editor	252	22085	0.20	0.14	87.64	2	2

Tabelle 2.8: Datenströme der MBone-Session FAU-TV

Datenstrom mit 66 kbps und einer durchschnittlichen Paketgröße von 332 Byte läßt darauf schließen, daß das Audio-Signal als 8-Bit mu-law encoded 8 kHz PCM-Signal mit einer Sample-Time von 40 ms pro Paket übertragen wurde. Die Paketrate von 25 pps bestätigt dies.

Der RTP-Standard legt fest, daß der zum RTP-Datenstrom gehörige RTCP-Datenstrom maximal 5% der Datenrate des RTP-Datenstroms betragen darf [SCFJ96, Abschnitt 6.2]. Diese Marke wird von den Video- und Audio-Datenströmen deutlich unterschritten.

Die Analyse der Sender-Zahlen ergibt, daß Pakete von 23 unterschiedlichen Sende-Transportadressen empfangen wurden. Dabei konnten 11 Sender-IP-Adressen unterschieden werden. Der Audio-Datenstrom wurde vom Transmitter ausgesendet und von 8 Sitzungsteilnehmern empfangen. Der vom Sender übermittelte Video-Datenstrom wurde von 9 Mitgliedern der Sitzung empfangen. Diese Zahlen ergeben sich aus der Betrachtung der Sender-Zahlen nach IP-Adresse für alle RTCP-Sender.

Interessant ist weiter die Betrachtung der durchschnittlichen Paketgrößen pro Multicast-Transportadresse. Es lassen sich drei Gruppen identifizieren. Eine Gruppe bildet der Video-Datenstrom. Hier liegt die durchschnittliche Paketgröße bei 929 Byte UDP-Payload. Die nächste Gruppe bildet der Audio-Datenstrom mit 332 Byte UDP-Payload im Mittel. Die durchschnittlichen Paketgrößen aller weiteren Multicast-Transportadressen, d.h. denen der RTCP-Datenströme und des Network-Editors, liegen in der Größenordnung von 100 Byte UDP-Payload pro Paket.

Hong Kong 1997 Handover

Die Übergabe der britischen Kronkolonie Hong-Kong an die Volksrepublik China wurde in Kooperation von Radio-Television-Hong-Kong und der Universität Hong-Kong über den MBone übertragen. Im Rahmen der Messung wurde ein Teil der Übertragung am 01. Juli 1997 in der Zeit von 19:10 bis 19:33 MET DST protokolliert. Die Messung dauerte 1337 Sekunden und lieferte die in Tabelle 2.9 dargestellten Ergebnisse.

Die Gesamtdatenrate bei dieser Sitzung ist mit etwa 180 kbps deutlich geringer als beim vorangegangenen Beispiel. Dies begründet sich durch den geringeren Video-Datenstrom von 126 kbps und den geringeren Audio-Datenstrom von 33 kbps. Der Audio-Datenstrom wurde unter Nutzung der Intel-DVI-ADPCM-Codierung übertragen. Die durchschnittliche Paketgröße von 96 Byte, die Paketrate von 43 pps sowie die Datenrate weisen darauf

	Empfangene Pakete	Empfangene Bytes	Paketrate (pps)	Datenrate (kbps)	Durchschn. Paketgröße (Bytes)	Anzahl Sender (transport)	Anzahl Sender (host)
Sitzung (Audio, Video, Whiteboard)	103503	30008139	77.41	179.56	289.93	121	57
Transmitter Audio/RTP	57604	5529984	43.08	33.09	96.00	1	1
Transmitter Audio/RTCP	261	28996	0.20	0.17	111.10	1	1
Alle Audio/RTCP	8485	730889	6.35	4.37	86.14	46	43
Transmitter Video/RTP	23533	21102222	17.60	126.27	896.71	1	1
Transmitter Video/RTCP	269	30116	0.20	0.18	111.96	1	1
Alle Video/RTCP	7626	641520	5.70	3.84	84.12	59	51
Alle Whiteboard	6092	1897248	4.56	11.35	311.43	14	14

Tabelle 2.9: Datenströme der MBone-Session Hong Kong 1997 Handover

	Empfangene Pakete	Empfangene Bytes	Paketrate (pps)	Datenrate (kbps)	Durchschn. Paketgröße (Bytes)	Anzahl Sender (transport)	Anzahl Sender (host)
Sitzung (Audio, Video)	98505	31513517	54.82	140.29	319.92	386	210
Transmitter Audio/RTP	42244	7265968	23.51	32.35	172.00	1	1
Transmitter Audio/RTCP	214	21628	0.12	0.10	101.07	1	1
Alle Audio/RTCP	21264	1818531	11.83	8.10	85.52	179	165
Transmitter Video/RTP	24772	21543058	13.79	95.91	869.65	1	1
Transmitter Video/RTCP	270	29416	0.15	0.13	108.95	1	1
Alle Video/RTCP	10225	885960	5.69	3.94	86.65	205	192

Tabelle 2.10: Datenströme der MBone-Session STS-94-Mission

hin. Die Differenzen zu der in Tabelle 2.7 genannten Werte ergeben sich durch Paketverluste.

Aus der Anzahl der RTCP-Sender ergibt sich, daß die Sitzung von bis zu 50 Nutzern je Datenstrom verfolgt wurde. In dieser Sitzung wurde die Vorgabe, daß der RTCP-Datenstrom maximal 5% des RTP-Datenstromes betragen soll, beim Audio-Datenstrom verletzt.

Durch die geringere Paketgröße der Audio-Pakete bei erhöhter Frequenz ist die durchschnittliche Paketgröße für die Sitzung deutlich kleiner als beim vorigen Beispiel. Dabei ist zu beachten, daß auch hier der Video-Datenstrom eine durchschnittliche Paketgröße von fast 900-Byte und somit die gleiche Größenordnung wie beim FAU-TV hat, allerdings ist die Paketrate deutlich kleiner.

Anzumerken bleibt, daß bei dem in dieser Sitzung eingesetzten Whiteboard WB durchschnittlich deutlich größere Datenpakete als beim Network-Editor NT übertragen werden.

NASA Space-Shuttle Mission STS-94

Das NASA-Ames Research Center übertrug den Flug des Space-Shuttle in der Mission STS-94, Microgravity Science Laboratory-1, im MBone. Im Rahmen der Messung wurde die Übertragung am 08.07.97 in der Zeit von 17:35 bis 18:05 MET DST protokolliert. Die Messung dauerte 1797 Sekunden und ergab die in Tabelle 2.10 dargestellten Ergebnisse.

Der Gesamtdatenstrom bei dieser Sitzung ist mit ca. 140 kbps noch einmal geringfügig kleiner als bei der Übertragung des Hong-Kong-Handover. Ursächlich ist der kleinere

	Empfangene Pakete	Empfangene Bytes	Paketrate (pps)	Datenrate (kbps)	Durchschn. Paketgröße (Bytes)	Anzahl Sender (transport)	Anzahl Sender (host)
Alle Sender	7149	4830527	0.51	2.74	675.69	5835	29
RVS-Proxy/Cache	5715	4012547	0.41	2.28	702.11	5715	1
Andere Sender	1434	817980	0.10	0.46	570.42	120	28

Tabelle 2.11: Datenströme des SAP/SDP-Protokolls

Video-Datenstrom. Zudem fehlt das Shared-Tool.

Hervorzuheben ist die große Teilnehmerzahl an dieser Sitzung. Die Zahl der Sender-Hosts für Video/RTCP weist darauf hin. Auch hier wird die Regel, daß nur 5% der RTP-Datenrate für RTCP aufgewendet werden sollten, vom Audio-Datenstrom verletzt. Offensichtlich haben die eingesetzten Audio-Tools Skalierungsprobleme. Der RTCP-Datenratenbedarf steigt in den drei Sitzungen mit der Teilnehmerzahl, wohingegen der RTCP-Datenstrom beim Video klein bleibt.

Eine weitere Besonderheit dieser Übertragung läßt sich nicht aus den Meßergebnissen erkennen, soll aber nicht verschwiegen werden. Die Multicast-Transportadressen des Audio- und Video-Datenstroms unterscheiden sich bei dieser Übertragung nur im Port-Teil. Es wurde die gleiche IP-Multicast-Adresse benutzt. Dadurch ist die Unabhängigkeit der Datenströme bei der Vermittlung nicht gegeben. Grund hierfür ist, daß das Multicast-Routing auf der Basis der Multicast-Adressen und nicht der Transportadressen darüber entscheidet, ob ein Datenstrom zu vermitteln ist. Abonniert ein Nutzer den Audio-Datenstrom, so gelangt auch der Video-Datenstrom in sein Subnetz und umgekehrt. Diese Problematik wird in Abschnitt 2.5.2, Seite 49 detaillierter behandelt.

Ergänzend zur Ermittlung der Datenströme in Sitzungen auf dem MBone wurde auch der durch das Session-Directory-Protocol auf dem MBone entstehende Datenstrom ermittelt. Dazu wurde der Verkehr auf der Multicast-Transportadresse für das SAP/SDP-Protokoll (SAP.MCAST.NET/9875) vom 11.07.1997 um 21:31 bis zum 12.07.97 um 01:26 protokolliert. Die Meßdauer betrug 14079 Sekunden. Es ergaben sich die Tabelle 2.11 dargestellten Ergebnisse.

Die erste Zeile der Tabelle gibt die Gesamtheit des empfangenen Datenstroms an. Er zerfällt in zwei Komponenten:

- Der größte Teil des Datenstroms wird von einem am RVS betriebenen SAP/SDP-Proxy/Cache verursacht.
- Die kleinere Komponente umfaßt originäre Session-Announcements, die aus dem internationalen MBone empfangen wurden.

Der am RVS betriebene SAP/SDP-Cache hat die Aufgabe, MBone-Nutzer nach dem Start eines beliebigen MBone-Session-Directory schneller mit dem aktuellen Inhalt des Directories zu versorgen. Der SAP/SDP-Proxy übernimmt für lokale Nutzer automatisch die Ankündigung von Sitzungen für den Fall, daß der ankündigende Nutzer sein Werkzeug zum Session-Announcement verläßt, z.B. nachts. Dieses Werkzeug ist im Rahmen einer Diplomarbeit am RVS entstanden und in [Vöc97] dokumentiert.

Der durch das System verursachte Verkehrsfluß entsteht im wesentlichen durch seine Cache-Funktion. Das System basiert auf der Idee, empfangene Sitzungsankündigungen aus dem Internet mit erhöhter Frequenz im universitären Multicast-Netz auszusenden. Die Session-Directories der Nutzer an der Universität aktualisieren sich dadurch innerhalb weniger Minuten. Diesem Vorteil steht eine erhöhte Belastung des Multicast-Kanals für das Session-Announcement im lokalen Bereich gegenüber.

Die aus dem weltweiten MBone empfangenen, originalen Sitzungsankündigungen belasten den Multicast-Kanal mit ca. 500 bps.

2.5 Bestehende Ansätze zum Anschluß von Heimarbeitsplätzen an den MBone

Der Anschluß von Einzelplatzsystemen und privaten Netzwerken über Modemverbindungen und ISDN-Verbindungen an den MBone ist ein Bereich, mit dem sich bereits einige Arbeiten beschäftigt haben. Die bisher entwickelten und in Entwicklung befindlichen Systeme lassen sich in drei Gruppen gliedern:

- Application-Layer-Gateways, die oberhalb des IP-Transportprotokolls (UDP oder RTP) operieren.
- Ansätze, welche die etablierten Methoden der Multicast-Datenvermittlung auf der Schicht des Internet-Protokolls nutzen, d.h. auf Schicht 3 im ISO-OSI-Referenz-Modell [Tan90, Seite 17ff.] arbeiten.
- Konzepte, die sich mit dem Transport der Daten auf dem Link-Level, d.h. der Schicht 2 im ISO-OSI-Referenz-Modell beschäftigen.

Für den konkreten Betrieb eines Systems in der hier betrachteten Anwendung ist zu beachten, daß in der Regel die Kopplung mehrerer Konzepte erforderlich ist. Im folgenden werden einige dieser Ansätze im Detail vorgestellt, um die konzeptionellen Wirkungsweisen zu verdeutlichen. Zudem werden implementierte Systeme referenziert. Ziel ist dabei, einen möglichst umfassenden Überblick zu geben. Da dieser Bereich jedoch im Blickpunkt der aktuellen Forschung steht, kann die Vollständigkeit nicht gewährleistet werden.

2.5.1 Gateways auf RTP- und UDP-Ebene

Bereits in der Spezifikation des RTP-Protokolls werden zwei Ausprägungen von RTP-Gateways eingeführt: *Translatoren* und *Mixer*. Translatoren unterscheiden sich von Mixern dadurch, daß sie die *Synchronisation Source* (SSRC) nicht verändern. Die Daten einer Quelle werden als ein Datenstrom weitergegeben. Das Vorhandensein eines Translators in einem Netzwerkpfad ist für den Empfänger nicht erkennbar. Mixer überlagern die Datenströme mehrerer Sender zu einem Ergebnisdatenstrom. Translatoren und Mixer untergliedern sich in Untergruppen, die im folgenden vorgestellt werden.

2.5.1.1 Translatoren

In der einfachsten Form wird der RTP-Datenstrom durch einen Translator unverändert weitergeleitet und lediglich eine Adreßumsetzung auf UDP-Ebene durchgeführt. Damit ist die Ankopplung von Partnern an Multicast-Konferenzen möglich, die sich mit ihrem System nicht in einem Multicast-fähigen Netz befinden. Ebenfalls lassen sich auf diesem Weg Multicast-Inseln über eine Unicast-Infrastruktur miteinander koppeln, wenngleich hierfür die Kopplung der Netze über Multicast-Router mit IP-Tunneln in der Regel vorzuziehen ist. Eine letzte Anwendung ist der Aufbau von Konferenz-Szenarien, in denen zentrale Vermittler die RTP-Unicast-Datenströme an alle Partner einer Konferenz verteilen. Systeme dieser Art lassen sich mit verhältnismäßig kleinem Aufwand realisieren, da lediglich Abbildungen auf UDP-Ebene erforderlich sind; die RTP-Daten können unverändert bleiben. Gleiches gilt für den RTCP-Kontrolldatenfluß, wobei die Zusammenfassung von RTCP-PDUs zulässig ist.

Wichtig ist generell, daß auch einfache Systeme Schleifen erkennen sollten, die in aller Regel zu einer Flutung des Netzwerkes führen. Wenn Schleifen erkannt werden, muß die Vermittlung der entsprechenden Datenströme unterbunden werden. Dazu ist eine minimale Interpretation des RTP-Datenstromes erforderlich. Die RTP-Spezifikation enthält Hinweise zur Implementierung der Schleifenerkennung. Im folgenden werden Systeme, die lediglich Adreßumsetzungen auf UDP-Ebene realisieren als *Transmitter* bezeichnet.

Eine andere Klasse von Translatoren modifiziert die Codierung von RTP-Datenströmen. Neben der Änderung des Codierungsformats, beispielsweise der Umsetzung eines MJPEG-Video-Datenstroms auf einen H.261-Video-Datenstrom, ist auch die Reduktion der Datenrate oder die Auflösungsänderung eines Video-Datenstroms möglich. In der Regel umfassen Systeme dieser Klasse auch die Merkmale von Transmittern. Systeme mit diesen Merkmalen werden häufig aus als *Transcoder* bezeichnet.

2.5.1.2 Mixer

Mixer fassen die Medienströme einer Medienklasse von mehreren Sendern zu *einem* neuen Medienstrom zusammen. Anwendungen sind das Zusammenfassen mehrerer Audio-Datenströme, wobei sich die Signale der Sender überlagern. Vorteilhaft ist dies bei der Ankopplung von Teilnehmern über "Low-Speed-Networks" wie Modemstrecken, da die Signale beim Ausspielen durch den Empfänger in der Regel ohnehin überlagert werden.[41] Eine andere Anwendung ist das Zusammenfassen mehrerer Video-Datenströme, wobei der resultierende Datenstrom aus skalierten und überlagerten Bildern der originären Datenströme zusammengesetzt ist. Der Vorteil beider Verfahren ist, daß das zu übertragende Datenvolumen signifikant verringert wird.

Neben den Veränderungen auf der Basis der Codierung müssen Mixer dafür sorgen, daß den Empfängern des Datenstroms Informationen über die originären Sender zukommen. Dazu muß neben dem RTP-Datenstrom auch der RTCP-Datenstrom verändert werden. Im ausgehenden RTP-Datenstrom erscheint der Mixer als Synchronisation-Source. Die ursprünglichen Sender werden in einer bis zu 15 Elementen umfassenden Liste als Contributing-Sources aufgeführt.

[41] Eine Ausnahme entsteht beim Einsatz bestimmter "Floor-Controller" zur Gesprächssteuerung, die beim Empfänger das Ausspielen der Daten einzelner Sender auf Basis der Synchronisation-Source unterdrücken.

Sind mehr als 15 Sender vorhanden, wird die Anzahl zwar korrekt übermittelt, die Quellen werden jedoch nicht mehr dargestellt, womit das Problem einer zuverlässigen Schleifenerkennung entsteht.

Bezüglich des RTCP-Datenstroms wird folgendermaßen vorgegangen: Sender- und Receiver-Reports werden selbst generiert, die Reports der originären Sender werden verworfen. Die Quellenbeschreibungen der anderen Gruppenmitglieder werden unverändert weitergegeben. Entsprechendes gilt für BYE-Nachrichten, wobei der Mixer zudem selbst BYE-Nachrichten erzeugt.

Mixer sind aufwendiger in der Implementierung als Transcoder. Zudem sind sie für die Empfänger nicht transparent, so daß bei kaskadierter Anwendung von Mixern zusätzlicher Aufwand entsteht. Für die Ankopplung von Heimarbeitsplätzen über "Low-Speed-Networks" bieten sie aufgrund der nachhaltigen Reduktion des zu übertragenden Datenvolumens jedoch deutliche Vorteile gegenüber Translatoren.

2.5.1.3 Verfügbare Translatoren und Mixer

Zwischenzeitlich sind einige Systeme in diesem Umfeld entwickelt worden, die im folgenden kurz vorgestellt werden.

Mash

Bereits 1992 entwickelte Jon Crowcroft das Programm *mash* [HC92]. Dabei handelt es sich um einen einfachen UDP-Transmitter, der auf der Basis der Transportadressen die Verbindung von Multicast-Inseln erlaubt. *mash* implementiert keine Schleifenerkennung. Die Nutzung von *mash* ist mühsam und fehlerträchtig, da auf beiden zu verbindenden Rechnern manuell für jeden Transportkanal entsprechend konfigurierte *mash*-Instanzen zu starten sind. Es handelt es sich um ein einfaches Programm, das für den hier betrachteten Einsatz kaum nutzbar ist.

Rtptrans

Einen ähnlichen Ansatz verfolgt das im Paket *rtptools* [SSC] enthaltene Programm *rtptrans*. Auch hierbei handelt es sich um einen einfachen UDP-Transmitter, der ebenfalls keine Schleifenerkennung bietet und daher für den Produktionseinsatz ungeeignet ist. Erweiterungen gegenüber *mash* sind die einfachere Aktivierung, da eine Programminstanz den RTP- und RTCP-Datenstrom vermittelt, die Fähigkeit, RTPv0-PDUs in RTPv2-PDUs umzusetzen und die Vermittlung der Datenströme zu mehr als einem Partner.

Cms.connectd

Das am Lehrgebiet Rechnernetze und Verteilte Systeme entstandene Programm *cms.connectd* [Ber94] wurde zur zentralen Vermittlung RTP-basierter Datenströme im Kontext von ATM-Netzen entwickelt. Hintergrund für die Entwicklung des Systems war, daß ATM-Netze zum Zeitpunkt der Entwicklung des Systems keinerlei Unterstützung für Multicast boten. Die dem System zugrunde liegende Idee ist, daß die Medien-Werkzeuge aller Konferenzteilnehmer über Unicast-Datenströme mit dem zentralen Vermittler verbunden werden und dieser die weitere Verteilung an die Konferenzpartner übernimmt.

cms.connectd ist ebenfalls ein einfacher UDP-Transmitter ohne Schleifenerkennung. Eine wichtige Erweiterung gegenüber *mash* und *rtptrans* ist, daß der Vermittler über ein Konfigurationsprotokoll gesteuert wird. Die Einbettung des Konfigurationsprotokolls in das

Konferenz-Management-System *Confman* (vgl. [Fri96], [BFGP96]) ermöglicht die einfache Nutzung des Systems und hat seine Eignung für den praktischen Einsatz prinzipiell nachgewiesen.

RTP-Gateway

Ebenfalls verfügbar sind Implementierungen von Transcodern. Das Paket *RTP-Gateway* (RTPGW) (vgl. [AM], [Ami95], [AMZ95]) realisiert einen Transcoder für Audio- und Video-Datenströme. Das System besteht aus drei Komponenten:

- Einem Video-Gateway, das MJPEG-, H.261- und NV-codierte Datenströme in H.261-Datenströme wandeln kann und dabei eine Datenratenbegrenzung realisiert,
- einem Audio-Gateway, das PCM-kodierte Datenströme in LPC-kodierte Datenströme wandelt und
- dem Steuerungswerkzeug *rtpgw_ui*. Es dient der Steuerung der Medien-Gateways.

Die Komponenten sind als eigenständige Prozesse implementiert, die über den in [MJ95] beschriebenen Conference-Bus miteinander in Verbindung stehen. Die Steuerungskomponente erlaubt die Umschaltung zwischen verschiedenen Sendern einer Sitzung sowie die Steuerung der Datenratenbegrenzung über eine grafische Benutzerschnittstelle. Die resultierenden Datenströme können, wie bei einem Transmitter, an beliebige Unicast-Empfänger oder Multicast-Gruppen vermittelt werden. Positiv hervorzuheben ist die in RTPGW realisierte Schleifenerkennung. Die Datenvermittlung wird beim Erkennen einer Schleife mit einer Fehlermeldung abgebrochen. Da die Steuerungskomponente auf dem als Gateway fungierenden Rechner ablaufen muß, sind die Einsatzmöglichkeiten für RTPGW, zumindest für die hier betrachteten Szenarien, stark eingeschränkt.

Robust Audio Tool

Ein Beispiel für einen Mixer ist das Robust-Audio-Tool (*rat*) [SHK+]. Neben seiner üblichen Funktion als Medienwerkzeug für Audio-Übertragung kann *rat* als Mixer betrieben werden. Dabei wird der von einer Multicast-Gruppe oder über eine Unicast-Transportadresse empfangene Datenstrom einem Transcoding unterzogen und dann entsprechend den Vorgaben im RTP gemixt. Dabei wird das Werkzeug durch Parameter beim Programmstart konfiguriert. Die Änderung der Parameter im laufenden Betrieb ist nicht möglich. Da *rat* einen vollständigen RTP-Mixer implementiert, ist es ein wichtiger Baustein für Wirkungsketten zur Übertragung von MBone-Sitzungen über "Low-Speed-Networks".

Ein Mixer für Video-Datenströme ist zur Zeit nicht bekannt.

mTunnel

Neben Einzelbausteinen sind in den letzten Monaten auch kombinierte Systeme entwickelt worden, die dem Nutzer Zugriff auf MBone-Sitzungen auf der Basis von Gateways auf der Anwendungsschicht bereitstellen. Im März 1997 wurde von Mitarbeitern der Luleå University of Technology in Schweden das System *mTunnel* [PSS97b] vorgestellt. Dieses System wurde vollständig in der Sprache Java entwickelt und wird in [PSS97a] beschrieben. *mTunnel* besteht aus vier wesentlichen Komponenten:

- Der *Tunneler* übernimmt das Tunneling der Datenströme über ein "Low-Speed-Network". Er multiplext alle Datenströme über einen UDP-Port, wodurch der Einsatz im Zusammenhang mit Firewalls erleichtert wird, da nur ein UDP-Port zwischen auf dem Firewall-System freigegeben werden muß. Zur Übertragung der Datagramme wird die originäre Payload in Unicast-Datagramme gekapselt und um die ursprüngliche Transportadresse ergänzt. Damit entsteht ein zusätzlicher Overhead von mindestens 6 Byte.
- Die zweite Komponente des Systems ist der *Controller*. An jedem Ende des Tunnels ist ein Controller installiert. Beide Controller sind zum Austausch von Kontrollinformationen durch einen TCP-Stream miteinander verbunden. Über diese TCP-Verbindung werden Gruppen angefordert und abbestellt, die Sitzungen mit Prioritäten versehen und Translatoren und Mixer für einzelne Gruppen konfiguriert.
- Dritter Baustein des Systems ist ein *Web-Interface* zur Steuerung des Systems. Das Web-Interface wird durch einen World-Wide-Web-Server zugänglich gemacht, der in die Anwendung integriert ist.
- Die letzte und vierte Komponente des Systems wird als *Translator* bezeichnet und beinhaltet
  - einen Recoder zur Umsetzung von PCM-codierten Audio-Stream in GSM-Kodierung,
  - einen Mixer, der zur Zeit nur für Audio-Datenströme operabel ist,
  - einen Switch, der auf Basis des aktuellen Sprechers jeweils nur den zugehörigen Video-Datenstrom übermittelt, und
  - einen Scaler, der bestimmte Teile eines Datenstroms vernichtet, um den Video-Datenstrom hinsichtlich des zu übertragenden Datenvolumens zu begrenzen.

Wenngleich die Implementierung auf Basis von Java die weitgehende Unabhängigkeit von der Systemplattform bedeutet, ist dieser Weg dennoch in Frage zu stellen: Java-Anwendungen weisen gegenüber plattformspezifischen Programmen ein deutlich schlechteres Laufzeitverhalten auf. Die erforderlichen Transcoder und Mixer erfordern teilweise umfangreiche Berechnungen, die bei geringer Systemleistung zu kritisch einzustufenden Verzögerungen führen.

Unabhängig von der Java-Implementierung besteht folgende Kritik: Da der Tunneler bei der Kapselung der RTP-Daten zusätzliche Bytes für die originäre Multicast-Gruppe ergänzt, erhöht sich die Serialisierungszeit und die nutzbare Datenrate wird reduziert.

MERCI: Accessing MBone Sessions over N-ISDN

Einen ähnlichen Ansatz wie *mTunnel* verfolgt ein im Rahmen des im EU-Projekt MERCI entwickeltes Werkzeug. Es ist in [Yan97] dokumentiert. Ziel ist die Anbindung eines unter Windows-95 betrieben PCs über 2 ISDN-B-Kanäle an den MBone. Ein Rechnersystem im regulären Internet mit direktem Zugriff auf den MBone fungiert als Partner. Auf dem PC am Heimarbeitsplatz kommen für den Zugriff auf MBone-Konferenzen unmodifizierte Versionen der MBone-Tools zum Einsatz. Sie werden durch einen Controller ergänzt, der für die Steuerung des Systems verantwortlich ist. Dieser Prozeß steht mit einer Partnerinstanz auf dem Rechner im regulären Internet über eine TCP-Verbindung in Kontakt. Diese Partnerinstanz kommuniziert über den bereits zitierten Conferencing-Bus mit

verschiedenen Adaptionswerkzeugen, die den Heimarbeitsplatzrechner entsprechend der Wahl des Nutzers mit Datenströmen aus dem MBone versorgen. Diese Programme wurden teilweise neu entwickelt, teilweise werden bestehende Entwicklungen genutzt, wie das Video-Gateway von Elan Amir (vgl. [Ami95],[AM]).

Zur Zeit ist die Vermittlung von Audio- und Video-Datenströmen sowie der Zugriff auf das MBone-Session-Directory über dieses System möglich. In Aussicht gestellt wird die Unterstützung des in Entwicklung befindlichen Shared-Whiteboard *Teledraw*.

Grundsätzlich ist die Architektur des Systems geeignet, MBone-Sitzungen für ein über ISDN angekoppeltes Einzelplatzsystem zugänglich zu machen. Einschränkend bleibt anzumerken, daß das System nicht für den Transport von MBone-Sitzungen in privaten Internets (wie auf Seite 9 beschrieben) geeignet ist. Ein weiteres Problem entsteht durch den hohen Protokoll-Overhead, insbesondere bei der Vermittlung von Audio-Datenströmen. Aufgrund der wenig effektiven Kanalauslastung ist es nicht möglich, das System über nur einen ISDN-B-Kanal zu betreiben.

2.5.2 Gateways auf IP-Ebene

Einen grundsätzlich anderen Weg zur Anbindung privater Internets und von Heimarbeitsplätzen über "Low-Speed-Networks" an den MBone ist die Behandlung der Problematik auf der IP-Schicht.

Bereits einführend wurde erklärt, daß der MBone heute aus einer großen Zahl Multicast-fähiger Subnetze besteht, die durch MBone-Router und IP-Tunnel miteinander verbunden sind. Es liegt nahe, dieses Verfahren auch für die Anbindung von Heimarbeitsplätzen zu nutzen. Dabei wird ein IP-Tunnel zwischen dem privaten Internet, in dem sich der Heimarbeitsplatzrechner befindet, und einem im regulären Internet befindlichen MBone-Router konfiguriert. Damit erlangt der Heimarbeitsplatz bzw. das private Internet den Status eines regulären Teilnetzes des MBone. Die Vorteile dieser Lösung sind evident. Es können alle Anwendungen im MBone ohne Modifikation und ohne Abhängigkeit von der jeweiligen Anwendung direkt zum Einsatz kommen. Trotzdem ist dieser Ansatz aus folgenden Gründen praktisch kaum nutzbar:

- In Abhängigkeit vom eingesetzten Routing-Protokoll (DVMRP, PIM Dense-Mode[42], PIM Sparse-Mode[43]) entsteht auf der Verbindung eine ständige Grundlast, welche die für die Nutzung in anderen Sitzungen bereitstehende Datenrate begrenzt.

- Zum Zeitpunkt der Installation eines Tunnels erfolgt bei den heute üblichen DVMRP-Routern eine Flutung des Netzwerkes mit allen momentan bekannten Gruppen. Erst danach werden nicht erforderliche Gruppen durch Pruning-Meldungen vom untergeordneten Router an den *Up-Stream-Router*[44] der abbestellt. Solange diese Phase nicht abgeschlossen ist, kann die Verbindung nicht für eine reale Anwendung genutzt werden.

[42] PIM Dense-Mode steht für *Protocol Independent Multicast-Dense Mode*. Siehe hierzu [Dee97].

[43] PIM Sparse-Mode steht für *Protocol Independent Multicast-Sparse Mode.* Siehe hierzu [Est97].

[44] Unter dem *Up-Stream-Router* wird hier der Router vertstanden, der hinischtlich des jeweiligen Senders näher zur Wurzel des Multicast-Verteilungsbaums liegt.

- Der Zugang zum regulären Internet von Heimarbeitsplätzen erfolgt über das Wählnetz; die Verbindungen bestehen nur für einen kurzen Zeitraum. Durch den regelmäßigen Verbindungsaufbau würde der im regulären Internet befindliche Multicast-Router praktisch kaum Pruning-Nachrichten versenden und über lange Zeitintervalle hinweg den vollen Verkehrsstrom aus dem MBone erhalten. Dies würde neben dem universitären Netz auch weitere Netzwerkteile hin zu den jeweiligen Sendern belasten.

- Wählzugänge sind heute üblicherweise so konfiguriert, daß Nutzern dynamisch IP-Adressen zugeordnet werden. Die Adreßzuordnung zwischen Nutzer und IP-Adresse ist für den die Heimarbeitsplätze versorgenden MBone-Router nicht transparent. Er müßte bei jedem Zugriff eines für den Dienst autorisierten Nutzers in seiner Konfiguration verändert und neu gestartet werden. Dies verursacht technische und administrative Probleme, die den Nutzen des Dienstes generell in Frage stellen.

- Es gibt nur wenige stabile Implementierungen von MBone-Routern für typische Heimarbeitsplätze. Namentlich gab es zum Zeitpunkt des Entstehens dieser Arbeit keine Implementierungen für die weit verbreiteten Betriebssysteme Windows-95 und Windows-NT.

- Der Datenstrom einer am Medium orientierten Komponente einer MBone-Konferenz wird über eine oder zwei Transportadressen übertragen. Dies sind eine IP-Adresse aus dem Class-D-Bereich sowie ein oder zwei Port-Adressen. Das Multicast-Routing berücksichtigt die Port-Adressen jedoch nicht. Werden mehrere Komponenten beliebiger Sitzungen über eine Class-D-Adresse transportiert, wird der Nutzer am Heimarbeitsplatz neben dem gewünschten Datenstrom auch alle anderen Datenströme vermittelt bekommen, die unter Nutzung dieser Class-D-Adresse übertragen werden.

- Selbst wenn es gelingt, alle bisher dargelegten Probleme zu überwinden, würden die Datenraten der im regulären MBone üblichen Sitzungen in vielen Fällen die Kapazität von zwei parallel betriebenen ISDN-B-Kanälen überschreiten. Eine Übertragung der Sitzungen über einen B-Kanal oder eine Modem-Verbindung ist ohne den Einsatz von Mixern und Transcodern, d.h. Application Level Gateways, nicht möglich.

Obwohl alle dargestellten Punkte dafür sprechen, daß die Kopplung von über Wählnetze angeschlossenen Heimarbeitsplätzen auf der Basis von IP-Multicast-Routing und -Tunneln nicht handhabbar ist, erscheinen in den entsprechenden Mailing-Listen vereinzelt immer wieder Kommentare, die über erfolgreiche Kopplungen auf diesem Weg für einzelne Sitzungen berichten. Namentlich die Übertragungen der Space-Shuttle-Missionen wurden hier wiederholt genannt. Ursächlich hierfür ist, daß die Multicast-Router die Datenströme auf der Basis der verwendeten Port-Komponente der Transportadresse unterschiedlich behandeln. Der Port-Bereich ist wie folgt untergliedert:

Portbereich	Priorität
0 – 16383	Niedrigste Priorität, unklassifizierte Daten.
16384 – 32767	Höchste Priorität, Audio-Daten.
32768 – 49151	Mittlere Priorität, Whiteboard-Daten.
49152 – 65536	Niedrige Priorität, Video-Daten.

Sofern bei der Ankündigung der Sitzung diese Regeln beachtet wurden, wird der Multicast-Router die Audio-Daten bevorzugt übertragen und den Video-Datenstrom zurückstellen. Damit hat er die gleiche Wirkung wie der *Scaler* des Gateway-System *mTunnel*. Die prinzipiellen Kritikpunkte bleiben von diesen Ausnahmen unberührt.

Ein gänzlich anderer Ansatz basiert auf der Nutzung des in Entwicklung befindlichen *Layer Two Tunneling Protocol* (L2TP) [Val97]. Die wesentliche Funktion dieses Protokolls ist die Kapselung von PPP-Datagrammen in IP-Datagrammen mit dem Ziel, einen Nutzer an einem Heimarbeitsplatz oder einen mobilen Nutzer über Internet-Service-Provider transparent an ein lokales Netz anzukoppeln. Die Besonderheit ist hierbei, daß der Nutzer auch andere als IP-Protokolle nutzen kann und das System Sicherheitskriterien erfüllt, die von professionellen Nutzern an die Sicherheit des Transports von unternehmensrelevanten Daten gestellt werden. Damit fällt es in die Gruppe von Protokollen zum Aufbau von virtuellen privaten Netzen auf der Infrastruktur des Internet.

Bei der Nutzung dieses Protokolls wählen sich die Endnutzer über einen nahe gelegenen Internet-Service-Provider ein, der als besonderen Service das L2TP unterstützt. Von hier aus werden ihre PPP-Datagramme, verpackt in IP-Datagramme, an einen L2TP-Network-Server im heimatlichen LAN transportiert. Dort wird die PPP-PDU dem IP-Paket entnommen und so auf das LAN weitergeleitet, als ob der Nutzer sich direkt per PPP in den L2TP-Server eingewählt hätte. Als L2TP-Server kann ein Microsoft-NT-Server ab der Version 4.0 dienen. Den Äußerungen auf der MBone-Mailing-Liste zufolge soll auf diesem Weg auch die Übertragung von IP-Multicast-Datagrammen möglich sein. Leider konnte dieser Ansatz aus Mangel an entsprechender Infrastruktur nicht praktisch erprobt werden. Letztlich gelten hier jedoch die gleichen Bedenken bezüglich der anfallenden Verkehrsflüsse, wie bei der direkten Verbindung des Heimarbeitsplatzes mit dem Internet über DVMRP-Tunnel.

2.5.3 Entwicklungen auf PPP-Ebene

Auch Ansätze zur Lösung der Problematik auf der Ebene des Link-Protokolls – PPP – sind erkennbar. Fordert ein Endsystem in einem üblichen lokalen Netzwerk, wie Ethernet, eine Multicast-Gruppe an, sendet es dazu einen Request des *Internet Group Management Protocol* (IGMP) aus (vgl. [Dee89], [Fen97]). IGMP ist, wie UDP, TCP und ICMP, ein IP-Protokoll. Der Request wird im Fall des lokalen Netzes von einem Multicast-Router empfangen und über ein entsprechendes Routing-Protokoll wird die Multicast-Gruppe abonniert und in das Subnetz vermittelt. Bei PPP-Verbindungen war es bis vor kurzem üblich, IGMP-Anfragen nicht zu übermitteln oder seitens der Access-Routers nicht darauf zu reagieren. Neuerdings bieten die Hersteller von Access-Routern allerdings die Unterstützung von IGMP an und erlauben damit, den Zugriff auf Multicast-Gruppen über PPP-Verbindungen. Sofern der über PPP angebundene Rechner oder Router dies unterstützt, werden IGMP-Anfragen über den PPP-Link zum Access-Router geleitet. Dieser abonniert seinerseits im lokalen Netzwerk die Multicast-Gruppe. Werden fortan IP-Datagramme für diese Gruppe empfangen, werden sie an den über PPP angeschlossenen Rechner weitergeleitet. Somit wird der Access-Router zum Multicast-Router.

Besonders elegant bei diesem Verfahren ist, daß die Schwierigkeiten beim Aufbau von DVMRP-Tunneln über Wählverbindungen entfallen und keinerlei administrative Schwierigkeiten entstehen. Weiterhin entfällt der Overhead durch Einkapselung der IP-Multicast-Datagramme in IP-

Unicast-Datagramme. Der mit dem IP-Multicast-Routing verbundene Datenstrom reduziert sich ebenfalls deutlich.

Die Firma Ascend bietet in ihren Zugangsroutern des Typs *MAX 4000* im *IP-only Release* die eingeschränkte Unterstützung für diesen Dienst ab dem Softwarelevel 4.6C an [Asc96]. Die Einschränkung besteht darin, daß der Multicast-Verkehr nur vom Access-Router zum Client geleitet wird. Damit können MBone-Sitzungen verfolgt werden, es können jedoch keine eigene Beiträge geliefert werden. Merkmale der Implementierung sind, daß das IGMP in den Versionen 1 und 2 unterstützt wird und priorisiertes Packet-Dropping auf Basis der Multicast-Port entsprechend der Implementierung der DVMRP Multicast-Router realisiert wird.

Zusätzliche Attraktivität erlangt diese Lösung durch die in Entwicklung befindlichen Ansätze zur komprimierten Übertragung von IP-, UDP- und RTP-Headern über PPP-Verbindungen. Diese Ansätze sind in [Bor97b], [Bor97a], [ECB97] und [CJ97] beschrieben. Mit den in diesen Internet-Drafts beschriebenen Verfahren wird es gelingen, die erzielbare Datenübertragungsrate auf PPP-Verbindungen der theoretisch möglichen Übertragungsrate anzunähern. Durch die Einführung von Suspend/Resume-Techniken sowie der Fragmentierung im Subframe-Bereich wird zudem die Einführung von Dienstklassen mit geringeren Verzögerungszeiten für ausgezeichnete Dienste möglich.

Leider erfordern diese Verbesserungen nachhaltige Änderungen an der Hard- und Software auf Link-Ebene. Während die Ansätze zur Header-Compression lediglich die Implementierung entsprechender Algorithmen auf den Access-Routern und Endsystemen sowie die Einführung neuer Link- und Network-Layer-Optionen und -Prozeduren im PPP erfordern, sind für die Einführung der Suspend/Resume-Techniken in Verbindung mit der Fragmentierung im Subframe-Bereich noch weitergehende Anpassungen erforderlich. Somit steht zu erwarten, daß noch einige Zeit vergeht, bis eine plattformübergreifende und geographisch nahezu flächendeckende Implementierung dieser Techniken zur Verfügung steht.

Letztlich lösen diese Technologien aber nicht das grundsätzliche Problem, daß die bereits heute im MBone üblichen Datenübertragungsraten deutlich über den theoretisch möglichen Übertragungsraten im Bereich der “Low-Speed Serial Links” liegen. Auch wenn in den kommenden Jahren Digital-Subscriber-Lines mit Datenübertragungsraten im 2 Mbps Bereich üblich werden und die Codierungen für Video- und Audio-Datenströme weiter verbessert werden können, bleibt dieses grundsätzliche Ungleichgewicht erhalten. Letztlich kann auf Application-Layer-Gateways zur Anpassung der Datenraten nicht verzichtet werden. Zudem ist davon auszugehen, daß auch in nächster Zeit das Multicast-Routing nur in Teilbereichen des Internets zur Verfügung stehen wird.

2.6 Anforderungen an das System

Aus den in den vorangegangenen Abschnitten beschriebenen Sachverhalten ergeben sich folgende Anforderungen an ein System zur Übertragung multimedialer Echtzeitdatenströme über Internet-Zugangsnetze geringer Datenrate:

Das zu realisierende System soll den Nutzern von Heimarbeitsplätzen ermöglichen, an multimedialen Konferenzen im Internet aktiv teilzunehmen. Der Teilnehmer befindet sich entweder

an einem Einzelplatzsystem, das über einen ISDN-B-Kanal und einen Access-Router an das Internet gekoppelt ist oder in einem privaten Internet, das über einen ISDN-Router und einen Access-Router mit dem Internet verbunden ist. Bei der ISDN-Verbindung handelt es sich um eine Wählverbindung, die Zuordnung der IP-Adresse des Routers im privaten Internet oder dem Endsystem erfolgt dynamisch beim Verbindungsaufbau. Auf der Wählverbindung wird als Layer-2-Protokoll das Point-to-Point-Protocol (PPP) benutzt. Als Netzwerkprotokoll kommt das Internet-Protokoll (IP) zum Einsatz. Das zu realisierende System muß also auf Basis des Internet-Protokolls arbeiten.

Es ist davon auszugehen, daß an den Konferenzen häufig mehr als zwei Personen teilnehmen. Die Datenverteilung erfolgt über IP-Multicast. Bezugssystem ist der MBone im Internet. Die in den Konferenzen entstehenden Datenraten übersteigen die nominellen Übertragungskapazitäten eines ISDN-B-Kanals um eine Größenordnung – auf der Basis von Application-Layer-Gateways ist eine entsprechende Datenratenanpassung vorzunehmen. Für diesen Zweck stehen entsprechende UNIX-Systeme im Internet, nahe dem Anschlußpunkt des Access-Routers, zur Verfügung. Diese Systeme besitzen direkten Zugang zum MBone. Der Access-Router selbst kann nicht zur Konvertierung von Daten genutzt werden.

Die in den multimedialen Konferenzen anfallenden Audio- und Video-Datenströme nutzen als Transportprotokoll RTP. Zudem entstehen durch Shared-Tools weitere Multicast- und Unicast-Datenströme, die zu übertragen sind.

Der Nutzer soll mit hohem Komfort auf Konferenzen zugreifen können. Die dazu üblicherweise eingesetzten Werkzeuge sollen ohne Modifikation weitergenutzt werden können. Hinsichtlich der Übertragungsqualität für Audio- und Video-Datenströme sind Einschränkungen unvermeidlich. Mit höchster Priorität sind Audio-Daten zu übertragen, als Codierung soll mindestens GSM geboten werden. Video-Daten sollen in H.261-Codierung übertragen werden, wobei die Übertragung des Video-Datenstroms die geringste Priorität hat.

Das zu entwickelnde System soll Konferenzen gemäß dem Internet-Conferencing-Protocol-Stack unterstützen, darüber hinaus aber flexibel genug sein, um auch andere Konferenz-Modelle prinzipiell zu unterstützen, insbesondere Konferenzen nach ITU H.323.

Es ist davon auszugehen, daß sich potentielle Kommunikationspartner in interaktiven Sitzungen im B-WiN befinden und die Laufzeiten der Pakete im B-WiN in der Größenordnung von 20 – 30 ms liegen. Die Gesamtlaufzeit von Paketen soll unter Einbeziehung des Systems 400 ms nicht übersteigen, wobei eine maximale Ende-zu-Ende-Verzögerung von 150 ms angestrebt werden sollte (vgl. [Int93]).

Als Endsysteme sollen Personal-Computer dienen, die unter den Betriebssystemen Windows-95, Windows-NT und Linux betrieben werden. Multicast-Unterstützung im Betriebssystem ist erforderlich, die angeschlossen Network-Devices müssen Multicast nicht notwendigerweise unterstützen.

Im folgenden Kapitel 3 wird ein System entwickelt, das diesen Anforderungen gerecht wird. Kapitel 4 beschreibt die exemplarische Implementierung dieses Systems und enthält eine Bewertung auf der Basis von Messungen, die am realisierten System vorgenommen wurden.

Kapitel 3

Entwurf

3.1 Systemstruktur

Das zu entwickelnde System verfolgt das Ziel, Nutzern von Heimarbeitsplätzen den Zugriff auf MBone-Konferenzen über Zugangsleitungen zu ermöglichen, die keine oder unzureichende Unterstützung für IP-Multicast bieten. Der Schwerpunkt liegt bei Zugangssystemen mit geringen Datenraten. Dies sind in erster Linie Zugänge über das ISDN-Netz, die mit der Nutzung eines ISDN-B-Kanals auskommen. Wenngleich die Architektur des vorliegenden Systems auf diese Anwendung fokussiert, soll die Nutzung auch über andere Zugangsnetze, wie Digital-Subscriber-Lines (xDSL), möglich sein. Die Datenvermittlung erfolgt auf Basis des Internet-Protokolls, als Transportprotokoll für Nutzdaten wird das User-Datagram-Protocol eingesetzt.

3.1.1 Generelle Systemstruktur

Das System dient dem Zugriff auf den MBone und führt dementsprechend den Namen *MBone Access Gateway* (MAGW). Es handelt sich um ein verteiltes System, bestehend aus den beiden folgenden Diensten:

MBone-Access-Server (MAS)
: Dieser Dienst befindet sich auf einem Rechner im regulären Internet mit direktem Zugriff auf den MBone. Er hat die Aufgabe, Datenströme aus dem MBone an Nutzer an Heimarbeitsplätzen zu vermitteln. Eng damit verknüpft sind Aufgaben wie Datenratenanpassung, Kompression und Ressourcenverwaltung. Im weiteren Verlauf dieses Abschnitts wird genauer auf diese Punkte eingegangen.

MBone-Access-Gate (MAG)
: Dieser Dienst befindet sich auf dem Rechner des Nutzers am Heimarbeitsplatz oder im Fall privater Internets auf einem Vermittlungsrechner. Es ist zu berücksichtigen, daß auch mehrere Nutzer in einem privaten Internet über eine Instanz dieses Dienstes mit Konferenzpartnern im regulären Internet verbunden werden können. Zu beachten ist ferner, daß Multicast von PCs unter Windows-95 zwar generell unterstützt wird, jedoch nicht alle

Netzwerkkartentreiber Multicast-Unterstützung bieten. Auch Rechner mit solchen Netzwerkkarten sollen durch das MAG Zugriff auf MBone-Konferenzen erhalten.

Viele private Internets werden über Firewalls mit dem Internet verbunden. Das zu entwickelnde System soll diesen Anwendungsfall ausdrücklich unterstützen.

Das MBone-Access-Gate und der MBone-Access-Server werden, von außen betrachtet, durch UDP-Tunnel verbunden, die auf der Basis von Transportadressen-Tupeln operieren. Die Struktur der Tunnel wird im Abschnitt 3.1.3, Seite 59 erläutert.

Da die Datenvermittlung auf Basis von Transportadressen und nicht, wie beim Multicast-Routing üblich, unter Nutzung der Internet Class-D Adressen erfolgt, ist das Internet-Group-Management-Protocol zur Anforderung der Datenströme ungeeignet. Statt dessen wird ein spezielles Anwendungsprotokoll entwickelt[1], das die explizite Anforderung der Datenströme durch den Nutzer erforderlich macht. Um den Anwender von diesen Aufgaben zu entlasten, ohne bestehende MBone-Anwendungen modifizieren zu müssen, wurden hierfür sogenannte Wrapper-Anwendungen und Plug-ins entwickelt.

3.1.2 Einsatzszenarien

Das MBone-Access-Gateway soll, wie bereits dargestellt, in verschiedenen Szenarien einsetzbar sein. Stellvertretend werden hier drei Varianten vorgestellt.

3.1.2.1 Anschluß eines Einzelheimarbeitsplatzes

Dieses Szenario unterstellt, daß der Nutzer an seinem Heimarbeitsplatz über einen Personal-Computer verfügt, der unter Windows-95, Windows-NT oder unter einem UNIX-Derivat betrieben wird. Der Zugriff auf das Internet erfolgt unter Nutzung einer ISDN-Wählleitung und eines Access-Routers, von dem der Nutzer eine Internet-Adresse für die Dauer der Sitzung dynamisch zugewiesen bekommt. Über den Access-Router können Prozesse auf dem Rechner des Nutzers IP-Unicast-Verbindungen zum MBone-Access-Server aufbauen. Die entsprechende Netz- und Anwendungsstruktur zeigt Abbildung 3.1.[2]

Auf den ersten Blick mag es ungünstig erscheinen, die zu vermittelnden Daten nicht direkt an die Medien-Werkzeuge zu vermitteln, sondern durch eine Instanz des MBone-Access-Gates zu leiten. Dies ist jedoch die Voraussetzung dafür, daß auf der ISDN-Leitung fortschrittliche Kompressionsmethoden angewendet werden können und eine Flußsteuerung zwischen MBone-Access-Server und MBone-Access-Gate einsetzbar ist. Auf diese Aspekte wird in den folgenden Abschnitten detaillierter eingegangen.

Es ist davon auszugehen, daß diese Einsatzvariante am häufigsten angewendet wird.

[1] Das Protokoll trägt den Namen *MBone Access Server Control Protocol* (MASCP) und wird in Abschnitt 3.2.10, Seite 119 vorgestellt.

[2] In dieser sowie den beiden folgenden Abbildungen wurden zur besseren Übersicht nur die Datenpfade der Nutzdatenströme berücksichtigt. Kontrollflüsse und Prozeß-Steuerungsbeziehungen fehlen. Auf diese Aspekte wird im weiteren genauer eingegangen.

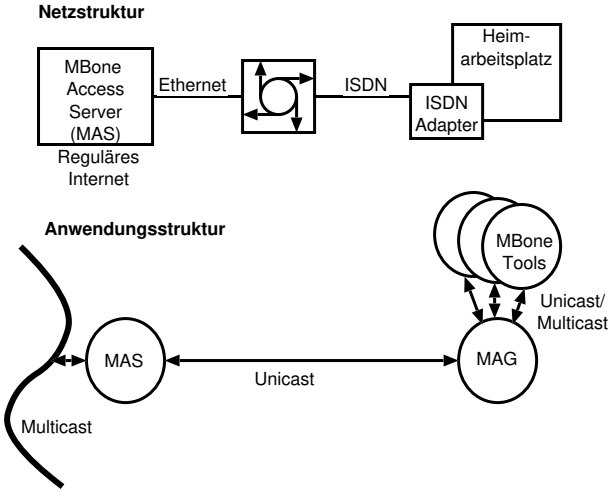

Abbildung 3.1: Zugang für einen Einzelheimarbeitsplatz

3.1.2.2 Anschluß eines privaten Internets

Wesentliches Merkmal dieser Variante ist, daß der Internet-Router des privaten Internets die Adreßräume des regulären und des privaten Internets voneinander trennt. Es handelt sich hierbei um ein UNIX-System, das mit beiden Netzen verbunden ist und auf dem eine Instanz des MBone-Access-Gates betrieben wird. Die Rechner der Konferenzteilnehmer befinden sich einerseits im Multicast-fähigen privaten Internet und andererseits im regulären MBone. Das MBone-Access-Gate setzt die vom MBone-Access-Server erhaltenen Datenströme aus dem MBone wieder in Multicast-Datenströme um: es ist somit ein Spiegelbild des MBone-Access-Servers. Wenngleich dieses Szenario dem Betrieb von Multicast-Routern gleicht, gibt es doch zwei wesentliche Unterschiede: Zum einen überbrücken die Datenströme die Grenzen von Adreßräumen. Zum anderen erfolgt die Anforderung der Datenströme nicht auf Basis des IGMP, sondern durch ein spezifisches Anwendungprotokoll auf der Basis von Transportadressen. Die entsprechende Netz- und Anwendungsstruktur zeigt Abbildung 3.2.

3.1.2.3 MBone-Zugriff über dedizierte Firewalls

Diese Variante ähnelt von ihrer Funktionalität dem vorangegangenen Szenario. Ein wichtiger Unterschied ist, daß das MBone-Access-Gate nicht auf dem System betrieben wird, das das private Internet vom regulären Internet abgrenzt. Um dennoch die Vermittlung der MBone-Datenströme zu erlauben, ist die Nutzung eines sogenannten *Circuit Relays* (vgl. [Kya96], Seite 160ff.) erforderlich. Eines der verbreitesten Systeme dieser Kategorie ist *SOCKS*. Seit der Version 5 wird hier auch die Vermittlung von UDP-Datenströmen unterstützt.

Das MBone-Access-Gate soll den Betrieb über Firewalls, die SOCKS in der Version 5 [LGL+96] unterstützen, erlauben. Abbildung 3.3 zeigt die entsprechende Netz- und Anwendungsstruktur.

Ein Nachteil dieser Lösung gegenüber dem Betrieb des MAG auf dem Firewall-Rechner ist, daß die Datenströme ein weiteres Mal über einen Forwarder fließen. Dadurch wird die Paketlaufzeit

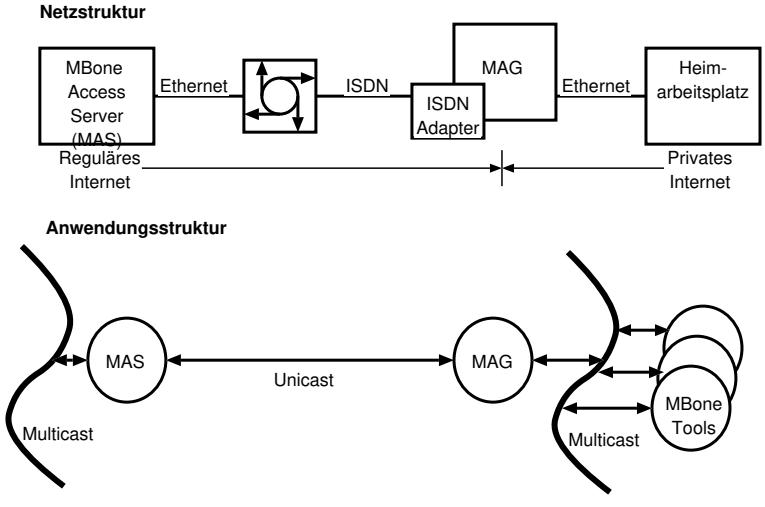

Abbildung 3.2: Zugang für ein privates Internet

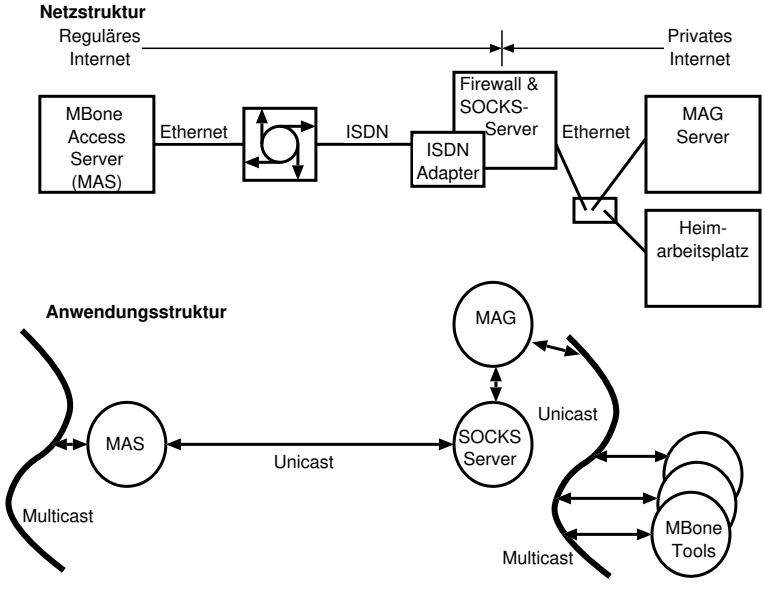

Abbildung 3.3: Zugang über dedizierte Firewalls

verlängert. Dieser Nachteil muß unter den gegebenen Rahmenbedingungen akzeptiert werden.

3.1.3 Tunnelstruktur zwischen MAS und MAG

3.1.3.1 Problematik der Transportadressen

Der MBone-Access-Server und das MBone-Access-Gate werden über UDP-Tunnel verbunden. Sie dienen dem Transport der Nutzdaten und der Flußsteuerung. Ergänzt werden die Tunnel durch eine TCP-Verbindung zwischen den Prozessen, die der Etablierung der Tunnel sowie der Anforderung von Datenströmen dient.

Aus Sicht der Anwendungen im MBone und am Heimarbeitsplatz müssen die Tunnel die Nutzdaten transparent übermitteln und eindeutige und differenzierbare Transportadressen bereitstellen. Jede PDU, die über den MBone übertragen wird, läßt sich anhand der IP-Multicast-Adresse, dem zugehörigen Port, der Absenderadresse und dem Sende-Port eindeutig einem Datenfluß oder *Flow*[3] zuordnen (vgl. [Bra97, Abschnitt 1.1]). Das MBone-Access-System muß diese Flows transparent übertragen. Dabei stellt sich folgendes Problem: Im MBone unterscheiden sich Flows neben der Multicast-Transportadresse vor allem durch die IP-Adresse des Absenders. Das Protokoll der Space-Shuttle Mission STS-94 (Abschnitt 2.4.3, Seite 42) zeigt 210 Sender mit unterschiedlichen IP-Adressen bei 386 Sendern, die hinsichtlich ihrer Transportadresse zu unterscheiden waren.

Bei einer Übertragung dieses Datenstroms über das MBone-Access-System an einen Heimarbeitsplatz hätten alle Flows die gleiche Absendeadresse – die des MAG-Servers. Sollen die Flows weiterhin zu unterscheiden sein, muß das MAG die PDUs eines jeden Flow von einem anderen Port versenden.

Auch zwischen MAG und MAS müssen die Flows differenzierbar sein. Dies kann ebenfalls dadurch gewährleistet werden, daß die PDUs eines jeden Flows von einem anderen Port des MAS-Rechners versendet werden oder an unterschiedliche Ports des MAG-Rechners gesendet werden. Beide Ansätze für sich sind ungünstig, da durch die in Abschnitt 3.2.5, Seite 97, vorgestellte Header-Compression der Bedarf zur weiteren Unterscheidung einzelner Komponenten der Flows entsteht: Es entsteht ein um den Faktor 3-4 größerer Bedarf an Ports. Für die zitierte Space-Shuttle-Mission müßten damit über 1500 Ports für einen Nutzer bei MAG- oder MAS-Server breitgestellt werden. In beiden Fällen sind Probleme mit Limitierungen des Betriebssystems zu erwarten.[4]

[3] Durch einen *Flow* wird ein gerichteter Baum-Graph zwischen der Quelle und der Gruppe der Empfänger definiert.

[4] Die Ports werden durch die in heutigen UNIX-Betriebssystemen übliche Berkeley-UNIX Socket-Schnittstelle zugänglich gemacht [WS95, Seite 9ff.]. Die Verbindung zwischen Nutzer-Prozeß und Socket wird über Datei-Deskriptoren realisiert. Jeder Prozeß unterliegt einer Limitierung der offenen Datei-Deskriptoren. Die Limitierung untergliedert sich in ein Hard- und ein Soft-Limit. Das Soft-Limit kann mittels der *setrlimit*-Routine des Betriebssystems auf das Hard-Limit erhöht werden. Übliche Standard-Soft-Limits liegen im Bereich von 64 bis 256 Datei-Deskriptoren pro Prozeß. Die Hard-Limits bewegen sich im Bereich von 256 bis 1024. Ohne eine Modifikation des Betriebssystem-Kerns lassen sich diese Begrenzungen nicht überwinden. Eine weitere Limitierung resultiert aus der für den asynchronen I/O über das Netzwerk unter den meisten Betriebssystemen unverzichtbaren *select*-Routine; unter dem Sun-Betriebssystem Solaris kann sie beispielsweise nicht mehr als ca. 1500 Datei-Deskriptoren verwalten.

Alternativ können MAG- und MAS-Server ein UDP-Port-Paar für die Übertragung nutzen und durch einen Protokoll-Kopf in jeder PDU die Flow-Zuordnung signalisieren. Abgesehen von dem nicht auszuschließenden Problem der Fragmentierung würde diese Lösung jede PDU um 1-2 Bytes vergrößern. Das hätte bei der Übertragung über einen ISDN-B-Kanal eine zusätzliche Verzögerung von 0.125 ms bzw. 0.25 ms zur Folge. Wenngleich diese Verzögerung relativ klein ist, sollte sie nur dann akzeptiert werden, wenn alle anderen Möglichkeiten ausgeschöpft sind.

3.1.3.2 Nutzung der UDP-Ports zur Nachrichtenübertragung

Eine günstige Lösung ergibt sich aus der Kombination beider Verfahren. Dazu werden beim MAG- und MAS-Rechner 64 Ports für UDP-Tunnel bereitgestellt. Die sich ergebenden Flows werden jedoch nicht explizit zur Codierung des Flows, sondern zur Übertragung der Information genutzt. Die Tunnel-Schicht bietet den darüber liegenden Schichten ganz allgemein den Transport von PDUs unterschiedlicher Payload-Typen an. Bei der Übertragung einer PDU wird der Payload-Typ mit 4 Bit Länge sowie das erste Byte der Payload durch den gewählten Flow zwischen MAS und MAG codiert. Damit wird den höheren Schichten des Systems ein transparenter Tunnel zwischen MAS und MAG angeboten und trotzdem eine bzgl. der Verzögerung verbesserte Übertragung realisiert. Durch die Nutzung der Port-Felder der UDP-PDUs zur Übertragung von Payload-Informationen werden 12 Bit Paketlänge pro Paket auf dem Tunnel eingespart.

Die Wirkungsweise des Verfahrens wird am folgenden Beispiel verdeutlicht. Der MAS hat zwei Datenpakete mit den folgenden Inhalten $\{C_{16}, AF_{16}, 3B_{16}, 7C_{16}\}$ und $\{5_{16}, 64_{16}, E1_{16}, 4F_{16}\}$ an das MAG zu übertragen.[5] Jedes Paket hat eine Länge von 28 Bit. Die ersten 12 Bit der Pakete ($\{C_{16}, AF_{16}\}$ bzw. $\{5_{16}, 64_{16}\}$) werden gemäß des vorgestellten Verfahrens nicht als Payload innerhalb der Tunnel-PDU, sondern durch die Wahl des Sende- und Empfangsports übertragen. Zwecks einfacher Betrachtung wird unterstellt, daß die Sende und Empfangsports bei Sender und Empfänger beginnend mit einem Port A in Folge reserviert wurden. Die Portnummern des Empfangsports (P_R) und des Sende-Ports (P_S) ergeben sich für das erste Paket zu

$$P_R = P_{A_R} + (CAF_{16} div 40_{16}) = P_{A_R} + 32_{16} = P_{A_R} + 50_{10}$$

$$P_S = P_{A_S} + (CAF_{16} mod 40_{16}) = P_{A_S} + 2F_{16} = P_{A_S} + 47_{10}$$

und entsprechend für das zweite Paket

$$P_R = P_{A_S} + (564_{16} div 40_{16}) = P_{A_R} + 15_{16} = P_{A_R} + 21_{10}$$

$$P_S = P_{A_S} + (564_{16} mod 40_{16}) = P_{A_S} + 24_{16} = P_{A_S} + 36_{10}$$

Der Wert des ersten Halb-Bytes des Pakets wird arithmetsich um 8 Bit nach links verschoben und der Wert des ersten Bytes wird hierzu addiert. Das Ergebnis dieser Operation wird gannzahlig durch den Wert 64_{10} bzw. 40_{16} dividiert. Das Ergbnis der ganzahligen Division liefert den Index des Empfangsports (P_R). Der ganzahlige Rest der Division liefert den Index des Sende-Ports (P_S).

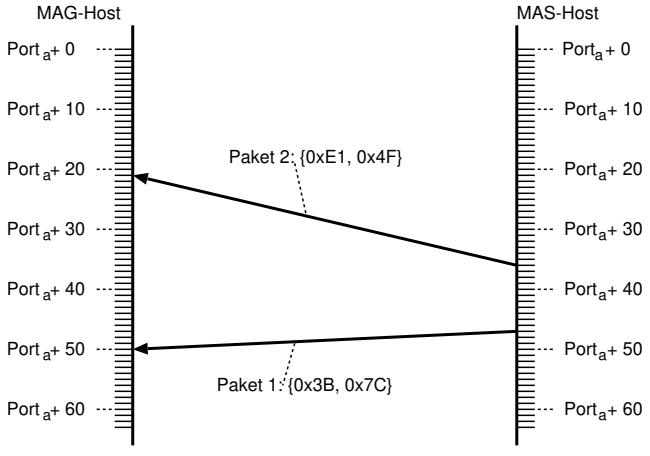

Abbildung 3.4: Nutzung von Ports zur Nachrichtenübertragung

Die Payload der UDP-Pakete besteht dann lediglich aus den Bytes $\{0x3B, 0x7C\}$ bzw. $\{0xE1, 0x4F\}$. Abbildung 3.4 stellt die Übertragung der beiden Datenpakete grafisch dar.

Das vorgestellte Verfahren zur Mitnutzung der Ports zur Nachrichtenübertragung harmoniert optimal mit IP-Routern, die reines FIFO-Queuing nutzen (vgl. Abschnitt 3.2.1, Seite 65f.). Dies ist heute Standard. Wenn in zukünftigen Systemen faire Queuing-Verfahren pro Flow zum Einsatz kommen, können im Fall von Stauungen Pakete eines externen Flows nicht in der Reihenfolge beim Empfänger angelangen, in der sie versendet wurden. Hier gilt, daß das MBone-Access-Gateway Stauungen weitgehend vermeidet (Siehe hierzu Abschnitt 3.2.1, Seite 65) und zudem durch das überlagerte *Header-Compression*-System (Siehe Abschnitt 3.2.5, Seite 97) eine Methodik zur Behebung dieser Defizite bereitgestellt wird.

3.1.4 Wrapper-Anwendungen

3.1.4.1 Grundlegendes Konzept

Ein wesentliches Ziel des MAG-Service ist, daß am Heimarbeitsplatz unveränderte MBone-Werkzeuge für die interaktive interpersonelle Kommunikation zum Einsatz kommen können. Andererseits muß das MAG die erforderlichen Datenströme explizit beim MAS anfordern. Dazu muß dem MAG mitgeteilt werden, welche Datenströme anzufordern sind. Die notwendigen Parameter werden dem MBone-Werkzeug vom Nutzer am Heimarbeitsplatz beim Programmstart mitgegeben, es fehlt jedoch die Verbindung zum MAG. Diesem Zweck dienen die *Wrapper*-Anwendungen.

Die grundlegende Idee der Wrapper-Anwendungen ist, daß der Benutzer anstelle des originären MBone-Tools eine entsprechende Wrapper-Anwendung startet, die mit den gleichen Parametern wie das MBone-Tool konfiguriert wird. Die Wrapper-Anwendung wertet die Parameter aus und

[5] Jedes Paket besteht aus einem Halb-Byte und 3 Bytes, deren Wert in hexadezimaler Schreibweise angegeben ist.

leitet daraus die Definition der benötigten Datenströme ab. Diese Daten werden an das zugeordnete MAG weitergeleitet, welches die Anforderung wiederum an den MAS weitergibt. Erst nachdem eine Antwort auf die Anforderung vorliegt, startet die Wrapper-Anwendung das originale MBone-Tool beim Nutzer. Beendet der Nutzer das MBone-Tool, signalisiert die Wrapper-Anwendung dies dem MAG und beendet sich dann ebenfalls.

Die Wrapper-Anwendungen fungieren folglich als Bindeglied zwischen dem jeweiligen MBone-Werkzeug und dem MAG und sind daher ein wesentliches Element des Gesamtsystems.

3.1.4.2 Anforderungen an die Wrapper-Anwendungen

Wenngleich sich die MBone-Werkzeuge in ihrer Parameterisierung ähneln, sind dennoch Unterschiede zu erkennen, so daß für jedes am Heimarbeitsplatz einzusetzende MBone-Werkzeug auch eine entsprechende Wrapper-Anwendung zu erstellen ist. Die Werkzeuge unterscheiden sich neben der Parameterisierung teilweise auch in der Art und Weise des Netzwerk-Zugriffs. Dies ist bei der Entwicklung der Werkzeuge zu beachten.

Ebenfalls ist zu berücksichtigen, daß einige MBone-Tools selbst wieder andere MBone-Tools starten. Ein weithin bekanntes Beispiel hierfür ist das Session-Directory-Tool *sdr*. Hierfür sind neben den Wrapper-Anwendungen entsprechende *Plug-Ins* zu erstellen.

Die MBone-Werkzeuge sind für eine breite Palette von Rechnerarchitekturen und Betriebssystemen verfügbar. Daraus ergibt sich die Forderung, daß auch Wrapper-Anwendungen zumindest für die wichtigsten Plattformen bereitsgestellt werden. Dazu gehören Microsoft Windows, Microsoft NT, Linux auf Intel-Rechnern, Solaris auf Sun-Rechnern sowie IRIX auf Silicon-Graphics-Systemen. Aus dieser Aufzählung wird deutlich, daß Portabilität eine wichtige Anforderung an die Wrapper-Anwendungen ist.

Ein weiterer Gesichtspunkt ergibt sich aus der Rolle der Werkzeuge im Gesamtszenario. In der Regel startet der Nutzer während einer Konferenz mehrere Werkzeuge, z.B. ein Session-Directory-Tool, ein Audio-Tool, ein Video-Tool sowie ein Whiteboard-Tool. Im vorliegenden Fall muß für jedes dieser Werkzeuge eine entsprechende Wrapper-Anwendung gestartet werden. Folglich sollten die Wrapper-Anwendungen effizient bezüglich Start-Up-Zeit, Speicherbelegung und benötigter Rechenleistung sein. Die Implementierung in einer Interpreter-Sprache wie *Perl* [Wal96], *Tcl* [Ous94] oder *Java* [Sun97] ist daher mit Problemen behaftet. Deshalb ist die Realisierung in der Sprache *C* [KR90] oder *C++* [Str91] anzustreben.

3.1.4.3 Architektur der Wrapper-Anwendungen

In Abbildung 3.5 ist die Einbettung der Wrapper-Anwendung *rvat* in das Gesamtsystem in drei unterschiedlichen Betriebssituationen dargestellt. *rvat* ist ein Wrapper für das MBone-Audio-Tool *vat* [JM].

Die erste Betriebssituation zeigt den Einsatz des System in einem privaten Internet, welches nicht Multicast-fähig ist. Der Audio-Datenstrom wird der *vat*-Instanz über Unicast-UDP zugeführt. Das zweite Bild zeigt das gleiche Szenario. Hier sind das private Internet sowie der Rechner des Nutzers Multicast-fähig und die Weiterleitung des Datenstroms vom MAG zum *vat*

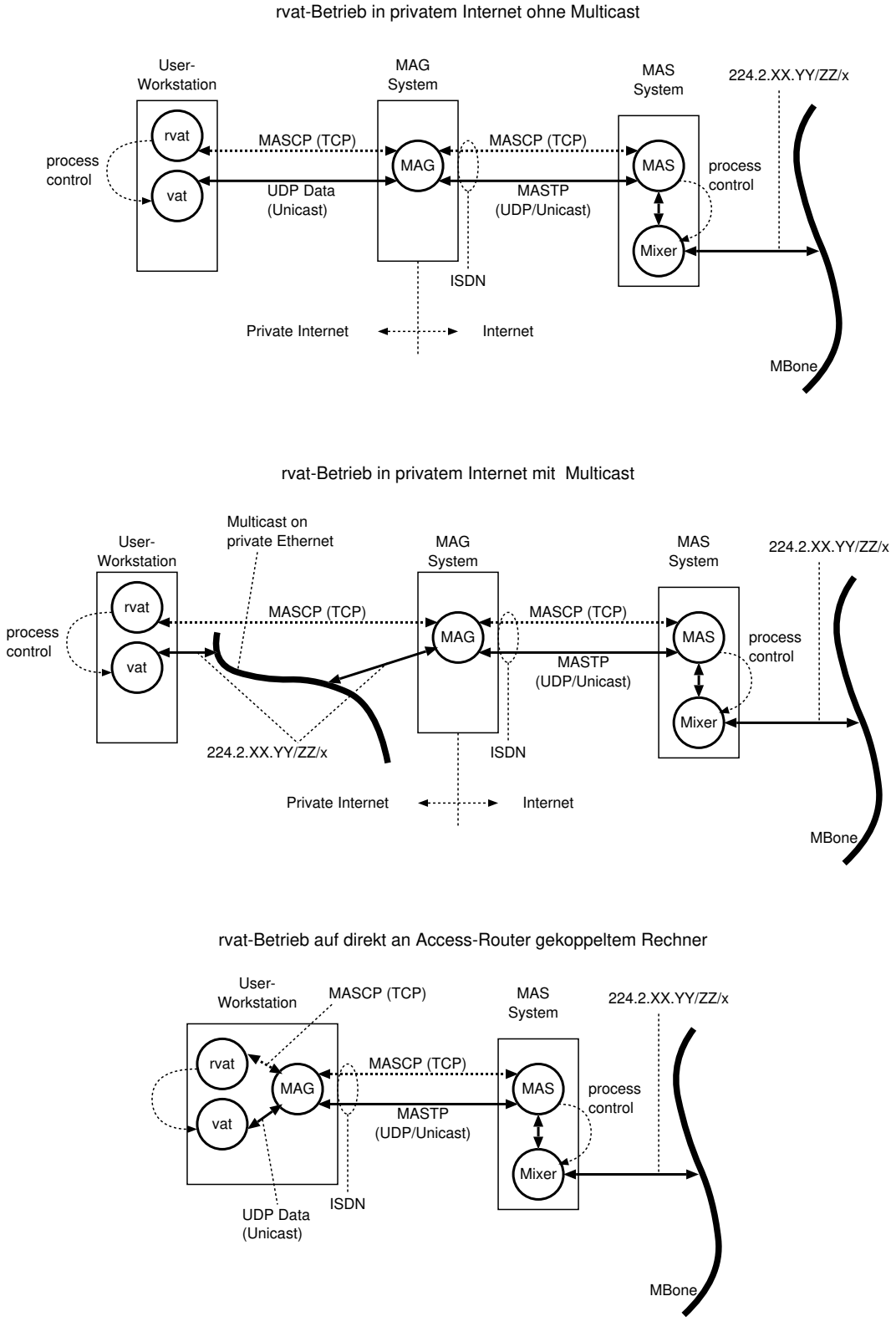

Abbildung 3.5: Einbettung der Wrapper-Anwendungen in das Gesamtsystem anhand des Beispiels *rvat* in drei Betriebssituationen

erfolgt per Multicast. Das dritte Bild zeigt schließlich den Betrieb des Systems für den Fall, daß der Rechner des Nutzers direkt über den Access-Router mit dem MAS im Internet verbunden ist.

Die Wrapper-Anwendung *rvat* hat in allen Betriebsfällen die gleichen Aufgaben:

- Aus den vom Nutzer angegebenen Parametern wird der aus dem MBone zu beziehende Datenstrom festgelegt. Die Wrapper-Anwendung kommuniziert eine entsprechende Anfrage über das MAG an den MAS.

  Zur Kommunikation mit dem MAG nutzt *rvat* das in Abschnitt 3.2.10, Seite 119, vorgestellte MAS-Kontroll-Protokoll. Wenngleich es originär für die Kommunikation zwischen MAG und MAS entworfen wurde, gestattet es aufgrund der Verwandtschaft der Anforderungen auch die Kommunikation zwischen Wrapper-Anwendung und MAG.

- Nachdem die Anfrage beantwortet wurde, legt *rvat* die Parameter für die zu startende *vat*-Instanz fest und startet das Werkzeug.

- Bei Beendigung der *vat*-Instanz durch den Nutzer signalisiert die Wrapper-Anwendung dieses Ereignis an das MAG. Die Wrapper-Anwendung steuert den *vat*-Prozeß durch die üblichen Prozeßkontrollmechanismen.

Das für die unterschiedlichen Betriebsmodi erforderliche Verhalten der Wrapper-Anwendung wird durch eine Konfigurationsdatei gesteuert.

Der Betriebszyklus der Wrapper-Anwendung untergliedert sich in 8 Phasen:

- Auswertung der Kommandozeilen-Parameter und Einlesen der Konfiguration.
- Festlegung der Betriebsparameter.
- Kommunikationsaufbau zum MAG nebst ggf. erforderlichen Authentifizierung und Austausch der Konfigurationsparameter.
- Anforderung der Datenströme vom MAG.
- Optionale Installation eines internen Transmitters für besondere Betriebsfälle zur lokalen Umsetzung von Unicast in Multicast.
- Start des originären MBone-Werkzeugs.
- Betrieb des Transmitters, falls erforderlich.
- Nach Beendigung des MBone-Werkzeugs durch den Nutzer: Freigabe belegter Ressourcen und Signalisierung des Statuswechsels an das MAG.

Die Struktur der Wrapper-Anwendungen orientiert sich an diesen Phasen. Nachdem der Start der MBone-Anwendung erfolgt ist, übernimmt ein Dispatcher die Kontrolle über das Programm und leitet die erforderlichen Aktionen auf der Basis eines Event-Modells ein.

Im Rahmen dieser Arbeit wurden Wrapper-Anwendungen für die MBone-Werkzeuge *vat* [JM], *rat* [SHK+], *vic* [MJ95], *wb* [FJM+95] und *sdr* [Han96b] entwickelt.

Anzumerken bleibt, daß die für die Integration der Wrapper-Anwendungen in das Session-Directory *sdr* erforderlichen Plug-Ins aus wenigen Zeilen Tcl-Code bestehen, die sich von den in der *sdr*-Distribution enthaltenen Beispielen ableiten.

3.2 Grundlegende Systembausteine

3.2.1 Queuing

Die Weiterleitung von Datagrammen in paketvermittelnden Netzen erfolgt durch Zellvermittler, Rahmenvermittler, Brücken, Router oder andere Gateways. Diese Komponenten enthalten Warteschlangen, denen im Internet-Kontext bisher wenig Beachtung geschenkt wurde. Wenngleich die Warteschlangen-Verwaltung bereits seit langem ein fest etabliertes Forschungsgebiet ist, findet es erst durch die Notwendigkeit der Übertragung multimedialer Datenströme praktische Anwendung im Internet-Kontext.

Das Internet arbeitet seit seiner Entstehung nach dem "Best Effort"-Ansatz. Die Datenströme vermittelnden Komponenten arbeiten nach dem FIFO-Konzept[6] und die Steuerung des Verkehrsflusses erfolgt durch Rückkopplung zwischen Sender und Empfänger auf der Ebene des Transportprotokolls zwischen den Endpunkten des Kommunikationspfades.[7] Dieses Verfahren bewährte sich für Anwendungen wie File-Transfer (FTP) und entfernter Rechnerzugriff (Telnet). Der dabei entstehende Datenstrom wird als *elastisch* bezeichnet, da er sich auf verändernde Lastsituationen im Netzwerk einstellt, ohne seine Nützlichkeit zu verlieren.

Im Gegensatz zu elastischen Datenströmen werden UDP-Echtzeitdatenströme ohne Flußkontrolle, wie sie bei Audio- und Video-Übertragungen entstehen, als *unelastische* Datenströme bezeichnet. Bei der Übertragung von Audio- und Video-Daten ist die Minimierung der Paketlaufzeit sowie ihrer Varianz, dem Jitter, wichtig. Zudem sind insbesondere Audio-Datenströme empfindlich gegenüber Paketverlusten. Das FIFO-Queuing in den vermittelnden Einheiten ist mit diesen Datenströmen im Fall der Überlastung einzelner Pfadsegmente nicht in Einklang zu bringen:

- Unelastische Datenströme verhalten sich gegenüber elastischen Datenströmen unfair, da im Falle von Paketverlusten keine automatische Reduktion des emittierten Datenstroms erfolgt und damit die elastischen Datenströme in erhöhtem Maße ihren Verkehrsfluß reduzieren.

[6] FIFO steht für "first-in-first-out".

[7] Ein Beispiel hierfür ist das im Internet allgemein übliche *Transmission Control Protocol* (TCP), bei dem der Datenfluß des Senders an den Empfänger durch die Bestätigungen des Empfängers an den Sender gesteuert wird. Die wesentlichen Techniken sind hierbei der Slow-Start sowie die Congestion-Avoidance [Ste94, Seite 285ff., Seite 310ff.].

- Elastische Datenströme mit Burst-Verhalten, verursacht durch File-Transfer oder WWW-Anwendungen, generieren *Packet Trains*[8]. Sind die Pakete der Packet-Trains *zusammenhängend* [Ste93, Seite 27], verursachen sie bei FIFO-Queuing im Vermittler Stockungen für unelastische Datenströme. Die Folge sind längere Paketlaufzeiten und in aller Regel ein großer Jitter.

Zur Überwindung dieser Problematik wurden in der letzten Dekade Queuing-Verfahren entwickelt, deren wichtigsten Vertreter hier kurz vorgestellt werden.

3.2.1.1 Simple Priority Queuing

Bei diesem Verfahren wird der zu übertragende Paket-Datenstrom in Abhängigkeit von konfigurierbaren Parametern wie Absender- oder Ziel-Adresse, Protokoll oder Datenstrom vor der Übertragung in unterschiedliche Warteschlangen eingereiht. Diese Warteschlangen werden bei der Übertragung nach Prioritäten geordnet geleert.

Es handelt sich nach wie vor um einen FIFO-Queuing-Algorithmus, einzelne Paketströme werden jedoch bevorzugt gegenüber anderen behandelt. Eine wesentliche Entscheidung bei der Anwendung dieses Verfahrens ist die Festlegung der Prioritäten, zumal sich das Lastverhalten im Netzwerk im Laufe der Zeit i.a. verändert. Dieses Verfahren wird von dem im Internet häufig eingesetzten DVMRP-Multicast-Router *mrouted* benutzt.

Bei der Implementierung im *mrouted* werden Audio-Datenströme mit höchster Priorität behandelt, gefolgt von durch Shared-Tools verursachten Datenströmen und Video-Datenströmen. Die geringste Priorität haben unklassifizierte Daten. Dazu gehören auch Ankündigungen im MBone-Session Directory. Der PDU-Typ wird anhand des benutzten Multicast-Ports bestimmt.

Es gibt eine Queue fester Größe für jeden Tunnel. Diese Queue wird vom Übertragungssystem entsprechend der definierten Datenratenbegrenzung geleert. Gefüllt wird die Queue vom Forwarding-System. Ist die Warteschlange beim Eintreffen eines neuen Pakets vom Forwarder gefüllt, wird überprüft, ob Datenpakete niedrigerer Priorität in der Warteschlange enthalten sind. Ist das der Fall, wird das Paket mit der geringsten Priorität entfernt und verworfen. Enthält die Queue mehrere Pakete geringster Priorität, wird das zuletzt eingelagerte Paket entfernt. Die zu übertragende PDU höherer Priorität wird am Ende der Queue eingereiht.[9]

3.2.1.2 Class-Based Queuing

Auch bei diesem Verfahren wird der Paket-Datenstrom in Abhängigkeit konfigurierbarer Parameter in unterschiedliche Warteschlangen eingereiht. Die Warteschlange wird nach dem Round-Robin-Verfahren geleert: Die jeweils entnommene Menge an Paketen kann gesteuert werden,

[8] Eine Gruppe von Paketen, die einem Datenstrom angehören und zusammenhängend durch das Netz transportiert werden, werden gemäß [JR86] als Packet-Train bezeichnet.

[9] Nach Auskunft von Bill Fenner, Xerox PARC, der zur Zeit den DVMRP-Router weiterentwickelt, enthält die aktuelle Implementierung einen Fehler, der dem Autor dieser Arbeit im Rahmen einer persönlichen Kommunikation mit Bill Fenner bekannt wurde. Es wird nicht die PDU mit der kleinsten Priorität entfernt, sondern die erste PDU vom Ende der Schlange, die eine kleinere Priorität als die einzureihende PDU hat. Damit verdrängen beim DVMRP-Router Audio-PDUs im Falle einer Stauung alle anderen Typen. Shared-Tools-PDUs verdrängen Video-PDUs u.s.w..

womit sichergestellt wird, daß jede Klasse im Fall einer Stauung anteilig Daten übertragen kann. Wird das Verfahren um Timer ergänzt, kann sichergestellt werden, daß keine Klasse einen zu großen Anteil der Übertragungskapazität in Anspruch nimmt.

Innerhalb der Klassen wird nach dem FIFO-Queuing-Algorithmus gearbeitet, die Queues der Klassen werden fair zueinander behandelt. Letztlich liegt auch hier die Schwierigkeit in der Parameterisierung der Klassen. Das Verfahren wird in IP-Routern von *Internet Service Providern* (ISPs) zur Zusicherung fester Datenraten gegenüber Kunden benutzt.

3.2.1.3 Stochastic Fairness Queuing

Dieses Verfahren wurde zur automatischen Klassifizierung elastischer Datenströme und deren fairer Behandlung auf Basis der Paketraten entwickelt. Hier erfolgt die Einordnung der Pakete in verschiedene Warteschlangen in Abhängigkeit von Sende- und Zieladresse. Die Warteschlangen werden nach dem Round-Robin-Verfahren geleert, wodurch Packet-Trains aufgebrochen werden. Die Paketgröße und damit das Datenvolumen haben keinen Einfluß auf das Queuing.

3.2.1.4 Bitwise Round-Robin Fair Queuing

Dieser Nachteil wird beim *Bitwise Round-Robin Fair Queuing* überwunden, indem jedem Paket beim Eintreffen eine Sequenznummer zugeordnet wird. Sie berechnet sich aus der Sequenz-Nummer des vorhergehenden Pakets in der jeweiligen Warteschlange, vergrößert um die Größe des Pakets. Die Warteschlange wird geordnet nach den Sequenz-Nummern aller Pakete geleert. Dadurch erfolgt eine faire Aufteilung der Übertragungskapazität auf die Datenflüsse.

3.2.1.5 Weighted Fair Queuing

Mit der Einführung des *Resource Reservation Protocol* (RSVP) [Bra97] wird die gewichtete Behandlung von Datenströmen erforderlich. Dieses leistet *Weighted Fair Queuing* (WFQ). Es handelt sich um ein erweitertes Bitwise-Round-Robin-Fair-Queuing, indem den Paketen der Datenströme in Abhängigkeit der Gewichtung modifizierte Sequenznummern zugeordnet werden.

Weighted-Fair-Queuing wird bereits in Routern installiert und ist für die Nutzung von RSVP dringend anzuraten, wenngleich noch Performance-Probleme zu überwinden sind.[10]

3.2.1.6 Queuing im MBone-Access-Gateway

Bereits im heutigen Internet werden elastische und unelastische Datenströme gemischt übertragen. Solange die Übertragungskapazität ausreichend ist und Packet-Trains die Queues selten füllen, lassen sich nur wenige negative Folgen des üblichen FIFO-Queuing erkennen. Aber auch

[10] Gemäß der Beschreibung in [Cis96] wird die CPU des Routers beim Einsatz von Custom-Queuing-Verfahren, zu denen auch Weighted-Fair-Queuing zählt, mit der Verwaltung der Queue belastet, wenn die Warteschlange des Ausgabe-Interfaces überlastet ist. Die Nutzung der zentralen Router CPU zur Verwaltung der Warteschlangen ist als kritisch einzuschätzen.

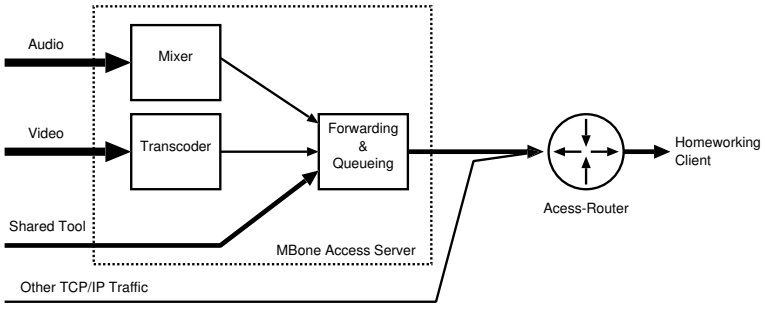

Abbildung 3.6: Datenfluß vom MBone zum Nutzer am Heimarbeitsplatz

heute sind schon Überlastsituationen im Zugangsbereich des B-WiN und in universitären Datennetzen zu erkennen. Beispiele hierfür sind 2 Mbps Anbindungen von Partnern im Norddeutschen Rechnerverbund und Institute der Universität Hannover, die ähnlich angebunden sind und zu denen DVMRP- oder PIM-Multicast-Tunnel betrieben werden. Hier konkurrieren elastische und unelastische Datenströme um die Übertragungskapazität, und es lassen sich in Überlastsituationen die beschriebenen Auswirkungen des FIFO-Queuing beobachten (Großer Jitter durch Packet-Trains und Rückwirkung unelastischer Datenströme auf elastische Datenströme). Zur weitgehenden Vermeidung dieser Effekte sind auf Multicast-Tunneln Datenratenbegrenzungen aktiv, die den Tunnel-Verkehr auf einen Teil der Übertragungskapazität der Strecke begrenzen. Dennoch tritt das Phänomen auf.

Anders stellt sich die Situation bei der Vermittlung von MBone-Konferenzen über ISDN-Kanäle zu Nutzern an Heimarbeitsplätzen dar. Aufgrund der stark begrenzten Kapazität des Übertragungskanals wird der Kanal zum größten Teil für die Übertragung der Konferenz-Datenströme ausgenutzt. Auch wenn vorgeschaltete Mixer und Transcoder die Audio- und Video-Datenströme in ihrer Datenrate begrenzen, liegt die Summe der Datenströme im Bereich der Übertragungskapazität des ISDN-Kanals. Während die Datenrate des Audio-Datenstroms nahezu konstant ist, schwankt sie beim Video-Datenstrom im Zeitverlauf.

Im Forwarding-System entstehen zusätzliche Schwankungen der Datenrate durch die Komprimierung der Paketköpfe (Header-Kompression, vgl. Abschnitt 3.2.5, Seite 97ff.). Hinzu kommen von Shared-Tools verursachte Datenströme, die in Abhängigkeit vom jeweils implementierten Transportprotokoll mehr oder weniger starken Burst-Charakter aufweisen. Das Forwarding- und Queuing-System faßt diese Datenströme zusammen und überträgt sie gebündelt zum Client. Eine Datenratenbegrenzung für den summierten Datenstrom sorgt dafür, daß das System die bereitstehende Übertragungskapazität nutzt. Die Datenratenregelung erkennt parallele, elastische Datenströme, die am MBone-Access-Server vorbei zum Nutzer am Heimarbeitsplatz fließen und sich der Kontrolle des MBone-Access-Servers (MAS) entziehen. Abbildung 3.6 skizziert den Verlauf der Datenströme.

Eine wichtige Entwurfsentscheidung ist die Frage, nach welchem Algorithmus das Queuing-System im MBone-Access-Server arbeitet. Einfache Transmitter vermitteln jeden Datenstrom isoliert und sind darauf ausgerichtet, daß vorgeschaltete Mixer und Transcoder die Datenrate so begrenzen, daß Stauungen vermieden werden. Mit Hilfe des MBone-Access-Gateway soll die Übertragungsstrecke bis zur Übertragungsgrenze ausgelastet werden. Hierfür ist der Einsatz eines Queuing-Verfahrens erforderlich.

Weighted-Fair-Queuing ist das umfassendste Queuing-Verfahren und bietet den Vorteil, daß alle Datenströme in fairer Weise übertragen werden. Durch eine entsprechende Gewichtung können Audio- und Shared-Tools-Datenströme gegenüber dem Video-Datenstrom vorgezogen werden. Nachteilig ist, daß bereits beim Einfügen von Datagrammen in die Queues eine relativ aufwendige Flow-Klassifizierung auf der Basis des Transport-Adressen-Tupels (Sendeadresse, Sende-Port, Zieladresse, Zielport) erforderlich ist. Diese Klassifizierung kann auch beim Einsatz von Schlüsseltransformationsverfahren aufgrund des langen Schlüssels leicht zu Kollisionen in der Berechnung des primären Schlüssels führen und erfordert somit auf Workstations gegenüber der Klassifizierung der Media-Streams höhere Rechenleistung.

Für den Einsatz im vorliegenden Fall scheint die Nutzung eines modifizierten Priority-Queuing mit Ähnlichkeit zur Implementierung im DVMRP-Router daher ausreichend. Bei den heute auf dem MBone üblichen Konferenzen hat die Audio-Komponente den höchsten Stellenwert. Wenn Audio-Daten verloren gehen oder verzögert eintreffen, sinkt die Sprachqualität. Dies ist zu vermeiden. Nahezu ebenso wichtig sind die Datenströme für Shared-Tools, da sie eine wichtige Grundlage für die Zusammenarbeit sind. Wenn Daten zur Übertragung anstehen, konzentrieren sich die Konferenzteilnehmer in der Regel auf die Shared-Tools und nicht auf das Video-Bild. Während unklassifizierte Datenströme vor dem Eintritt in eine Konferenz wichtig sind, sind sie dies in einer laufenden Konferenz, zumindest bei loser Koppelung der Konferenzen, nicht mehr, d.h. der vom MBone-Session-Directory verursachte Datenstrom sollte die niedrigste Priorität haben. Dies gilt nicht für Konferenz-Steuerungsprotokolle, wie das von *Confman*. Sie sind in der Klasse der Shared-Tools abzuwickeln.

Die Klassifizierung der Media-Streams erfolgt auf der Basis der Zieltransportadresse (Zieladresse, Zielport). Diese ist bereits beim Empfang eines Datagramms durch den Empfangs-Socket eindeutig festgelegt. Daher erfordert die Klassifizierung des Media-Streams keinen nennenswerten Rechenaufwand.

Die Klassifizierung der Media-Streams allein nach dem Zielport ist jedoch unzureichend, da die Lage der Ports bei Unicast-Konferenzen sich häufig nicht an den üblichen Prioritäten-Bereichen orientiert. Dieser Modus soll jedoch explizit unterstützt werden. Zudem besteht die Notwendigkeit zur Anhebung der Priorität von Steuerungsdatenströmen für Konferenzen. Daher erfolgt die Zuordnung der Priorität nicht über die Lage der verwendeten Ports, sondern explizit über einen mit dem Empfangs-Socket verbundenen Media-Stream-Kontext. Die Kontexte werden über das *MAS Control Protocol* (MASCP) zwischen den Endinstanzen des Tunnels ausgetauscht.

3.2.2 Vermeidung von Stauungen

Wie im vorangegangenen Abschnitt dargelegt wurde, können neben dem vom MBone-Access-Server emittierten Datenstrom nebenläufige Datenströme um die Übertragungskapazität der ISDN-Strecke konkurrieren. Hierbei handelt es sich in der Regel um vom Nutzer am Heimarbeitsplatz initiierte *HTTP-*, *FTP-* oder *Telnet*-Verbindungen. Zudem kann der Nutzer bei der Initialisierung des Systems versehentlich einen zu hohen Wert für die Übertragungskapazität angegeben haben. Das Ergebnis ist in beiden Fällen gleich: In den IP-Vermittlern an den Grenzen der Übertragungsstrecke entstehen Stauungen, die letztlich zu unzulässigen Verzögerungen und Paketverlusten führen können. Parallele, elastische Datenströme reagieren darauf mit einer Reduktion der Sendedatenrate, jedoch in der Regel erst nach dem Auftreten von Paketverlusten.

Obwohl der MBone-Access-Router und sein Gegenstück beim Nutzer durch den Queuing-Mechanismus auch ohne erneute Parameterisierung der Mixer und Transcoder über die Möglichkeit zur kurzfristigen Änderung der emittierten Datenrate verfügen, erkennen sie eine Stauung nicht ohne Weiteres. Es fehlt die entsprechende Rückkopplung, die auf vier Wegen integriert werden kann:

- Werden IP-Router mit Datagrammen überflutet, können sie gemäß RFC1009 (siehe dazu [BP87], Sektion 2.2.3.), eine ICMP *Source Quench* Nachricht[11] senden. Beim Empfang dieser Nachricht kann der Datenstrom in geeigneter Weise begrenzt werden.

  Der Nutzen von Source-Quench-Meldungen ist umstritten. Nach RFC1812, dem Nachfolger von RFC1009, soll ein Router keine Source-Quench-Meldungen versenden (Siehe hierzu [Bak95], Sektion 4.3.3.3.). Sie gelten als ineffizienter Weg zur Verhinderung von Stauungen und Paketverlusten. Hinzu kommt, daß es Unterschiede in den TCP/IP-Implementierungen der Endsysteme gibt: Einige leiten Source-Quench-Meldungen für UDP an die Anwendung weiter, andere nicht (Siehe hierzu [Ste94], Seite 160ff.).

  Letztlich wird die Source-Quench-Meldung erst dann versandt, wenn die Router-Queue überläuft. Es scheint daher nicht ratsam, ICMP-Source-Quench-Meldungen zur Steuerung des Datenstroms heranzuziehen.

- Für die Übertragung der Audio- und Video-Datenströme wird RTP als Transportprotokoll benutzt. Das begleitende Kontroll-Protokoll (RTCP) bietet mit seinen Sender- und Receiver-Reports die Möglichkeit zur Beobachtung der Qualität der Datenübertragung. In [SCFJ96], Sektion 6, wird angemerkt, daß diese Daten zur Steuerung adaptiver Codierungen geeignet sind. Ein MBone-Access-Server kann diese Meldungen auswerten und den Verkehrsstrom entsprechend steuern.

  Hierbei ist jedoch zu bedenken, daß Sender- und Receiver-Reports die Bestimmung von Paketverlusten und Jitter auf dem Gesamtübertragungsweg ermöglichen, nicht aber für den hier betrachteten Teil des Übertragungsweges zwischen MAS und MAG. Zudem kann nur ein Teil der Datenströme beobachtet werden. Der in der Priorität hoch angesiedelte Datenstrom für Shared-Tools läßt sich nicht beobachten. Die Nutzung der RTP/RTCP-Sender und Receiver-Reports erscheint daher nicht zur Regelung des Systems geeignet.

- Für die Signalisierung besteht zwischen den Endpunkten des Tunnels, d.h. dem MBone-Access-Server sowie seinem Gegenstück am Heimarbeitsplatz, eine TCP-basierte Verbindung. Auch diese kann zur Meldung von Stauungen benutzt werden. Es besteht jedoch die Gefahr, daß die erforderlichen Acknowledges auf der belasteten Übertragungsstrecke stark verzögert werden und damit die Nachricht erst dann eintrifft, wenn sich die Situation wieder verbessert hat oder aber die TCP-Verbindung aufgrund dauerhafter Paketverluste abgebrochen wird. Auch dieses Verfahren scheint daher ungeeignet.

- Die verschiedenen Datenströme einer Konferenz werden zum Transport durch den Tunnel zu einem Datenstrom zusammengemischt. Es ist möglich, das RTP-Konzept der Sender-Reports aufzugreifen und diese Nachrichten mit hoher Priorität ebenfalls über den Tunnel

[11] ICMP *Source Quench* Nachrichten können von einem Router oder Host erzeugt werden, wenn IP-Datagramme mit einer höheren Datenrate eintreffen als sie verarbeitet werden können.

zu transportieren. Vorteilhaft ist, daß eine kontinuierliche Kontrolle für die Übertragungsqualität vorhanden ist, Stauungen schnell erkannt werden können und eine Statistik für den gesamten Datenstrom erhoben wird. Die einzelnen RTP-Datenströme bleiben davon unberührt. Nachteilig ist die zusätzliche Reduktion der Übertragungskapazität durch die erforderlichen Reports.

Die letztgenannte Verfahren ist der beste Weg zur Realisierung der Flußkontrolle und wird daher im MBone-Access-Gateway eingesetzt.

Wenn eine Stauung erkannt wird, muß der Tunnel-Datenstrom begrenzt werden. Beim TCP-Protokoll wird hierfür der Slow-Start-Mechanismus (vgl. [Ste94], Seite 285ff.) gekoppelt mit *Congestion Avoidance* (vgl. [Ste94], Seite 310ff.) eingesetzt. Dieses Verfahren ist für den vorliegenden Fall nicht ohne weiteres zu adaptieren, da beim Slow-Start der Datenstrom kurzfristig sehr stark begrenzt wird. Dies führt unweigerlich zum Verlust einer großen Zahl von Audio-Datenpaketen und somit zu Unverständlichkeiten bei der Sprachkommunikation.

Erforderlich ist, daß im Falle moderater Paketverzögerung der Sender die Datenrate deutlich senkt, damit die bestehende Stauung schnell abgebaut wird. Sobald die Stauung abgebaut ist, sollte die Datenrate langsam wieder angehoben werden. Die eingehende Beschreibung des Regelungsverfahrens erfolgt in Abschnitt 3.2.2.4, Seite 86ff.

Mit dem Verfahren ist es möglich, die Datenrate dauerhaft im Bereich des Übertragbaren zu halten und gleichzeitig die weitgehende Kontrolle darüber zu behalten, welche der anstehenden Daten tatsächlich übertragen werden. Die wesentliche Schwierigkeit ist, festzustellen, wann eine Stauung vorliegt.

3.2.2.1 Erkennung von Stauungen

Das im Rahmen dieser Arbeit entwickelte Verfahren zur Erkennung von Stauungen basiert auf der Abschätzung von Paketlaufzeiten zwischen MAS und MAG und nicht auf der Erkennung von Paketverlusten. Dies begründet sich aus dem in Abbildung 2.15, Seite 33, zu erkennenden Queuing-Verhalten des Access-Routers: Bevor es in einer Überlastsituation zu Paketverlusten kommt, vergrößert sich die Paketlaufzeit. Daher erlaubt das hier entwickelte Verfahren die schnellere Erkennung einer Stauung als ein Verfahren, das auf der Detektion von Paketverlusten basiert.

Wenn die Uhren der Rechner, auf denen MAS und MAG ablaufen, synchron gehen, ist die Paketlaufzeit einfach zu ermitteln. Dazu wird in jedem Paket ein Zeitstempel mit der Systemzeit integriert, der den Absendezeitpunkt des Pakets markiert. Der Empfänger kann durch den Vergleich dieses Zeitstempels mit seiner Systemzeit die Laufzeit des Pakets bestimmen.

In der Praxis kann nicht unterstellt werden, daß die Uhren der Rechner hinreichend synchron sind, d.h. sie weisen eine Zeitdifferenz auf. Damit ist das beschriebene Verfahren zur Paketlaufzeitbestimmung nicht anwendbar. Daher wird bei dem hier entwickelten Verfahren nicht die absolute Paketlaufzeit, sondern die Änderung der Paketlaufzeit im Zeitverlauf zur Erkennung von Stauungen genutzt. Dabei wird angenommen, daß die Änderung der Paketlaufzeit im wesentlichen durch die Verzögerung der Pakete in den Queues der Sender am Rande der ISDN-Übertragungsstrecke verursacht wird.

In den folgenden Absätzen wird ein entsprechendes Verfahren zur Bestimmung dieses Queuing-Delays entwickelt. Ausgangspunkt ist die Darstellung der Eigenschaften von Rechneruhren, die hier von Relevanz sind. Daran schließt sich eine Betrachtung der Komponenten der Paketlaufzeit an. Darauf aufbauend wird das Verfahren zur Abschätzung des Queuing-Delays vorgestellt. Die Berechnung des Queuing-Delays erfordert die Schätzung der Übertragungskapazität des Netzabschnitts mit der kleinsten Datenübertragungsrate zwischen MAS und MAG sowie der Serialisierungszeit für Pakete auf diesem Abschnitt. Die Darstellung dieser Schätzungen erfolgt ebenfalls in einem eigenen Abschnitt. Schließlich werden die Ergebnisse aller vorhergehenden Abschnitte zu einem Berechnungsverfahren für die Ermittlung eines Schätzwertes des Queuing-Delays eines Pakets zusammengefaßt.

Eigenschaften von Rechneruhren

Jede Uhr zeigt mit ihrer Uhrzeit eine Schätzung der absoluten Zeit an. Die Differenz zwischen angezeigter und tatsächlicher Uhrzeit ist die *Zeitabweichung* der Uhr.

Rechneruhren weisen, wie auch andere Uhren, *Gangabweichungen* gegenüber der absoluten Zeit auf. Dies wird ausgedrückt durch die erste Ableitung der Zeitabweichung nach der Zeit.

$$\text{Gangabweichung:} \quad R(t) = \frac{d(T(t) - t)}{dt} = \frac{dT(t)}{dt} - 1 \tag{3.1}$$

Nach [Lam78], Seite 563, gilt für Quarzuhren eine Gangabweichung von $|R| \leq 10^{-6}$.

Wenn Rechner in Betrieb sind, wird die Systemzeit ohne weitere Maßnahmen nicht mehr durch die Quarzuhr des Rechners, sondern durch eine Software-Uhr gesteuert. Eigene Beobachtungen zeigen, daß hier mit Gangungenauigkeiten in der Größenordnung von $|R| \leq 10^{-4}$ im Tagesverlauf zu rechnen ist.

Durch die Gangabweichung einer Uhr in einem vorgegebenen Zeitintervall können danach maximal folgende Zeitabweichungen entstehen:

Zeitintervall	Zeitabweichung bei $\|R\| = 10^{-6}$	Zeitabweichung bei $\|R\| = 10^{-4}$
$1s$	$1\mu s$	$1ms$
$60s$	$60\mu s$	$6ms$
$1h$	$3.6ms$	$360ms$
$24h$	$86.4ms$	$8.64s$

Bei der Ausweitung der Betrachtung auf den Vergleich von zwei Rechneruhren wird deutlich, daß die Uhren zu jedem Zeitpunkt unterschiedliche Uhrzeiten anzeigen. Die Differenz der angezeigten Uhren wird als *Anzeigeunterschied* bezeichnet. Dieser Anzeigeunterschied wird bei frei laufenden Uhren durch einen beim Stellen der Uhren unvermeidlichen Anteil verursacht. Des weiteren verändert sich der Anzeigeunterschied im Zeitverlauf durch die Gangabweichung der Uhr über die Zeit.

Der Anzeigeunterschied der Uhren des Senders und Empfängers im MBone-Access-Gateway wird definiert als:

$$\text{Anzeigeunterschied:} \quad E(t_i) = T_R(t_i) - T_S(t_i) \tag{3.2}$$

Dabei bedeuten:

$T_R(t_i)$ Uhrzeit des Empfängers zum Zeitpunkt t_i
$T_S(t_i)$ Uhrzeit des Senders zum Zeitpunkt t_i

Demnach ist der Anzeigeunterschied eine Funktion der Zeit. Für die folgenden Betrachtungen stellt sich die Frage, in welchen Größenordnungen sich $E(t)$ ändert. Hier sind zwei Fälle zu unterscheiden:

- Die Uhr auf einem der beteiligten Rechner wird *gestellt*. Dadurch ändert sich der Anzeigeunterschied sprunghaft.
- Die Gangabweichung der Uhren verändert den Anzeigeunterschied. Im ungünstigsten Fall ergibt sich über den Zeitverlauf ein Anzeigeunterschied von

  $$\Delta E = 2 \cdot R_{max} \cdot \Delta t \tag{3.3}$$

  Für diesen Fall läßt sich ermitteln, um welchen Betrag sich der Anzeigeunterschied in einem Zeitintervall maximal ändert.

Eine nachhaltige Verbesserung bezüglich der Veränderung des Anzeigeunterschiedes im Zeitverlauf läßt sich durch den Einsatz von Regelungssystemen wie dem Network-Time-Protocol (NTP) [Mil92] erreichen. Eine Verbesserung ergibt sich bereits dann, wenn eine der Uhren über ein Zeitsynchronisierungsverfahren an einen Normalzeitsender gekoppelt wird.

Im weiteren wird davon ausgegangen, daß der Anzeigeunterschied im Zeitverlauf konstant ist. Dies ist eine Idealisierung, die nur abschnittsweise zulässig ist. Die zeitliche Ausdehnung des Abschnitts ist abhängig davon, ob ΔE gegenüber anderen Zeit-Größen nach einer *Worst-Case*-Annahme vernachlässigbar ist. Generell zulässig ist die Annahme, wenn die beteiligten Rechnersysteme durch ein Zeitsynchronisierungsverfahren gekoppelt sind, welches die Systemzeiten der Rechner monoton und quasi-statisch annähert. Dies ist jedoch nicht als Regelfall anzunehmen.

Paketlaufzeit zwischen MAS und MAG

Die Laufzeit eines Pakets zwischen Sender und Empfänger setzt sich aus verschiedenen Komponenten zusammen. Abbildung 3.7 stellt diese Komponenten für den Down-Stream dar. Hier wurde angenommen, daß sich der MAS und der Access-Router im gleichen IP-Subnetz befinden.

Die Pakete werden vom MAS ausgesendet und gelangen über eine Netzwerkverbindung hoher Datenrate zum Access-Router. Hierbei entstehen Verzögerungen durch die Serialisierung des Pakets sowie der Übertragungsdauer für ein Bit. Im Access-Router werden die Pakete weitergeleitet und in der Output-Queue des Routers ggf. zwischengespeichert. Schließlich erfolgt die Übertragung über die ISDN-Strecke. Die hier enstehende Verzögerung ergibt sich aus der Serialisierungszeit des Pakets und der Übertragungsdauer eines Bits.

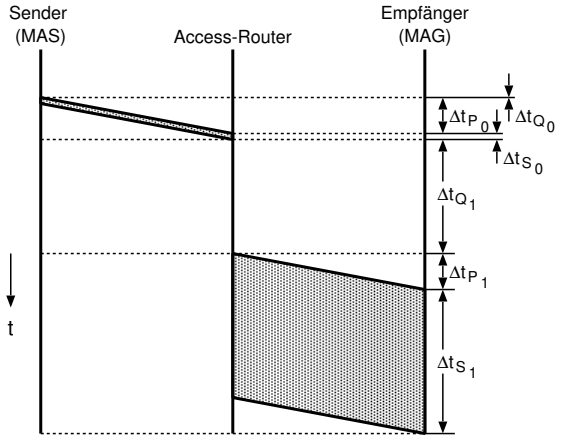

Abbildung 3.7: Komponenten der Paketlaufzeit zwischen MAS und MAG

Für ein Paket i ergibt sich daraus die Paketlaufzeit:

$$\text{Paketlaufzeit:} \quad D_i = \Delta t_{Q_{0_i}} + \Delta t_{P_0} + \Delta t_{S_{0_i}} + \Delta t_{Q_{1_i}} + \Delta t_{P_1} + \Delta t_{S_{1_i}} \tag{3.4}$$

Dabei bedeuten:

$\Delta t_{Q_{0_i}}$	Verweilzeit des Pakets in der Output-Queue des MAS.
Δt_{P_0}	Übertragungsdauer eines Bits zwischen MAS und Access-Router (LAN, $C \geq 10Mbps$) .
$\Delta t_{S_{0_i}}$	Serialisierungszeit des Pakets auf dem Netz zwischen MAS und Access-Router.
$\Delta t_{Q_{1_i}}$	Verweilzeit des Pakets in der Output-Queue des Access-Routers.
Δt_{P_1}	Übertragungsdauer eines Bits zwischen Access-Router und MAG.
$\Delta t_{S_{1_i}}$	Serialisierungszeit des Pakets auf dem Netz zwischen Access-Router und MAG.

Da die Datenübertragungskapazität des Netzes zwischen MAS und Access-Router deutlich größer als die der Übertragungsstrecke zwischen Access-Router und MAG ist, wird der Datenstrom zwischen MAS und Access-Server unzusammenhängend sein. Daraus folgt, daß die Output-Queue beim MAS in normalen Betriebssituationen i.a. leer ist. Daher wird angenommen, daß $\Delta t_{Q_{0_i}} = 0$ gilt.

Die Übertragungsdauer eines Bits zwischen MAS und Access-Router ist von der Länge des Übertragungsweges abhängig. Der Wert von Δt_{P_0} ist konstant. Gleiches gilt für Δt_{P_1}.

Die Serialisierungszeit des Pakets auf dem Netz zwischen MAS und Access-Router sowie zwischen Access-Router und MAG verhält sich proportional zur Länge des Pakets und umgekehrt proportional zur Übertragungskapazität des Mediums. Da die Übertragungskapazität des Netzes zwischen MAS und Access-Router deutlich größer als die des Netzes zwischen Access-Router und MAG ist, gilt, daß $\Delta t_{S_{0_i}}$ deutlich kleiner als $\Delta t_{S_{1_i}}$ ist.

$\Delta t_{Q_{1_i}}$ beinhaltet neben der Verweildauer des Pakets in der Output-Queue des Access-Routers auch die für das Forwarding des Pakets erforderliche Zeitspanne. Anders als die Verweildauer des Pakets in der Output-Queue des MAS ist $\Delta t_{Q_{1_i}}$ nicht zu vernachlässigen. Wenn der MAS mit einer höheren Datenrate, als der Access-Router über die ISDN-Strecke übertragen kann, sendet, kommt es in der Output-Queue des Access-Routers zur Zwischenspeicherung von Paketen. Der Wert von $\Delta t_{Q_{1_i}}$ dient als Indikator für das Vorliegen einer Stauung. Ziel ist es, $\Delta t_{Q_{1_i}}$ möglichst genau zu schätzen.

Für die weiteren Betrachtungen wird die Berechnung der Paketlaufzeit vereinfacht, indem gleichartige Komponenten zusammengefaßt werden:

$$\text{Paketlaufzeit:} \quad D_i = \Delta t_{Q_i} + \Delta t_P + \Delta t_{S_i} \tag{3.5}$$

Dabei bedeuten:

n	Anzahl der Hops zwischen MAS und MAG.
$\Delta t_{Q_i} = \sum_{k=1}^{n} \Delta t_{Q_{k_i}}$	Summierte Verweilzeit des Pakets in den Output-Queues vermittelnder Komponenten.
$\Delta t_P = \sum_{k=1}^{n} \Delta t_{P_k}$	Übertragungsdauer eines Bits auf dem Netzpfad zwischen MAS und MAG.
$\Delta t_{S_i} = \sum_{k=1}^{n} \Delta t_{S_{k_i}}$	Summierte Serialisierungsdauer des Pakets auf den Netzabschnitten zwischen MAS und MAG.

Unter der Annahme, daß das Netz, in dem sich MAS und Access-Router befinden, nicht überlastet ist, gilt, daß Δt_{Q_i} im wesentlichen durch $\Delta t_{Q_{1_i}}$ bestimmt wird, da die ISDN-Strecke die kleinste Übertragungskapazität bietet.[12]

Δt_P ist für die Dauer einer Sitzung als konstant anzunehmen, da die Summanden konstant sind.

Der wesentliche Anteil von Δt_{S_i} wird durch $\Delta t_{S_{1_i}}$ bestimmt. Grund hierfür ist, daß die Serialisierungszeit durch die Übertragungskapazität des jeweiligen Netzes bestimmt wird. Bei 10 Mbps Ethernet und ISDN stehen sie nominell im Verhältnis 1:150. Diese Aussage wurde in der Analyse, Abschnitt 2.3.1, Seite 20ff, belegt. Besonders deutlich wird dies in Abbildung 2.7, Seite 24, sichtbar.

Verfahren zur Abschätzung des Queuing-Delays Δt_{Q_i}

Die Aufgabe des hier vorgestellten Verfahrens ist, Stauungen zu erkennen. Eine Stauung macht sich durch wachsende Δt_{Q_i} bemerkbar. Wenn die Schätzung der Serialisierungszeit Δt_{S_i} möglich ist, kann gemäß Gleichung 3.5 aus der Paketlaufzeit D_i die Verweilzeit in den Queues, Δt_{Q_i}, berechnet werden. Wie eingangs bereits dargestellt wurde, ist die Ermittlung der absoluten Paketlaufzeit nicht auf einfache Weise möglich. Ursachen hierfür sind, daß der Anzeigeunterschied der Uhren (E) und die Übertragungsdauer eines Bits über den Netzpfad zwischen MAS und MAG (Δt_P) nicht bekannt sind.

Eine Lösung zur Schätzung des Queuing-Delays Δt_{Q_i} ergibt sich durch das folgende Gedankenmodell: Unmittelbar vor dem Aussenden wird jedes Paket beim Sender mit einem Zeitstempel versehen. Direkt nach dem Eintreffen beim Empfänger wird für das Paket ebenfalls ein Zeitstempel genommen. Beide Zeitstempel geben die Systemzeit des jeweiligen Rechners in hinreichender Auflösung an. Damit gilt

$$T_{R_i} - T_{S_i} = E + D_i = E + \Delta t_{Q_i} + \Delta t_P + \Delta t_{S_i} \tag{3.6}$$

[12] Generell ist die additive Zusammenfassung der Verweilzeiten der Pakete in den Ouput-Queues der vermittelnden Komponenten (Δt_{Q_i}) nur dann korrekt, wenn die Queues nach dem FIFO-Verfahren verwaltet werden und Queue-Überläufe verhindert werden.

Dabei bedeuten:

T_{R_i}	Zeitstempel beim Empfang, Systemzeit des Empfängers.
T_{S_i}	Zeitstempel beim Senden, Systemzeit des Senders.
E	Anzeigeunterschied zwischen Uhren des Empfängers und des Senders.
D_i	Paketlaufzeit des Paketes i.
Δt_{Q_i}	Verweilzeit des Pakets in den Output-Queues vermittelnder Komponenten.
Δt_P	Übertragungsdauer eines Bits auf dem Netzpfad zwischen MAS und MAG.
Δt_{S_i}	Summierte Serialisierungszeit des Pakets auf den Netzabschnitten zwischen MAS und MAG.

Wird ferner angenommen, daß stets Pakete gleicher Größe übertragen werden und die Zeitstempel von jeweils zwei Paketen i und j verglichen werden, ergibt sich:

$$(T_{R_j} - T_{S_j}) - (T_{R_i} - T_{S_i}) = D_j - D_i = (\Delta t_{Q_j} - \Delta t_{Q_i}) + (\Delta t_{S_j} - \Delta t_{S_i}) \tag{3.7}$$

Da die Pakete die gleiche Größe haben und damit die Serialisierungszeiten gleich sind, gilt folgende Gleichung für die Ermittlung der Paketlaufzeitdifferenz:

$$\text{Paketlaufzeitdifferenz:} \quad (T_{R_j} - T_{S_j}) - (T_{R_i} - T_{S_i}) = D_j - D_i = D_{ij} = \Delta t_{Q_j} - \Delta t_{Q_i} \tag{3.8}$$

Dabei bedeuten:

T_{R_j}	Zeitstempel beim Empfang des Pakets j, Systemzeit des Empfängers.
T_{S_j}	Zeitstempel beim Senden des Pakets j, Systemzeit des Senders.
T_{R_i}	Zeitstempel beim Empfang des Pakets i, Systemzeit des Empfängers.
T_{S_i}	Zeitstempel beim Senden des Pakets i, Systemzeit des Senders.
D_j	Paketlaufzeit des Paketes j.
D_i	Paketlaufzeit des Paketes i.
D_{ij}	Paketlaufzeitdifferenz der Pakete i und j.
Δt_{Q_j}	Verweilzeit des Pakets i in den Output-Queues vermittelnder Komponenten.
Δt_{Q_i}	Verweilzeit des Pakets i in den Output-Queues vermittelnder Komponenten.

Mit $j = i + 1$ entspricht diese Formel der in [SCFJ96], Abschnitt 6.3.1, angegebenen Vorschrift für die Bestimmung des Zwischenankunfts-Jitter. Bei dem hier vorgestellten Verfahren dient die Paketlaufzeitdifferenz der Berechnung unterschiedlicher Verweilzeiten von Paketen gleicher Größe in den Router-Queues und bildet die Grundlage für die Erkennung von Stauungen.

Der dem Verfahren zugrunde liegende Gedanke ist, die absolute Verweilzeit eines Pakets i in den Router-Queues zu bestimmen, indem der aktuelle D_i-Wert mit einem minimalen Wert für D verglichen wird. Ein minimales D entsteht, wenn ein Paket übertragen wird, ohne in Router-Queues zu verweilen.

Da das D_i eines einzelnen Pakets nicht bestimmt werden kann, andererseits aber ein Maß für die Paketlaufzeit der einzelnen Pakete für die folgenden Berechnungen erforderlich ist, wird im folgenden mit der Differenz zwischen Empfangs- und Sendezeitstempel gerechnet.

$$\text{Zeitstempeldifferenz:} \quad L_i = E + D_i = T_{R_i} - T_{S_i} \tag{3.9}$$

Mit der Zeitstempeldifferenz läßt sich ein zur Paketlaufzeit proportionales Minimum bilden:

$$L_{min_i} = \min_{k \in N_i} L_k \quad \text{mit} \quad N_i = \{k | k \in \mathsf{N} \wedge k < i\} \tag{3.10}$$

Der L_i-Wert des aktuellen Pakets wird im Betrieb mit L_{min_i} verglichen. Wächst $L_i - L_{min}$ über einen zu konfigurierenden Schwellwert, liegt eine Stauung vor.

Diese Annahme ist allerdings nur dann zulässig, wenn stets Pakete gleicher Größe übertragen werden oder die Serialisierungszeit Δt_{Q_i} gegenüber der Queue-Verweilzeit Δt_{S_i} vernachlässigbar ist. Tatsächlich werden Pakete unterschiedlicher Größe übertragen und die Serialisierungszeit ist nicht vernachlässigbar. Um Δt_{Q_i} bestimmen zu können, ist die Schätzung von Δt_{S_i} erforderlich. In den folgenden Absätzen wird daher ein Verfahren zur Berechnung von Δt_{S_i} entwickelt, indem die Übertragungskapazität geschätzt wird und darauf aufbauend die Serialisierungszeit berechnet wird. Der Abschnitt schließt mit der Vorstellung einer Berechnungsformel zur Schätzung der Verweilzeit eines Pakets in den Warteschlangen der Output-Queues vermittelnder Elemente auf dem Netzwerkpfad zwischen MAS und MAG.

Schätzung der Übertragungskapazität und der Serialisierungszeit

Es kann unterstellt werden, daß der Betrieb des MBone-Access-Gateways über nur eine Verbindung mit geringen Übertragungsraten erfolgt[13]. Zudem kann angenommen werden, daß das System im Laufe der Zeit die maximale auf dem Netzwerkpfad mögliche Übertragungsleistung ausschöpft.

Der Protokoll-Overhead ist in Größenordnungen bekannt: jedes Paket erfordert die Übertragung der Payload, des UDP-Headers (8 Byte), des IP-Headers (20 Byte) und 7 Byte PPP-Protokoll-Overhead. Werden Übertragungssysteme mit anderen Parametern benutzt, sind diese individuell zu konfigurieren.

Wenn keine PDU eines dem MBone-Access-Gateways unbekannten Datenstroms zwischen zwei Paketen $i-1$ und i übertragen wurde und die Paketsendefrequenz im Grenzbereich der Übertragungskapazität liegt, berechnet sich die Übertragungskapazität zu

$$\text{Übertragungskapazität:} \quad C_i = \frac{P_i + 35}{T_{R_i} - T_{R_{i-1}}} \ \text{[Byte/s]} \tag{3.11}$$

Dabei bedeuten:

T_{R_i}	Zeitstempel beim Empfang des Pakets i, Systemzeit des Empfängers.
$T_{R_{i-1}}$	Zeitstempel beim Empfang des Pakets $i-1$, Systemzeit des Empfängers.
P_i	Größe der UDP-Payload des Pakets i in Bytes.

Dieser Schätzwert ist ein Maß für die Übertragungskapazität des Kanals in $\frac{Byte}{s}$.

Da im Betrieb nicht unterstellt werden kann, daß stets optimale Randbedingungen vorliegen, wird C_i variieren. Ein stabiler Schätzer für die Übertragungskapazität ergibt sich durch die Bildung des Maximums von C_i über alle vorangegangenen Pakete.

[13] Der Betrieb des Systems über verkettete ISDN-Strecken verbietet sich, da die sich addierenden Serialisierungszeiten in Größenordnungen gelangen, die eine verbale Verständigung behindern.

Maximum der Übertragungskapazität:

$$C_{max_i} = \max_{k \in N_i} C_k \quad \text{mit} \quad N_i = \{k | k \in \mathsf{N} \wedge k < i\} \tag{3.12}$$

Mit C_{max_i} läßt sich die Serialisierungszeit des Pakets i abschätzen.

$$\text{Serialisierungszeit:} \quad \Delta t_{S_i} = \frac{P_i + 35}{C_{max_i}} \text{ [s]} \tag{3.13}$$

Schätzung des Queuing-Delays

Die Schätzung der Serialisierungszeit (Gleichung 3.13) und der Übertragungskapazität (Gleichung 3.12) gestattet letztlich die Schätzung des Queuing-Delays für das Paket i.

$$\begin{aligned} \text{Queuing-Delay:} \quad \Delta t_{Q_i} &\approx (L_i - L_{min_i}) - (\Delta t_{S_i} - \Delta t_{S_{min_i}}) \\ &= L_i - L_{min_i} + \frac{P(L_{min_i}) - P_i}{C_{max_i}} \end{aligned} \tag{3.14}$$

Dabei bedeuten:

L_i	Zeitstempeldifferenz des Pakets i.
L_{min_i}	Minimale Zeitstempeldifferenz.
$\Delta t_{S_{min_i}}$	Serialisierungszeit des Pakets, das L_{min_i} ergeben hat.
Δt_{S_i}	Serialisierungszeit des Pakets i.
$P(L_{min_i})$	Anzahl der UDP-Payload-Bytes des Pakets, das L_{min_i} ergeben hat.
P_i	Anzahl der UDP-Payload-Bytes des Pakets i.
C_{max_i}	Maximum der geschätzten Übertragungskapazität.

Die sich ergebende Schätzung für Δt_{Q_i} hat integrativen Charakter. Nachdem sich die grundlegenden Werte L_{min_i} und C_{max_i} stabilisiert haben, wird sich Δt_{Q_i} stetig ändern. Daher ist in den folgenden Betrachtungen die ausschließliche Berücksichtigung des aktuellen Wertes Δt_{Q_i} opportun.

Zu berücksichtigen bleibt, daß dieses Berechnungsverfahren für die Abschätzung des Queuing-Delays davon ausgeht, daß der Anzeigeunterschied zwischen den Rechneruhren von MAS und MAG konstant ist. Wie auf Seite 72 gezeigt wurde, ist diese Annahme nur zeitabschnittsweise zulässig. Durch Gangungenauigkeiten der Uhren verändert sich der Anzeigeunterschied E und damit die Zeitstempeldifferenz L (Gleichung 3.9, Seite 76). Damit der entstehende Fehler über den Zeitverlauf hinweg in tolerierbaren Grenzen bleibt, muß L_{min_i} abschnittsweise neu bestimmt werden.

Dazu werden in den Empfangsmoduln von MAS und MAG jeweils zwei Register für die Speicherung von L_{min_i} und $P(L_{min_i})$ vorgesehen. Nach einer kurzen Initialisierungsphase, auf die im folgenden Absatz eingegangen wird, dient ein Registersatz der Berechnung von Δt_{Q_i}, während der andere Registersatz im Stand-By-Mode nach vorheriger Initialisierung mitläuft.

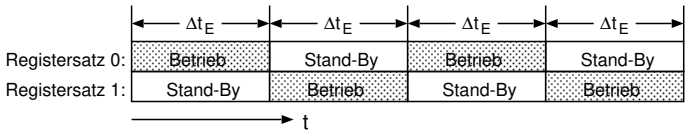

Abbildung 3.8: Phasenwechsel zwischen Betriebs- und Stand-By-Phasen der Registersätze für L_{min_i}

Nach Ablauf eines konfigurierbaren Zeitabschnitts Δt_E werden die Rollen der Registersätze getauscht. Bei jedem Rollenwechsel wird das L_{min_i}-Register des neuen Stand-By-Registersatz mit einem hohen Wert initialisiert, so daß im Zeitverlauf dieser Phase ein neuer L_{min_i}-Wert berechnet wird. Mit diesem Verfahren wird den Gangunterschieden zwischen den Uhren von MAS und MAG begegnet. Abbildung 3.8 zeigt den Phasenwechsel zwischen Betriebs- und Stand-By-Phasen der Registersätze.

Die Wahl des Zeitabschnitts Δt_E hat nachhaltigen Einfluß auf die Genauigkeit des Verfahrens. Wird er zu groß gewählt, verursacht der Gangunterschied der Uhren eine Veränderung des Anzeigeunterschieds, der bei der Registerumschaltung durch einen Sprung der minimalen Zeitstempeldifferenz (L_{min_i}) einen Regelvorgang auslöst. Wird das Zeitintervall zu klein gewählt, kann eine durch nebenläufige Datenströme verursachte Stauung im Access-Router oder im MAG dazu führen, daß in der Stand-By-Phase des Registersatzes kein Paket den Tunnel bei leeren Output-Queues passiert. In diesem Fall wird die während der Stand-By-Phase geschätzte minimale Zeitstempeldifferenz (L_{min_i}) Paketverögerungen in Warteschlangen beinhalten. Die Folge kann ein Sprung von L_{min_i} beim Phasenwechsel sein, der einen Regelvorgang des Systems auslöst.

Letztlich bleibt die Wahl des Parameters Δt_E dem Betreiber des MAG vorbehalten. Sein Wert ist davon abhängig, ob die Uhren des MAS-Rechners und des MAG-Rechners frei laufen oder durch ein Zeitsynchronisierungsverfahren gekoppelt sind und wie groß im Fall frei laufender Uhren die Gangunterschiede sind. Selbst bei Uhren mit hohen Gangabweichungen ist ein Wert von ca. 5 Minuten für Δt_E geeignet. Auf der anderen Seite gilt es abzuschätzen, wie lange ein Stauungen verursachender, nebenläufiger Datenstrom vom Nutzer akzeptiert wird, bevor er die Teilnahme an der Konferenz beendet. Auch hier scheint ein Δt_E-Wert von ca. 5 Minuten akzeptabel zu sein.

Die Schätzung der Übertragungskapazität des Netzwerkpfades (Gleichung 3.12) (C_{max_i}) zwischen MAS und MAG berücksichtigt ausschließlich die Zeitstempel aufeinanderfolgender Tunnel-Pakete. Die Zeitspanne zwischen dem Eintreffen von zwei aufeinanderfolgender Pakete ist so klein, daß der Gangunterschied der Uhren selbst im ungünstigsten Fall[14] nur marginale Fehler verursacht. Eine in Phasen gegliederte Neubestimmung von C_{max_i} ist daher nicht erforderlich.

Bevor auf die mit der Übertragung der Zeitstempel und der Erstellung der Tunnel-Reports verbundenen Fragestellungen sowie auf die Regelung des Systems eingegangen wird, wird zunächst die Zuverlässigkeit der Schätzung für Δt_{Q_i} untersucht. Sie ist eine grundlegende Voraussetzung für die weitere Diskussion.

[14]Dieser Fall ist charakterisiert durch $\Delta E = 2 \cdot R \cdot \Delta t$, vergl. Gleichung 3.3.

3.2.2.2 Zuverlässigkeit der Stauungserkennung

Wie bereits ausgeführt, kann das Queuing-Delay einzelner Pakete, Δt_{Q_i}, nur näherungsweise ermittelt werden. Für den stabilen Betrieb ist es erforderlich, die Zuverlässigkeit dieses Wertes zu untersuchen. Dazu werden exemplarisch folgende Fälle betrachtet:

1. Das Verhalten direkt nach der Etablierung des Tunnels.
2. Die Summe aller Flows unterschreitet die Kapazität des Tunnels.
3. Die Summe aller Flows übersteigt die Kapazität des Tunnels.
4. Konkurrierende TCP-Verbindungen überlagern den Verkehrsfluß.
5. Die Systemzeit auf einem der beteiligten Rechnersysteme wird verstellt.

Die Analyse dieser Betriebsfälle soll zeigen, ob das entwickelte Verfahren dazu geeignet ist, Stauungen zu erkennen und hinreichend zuverlässige Daten zur Regelung des Systems bereitzustellen. Dabei wird im Vorgriff auf die Datenraten-Regelung des Systems eingegangen, deren umfassende Beschreibung erst in Abschnitt 3.2.2.4, Seite 86ff, erfolgt.

Verhalten direkt nach der Etablierung des Tunnels

Bei der Etablierung des Tunnels wird das Register für die minimale Zeitstempeldifferenz (Gleichung 3.10) (L_{min_i}) mit dem größten darstellbaren Wert belegt. Die Register für die geschätzte Übertragungskapazität des Netzwerkpfades zwischen MAS und MAG (Gleichung 3.12) (C_{max_i}) wird mit dem Wert 0 initialisiert. Der Sender limitiert die Übertragungsrate entsprechend eines vom Nutzer am MAG zu konfigurierenden Wertes. Im Fall einer ISDN-Strecke sollten 64 kbps konfiguriert werden.

Die dynamische Regelung des Datenstroms kann durch den Sender erst dann erfolgen, wenn als Entscheidungsgrundlage ein Tunnel-Report vom Empfänger erhalten wurde. Der Empfänger verzögert die Erstellung und Übermittlung des Reports, bis er $16 \ldots 32$ Pakete mit Zeitstempeln vom Sender empfangen hat. Im weiteren sind 3 Fälle zu unterscheiden:

- Der Sender hat initial eine zu hohe Datenrate übertragen. Unter der Einschränkung, daß keine konkurrierende Datenströme über den "Slow-Speed-Serial-Link" in gleicher Richtung übertragen werden, repräsentiert C_{max_i} die Übertragungskapazität des Tunnels bereits nach dem Empfang weniger Datenpakete recht genau. Grund dafür ist, daß die zu hohe Sendedatenrate dafür sorgt, daß die volle Übertragungskapazität des "Slow-Speed-Serial-Link" genutzt wird.

  Die Zeitstempeldifferenz L_{min_i} ist schon nach dem Empfang des ersten Pakets hinreichend genau, da sich bei der Übertragung des ersten Pakets noch keine Warteschlange aufgebaut hat.

  Der erste Report vom Empfänger zum Sender beinhaltet daher einen zuverlässigen Wert für das Queuing-Delay, Δt_{Q_i}. Der Wert ist aufgrund der Überflutung der Übertragungsstrecke > 0.

- Der Sender hat initial eine der Tunnel-Kapazität entsprechende Datenrate übertragen. Hier gelten die gleichen Aussagen wie im vorhergehenden Fall. Einziger Unterschied ist, daß der Wert von $\Delta t_{Q_i} \approx 0$ ist, da keine Warteschlange aufgebaut wurde.[15]
- Der Sender hat initial mit einer kleineren Datenrate als der Tunnelkapazität gesendet. In diesem Fall ist L_{min_i} hinreichend genau, C_{max_i} hat allerdings einen zu kleinen Wert, da der Paketstrom über den Tunnel unzusammenhängend ist. Δt_{Q_i} ist ≈ 0, da keine Warteschlange aufgebaut wurde.

Im ersten Fall muß der Sender beim Empfang des Tunnel-Reports regelnd eingreifen. Der zweite Fall ist der gewünschte Fall. Es ist keine Regelung erforderlich. Problematisch ist der dritte Fall. Er läßt sich für den Sender nicht vom zweiten Fall unterscheiden, trotzdem könnte die Datenrate durch den Sender angehoben werden. Hier gilt jedoch die Vereinbarung, daß die durch den Nutzer konfigurierte Datenratenlimitierung in jedem Fall Priorität hat. Eine automatische Erhöhung über diesen Wert hinaus erfolgt nicht.

Summe der Flows unterschreitet die Tunnel-Kapazität

In diesem Fall liefert L_{min_i} eine richtige Schätzung für die Zeitstempeldifferenz, da keine Warteschlangen aufgebaut wurden. C_{max_i} ist kleiner als die tatsächliche Übertragungskapazität des Tunnels zwischen MAS und MAG. Da keine Warteschlange entstanden ist, ist auch $\Delta t_{Q_i} \approx 0$.

Wenn der Wert von Δt_{Q_i} über einen Zeitraum in der Größenordnung einiger Minuten ≈ 0 bleibt, kann der Sender die Transcoder und Mixer so rekonfigurieren, daß ein stärkerer Datenstrom entsteht. Die Datenrate darf allerdings nicht über die vom Benutzer konfigurierte Begrenzung hinaus angehoben werden.

Summe der Flows übersteigt die Tunnel-Kapazität

Entwickelt sich die Stauung aus einer Situation heraus, in der die Gesamtdatenrate unter der Tunnelübertragungskapazität liegt, ist L_{min_i} eine zuverlässige Schätzung für die minimale Zeitstempeldifferenz. C_{max_i} ist kleiner als die tatsächliche Übertragungskapazität des Tunnels. Während sich die Überflutung entwickelt, steigt C_{max_i} auf den tatsächlichen Wert der Tunnelübertragungskapazität. Die Erkennung wachsender Δt_{Q_i} ist sichergestellt. Beim Empfang des nächsten Tunnel-Reports kann der Sender die Übertragungsrate entsprechend reduzieren.

Dauert die Überlastsituation über den Zeitraum mehrerer Δt_E-Intervalle[16] hinweg an, ermittelt der Empfänger eine Schätzung für die minimale Zeitstempeldifferenz (L_{min_i}), die Queuing-Delays beinhaltet. Während die Schätzung für die Tunnelübertragungskapazität C_{max_i} dem tatsächlichen Wert entspricht, ergibt die Berechnung des Queuing-Delays Δt_{Q_i} falsche Werte. Durch die sich ergebenden Paketverluste[17] wird die Überlastsituation für den Sender dennoch offensichtlich. Er reagiert darauf durch die Reduktion des Datenstroms.

[15] Der Wert von Δt_{Q_i} ist nicht genau 0, da durch Randeffekte in Grenzen variierende Paketlaufzeiten entstehen. Abbildung 2.6 auf Seite 22 zeigt diesen Effekt am Beispiel der Ermittlung von Round-Trip-Times.

[16] Zur Erläuterung von ΔT_E siehe Abbildung 3.8, Seite 79.

[17] Die Erkennung von Paketverlusten wird in Abschnitt 3.2.3, Seite 93ff, behandelt.

Konkurrierende TCP-Verbindungen überlagern den Verkehrsfluß

Diese Situation entsteht, wenn der Nutzer am Heimarbeitsplatz durch nebenläufige Werkzeuge, wie einen World-Wide-Web-Browser, einen TCP-Datenstrom initiiert.

Der konkurrierende Datenstrom wird durch wachsende Queuing-Delays (Δt_{Q_i}) erkannt. Das Regelsystem reagiert auf Senderseite nach dem Empfang eines Tunnel-Reports mit einer Begrenzung der Datenrate für den Tunnel-Datenstrom. Der konkurrierende TCP-Datenstrom reagiert darauf hin elastisch und steigt an. Daher steigt das Queuing-Delay Δt_{Q_i} weiterhin an oder bleibt unverändert hoch. Ohne weitere Maßnahmen wird der Tunnel-Datenstrom gänzlich durch den TCP-Datenstrom verdrängt.

Das Regelungsverfahren berücksichtigt diesen Fall und reagiert weitgehend elastisch auf den konkurrierenden TCP-Datenstrom. Eine weitergehende Betrachtung dieser Situation erfolgt im Abschnitt 3.2.2.4, Seite 86ff.

Aus der elastischen Reaktion des Tunnel-Systems verändern sich die Parameter zur Erkennung von Stauungen wie folgt:

- Der Schätzwert für die Übertragungskapazität des Tunnels, C_{max_i}, hat einen den Gegebenheiten entsprechenden Wert, sofern der konkurrierende TCP-Datenstrom erst nach der Initialisierung des Tunnels etabliert wurde. Andernfalls wird sein Wert kleiner als die tatsächliche Tunnelübertragungskapazität sein.
- Die minimale Zeitstempeldifferenz (L_{min_i}) beinhaltet einen Teil des Queuing-Delays, sofern die Überlastsituation über mehrere Δt_E-Intervalle anhält.

Diese Fehler führen dazu, daß das berechnete Queuing-Delay (Δt_{Q_i}) in einzelnen Überlastsituationen einen zu kleinen Wert hat. In diesen Fällen treten jedoch Paketverluste auf, die das Regelungssystem die Stauung erkennen lassen. Zur weiteren Reduktion des Tunnel-Datenstroms kommt es jedoch nur dann, wenn nicht bereits zuvor der Tunnel-Datenstrom minimiert wurde.

In diesem Unterabschnitt wurde ausschließlich die Überlagerung des Tunnel-Datenstroms durch TCP-Datenströme betrachtet. Dies ist zum einen der wahrscheinlichere Fall, zum anderen aber auch der interessantere Fall, da TCP durch den Slow-Start-Mechanismus [Ste94, Seite 285 ff.] einen Datenstrom verursacht, der elastisch auf Stauungen reagiert. Dennoch ist zu berücksichtigen, daß auch UDP-Datenströme dem Tunnel-Datenstrom überlagert werden können. Da der Tunnel-Datenstrom elastisch auf alle konkurrierende Datenströme reagiert, entsteht dabei eine Situation, die mit dem eingehend betrachteten Fall vergleichbar ist.

Verstellung der Systemzeit auf einem Rechnersystem

Die grundlegenden Eigenschaften von Rechneruhren wurden bereits auf Seite 72 vorgestellt. Das hier entwickelte Verfahren zur Erkennung von Stauungen erfordert, daß der Anzeigeunterschied zwischen MAG und MAS näherungsweise konstant bleibt. Diese Forderung ist mit den heute üblichen, frei laufenden Uhren jedoch nicht über ein Zeitintervall von Stunden oder Tagen mit

hinreichender Genauigkeit zu erfüllen. Daher beinhaltet das Verfahren zu Stauungserkennung eine zeitabschnittsweise Neubestimmung der minimalen Zeitstempeldifferenz (L_{min_i}).

Ein durch Gangunterschiede verursachter Anzeigeunterschied der Rechneruhren führt aber nicht nur bei der hier betrachteten Datenvermittlung zu Schwierigkeiten: Nahezu alle Anwendungen in Rechnernetzen erfordern für den zuverlässigen Betrieb die abschnittsweise Synchronisierung der Uhren, um Gangunterschiede auszugleichen. Im Zeitverlauf wurden verschiedene Verfahren zur Lösung dieses Problems für unterschiedliche Zwecke entwickelt. Eine Übersicht ist [Kar93] zu entnehmen.

Die übliche Forderung an die Synchronisierung ist, daß die Uhren der Rechner in einer Arbeitsgruppe "ungefähr synchron" sein sollen. Die Zeit soll der Wanduhr entsprechen und die Rechneruhren sollten maximale Anzeigeunterschiede im Sekundenbereich aufweisen. Zur Synchronisierung der Zeit werden einfache Programme wie *rdate* [UNIb, Pos83] oder *netdate* [UNIa, Pos83] benutzt, die die Uhr eines Rechners auf die Uhrzeit eines anderen Rechners stellen. Die Programme werden mehrfach stündlich aufgerufen. Damit stimmen die Zeiten "gut" überein. Das Problem bei diesem Verfahren ist das "Stellen" der Uhr. Eilt die Uhr des betrachteten Rechners voraus, wird die in [Lam78] beschriebene *Clock Condition*[18] verletzt. Letztlich verletzt jedes Stellen der Uhr die Kontinuität der Zeit auf dem System. Obgleich die Nachteile des Verfahrens evident sind, ist es gängige Praxis.

Dieses offensichtliche Problem löst beispielsweise der Einsatz des *Network Time Protocol* (NTP) [Mil92]. Aufgrund des Installationsaufwandes wird NTP jedoch selten eingesetzt, insbesondere an Heimarbeitsplätzen.

Somit besteht das Risiko, daß im Laufe einer Konferenz die Uhr eines der beteiligten Rechner gestellt wird. Die Rückwirkungen auf das System variieren in Abhängigkeit vom Zeitpunkt und dem Betrag der sich ergebenden Zeitänderung. In jedem Fall wird die Aussagekraft des Schätzwertes für die minimale Zeitstempeldifferenz (L_{min_i}), die maximale Übertragungskapazität des Tunnels (C_{max_i}) und damit auch des Queuing-Delays (Δt_{Q_i}) nachhaltig beeinflußt. Wäre die Änderung der Uhrzeit sicher zu erkennen, könnten die aktuellen Werte als ungültig erklärt und neue Werte erstellt werden. Leider ist dies nicht möglich. Eine Verbesserung ergibt sich durch die Neubestimmung von L_{min_i} in regelmäßigen Intervallen, trotzdem existieren Zeitbereiche, in denen die Regelung auf falschen Randwerten basiert. Wenngleich der Schätzwert für die maximale Übertragungskapazität des Tunnels (C_{max_i}) weitgehend robust gegen monotone Änderungen der Systemzeit ist, führen sprunghafte Änderungen in aller Regel zu Fehlern, insbesondere dann, wenn es sich um betragsmäßig kleine Sprünge handelt.[19]

Da die Lösung des Problems innerhalb des Systems nicht möglich ist, kann nur der Hinweis gegeben werden, daß für den Betrieb des MBone-Access-Gateways die beteiligten Rechner (MAG-Rechner und MAS-Rechner) entweder unter Nutzung eines Verfahrens wie NTP synchronisiert werden[20], welches einen monotonen und stetigen Uhrengang sicherstellt, oder auf die Synchronisierung der Rechneruhren während der Laufzeit des MBone-Access-Gateways verzichtet wird

[18] Die Clock-Condition besagt für zwei beliebige Ereignisse a und b: Wenn $a \rightarrow b$, dann muß $C(a) < C(b)$ sein.

[19] Ein betragsmäßig großer Sprung zeigt sich in einer Fehlberechnung von C_i, die oberhalb der vom Nutzer eingestellten Übertragungskapazität liegt, oder deren Wert so klein ist, daß er ohne Einfluß für die Bildung von C_{max_i} ist. Ein betragsmäßig kleiner Zeitsprung kann jedoch zu Fehlern bei der Bestimmung der Serialisierungszeit führen, die wiederum Einfluß auf auf die Schätzung des Queuing-Delays hat.

[20] Damit ist gemeint, daß auf beiden Rechnern dauerhaft eine *xntpd*-Instanz läuft.

und die Rechneruhren frei laufen.

Zusammenfassung

Die vorstehenden Betrachtungen zeigen, daß das beschriebene Verfahren grundsätzlich zur Bestimmung der Verweildauer der Pakete in den Queues der Router geeignet ist. Der Schätzwert für die Übertragungskapazität des Tunnels (C_{max_i}) kann kleiner als die tatsächliche Tunnel-Übertragungskapazität sein, auf die Zuverlässigkeit der Schätzung des Queuing-Delays (Δt_{Q_i}) hat dies jedoch keinen negativen Einfluß. Zu Fehlschätzungen des Queuing-Delays kommt es, wenn eine Stauung über ein Zeitintervall $> \Delta t_E$ anhält. In diesem Fall müssen die für den Sender erkennbaren Paketverluste zur Regelung des Systems herangezogen werden. Das zwischenzeitliche Stellen der Uhren der beteiligten Rechner kann ebenfalls zu Fehlschätzungen der Parameter führen. Daher sollte zur Uhren-Synchronisierung ein Verfahren gewählt werden, daß einen monotonen und stetigen Zeitverlauf garantiert. Andernfalls sollte auf die Synchronisierung verzichtet werden.

3.2.2.3 Übertragung der Zeitstempel

Das entwickelte Verfahren zur Erkennung von Stauungen in Routern auf dem Übertragungsweg basiert auf der Übertragung der Systemzeit in jedem über den Tunnel übertragenen Paket. Wenngleich die regelmäßige Übertragung der Zeitstempel für eine schnelle Reaktion des Systems wichtig ist, bedeutet sie, insbesondere bei der Übertragung kleiner Pakete, einen nicht vernachlässigbaren Overhead. In diesem Abschnitt wird daher untersucht, wie dieser Overhead zu minimieren ist und wie die Zeitstempel einfach und transparent in den Paketdatenstrom zwischen Sender und Empfänger zu integrieren sind.

Format und Auflösung der Zeitstempel

In den bisherigen Betrachtungen wurde davon ausgegangen, daß jeweils die Systemzeit des sendenden Rechners übertragen wird. Diese Systemzeit wird bei den hier geforderten Auflösungen durch den System-Aufruf *gettimeofday* [Ste92, Seite 155] ermittelt. Das Ergebnis ist eine Struktur vom Typ *timeval*. Sie besteht aus zwei 32-Bit breiten Integer-Werten, die die aktuelle Systemzeit in Mikrosekunden-Auflösung seit dem 01. Januar 1970, $00:00:00$ angeben.

Das einfachste Verfahren besteht in der Übertragung dieses Zeitstempels. Nachteilig ist, daß 64 Bit für die Übertragung des Zeitstempels erforderlich sind. Dies begrenzt die für Nutzdaten verfügbare Übertragungskapazität unnötig, da Timer-Auflösungen im Bereich von Millisekunden ausreichend sind. Zudem ist eine Epoche von 136 Jahren für diese Anwendung nicht erforderlich.

Da das vorgeschlagene Verfahren nicht auf der absoluten Systemzeit, sondern auf der Bildung von Zeitdifferenzen basiert, können alternativ diskrete Uhren mit reduzierter Auflösung und begrenztem Wertevorrat eingeführt werden. Geeignet erscheinen 24-Bit Werte für die Angabe der Uhrzeit. Bei einer Auflösung von einer Millisekunde ergibt sich eine Epoche von 4.6 Stunden.

Bei der Initialisierung des Systems registrieren die Rechner an den Tunnelenden die aktuelle, lokale Systemzeit als *timeval*. Zur Bildung eines Zeitstempels wird die aktuelle Systemzeit ermittelt und die Differenz zur Systemzeit während der Initialisierung in Millisekunden gebildet. Das 24-Bit Modulo dieser Differenz wird als Zeitstempel übertragen.

Der Empfänger initialisiert eine *timeval*-Struktur sowie ein Zeitstempel-Register mit dem Wert Null. Beim Eintreffen eines Zeitstempels wird die Zeitdifferenz zwischen Zeitstempel-Register und aktuellem Zeitstempel gebildet und der Wert in der *timeval*-Struktur um die sich ergebenden Anzahl von Millisekunden erhöht. Das Zeitstempel-Register ist erforderlich, um einen Überlauf der Zeitstempel beim Sender zu erkennen und korrekt handhaben zu können. Wenn Pakete nicht in der Sendereihenfolge eintreffen, werden die darin enthaltenen Zeitstempel ignoriert und nicht zur Schätzung der Tunnelübertragungskapazität C_{max_i} herangezogen.

Frequenz der Zeitstempel im Paketstrom

Weitere Einsparungen lassen sich erzielen, indem nicht jedes über den Tunnel übertragene Paket einen Zeitstempel enthält. Dies ist unkritisch bezüglich der Bestimmung von L_i. Einfluß hat diese Optimierung jedoch auf die Bestimmung von C_i, da die einfache Formel

$$C_i = \frac{P_i + 35}{T_{R_i} - T_{R_{i-1}}} \text{ [Byte/s]}$$

nicht mehr anwendbar ist. Sie wird durch den folgenden Term ersetzt. Seien T_{R_i} und T_{R_j} die Empfangszeitpunkte zweier Pakete, wobei $j - i \geq 1$ gilt. Dann ergibt sich

$$C_j = \frac{\sum_{n=i}^{j} P_n + 35}{T_{R_j} - T_{R_i}} \text{ [Byte/s]} \tag{3.15}$$

und

$$C_{max_j} = \max_{k \in N_j} C_k \quad \text{mit} \quad N_j = \{k | k \in \mathsf{N} \wedge k < j\} \tag{3.16}$$

Obwohl die Berechnung von C_{max_j} problemlos durchzuführen ist, bleibt zu berücksichtigen, daß durch die Summenbildung eine Ungenauigkeit entsteht, die das Verfahren an sich nicht negativ beeinflußt, die Zeitdauer der Stabilisierung von C_{max} jedoch vergrößert. Die Ungenauigkeit resultiert aus der größeren Wahrscheinlichkeit, daß der Tunnel-Datenstrom zwischen der Übertragung der Zeitstempel nicht zusammenhängend ist.

Mit dem Ziel, die Stabilisierungszeit der Kennwerte klein zu halten, wird direkt nach der Initialisierung des Systems eine Folge von Paketen mit Zeitstempeln übertragen. Nach Abschluß dieser Phase orientiert sich die Frequenz der Zeitstempel an der Menge der übertragenen Daten. Etwa alle 1500 bis 3000 Byte wird ein Zeitstempel in den Paketstrom eingefügt.

Alternativ könnte eine zeitgesteuerte Übertragung der Zeitstempel realisiert werden. Dies hätte allerdings den Nachteil, daß in Zeiträumen, während derer keine Nutzdaten zur Übertragung anstehen, trotzdem Pakete mit Zeitstempeln zu übertragen wären. Dadurch würde der Tunnel unnötig mit Datenpaketen belastet. Zudem wären diese Datenpakete ausschließlich für die Schätzung des Queuing-Delays nutzbar, da für die Schätzung der Übertragungskapazität des Tunnels ein zusammenhängender Datenstrom erforderlich ist. Daher wird hier eine datenvolumen-orientierte Übertragung der Zeitstempel vorgenommen.

Einbettung der Zeitstempel in den Paketstrom

Wie dargestellt, enthält nicht jedes über den Tunnel transportierte Paket einen Zeitstempel. Zudem müssen die Zeitstempel dem Absendezeitpunkt möglichst genau entsprechen. Diese Randbedingung erfordert, daß der Zeitstempel unmittelbar vor der Übergabe des Pakets an das Betriebssystem in das Paket integriert wird.

Zur Erfüllung dieser Anforderungen wurde der folgende Weg zur Einbettung der Zeitstempel in den Paketstrom eingeschlagen:

- Die Zeitstempelgenerierung erfolgt im Output-Modul des Systems, d.h. zum Zeitpunkt der Übergabe des Paketes an das Betriebssystem.
- Der Zeitstempel wird als Paketoption im Nutzdatenpaket übertragen. Der Empfänger kann die Option aus dem Paket entfernen, ohne daß der Nutzdatenfluß davon berührt wird. Dazu wird der Zeitstempel als Service-PDU dem komprimierten Nutzdatenpaket vorangestellt. Das Konzept der Service-PDUs wird in Abschnitt 3.2.5.1, Seite 100ff vorgestellt.
- Das Input-Modul des Empfängers erkennt den Zeitstempel und entfernt ihn aus dem Nutzdatenstrom. Hier wird auch die Verarbeitung des Zeitstempels initiiert.

Dieses Verfahren entkoppelt die Übertragung der Zeitstempel von der Behandlung der Nutzdatenströme und vereinfacht damit die Struktur des Gesamtsystems.

3.2.2.4 Datenratenregelung des Systems

Der Empfänger berechnet, wie ausführlich dargestellt, aus den empfangenen Paketen und Zeitstempeln die Kennwerte für den Übertragungsweg. Sie werden zusammen mit weiteren Daten, auf die im folgenden noch eingegangen wird, dem Sender über Tunnel-Reports zugeführt. Auf der Basis der Tunnel-Reports steuert der Sender den Datenfluß. Damit entsteht ein lose gekoppeltes Regelungssystem.

Entscheidend für die Eigenschaften des Gesamtsystems ist, nach welcher Strategie der Sender den Datenfluß steuert und wie häufig der Empfänger Tunnel-Reports erzeugt. Auf diese beiden Fragestellungen wird in den folgenden Abschnitten eingegangen.

Steuerung des Datenflusses beim Sender

Die Steuerung des Datenflusses erfolgt beim Sender im Output-Modul des Systems. Dazu werden die folgenden Parameter bereitgestellt:

C_{init} Dieser vom Systemnutzer konfigurierte Wert repräsentiert die abgeschätzte Übertragungskapazität. Die dadurch vorgegebene Datenrate wird vom Sender nicht überschritten.

$\Delta t_{Q_{min}}$ Unterer Schwellwert für das Queuing-Delay (Δt_Q). Wird dieser Wert unterschritten, versucht der Sender die Datenrate anzuheben. $\Delta t_{Q_{min}}$ wird durch den Nutzer am Heimarbeitsplatz konfiguriert.

$\Delta t_{Q_{max}}$ Oberer Schwellwert für das Queuing-Delay (Δt_Q). Wird dieser Wert überschritten, beginnt der Sender die Datenrate abzusenken, um einer drohenden Überflutung der Router-Queues auf dem Übertragungsweg entgegenzuwirken. Auch dieser Wert ist durch den Nutzer zu konfigurieren.

Δt_Q Dieser Wert ist eine Schätzung für die durch Verweilzeiten in Router-Queues entstanden Verzögerungen für Pakete auf dem Übertragungsweg. Er wird vom Empfänger berechnet und dem Sender durch Tunnel-Reports zugänglich gemacht.

C_{max_R} Dies ist die vom Empfänger abgeschätzte Kapazität des Übertragungsweges. Er wird dem Sender ebenfalls über Tunnel-Reports mitgeteilt.

LR Die kurzzeitige Paketverlustrate. Sie wird wird vom Sender und Empfänger beim Erhalt eines Tunnel-Reports berechnet. Das Verfahren zur Berechnung der Paketverlustraten wird in Abschnitt 3.2.3, Seite 93ff, dargestellt.

LR_{krit} Schwellwert für Paketverlustraten, die das System zur Regelung veranlaßt. Hierbei handelt es sich um einen Konfigurationsparameter, der durch das MAG festgelegt wird und während der Etablierung des Tunnels dem MAS übermittelt wird.[21]

Die Datenratensteuerung beim Sender erfolgt im wesentlichen auf der Basis der geschätzten Verweilzeiten der Pakete in Router-Queues. Paketverluste[22] führen nicht, wie bei vielen anderen Protokollen, zur wiederholten Übertragung von Paketen, sondern dienen als zusätzliches Entscheidungskriterium zur Erkennung von Stauungen in den Output-Queues der Router. Die Datenratensteuerung basiert auf der folgenden Strategie:

- Initial wird die Sendedatenrate auf den vom Benutzer vorgegebenen Wert begrenzt. Erst nach Erhalt des ersten Tunnel-Reports vom Empfänger wird die Datenrate u.U. verändert.
- Weist ein im Tunnel-Report enthaltenes Queuing-Delay (Δt_Q) oder eine überhöhte Paketverlustrate (LR) darauf hin, daß eine Stauung entstanden ist, wird die Datenrate auf 50% begrenzt. Dadurch soll ein rascher Abbau der Stauung gewährleistet werden. Gleichzeitig wird die Soll-Datenrate des Senders modifiziert, indem $C_{max_S} = \min(C_{max_R}, C_{init})$ gesetzt wird.
- Sinkt im folgenden der Wert von Δt_Q unter $\Delta t_{Q_{min}}$ und liegt die Paketverlustrate (LR) unterhalb des Schwellwertes LR_{krit}, wird die Datenrate nach einem binären Schrittverfahren angehoben.
- Konnte durch die Datenratenreduktion die Stauung nicht abgebaut werden, verzichtet der Sender auf die Übertragung der Audio- und Video-Datenströme, überträgt die entsprechenden RTCP-Datenströme aber weiterhin.

[21] Die Übermittlung von LR_{krit} erfolgt unter Nutzung des MASCP, siehe Abschnitt 3.2.10, Seite 119f.

[22] Paketverluste enstehen durch den Verwurf von Paketen in vermittelnden Komponenten entlang des Netzpfades zwischen MAS und MAG.

- Konnte durch diesen Schritt die Stauung beseitigt werden, beginnt der Sender unter Verzicht auf den Video-Datenstrom erneut mit der Übertragung des Audio-Datenstroms.
- Führt diese Anhebung zu keiner erneuten Überflutung des Routers, wird auch der Video-Datenstrom wieder übertragen.
- War der vorhergehende Schritt erfolgreich, wird die Sendedatenrate in Schritten der Soll-Sendedatenrate angenähert.

Dieses Regelungsverhalten begründet sich aus der Überlegung, daß grundsätzlich zwei Fälle unterschieden werden müssen, die die Regelung der Datenrate erfordern:

1. Der Nutzer kann versehentlich einen zu großen Wert für C_{init} angegeben haben. In diesem Fall ist es erforderlich, daß das System tolerant auf diesen Fehler reagiert und die Teilnahme an der Konferenz gestattet. Ohne die Regelung würde es zu einer latenten Überlastsituation kommen, deren Folge eine nachhaltige Verzögerung der Datenpakete durch die Zwischenspeicherung in den Router-Queues wäre.

2. Durch konkurrierende Datenströme kann es zum Aufbau von Stauungen kommen. Da sich die nebenläufigen Datenströme der Kontrolle des Systems entziehen, kann es sich ihnen gegenüber nur reaktiv verhalten. Dafür eröffnen sich zwei Möglichkeiten: Der Tunnel-Datenstrom kann sich gegenüber konkurrierenden Datenströmen nach der Absenkung auf ein Mindestmaß, z.B. $\frac{C_{max_i}}{2}$, unelastisch verhalten und dadurch zumindest bei TCP-Datenströmen eine Reduktion der Datenrate des konkurrierenden Datenstroms erzwingen. Leider wird dadurch die Situation nicht verbessert. Der TCP-Datenstrom wird über Paketverluste geregelt. Diese treten erst bei Überflutung der Access-Router-Queue ein. Durch das üblicherweise eingesetzte FIFO-Queuing entstehen für den Tunnel-Datenstrom unzulässige Verzögerungen. Diese Stauung ist nicht zu verhindern. Daher erscheint es im ersten Schritt günstiger, den Tunnel-Datenstrom stark zu begrenzen, so daß der TCP-Datenstrom die volle Übertragungskapazität nutzen kann und die Übertragung schneller beendet wird.

   Für den Nutzer am Heimarbeitsplatz wirkt dieses Steuerungsverhalten wie eine Netzpartitionierung. Für ihn ist keines der Konferenz-Werkzeuge mehr nutzbar, obwohl tatsächlich nur die bezüglich der Laufzeit kritischen Audio- und Video-Datenströme unbrauchbar sind.

   Eine günstige Lösung bei FIFO-Queuing im Access-Router ergibt sich aus der Kombination beider Verfahren unter gleichzeitiger Einflußnahme auf die Queuing-Strategie beim Sender: Wird durch wachsende oder unverändert hohe Δt_{Q_i} deutlich, daß eine Stauung vorliegt und der Sender die Datenrate bereits auf $\frac{C_{max_i}}{2}$ abgesenkt hat, wird keine weitere Reduktion der Sendedatenrate vorgenommen. Stattdessen wird die Queuing-Strategie umgestellt. Es wird weiterhin ein Simple-Priority-Queuing mit unveränderten Prioritäten benutzt, die RTP-Pakete der Audio- und Video-Datenströme werden jedoch beim Sender verworfen. Davon unberührt bleiben die RTCP-Pakete der entsprechenden Datenströme. Damit werden nur noch Daten übertragen, die vergleichsweise unempfindlich gegenüber Verzögerungen auf der Übertragungsstrecke sind. Sinken das Queuing-Delay bzw. die Paketverlustrate, wird die Datenrate wieder angehoben und es wird auf die ursprüngliche Queuing-Strategie zurückgeschaltet.

Der Nutzer am Heimarbeitsplatz erhält dadurch während der Dauer der konkurrierenden TCP-Übertragung keine Audio- und Video-Daten mehr, der Status der Konferenzpartner wird jedoch weiterhin aktualisiert. Da die Audio- und Video-Daten nicht vom Konferenz-Werkzeug des Nutzers am Heimarbeitsplatz empfangen werden, werden die entsprechenden RTCP-Reports den Paketverlust ausweisen. Dadurch erhalten auch die entfernten Partner eine entsprechende Rückkopplung. Die Shared-Tools bleiben nutzbar, so daß die Problematik notfalls auf diesem Weg kommuniziert werden kann.

Wenngleich die Problematik aus der Sicht des Nutzers am Heimarbeitsplatz dargestellt wurde, gelten die Aussagen auch für Stauungen in der umgekehrten Richtung.

Das vorgestellte Regelungsverfahren gestattet die angemessene Reaktion auf beide Überlastsituationen.

Beschreibung des Senderverhaltens durch einen Zustandsautomaten

Das Regelverhalten des Senders läßt sich präzise durch einen Zustandsautomaten beschreiben. Daher dient diese Beschreibungsform als Ausgangspunkt für die Implementierung.

Tabelle 3.1 enthält die Zustände des Automaten. Die möglichen Ereignisse sind in Tabelle 3.2 dargestellt. Die Zustandsübergänge werden durch Tabelle 3.3 beschrieben. Das Zustandsübergangsdiagramm zeigt Abbildung 3.9.

Ausgangspunkt ist der Zustand α. Der Sender vermittelt die Daten mit der Datenübertragungsrate C. Sie wird mit C_{init} initialisiert. Dieser Zustand wird durch das Ereignis a hergestellt. Wird durch wachsende Queuing-Delays (Δt_Q) oder eine Paketverlustrate $LR > LR_{krit}$ eine Stauung erkannt, löst Ereignis d den Wechsel in den Zustand β aus. Dabei wird die Datenrate auf 50% der Solldatenrate begrenzt. Genügt dieser Schritt zum Abbau der Stauung, wird durch Ereignis b zurück in den Zustand α gewechselt. Dabei wird gleichzeitig die Datenrate angehoben. Konnte die Stauung nicht abgebaut werden, löst das Ereignis d den Wechsel nach Zustand γ aus. Wenngleich die Übertragungsrate C hier nicht verändert wird, wird das zu übertragende Datenvolumen durch die Abschaltung der Audio- und Video-Datenströme weiter begrenzt.

Das System verharrt im Zustand γ, bis sich Stauungen aufgelöst haben. Danach wird durch das Ereignis b oder c in den Zwischenzustand δ gewechselt. Hier wird der Audio-Datenstrom wieder übertragen. Führt dieser Schritt erneut zum Aufbau von Warteschlangen, fällt das Output-Modul bis zur Bereinigung der Situation in den Zustand γ zurück. Wenn die Zuschaltung des Audio-Datenstroms in Zustand δ keinen negativen Einfluß auf die Verzögerung der Pakete oder die Paketverlustrate hat, wird durch Ereignis b der Wechsel in den Ausgangszustand α ausgelöst. In diesem Zustand wird die Sendedatenrate C mit einem binären Schrittverfahren der Soll-Sendedatenrate (C_{max_S}) angenähert.

Bei allen Zustandsübergängen wird die Soll-Datenrate des Senders der vom Empfänger und Sender abgeschätzten Maximaldatenrate des Tunnels angepaßt. Keinesfalls wird die durch den Benutzer vorgegebene Datenrate C_{init} überschritten.

Eine wichtige Eigenschaft des Systems ist, daß im Falle ausbleibender Tunnel-Reports die Datenübertragung eingestellt wird. Dieses Verhalten ist erforderlich, damit die Datenvermittlung

Zustand	Beschreibung
α	Sender sendet mit Datenrate C und $C > \frac{C_{max_S}}{2}$, mit Audio und mit Video
β	Sender sendet mit Datenrate C und $C = \frac{C_{max_S}}{2}$, mit Audio und mit Video
γ	Sender sendet mit Datenrate C, ohne Audio und ohne Video
δ	Sender sendet mit Datenrate C, mit Audio und ohne Video
ε	Ende

Tabelle 3.1: Zustandstabelle der Datenratenregelung beim Sender

Ereignis	Beschreibung
a	Start
b	Tunnel-Report empfangen, $\Delta t_Q < \Delta t_{Q_{min}} \wedge LR < LR_{krit}$
c	Tunnel-Report empfangen, $\Delta t_{Q_{min}} \leq \Delta t_Q < \Delta t_{Q_{max}} \wedge LR < LR_{krit}$
d	Tunnel-Report empfangen, $\Delta t_Q \geq \Delta t_{Q_{max}} \vee LR \geq LR_{krit}$
e	Timer ist abgelaufen

Tabelle 3.2: Ereignistabelle der Datenratenregelung beim Sender

über den Tunnel eingestellt wird, wenn die Partnerinstanz am anderen Ende des Tunnels durch einen Fehler terminiert wurde oder eine Netzwerkpartitionierung entstanden ist.

Das Ausbleiben von Tunnel-Reports wird erkannt, indem beim Empfang eines Tunnel-Reports ein Timer zurückgesetzt wird. Bleibt das Rücksetzen aus, tritt ein Timer-Ereignis ein. Es führt in allen Systemzuständen zum sofortigen Abbruch des Prozesses. Dadurch wird verhindert, daß andere Nutzer des Access-Routers durch den Betrieb des MAS beeinflußt werden.[23] Die Timer-Laufzeit wird als T_{stop} bezeichnet. Sie wird durch den Betreiber des MAS festgelegt und dem MAG bei der Initialisierung über das MASCP mitgeteilt. Der Wert sollte im Bereich zwischen 30 und 60 Sekunden liegen.

Emission von Tunnel-Reports

In Abschnitt 3.2.2.2 wurde erklärt, daß zur Stabilisierung des Systems der erste Tunnel-Report erst nach dem Empfang einer Folge von Datenpaketen durch den Empfänger generiert werden soll. Die in diesem Abschnitt untersuchte Frage ist, nach welchen Regeln fortan Tunnel-Reports generiert werden. Aus Sicht des Senders sind sie eine zentrale Entscheidungsgrundlage und daher ist eine möglichst hohe Frequenz anzustreben. Auf der anderen Seite beanspruchen die Reports einen Teil der Übertragungskapazität, die den Nutzdaten dann nicht mehr zur Verfügung steht.

Die minimale Frequenz der Tunnel-Reports wird durch den Konfigurationsparameter T_{stop} des

[23] Wenn die PPP-Verbindung vom Access-Router zum Nutzer am Heimarbeitsplatz gelöst wird, wird bei dynamischer IP-Adreßvergabe die gleiche Adresse nach einer kurzen Zeit einem neuen Nutzer zugeordnet. Wenn der MAS die Datenvermittlung nicht abbrechen würde, würden die Datenströme der Konferenz dem neuen Nutzer zugeleitet werden. Damit würde der neue Nutzer den Eindruck gewinnen, daß die Wählverbindung gestört ist. Dies gilt es zu vermeiden.

Zustand	Ereignis	Folgezustand	Aktion
	a	α	$C = C_{max_S} = C_{init}$ Timer-Restart
α	b	α	$C_{max_S} = \min(\max(C_{max_R}, C_{max_S}), C_{init})$, $C = C + \frac{C_{max_S} - C}{2}$, Timer-Restart
α	c	α	$C_{max_S} = \min(\max(C_{max_R}, C_{max_S}), C_{init})$, Timer-Restart
α	d	β	$C_{max_S} = \min(\max(C_{max_R}, C_{max_S}), C_{init})$, $C = \frac{C_{max_S}}{2}$, Timer-Restart
α	e	ε	
β	b	α	$C_{max_S} = \min(\max(C_{max_R}, C_{max_S}), C_{init})$, $C = C + \frac{C_{max_S} - C}{2}$, Timer-Restart
β	c	β	$C_{max_S} = \min(\max(C_{max_R}, C_{max_S}), C_{init})$, Timer-Restart
β	d	γ	$C_{max_S} = \min(\max(C_{max_R}, C_{max_S}), C_{init})$, Timer-Restart
β	e	ε	
γ	b	δ	$C_{max_S} = \min(\max(C_{max_R}, C_{max_S}), C_{init})$, Timer-Restart
γ	c	δ	$C_{max_S} = \min(\max(C_{max_R}, C_{max_S}), C_{init})$, Timer-Restart
γ	d	γ	$C_{max_S} = \min(\max(C_{max_R}, C_{max_S}), C_{init})$, Timer-Restart
γ	e	ε	
δ	b	α	$C_{max_S} = \min(\max(C_{max_R}, C_{max_S}), C_{init})$, $C = C + \frac{C_{max_S} - C}{2}$, Timer-Restart
δ	c	δ	$C_{max_S} = \min(\max(C_{max_R}, C_{max_S}), C_{init})$, Timer-Restart
δ	d	γ	$C_{max_S} = \min(\max(C_{max_R}, C_{max_S}), C_{init})$, Timer-Restart
δ	e	ε	

Tabelle 3.3: Zustandsübergangstabelle der Datenratenregelung beim Sender

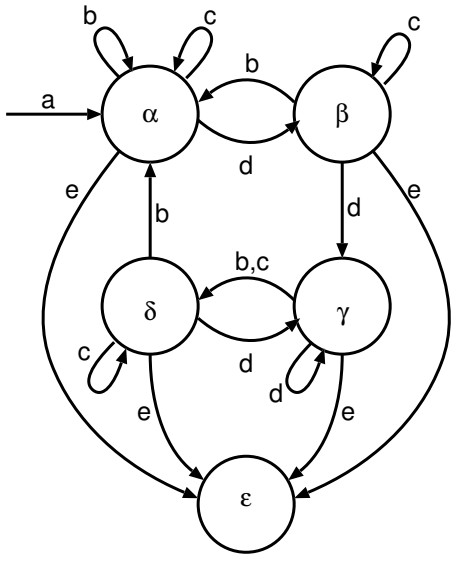

Abbildung 3.9: Zustandsübergangsdiagramm Datenratenregelung beim Sender

MAS bestimmt. Das Aussenden der Tunnel-Reports mit der Frequenz $\frac{1}{T_{stop}}$ ist jedoch nicht hinreichend. Tunnel-Reports werden über UDP übertragen und es gibt, wie bei allen UDP-Datagrammen, keine Sicherheit, daß sie den Empfänger erreichen. Zudem sind auch auf dem Übertragungsweg der Tunnel-Reports nachhaltige Verzögerungen in Folge von Stauungen nicht auszuschließen, wodurch die Auslieferung von Tunnel-Reports an den Empfänger verzögert werden kann und dadurch beim jeweiligen Nutzdatensender der Timer T_{stop} ablaufen kann. Daher wird die Emissionsfrequenz für Tunnel-Reports auf den Wert $\frac{10}{T_{stop}}$ festgelegt. Nach vorliegenden Erfahrungen wird dadurch sichergestellt, daß der Tunnel auch dann nicht abgebaut wird, wenn eine Reihe von Tunnel-Reports den Empfänger nicht erreicht oder die Reports um einige Sekunden verzögert ausgeliefert werden.

Die sich daraus ergebende Zeit zwischen der Emission von zwei Tunnel-Reports bewegt sich in der Größenordnung von 3 bis 6 Sekunden. In dieser Zeit kann sich der für die Regelung des Systems ausschlaggebende Wert Δt_Q nachhaltig verändern. Ziel muß es sein, bei entscheidenden Änderungen von Δt_Q unmittelbar einen Tunnel-Report zu übertragen. Dazu muß der Empfänger des Nutzdatenstroms jedoch erkennen, welches entscheidende Änderungen von Δt_Q sind.

Die Ereignistabelle der Datenratenregelung beim Sender (Tabelle 3.2) zeigt, daß Zustandsänderungen des Regelungssystems beim Sender immer dann auftreten, wenn Δt_Q die Schwellwerte $\Delta t_{Q_{min}}$ oder $\Delta t_{Q_{max}}$ über- oder unterschreitet oder die Paketverlustrate LR den Schwellwert LR_{krit} über- oder unterschreitet. Daraus läßt sich eine geeignete Emissionsstrategie für Tunnel-Reports ableiten:

- Der erste Tunnel-Report wird nach dem Empfang einer fest eingestellten Zahl von Paketen oder nach Ablauf der Zeitspanne $\frac{T_{stop}}{10}$ generiert.
- Neben den Werten Δt_{Q_i} und LR_i werden die Vorgängerwerte $\Delta t_{Q_{i-1}}$ und LR_{i-1} aufbewahrt.

- Zudem wird festgehalten, wann der letzte Tunnel-Report versendet wurde ($T(t_{ltr})$).
- Ein Tunnel-Report wird generiert, wenn der folgende Ausdruck zu einer positiven Bewertung führt:

$$\begin{array}{l}(T(t_i) \geq T(t_{ltr}) + \frac{T_{stop}}{10}) \vee \\ (\Delta t_{Q_i} < \Delta t_{Q_{min}} \wedge \Delta t_{Q_{i-1}} \geq \Delta t_{Q_{min}}) \vee \\ (\Delta t_{Q_i} \geq \Delta t_{Q_{min}} \wedge \Delta t_{Q_{i-1}} < \Delta t_{Q_{min}}) \vee \\ (\Delta t_{Q_i} \geq \Delta t_{Q_{max}} \wedge \Delta t_{Q_{i-1}} < \Delta t_{Q_{max}}) \vee \\ (\Delta t_{Q_i} < \Delta t_{Q_{max}} \wedge \Delta t_{Q_{i-1}} \geq \Delta t_{Q_{max}}) \vee \\ (LR_i \geq LR_{krit} \wedge LR_{i-1} < LR_{krit}) \vee \\ (LR_i < LR_{krit} \wedge LR_{i-1} \geq LR_{krit})\end{array} \tag{3.17}$$

  Der Term faßt die verbal formulierten Randbedingungen zur Emission eines Tunnel-Reports zusammen.

Mit diesem Verfahren wird sichergestellt, daß der Nutzdatenstromsender Tunnel-Reports erhält, wenn Statusveränderungen bei ihm zu erwarten sind. Eine weitere Optimierung bezüglich der Häufigkeit der Tunnel-Report-Übermittlung läßt sich erzielen, wenn dem Nutzdatenstromempfänger der aktuelle Status der Zustandsmaschine des Nutzdatenstromsenders bekannt wäre. Das einzusparende Datenvolumen muß dazu mit dem Aufwand zur sicheren Übertragung des Systemstatus in Relation gesetzt werden. Die Ermittlung dieser Werte ist von weiteren Randbedingungen wie der Verkehrscharakteristik des Nutzdatenstroms sowie dem Vorhandensein konkurrierender Datenströme abhängig und gestaltet sich damit außerordentlich schwierig. Zudem würde dadurch die Unabhängigkeit des Senders vom Empfänger weiter eingeschränkt, so daß im Rahmen dieser Arbeit auf eine derartige Optimierung verzichtet wurde.

Abschließend bleibt festzuhalten, daß die möglichst verzögerungsfreie Übertragung der Tunnel-Reports für die Datenratenregelung des Systems wichtig ist. Daher werden Tunnel-Reports mit höherer Priorität als Audio-Datenströme behandelt. Verzögerungen durch Stauungen in den Queues der beteiligten Router auf dem Übertragungsweg können nicht unterbunden werden. Die Sendedatenregelung sollte jedoch sicherstellen, daß die entstehenden Verzögerungen sich in engen Grenzen halten.

3.2.3 Erkennung von Paketverlusten

In den vorangegangenen Abschnitten wurde dargestellt, daß die Datenratenregulierung des MBone-Access-Service auf der Erkennung von Verweilzeiten der Pakete in Warteschlangen auf dem Übertragungsweg basiert. Daraus folgt, daß die Detektion von Paketverlusten für die Regelung des Systems von untergeordneter Bedeutung ist. Dennoch ist die Ermittlung der Paketverlustrate wichtig, da sie ein Qualitätskriterium für die Übertragungsstrecke ist. Aus gleichem Grund sollte sie dem Nutzer zugänglich sein.

Da die Tunnel-Pakete keine Sequenznummern tragen, ist die unmittelbare Ermittlung der Verlustrate nicht möglich. Sie wird daher mittelbar durch das folgende Verfahren ermittelt.

Ausgangspunkt ist, daß nicht der Verlust eines bestimmten Paketes erkannt werden muß, sondern die Summe der ausgesendeten Pakete, die den Empfänger in einem Zeitintervall nicht erreicht haben. Die grundlegende Idee ist, die Anzahl der vom Sender übertragenen Pakete und Bytes

mit den korrespondierenden Zahlen beim Empfänger zu vergleichen. Das Verfahren lehnt sich eng an das im RTP-Standard vorgeschlagenen Verfahren [SCFJ96, Abschnitt 6.3] an.

Zur Erkennung von Paketverlusten werden an beiden Tunnelenden Register für folgende Werte eingeführt:

- Anzahl der gesendeten Pakete (PCS).
- Summe der gesendeten Payload (BCS).
- Anzahl der empfangenen Pakete (PCR).
- Summe der empfangenen Payload (BCR).

Diese Register haben eine Kardinalität von 2^{32} und werden zur Initialisierungszeit des Tunnels mit dem Wert 0 initialisiert und fortan erhöht.

Bei der Generierung eines Tunnel-Reports überträgt der jeweilige Sender den Inhalt seiner Register PCS und BCS, also die Anzahl der gesendeten Datagramme und die Summe der darin enthaltenen Bytes. Durch einen einfachen Vergleich mit den Registern PCR und BCR kann der Empfänger die Paketverlustrate sowie die Datenverlustrate berechnen:

$$\text{Langfristiger Paketverlust: } PCS_{S_i} - PCR_{R_i} \tag{3.18}$$

$$\text{Kurzfristiger Paketverlust: } (PCS_{S_i} - PCS_{S_{i-1}}) - (PCS_{R_i} - PCS_{R_{i-1}}) \tag{3.19}$$

$$\text{Langfristiger Datenverlust: } BCS_{S_i} - BCR_{R_i} \tag{3.20}$$

$$\text{Kurzfristiger Datenverlust: } (BCS_{S_i} - BCS_{S_{i-1}}) - (BCS_{R_i} - BCS_{R_{i-1}}) \tag{3.21}$$

Das Verfahren wird am in Abbildung 3.10 dargestellten Beispiel für die Paketverlustrate erläutert. Auf der linken Seite sind die Werte des PCS-Registers des Senders im Zeitverlauf dargestellt. Auf der Seite des Empfängers sind die Werte seines PCR-Registers im Zeitverlauf dargestellt. Das dritte, siebte und neunte gesendete Paket erreicht den Empfänger nicht.

Der erste Tunnel-Report wird als sechstes Paket[24] vom Sender übertragen. Es gelangt zum Zeitpunkt t_1 beim Empfänger an. Das PCR-Register des Empfängers hat nach dem Empfang den Wert fünf. Langfristiger Paketverlust ($6 - 5 = 1$) und kurzzeitiger Paketverlust ($(6 - 0) - (5 - 0) = 1$) stimmen überein.

Der zweite Tunnel-Report des Senders erreicht den Empfänger nicht. Die Paketverlustwerte können nicht neu berechnet werden.

[24] In diesem Beispiel wurde zur besseren Illustration eine hohe Emissionsrate von Tunnel-Reports angenommen. Im normalen Betriebsfall wird die Rate geringer sein und der erste Tunnel-Report wird nicht bereits nach fünf übertragenen Paketen generiert.

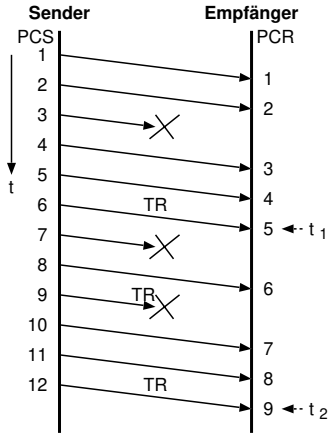

Abbildung 3.10: Ermittlung von Paket- und Datenverlustraten

Zum Zeitpunkt t_2 erreicht den Empfänger erneut ein Tunnel-Report. Die Berechnung des langfristigen Paketverlustes ergibt: $12 - 9 = 3$. Der kurzzeitige Paketverlust errechnet sich zu: $(12 - 6) - (9 - 5) = 2$. Dies ist die Anzahl der Pakete, die seit dem Empfang des vorhergehenden Tunnel-Reports den Empfänger nicht erreicht haben.

Es ist nicht auszuschließen, daß das Netz Pakete dupliziert. Dadurch können die Paketverlustzahlen negativ werden. Der Wert für den kurzzeitigen Paketverlust wird in diesem Fall auf den Wert 0 gesetzt. Der Wert für den langfristigen Paketverlust wird nicht korrigiert.

Die Regelung der Tunnel-Datenrate erfolgt beim Sender auf der Basis der kurzzeitigen Paketverlustrate (vgl. Abschnitt 3.2.2.2, Seite 80ff.) (LR). Dazu wird aus dem kurzzeitigen Paketverlust beim Empfänger die Paketverlustrate bestimmt und dem Sender im nächsten vom Empfänger erstellten Tunnel-Report übertragen. Die Paketverlustrate berechnet sich wie folgt:

$$\text{Paketverlustrate: } LR = \frac{(PCS_{S_i} - PCS_{S_{i-1}}) - (PCS_{R_i} - PCS_{R_{i-1}})}{PCS_{S_i} - PCS_{S_{i-1}}} \cdot 255.0 \qquad (3.22)$$

Der Bruch in Gleichung 3.22 ergibt einen Wert zwischen 0.0 (kein Paketverlust) und 1.0 (kein Paket wurde erhalten). Der sich daraus ergebende Wert wird mit 255.0 multipliziert. Der ganzzahlige Teil des Ergebnisses dieser Multiplikation wird als Maß für die Paketverlustrate in einem Byte des Tunnel-Reports an den Sender übertragen. Zusätzlich enthält der Tunnel-Report den Wert für den langfristigen Paketverlust sowie die Zahl der vom Empfänger empfangenen Datagramme. Ergänzt werden die Daten durch einen Zeitstempel der Systemzeit des Empfängers. Daraus kann der Sender die integrale Empfangsdatenrate, die integrale Empfangspaketrate jeweils kurz- und langfristig berechnen.

Dem Empfänger ist damit auch die Berechnung der kurz- und langfristigen Sendedatenrate des Senders möglich.

Da das System absolute Zähler für die über den Tunnel gesendeten Pakete und Bytes nutzt, ist ein Überlauf der Zähler unvermeidlich. Für den Fall einer ISDN-Übertragungsstrecke wird der Zähler für die gesendeten Bytes bei voller Auslastung des Kanals (64 kbps) aufgrund der

Feld	Bit-Breite	Bedeutung
TS	32	Timestamp des Senders (Sekunden seit dem 1. Januar 1970)
MSFRAC	8	Millisekunden Bruchteil des TS (Einheit: 1/256 s)
PCS	32	Anzahl der gesendeten Pakete
BCS	32	Anzahl der gesendeten Payload-Bytes
PCR	32	Anzahl der empfangenen Pakete
BCR	32	Anzahl der empfangenen Payload-Bytes
LR	8	Kurzzeitige Paketverlustrate
CPL	24	Anzahl der nicht empfangenen Pakete
TQ	16	Δt_Q in ms
CMAX	16	C_{max} in $10^2 bps = 0.1 kbps$

Tabelle 3.4: Inhalt eines Tunnel-Reports

Kardinalität von 2^{32} frühestens nach ca. 18 Stunden Betriebszeit des Tunnels überlaufen. Wenngleich die Teilnahme an einer Konferenz über diesen Zeitraum unwahrscheinlich ist, wurde ein Verfahren vorgesehen, das diesen Fall berücksichtigt.

Wenn der Sender einen Überlauf eines Zählers erkennt, unterbricht er die Übermittlung der Nutzdaten, setzt seine Zähler zurück und sendet in Folge *Counter-Reset-Requests* an den Empfänger. Hierbei handelt es sich um eine spezielle Ausprägung einer Tunnel-Service-PDU, die in Abschnitt 3.2.5.5, Seite 106, vorgestellt werden. Der Empfänger reagiert auf den Empfang dieser PDU ebenfalls mit dem Reset seiner Zähler und sendet bis zum Empfang einer von einem *Counter-Reset-Request* verschiedenen PDU *Counter-Reset-Confirmations* an den Sender. Auch hierbei handelt es sich um eine Tunnel-Service-PDU.

Der Nachteil dieses Verfahrens gegenüber dem im RTP vorgeschlagenen Algorithmus ist, daß nach dem Überlauf eines Zählers die Vermittlung der Nutzdaten unterbrochen wird. Dafür kann auf die Übertragung von Sequenz-Nummern und Zähler für Sequenz-Nummer-Zyklen verzichtet werden [SCFJ96, Abschnitt 6.3.1]. Dies trägt zur Verringerung des Protokoll-Overheads bei.

3.2.4 Tunnel-Reports

Tunnel-Reports dienen der Erkennung von Stauungen auf dem Übertragungsweg zwischen MAS und MAG und damit verbunden der Steuerung des Systems zur Vermeidung und dem Abbau von Stauungen. Zudem werden sie genutzt, um Paketverlustraten zu ermitteln und dem Nutzer Informationen über integrative Übertragungseigenschaften des Systems bereitzustellen. Tunnel-Reports werden im komprimierten Paketstrom zwischen MAS und MAG mit hoher Priorität als Service-PDUs übertragen. Auf die Integration in den Paketstrom wird im Abschnitt 3.2.5.1, Seite 100, eingegangen. Dieser Abschnitt dient der zusammenfassenden Darstellung des Inhalts der Tunnel-Reports.

Ein Tunnel-Report enthält die in Tabelle 3.4 dargestellten Daten. Alle Daten werden in Network-Byte-Order übertragen. Aufgrund der kompakten Größe eines Tunnel-Reports von 29 Bytes wird die ihn einbettende Service-PDU üblicherweise als Header-Option eines Tunnel-Pakets

übertragen. Nur wenn keine weiteren Nutzdaten zur Übertragung anstehen oder die Größe des sich ergebenden Pakets die *Path-MTU* übersteigt, erfolgt die Übertragung in einem einzelnen Tunnel-Paket. Tunnel-Reports haben Vorrang vor allen Nutzdaten-Paketen.

3.2.5 Komprimierung

Im Analyse-Teil wurde deutlich, daß bei der Übertragung von Echtzeitdatenströmen über serielle Leitungen geringer Übertragungskapazität ein Zielkonflikt entsteht:

- Die optimale Auslastung des Kanals ergibt sich, bedingt durch den Protokoll-Overhead für IP und UDP, wenn möglichst große PDUs übertragen werden.
- Die Verzögerung der PDUs korreliert eng mit ihrer Serialisierungszeit. Die Serialisierungszeit steigt linear mit der PDU-Größe. Besonders für die Übertragung von Audio-Datenströmen in interaktiven Konferenzen sind geringe Verzögerungen bei der Übertragung unabdingbar. Dementsprechend sind kleine PDUs vorteilhaft, die jedoch die effektive Auslastung des Kanals verringern.

Dieser grundsätzliche Konflikt läßt sich nicht lösen, zumindest nicht ohne direkten Zugriff auf das Framing unterhalb des Link-Layers. Ein Weg diesen Konflikt zu mildern ist, den zu übertragenden Nachrichtenstrom zu reduzieren.

Ein erster Schritt ist die verlustbehaftete Kompression der Datenströme durch Mixer und Transcoder. Diese Verfahren lassen sich bei Audio- und Video-Datenströmen anwenden und reduzieren den Payload-Anteil im jeweiligen Datenstrom, nicht jedoch den Protokoll-Overhead.[25]

Die Datenströme für Session-Management, Shared-Tools und Session-Advertising sind im Vergleich zu den Video- und Audio-Datenströmen klein. Nachteilig ist, daß in einigen Anwendungsfällen relativ große PDUs auftreten. Während diese serialisiert werden, ist der Kanal für Audio-PDUs blockiert. Daher scheint auch hier die Kompression der Daten sinnvoll, die jedoch verlustfrei erfolgen muß. Geeignete Verfahren, wie der Lempel-Ziv Algorithmus zur universellen sequentiellen Datenkompression [JZ77], sind verfügbar und sollen an dieser Stelle zum Einsatz kommen. Dabei ist zu beobachten, welchen Erfolg die Komprimierung hat. Wenn die Komprimierung keine Reduktion der Payload ergibt, ist die PDU unkomprimiert zu übertragen.

Wenn die Möglichkeiten zur Kompression der Nutzdaten ausgeschöpft sind, lassen sich weitere Optimierungen nur durch die Reduktion des Protokoll-Overhead erzielen. Auf den ersten Blick erscheint auch dabei die verlustfreie Intra-Paket-Kompression der Protokoll-Daten interessant. Bei näherer Betrachtung wird jedoch offensichtlich, daß erhebliche Teile der Protokoll-Köpfe über einen längeren Zeitraum konstant bleiben oder sich nur geringfügig ändern. Dieser Gedanke führt auf die Inter-Paket-Kompression der Protokoll-Header. Wenn darauf verzichtet werden kann, einen großen Teil des Protokoll-Headers in jedem Paket zu übertragen, ist dieses Verfahren der Intra-Paket-Kompression der Header überlegen.

[25] Tatsächlich reduzieren Mixer den Protokoll-Overhead, da die Sprach-Signale der Sender zu einem Datenstrom aggregiert werden. Zudem kann durch ein größeres Sampling-Intervall die Paketrate und damit der Protokoll-Overhead zu Lasten der Interaktivität in gewissen Grenzen reduziert werden.

Auf diesen Gedanken basiert die seit langem bekannte und vielfach nutzbringend eingesetzte *Van Jacobson Header Compression for TCP*, die in [Jac90] dokumentiert ist. In Entwicklung befindet sich ein ähnliches Verfahren für die Kompression von IP-, UDP- und RTP-Headern. Es ist in [CJ97] dokumentiert und fokussiert auf die Kompression der Protokoll-Header auf der Ebene des Data-Link-Layers. Das Hauptaugenmerk richtet sich dabei auf das Point-to-Point-Protocol als Data-Link-Layer-Protocol.

Mit diesem Verfahren kann die Summe der Header von IP, UDP und RTP von 40 Byte in vielen Fällen auf 2-4 Byte reduziert werden. Diese Reduktion wird erzielt, indem an beiden Verbindungsendpunkten Statusinformationen über die Verbindung verwaltet werden und nur Änderungen der Daten im Protokoll-Kopf übertragen werden. Eine Besonderheit des in [CJ97] beschriebenen Verfahrens ist, daß es auch auf UDP-Datagramme mit anderen Payloads als RTP anwendbar ist. Hier ist aufgrund des kleineren "komprimierten" Protokoll-Kopfes die Einsparung kleiner.

Dieser Ansatz löst das Problem grundsätzlich, da – wenn er auf dem Link-Layer angewendet wird – der Protokoll-Overhead reduziert wird und damit auch kleine PDUs mit hoher Kanalauslastung aus Sicht des Nutzdatenstroms übertragen werden können. Für das hier zu entwickelnde System gilt dies nicht, da der Link-Layer nicht zugänglich ist und jede PDU unter Nutzung von UDP über IP transportiert werden muß. Hierfür existieren bisher keine implementierten Header-Kompressionsverfahren. Dennoch kann durch die Header-Komprimierung für RTP der Protokoll-Overhead für Datenströme mit kleinen PDUs deutlich verringert werden. Dies wird am folgenden Beispiel deutlich.

Wenn ein Audio-Transcoder im Down-Stream den Datenstrom auf GSM-Codierung mit 40 ms Sample-Time begrenzt, werden mit einer Paketrate von 25 Paketen/s UDP-PDUs mit einer Payload von 78 Byte erzeugt und über den ISDN-Kanal übertragen. Hier entstehen unter Vernachlässigung des Bit-Stuffing HDLC-Frames mit 7 Byte HDLC- und PPP-Header und Trailer, 20 Byte IP-Header, 8 Byte UDP-Header, 12 Byte RTP-Header und 66 Byte GSM-codierte Audio-Daten. 66 Byte Nutzdaten stehen somit 40 Byte Protokoll-Overhead gegenüber. Die Serialisierungszeit für den HDLC-Frame beträgt 13.25 ms. Gelingt es, den RTP-Header auf 4 Byte zu komprimieren, würde die Serialisierungszeit auf 12.25 ms reduziert werden.

Die besten Ergebnisse lassen sich danach bei der kombinierten Komprimierung von IP-, UDP- und RTP-Headern erwarten. Die direkte Übertragung des in [CJ97] beschriebenen Verfahrens ist jedoch nicht möglich, da andere Rahmenbedingungen vorliegen. Der MAS erhält aus dem MBone IP-Datagramme, die den in Abbildung 3.11 dargestellen Aufbau haben. In der Skizze sind die dem Gateway auf Anwendung-Ebene zugänglichen Informationen markiert.

Wenngleich Abbildung 3.11 den Protokollkopf einer RTP-PDU darstellt, ist dies ein Spezialfall. Neben RTP-PDUs sind auch allgemeine UDP-PDUs zu transportieren. Anders als bei der in [CJ97] vorgestellten Header-Komprimierung ist dem MBone-Access-Gateway bekannt, ob ein bestimmter Flow ein RTP-Flow oder ein allgemeiner UDP-Flow ist und es kann diese Information nutzen. Somit definiert sich hier ein Flow über die IP-Adresse des Absenders, den Sende-Port, die IP-Zieladresse, den Zielport sowie die Information, ob es sich um einen RTP- oder einen UDP-Flow handelt. Letztere Information hat Einfluß auf die Kompression der Protokoll-Köpfe, die Struktur der Pakete während der Tunnel-Übertragung sowie dem eingesetzten Fehlerbehebungsverfahren.

Das grundlegende Verfahren zur Kompression ist für beide Fälle ähnlich:

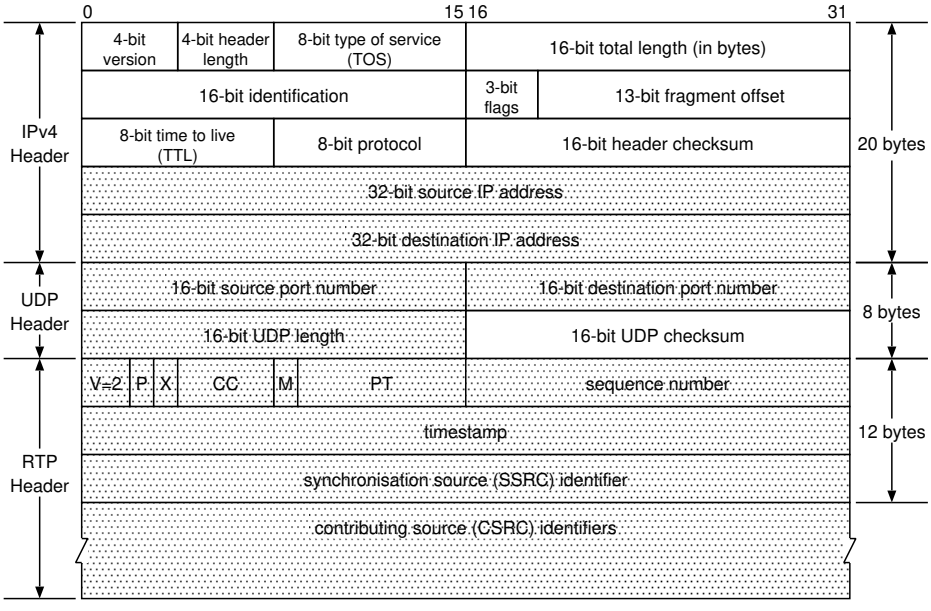

Abbildung 3.11: Header-Struktur von RTP-PDUs auf dem MBone

- Jeder PDU wird für die Übertragung über den Tunnel ein weiterer Protokollkopf vorangestellt, der den Payload-Type auf der Ebene des Tunnel-Protokolls, einen Flow-Identifier sowie ein 4-Bit Sequenznummer umfaßt.

- Die jeweils erste PDU eines jeden Flows wird mit unkomprimierten Headern übertragen, d.h. der Protokollkopf auf der Ebene des Tunnel-Protokolls wird um die IP-Adresse des Absenders, den Sende-Port, eine Media-Stream-Identifikation und einen Port-Index erweitert.

  Die Media-Stream-Identifikation und der Port-Index gestatten es dem Empfänger, am Tunnelende den engültigen Zielport sowie die Zieladresse zu rekonstruieren. Gegenüber der in [CJ97] beschriebenen Übertragung der Zieladresse und des Zielports in unkomprimierten Headern sind 4 Byte weniger zu übertragen.

  RTP-PDUs enthalten darüber hinaus noch den unkomprimierten RTP-Header. Der Empfänger bewahrt den unkomprimierten Protokoll-Kopf als Flow-Kontext auf.

  Durch die Übertragung der unkomprimierten Protokollköpfe wird die Synchronisierung der Kontexte zwischen Sender und Empfänger ermöglicht.

- Darauffolgende Pakete des gleichen Flows enthalten nur noch den Standard-Kopf des Tunnel-Protokolls sowie im RTP-Fall Änderungen einzelner Felder des RTP-Headers, deren Vorhandensein über ein entsprechend maskiertes Bit-Feld angezeigt wird.

  UDP-PDUs können beim Empfänger stets aus dem gespeicherten Kontext rekonstruiert und weitergeleitet werden. Bei RTP-PDUs wird der Kontext durch die empfangenen Änderungen fortgeschrieben und ermöglicht somit ebenfalls die Rekonstruktion des vollen Paketkopfes.

3.2.5.1 Struktur der Tunnel-Pakete

Der immer gleiche Protokoll-Kopf einer RTP oder UDP Tunnel-PDU besteht aus folgenden Komponenten:

Packet-Type
Dieses Feld dient der Markierung der folgenden Payload und ist 3-Bit breit. Die einzelnen Werte haben folgende Bedeutung:

Wert	Bedeutung
000_2	UDP, Header uncompressed, Payload uncompressed
001_2	UDP, Header uncompressed, Payload LZ-compressed
010_2	UDP, Header compressed, Payload uncompressed
011_2	UDP, Header compressed, Payload LZ-compressed
100_2	RTP, UDP-Header uncompressed, RTP-Header uncompressed
101_2	RTP, UDP-Header compressed, RTP-Header uncompressed
110_2	RTP, UDP-Header compressed, RTP-Header compressed
111_2	Service-Packet

Flow-ID-Size
Dieses Feld mit einer Breite von einem Bit zeigt die Breite des *Flow-ID*-Feldes an:

Wert	Bedeutung
0	4-Bit *Flow-ID*
1	12-Bit *Flow-ID*

Eine Ausnahme bilden hier die Service-Pakete. Bei ihnen dient das Flow-ID-Size-Bit nicht zur Beschreibung des Flows, sondern es gibt an, ob der Service-PDU eine weitere PDU folgt oder nicht. Hat das Bit den Wert 0, beinhaltet dieses Tunnel-Paket ausschließlich diese eine Service-PDU. Hat das Bit den Wert 1, folgt nach der Service-PDU eine weitere PDU. Dies kann ebenfalls eine Service-PDU sein, möglich sind aber auch UDP- oder RTP-PDUs.

Flow-ID
Die Breite des Flow-ID-Feldes ist vom Wert des Flow-ID-Feldes abhängig. Im Normalfall wird es für RTP-Flows 4-Bit breit sein, womit 16 parallele Flows unterschieden werden können.

Da die RTP-Flows durch Mixer bzw. Transcoder entstehen, ist in der Regel mit zwei Flows für Audio und Video in beide Richtungen zu rechnen und der Wertevorrat sollte daher für die meisten Anwendungen ausreichend sein.

Bei UDP-Flows, die auch der Übertragung von RTCP-PDUs dienen, ist mit größeren Flow-Zahlen zu rechnen. In diesen Fällen wird das Flow-ID-Bit einen Wert von 1 haben und der Flow wird über ein skalierbares Verfahren codiert:

- Hat das *Most Significant Bit* (MSB) des ersten Flow-Bytes den Wert 0, codieren die folgenden 11 Bit die Flow-ID.

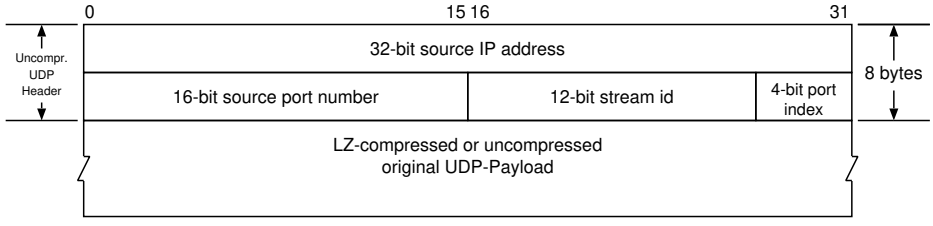

Abbildung 3.12: Unkomprimierter UDP-Header und Payload

- Hat das MSB des ersten Flow-Bytes den Wert 1, codieren das 2. Flow-Byte sowie ein weiteres Extension-Byte die Flow-ID mit 16 Bit.

Auch hier bilden die Service-PDUs eine Ausnahme. Das Flow-ID-Feld ist hier stets 4 Bit breit und gibt nicht den Flow, sondern den Typ der Service-PDU an.

Sequence-Number
Das Sequence-Number-Feld hat eine feste Breite von 4 Bit. Die Sequence-Number dient der Erkennung von Paketverlusten je Flow und kann in besonderen Fällen zur Korrektur der Reihenfolge von Paketen beim Empfänger dienen.

Service-PDUs enthalten keine Sequence-Number.

Das Packet-Type-Feld sowie das Flow-ID-Size-Feld werden den unterliegenden Schichten als 4-Bit-Wert codiert übergeben. Die Flow-ID sowie die Sequence-Number werden dem weiteren Paket vorangestellt und als Payload überreicht. Das Tunnel-System überreicht die Daten transparent an die Kompressionsschicht der Partner-Instanz. Der weitere Aufbau der Datagramme ist vom Packet-Type abhängig.

3.2.5.2 Header-Komprimierung bei UDP-Flows

Der Sender überträgt bei UDP-PDUs initial die Informationen des vollen Paketkopfes. Dazu wird beim Sender ein neuer Flow registriert und der Tunnel-Protokoll-Kopf um die Flow-Definition, wie in Abbildung 3.12 dargestellt, ergänzt.

Diese PDU wird zusammen mit der originalen Payload an den Empfänger übertragen. Weitere PDUs des gleichen Flows werden fortan ohne den unkomprimierten Header als Packet-Type 010_2 oder 011_2 übermittelt. Dabei wird die Sequenz-Nummer monoton erhöht. Bei Überlauf wird erneut mit der Sequenz-Nummer 0 begonnen. Der Anfangswert kann beliebig gewählt werden. Sollte der Empfänger die erste PDU mit der Flow-Definition nicht erhalten haben, kann er eine erneute PDU mit voller Flow-Definition durch die Emission eines Context-State-Requests anfordern. Dieser wird in Intervallen ausgesendet, bis die Flow-Definition erhalten wurde.

Wenn der UDP-Flow verebbt, kann der Sender die Flow-ID nach einer angemessenen Wartezeit im Minutenbereich durch die Emission einer UDP-PDU mit voller Flow-Definition erneut belegen.

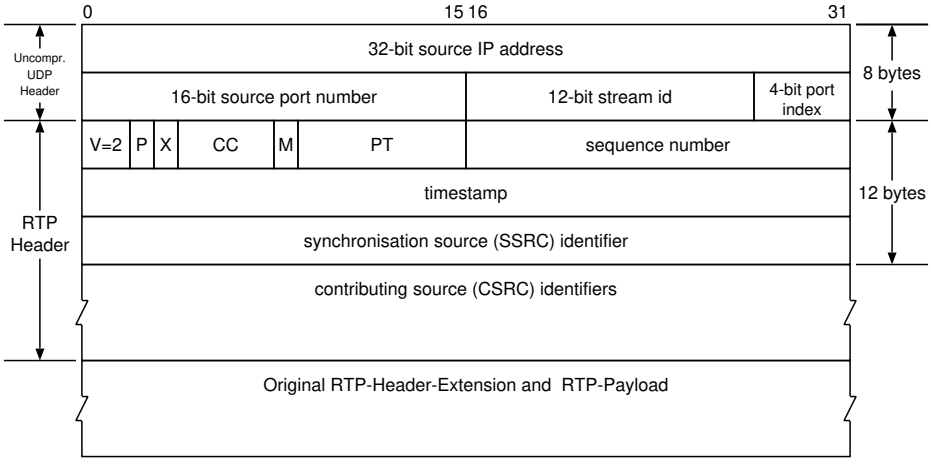

Abbildung 3.13: Unkomprimierter UDP-Header, RTP-Header und Payload

3.2.5.3 Header-Komprimierung bei RTP-Flows

Die Komprimierung von Datenströmen mit RTP-Payload wird in ähnlicher Weise realisiert. Initial wird eine PDU mit voller UDP-Flow-Definition und dem vollem RTP-Header übertragen. Abbildung 3.13 stellt die PDU ohne Tunnel-Protokollkopf dar.

Diese PDU ermöglicht es dem Empfänger, den Kontext für den jeweiligen RTP-Flow vollständig zu initialisieren. Der Sender bewahrt diesen Protokollkopf ebenfalls auf. Erreicht diese PDU den Empfänger nicht, fordert er sie ebenfalls mit einem Context-State-Request an.

Anders als bei der UDP-Header-Komprimierung ändern sich in jeder RTP-PDU einige Komponenten des RTP-Headers. Im einzelnen sind dies:

Version (V)
: Die Versionsbezeichnung bleibt konstant, sie braucht generell nicht übertragen zu werden. Ist die Version nicht 2, muß der Flow als UDP-Flow übertragen werden.

Padding (P)
: Das Padding-Bit im RTP-Header zeigt an, daß die Größe der originären Payload kleiner als die aktuelle Payload ist. Dem RTP-Standard nach können Pakete dieser Struktur entstehen, wenn die Payload blockverschlüsselt übertragen wird und die Nutzdatengrenze nicht auf eine Blockgröße fällt. Eine andere Anwendung ist die Übertragung mehrerer RTP-Payload-Blöcke in einem Paket. Ist das P-Bit gesetzt, zeigt das letzte Byte der Payload an, wieviele Bytes der aktuellen Payload nicht zur originären Payload gehören.

  Gängige Praxis ist, daß bei verschlüsselter Übertragung der Payload auch der Header verschlüsselt wird und daher seine Kompression nicht möglich ist. Der Datenstrom wird als UDP-Datenstrom übertragen.

  Obwohl die heute üblichen RTP-Payload-Typen das Padding-Bit nicht nutzen, kann es nicht ignoriert werden. Zwei Verfahren bieten sich zu seiner Behandlung an:

  - Die Padding-Bytes können bei der Übertragung ausgeschnitten werden. Die resultierende Payload wird dann ohne gesetztes P-Bit transportiert.

- Das P-Bit kann dem Kontext des Flows zugeschlagen werden. Bei einer Änderung ist ein unkomprimierter RTP-Header zu übertragen.

Das zuerst vorgestellte Verfahren hat den Vorteil, daß neben dem Protokoll-Kopf auch die Payload reduziert wird. Es besteht jedoch das Risiko, daß die Payload durch diese Änderung zerstört wird.

Nach der Diskussion dieser Thematik mit den Autoren von [CJ97] wurde der zweite Weg beschritten. Der Internet-Draft [CJ97] wurde entsprechend erweitert.

Header-Extension (X)
Das Header-Extension-Bit bleibt für den Flow unverändert. Es wird nicht im komprimierten Header übertragen. Änderungen an dieser Stelle erfordern die Übertragung eines Pakets vom Typ 101_2. Ist das X-Bit im Kontext gesetzt, wird automatisch die Übertragung einer RTP-Header-Extension am Ende des komprimierten Protokoll-Kopfes erwartet.

Contributing-Source-Count (CC)
Der Inhalt der Contributing-Source-List sowie der Contributing-Source-Count sollte sich zumindest für die Dauer von Talkspurts nicht ändern. Ein Bit im komprimierten Header (C-Bit) zeigt entsprechende Änderungen an. Der Paketkopf enthält dann den neuen CC-Wert sowie die aktualisierte CSRC-Liste.

Marker (M)
Das Marker-Bit wird heute üblicherweise bei der Übertragung von Video-Frames benutzt, um das Ende eines Frames zu markieren (vgl. [TH96]). Daher wird im komprimierten Protokollkopf ein Marker-Bit reserviert (M-Bit).

Payload-Type (PT)
Der Payload-Type bleibt für den Flow konstant und wird daher im komprimierten Protokollkopf nicht berücksichtigt.

Sequence-Number
Die Sequence-Number wird normalerweise bei der Übertragung jeder RTP-PDU um eins inkrementiert. Daher wird beim Empfänger das zu erwartende Sequenz-Inkrement bei der Synchronisierung mit eins initialisiert. Treten andere Inkremente während der Übertragung auf, wird ein entsprechendes Bit im komprimierten Header gesetzt und die Veränderung in einem Feld variabler Länge übertragen. Die Änderung des Sequence-Increment gegenüber eins wird durch ein gesetztes S-Bit im komprimierten Header angezeigt. Auf die Codierung der Werte wird im weiteren genauer eingegangen.

Timestamp
Auch der Zeitstempel ändert sich im Flow über die Zeit. Die Änderung ist jedoch von der Art des Medienstroms abhängig. Bei Audio-Datenströmen ändert sich der Zeitstempel üblicherweise mit konstanten Inkrementen, während bei Video-Datenströmen die RTP-Pakete eines Frames den gleichen Zeitstempel tragen. Darum erscheint es sinnvoll, das zu erwartende Zeitstempel-Inkrement programmierbar zu gestalten. Bei der Synchronisierung zwischen Sender und Empfänger wird das zu erwartende Inkrement mit 0 initialisiert. Ändert sich das Inkrement während der Übertragung, wird es in einem Feld variabler Länge an den Empfänger übertragen. Sein Vorhandensein wird durch ein gesetztes T-Bit

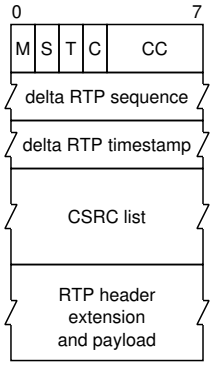

Abbildung 3.14: Komprimierter RTP-Header und Payload

im komprimierten Header signalisiert. Dieses neue Inkrement dient nun für die weiteren PDUs als Voreinstellung, d.h. wenn das T-Bit im komprimierten Header nicht gesetzt ist, gilt das zuletzt kommunizierte Inkrement.

Synchronisation-Source (SSRC)
: Die Synchronisation-Source ändert sich innerhalb des Flows nicht. Daher wird sie in komprimierten Headern nicht übertragen.

Contributing-Source-Identifiers (CSRC)
: Die Liste der Contributing-Source-Identifiers kann sich, wie bereits ausgeführt, an den Grenzen von Talkspurts ändern. Diese Änderung wird durch das C-Bit im komprimierten Header angezeigt. Ist dieses Bit gesetzt, wird zwischen komprimierten Protokoll-Kopf und der optionalen Header-Extension die aktualisierte Liste der Contributing-Source-Identifiers übertragen.

  Das C-Bit ist nötig, um einen Contributing-Source-Count von Null von einem nicht geänderten Contributing-Source-Count unterscheiden zu können.

Die resultierende Struktur des komprimierten Headers ist in Abbildung 3.14 dargestellt.

Sind Änderungen des RTP-Timestamps sowie der RTP-Sequenznummer zu übertragen, bietet sich der Einsatz einer Entropiecodierung an: Es ist davon auszugehen, daß kleine Änderungen häufig, größere Änderungen aber nur selten auftreten. Die folgende Tabelle gibt die Codierung an. Sie wurde unmodifiziert aus [CJ97] übernommen:

Decimal	Hex
-16384	C0 00 00
:	:
-129	C0 3F 7F
-128	80 00
:	:
-1	80 7F
0	00
:	:
127	7F
128	80 80
:	:
16383	BF FF
16384	C0 40 00
:	:
4194303	FF FF FF

Die Wandlung zwischen dem codierten Wort und dem originären Wert erfolgt nach folgendem Verfahren:

- Ist das MSB des ersten Byte nicht gesetzt, handelt es sich um einen in einem Byte codierten Wert, die verbleibenden 7 Bit des Bytes entsprechen dem originären Wert.
- Ist das MSB des ersten Byte gesetzt und das nächst niederwertigere Bit nicht gesetzt, handelt es sich um einen in zwei Bytes codierten Wert. Die verbleibenden 14 Bit dienen der Codierung des originären Wertes. Ergibt das Auslesen dieser 14 Bit in Network-Byteorder einen Wert < 128, ist das Endergebnis eine Zahl zwischen -128 und -1. Ist das Zwischenergebnis ≥ 128, so stellt sie den originären Wert im Bereich zwischen 128 und 16383 dar.
- Sind die beiden höchstwertigsten Bits des ersten Bytes gesetzt, handelt es sich um einen in 3 Bytes codierten Wert. Die verbleibenden 22 Bit dienen der Codierung des originären Wertes. Ergibt die Auswertung dieser Bits in Network-Byteorder eine Zahl < 16384, so handelt es sich beim originären Wert um eine negative Zahl im Wertebereich zwischen $-16384 \ldots -129$. Ist das Zwischenergebnis ≥ 16384, gibt es den originären Wert im Wertebereich zwischen $16384 \ldots 4194303$ an.

Die Umwandlung vom originären Wert zum codierten Wort erfolgt entsprechend.

3.2.5.4 Fehlerbehebung bei RTP

Die im weiteren vorgestellte Kompression operiert auf der Ebene der IP-Schicht, wohingegen die bereits zitierte Header-Komprimierung nach [CJ97] auf dem Link-Layer arbeitet. Daraus ergeben sich drei wesentliche Unterschiede. Auf dem Link-Layer sind Paketverluste selten. Nachhaltige Verzögerungen der Pakete durch Queues sind nicht zu erwarten. Zudem ist sicher, daß die Pakete in der Reihenfolge beim Empfänger angelangen, in der sie beim Sender abgeschickt wurden. Auf der IP-Schicht sind Paketverluste unvermeidbar und in der Analyse wurde gezeigt, daß PDUs im betrachteten Szenario stark verzögert werden können. Wenngleich das Internet-Protokoll keine Auslieferung der Daten in Sendereihenfolge gewährleistet, haben Tests im betrachteten Szenario gezeigt, daß die Empfangsreihenfolge praktisch stets der Sendereihenfolge entspricht.

Die Unterschiede wirken sich auch in den Verfahren zur Fehlerbehebung aus. Bei der Link-Layer-Kompression wird folgender Regelkreis realisiert: Der Empfänger erkennt verloren gegangene Pakete an der Unstetigkeit der Sequenznummern. Bis zum Empfang einer für die erneute Synchronisation zwischen Sender und Empfänger geeigneten PDU werden dieses und alle weiteren Datenpakete des Flows verworfen. Gleichzeitig wird eine Context-State-Nachricht an den Sender übertragen, um den Verlust des Kontextes anzuzeigen. Der Sender reagiert auf diese Nachricht mit der Übertragung einer unkomprimierten PDU. Der Empfang dieser PDU ermöglicht die Synchronisierung des Flow-Kontextes zwischen Sender und Empfänger.

Die Adaptierung dieses Verfahrens für die Header-Komprimierung über IP ist nicht möglich. Paketverluste treten im Fall von Stauungen ein, die zuvor die Paketlaufzeiten deutlich steigen lassen. Damit würde die Antwort auf eine Context-State-Nachricht, eine unkomprimierte PDU,

SPT	PDU-Typ
0000_2	Context-State-Request
0001_2	Timestamp
0010_2	Tunnel-Report
0011_2	Encapsulation
0101_2	Counter-Reset-Request
0110_2	Counter-Reset-Confirmation

Tabelle 3.5: Service-PDU Kennungen

erst mit großer zeitlicher Verzögerung beim Empfänger angelangen. Zuerst müssen die in der Queue gestauten Pakete ausgeliefert werden.

Günstiger ist es daher, anstelle einer Fehlerbehebungsprozedur ein fehlertolerantes Verfahren in den Vordergrund zu stellen. Dazu werden in jeden Flow gelegentlich PDUs mit unkomprimierten Protokoll-Köpfen eingefügt, die dem Empfänger die Synchronisierung auch ohne Rückkopplung erlauben. Die Frequenz dieser ausgezeichneten Pakete ist vom Datenstrom abhängig. Bei RTP-Datenströmen gilt, daß Audio-Daten wichtiger als Video-Datenströme sind und daher häufiger Synchronisationspunkte gegeben werden sollten.

Reine UDP-Datenströme benötigen, wie bereits dargestellt, keine implizite Übertragung der unkomprimierten Protokollköpfe, da die Komprimierung nur den Flow bezeichnet. Es sind keine Änderungen zu übertragen. Beim Verlust initialer Flow-Pakete, die einen unkomprimierten Header beinhalten, ist die explizite Wiederanforderung des Flow-Kontextes mittels Context-State-Nachrichten erforderlich.

Dem Problem von Unterschieden in Sende- und Empfangreihenfolge wird entgegengewirkt, indem der Empfänger auf Basis der Sequenznummern der Pakete die Ordnung des Paketstroms herstellt.

3.2.5.5 Service-Packets

Neben den bereits vorgestellten Klassen von Tunnel-PDUs gibt es noch die Klasse der Service-Packets. Sie unterscheiden sich von den vorhergehenden Typen, in dem nach dem Packet-Type nicht direkt eine Flow-ID folgt. Stattdessen wird eine 4-Bit breite Service-PDU-Typ-Kennung (SPT) übertragen. Tabelle 3.5 beschreibt die unterschiedlichen Service-PDUs.

In den folgenden Abschnitten wird der Aufbau der Service-PDUs dargestellt.

Context-State-Request

Der Context-State-Request kann jederzeit vom Empfänger zum Sender geschickt werden, um die Synchronisierung eines einzelnen oder mehrerer Flow-Kontexte zu initiieren.

Die PDU enthält Informationen darüber, ob es sich um einen RTP- oder einen allgemeinen UDP-Flow handelt sowie die jeweilige Flow-ID. Der Sender reagiert auf den Empfang einer solchen

Feld	Bit-Breite	Beschreibung
PT = 111_2	3	*Tunnel-Packet-Type*. Der Wert ist 111_2 für eine Service-PDU.
FLAG	1	Dieses Bit ist gesetzt, wenn weitere PDUs im Paket folgen.
SPT = 0000_2	4	*Service-Packet-Type*. Der Wert 0000_2 markiert einen *Context-State-Request*.
CDs	variabel	Context-Descriptions.

Tabelle 3.6: Struktur des Context-State-Request

Feld	Bit-Breite	Beschreibung
Flow-Type	1	Gibt an, ob es sich um einen UDP-Flow (0) oder einen RTP-Flow handelt.
Flow-ID-Size	1	Beinhaltet die *Flow-ID-Size* gemäß der Definition auf Seite 100.
M-Flag	1	Ein Wert von 1 gibt an, daß weitere Context Descriptions folgen.
Reserved	1	Dieses Feld wird zur Zeit nicht benutzt.
Flow-ID	4, 12, 20	Dieses Feld bezeichnet den Flow, dessen Kontext zu synchronisieren ist. Seine Bit-Breite ist variabel und entspricht den der Definition auf Seite 100.

Tabelle 3.7: Struktur der Context-Descriptions

PDU mit der sofortigen Zusendung einer entsprechenden Tunnel-PDU mit unkomprimierten Protokoll-Köpfen.

Den Aufbau eines Context-State-Request beschreibt Tabelle 3.6.

PDUs dieses Typs werden, wenn möglich, mit Nutzdaten-PDUs oder Service-PDUs zusammen in einem Tunnel-Paket übertragen. Folgt auf den Context-State-Request eine weitere PDU, ist $FLAG = 1$. Andernfalls ist $FLAG = 0$ und die PDU wird um 4 Füll-Bits erweitert.

Die Context-Descriptions sind das wesentliche Element des Context-State-Request. Ihren Aufbau beschreibt Tabelle 3.7.

Context-State-Requests werden beim Empfänger vom Dekomprimierungsmodul erzeugt. Sie werden dem Übertragungssystem durch eine besonders ausgezeichnete Queue zugeführt, die mit höchster Priorität behandelt wird.

Timestamp

Die Einflechtung von Zeitstempeln in den Datenstrom ist für die Bestimmung der Übertragungsqualität und damit für die Regelung des Systems essentiell. Die grundlegenden Konzepte und Anforderungen an die Übertragung der Zeitstempel wurden in Abschnitt 3.2.2.3, Seite 84ff, dargestellt. Hier wird nun auf das Format der Zeitstempel eingegangen.

Zeitstempel werden vom Output-Modul des Senders generiert und optional in den Datenstrom

Feld	Bit-Breite	Beschreibung
PT = 111_2	3	*Tunnel-Packet-Type*. Der Wert ist 111_2 für eine Service-PDU.
FLAG	1	Dieses Bit ist gesetzt, wenn weitere PDUs im Paket folgen.
SPT = 0001_2	4	*Service-Packet-Type*. Der Wert 0001_2 steht für einen Timestamp.
TS	24	24-Bit Timestamp gemäß Abschnitt 3.2.2.3, Seite 84ff.

Tabelle 3.8: Struktur eines Timestamp

eingebettet. Das Input-Modul des Empfängers erkennt Zeitstempel im Datenfluß und entfernt sie vor der weiteren Verarbeitung der Daten. Es ist daher wichtig, daß die Zeitstempel so eingebettet werden, daß sie leicht entfernt werden können. Tabelle 3.8 beschreibt den Anfang einer Tunnel-PDU, die einen Zeitstempel beinhaltet.

Die Größe der Timestamp-PDU beträgt 32-Bit. Damit ist sichergestellt, daß sich diese PDU leicht einer Service-PDU, einer RTP-Tunnel-PDU oder einer UDP-Tunnel-PDU voranzustellen ist. Problematisch ist die Übertragung von Zeitstempeln, wenn $FLAG = 0$ ist, d.h. keine weiteren PDUs im Paket enthalten sind. Für diesen Fall gilt, daß die PDU um 4 Füll-Bits erweitert wird.

Tunnel-Report

In Abschnitt 3.2.2.4, Seite 86, wurde die Datenratenregelung des Systems detailliert vorgestellt. Ein wesentliches Element ist die Rückkopplung vom Empfänger zum Sender über Tunnel-Reports. In Abschnitt 3.2.4, Seite 96, wurde der Inhalt der Tunnel-Reports zusammengefaßt. Hier wird nun die Einbettung dieser Reports in Service-PDUs dargestellt.

An Tunnel-Reports werden die gleichen Anforderungen wie an Timestamp-PDUs gestellt. Sie werden kooperativ vom Input- und Output-Modul des Empfängers erstellt und sollen möglichst als Option übertragen werden, wenngleich in Ruhe-Phasen, während derer keine Nutzdaten vermittelt werden, die gelegentliche Generierung von Tunnel-Reports erfolgt und alleine in Tunnel-Paketen zu übertragen sind.

Tabelle 3.9 zeigt den Anfang einer Tunnel-PDU, die zum Transport eines Tunnel-Reports dient. Die Gesamtlänge eines Tunnel-Reports beträgt 31 Byte. Auch hier gilt, daß die Tunnel-Reports sich von ihrer Struktur gut in ein Tunnel-Paket mit weiterer Payload integrieren lassen.

Encapsulation

Encapsulation ist eine weitere Optimierung in bezug auf die Auslastung des Übertragungskanals. Das Verfahren wird in Abschnitt 3.2.6, Seite 112 vorgestellt und weitergehend behandelt. Da die Encapsulation Service-PDUs generiert, wurde sie zur Vollständigkeit hier erwähnt.

Ein wesentlicher Unterschied zu den anderen Service-Packet-Types besteht darin, daß PDUs

Feld	Bit-Breite	Beschreibung
PT = 111_2	3	*Tunnel-Packet-Type*. Der Wert ist 111_2 für eine Service-PDU.
FLAG	1	Dieses Bit ist gesetzt, wenn weitere PDUs im Paket folgen.
SPT = 0010_2	4	*Service-Packet-Type*. Der Wert 0010_2 steht für einen Tunnel-Report.
0000_2	4	4 Bit reservierter Bereich (zur Vermeidung von Halb-Byte-Shifts).
TS	32	Timestamp des Senders (Sekunden seit dem 1. Januar 1970)
MSFRAC	8	Millisekunden Bruchteil des TS (Einheit: 1/256 s)
PCS	32	Anzahl der gesendeten Pakete
BCS	32	Anzahl der gesendeten Payload-Bytes
PCR	32	Anzahl der empfangenen Pakete
BCR	32	Anzahl der empfangenen Payload-Bytes
LR	8	Kurzzeitige Paketverlustrate
CPL	24	Anzahl der nicht empfangenen Pakete
TQ	16	Δt_Q in ms
CMAX	16	C_{max} in $10^2 bps = 0.1 kbps$
0000_2	4	4 Bit reservierter Bereich (zur Vermeidung von Halb-Byte-Shifts).

Tabelle 3.9: Struktur eines Tunnel-Report

Feld	Bit-Breite	Beschreibung
PT = 111_2	3	*Tunnel-Packet-Type*. Der Wert ist 111_2 für eine Service-PDU.
FLAG = 0_2	1	Dieses Bit ist gesetzt, wenn weitere PDUs im Paket folgen.
SPT = 0101_2	4	*Service-Packet-Type*. Der Wert 0101_2 steht für einen *Counter-Reset-Request*.
PAD = 000_{16}	12	Padding Bits

Tabelle 3.10: Struktur eines Counter-Reset-Request

Feld	Bit-Breite	Beschreibung
PT = 111_2	3	*Tunnel-Packet-Type*. Der Wert ist 111_2 für eine Service-PDU.
FLAG = 0_2	1	Dieses Bit ist gesetzt, wenn weitere PDUs im Paket folgen.
SPT = 0110_2	4	*Service-Packet-Type*. Der Wert 0110_2 steht für eine *Counter-Reset-Confirmation*.
PAD = 000_{16}	12	Padding Bits

Tabelle 3.11: Struktur einer Counter-Reset-Confirmation

dieses Typs nicht als Option übertragen werden, d.h. nach einer Encapsulation-PDU kann keine weitere Service- oder Payload-PDU im Tunnel-Paket folgen.

Encapsulation-PDUs haben einen SPT-Wert von 0011_2.

Counter-Reset-Request

Ein Counter-Reset-Request wird vom MAS oder MAG an die jeweilige Partnerinstanz gesendet, wenn einer der Zähler für die Anzahl der gesendeten Pakete, die Summe der gesendeten Bytes, die Anzahl der empfangenen Pakete oder die Summe der empfangenen Bytes überläuft (vgl. Abschnitt 3.2.3, Seite 93ff.). Gleichzeitig wird die Vermittlung aller Nutzdatenströme unterbrochen. Pakete dieses Typs werden wiederholt übertragen, bis die Partnerinstanz mit einer Counter-Reset-Confirmation antwortet. Danach wird mit der Übertragung der Nutzdatenströme fortgefahren. Gleichzeitig werden alle Sende- und Empfangszähler auf 0 zurückgesetzt.

Tabelle 3.10 beschreibt den Aufbau einer Counter-Reset-Request PDU.

Counter-Reset-Confirmation

Eine Counter-Reset-Confirmation wird vom MAS oder MAG als Reaktion auf den Empfang eines Counter-Reset-Requests generiert. Gleichzeitig werden alle Sende- und Empfangszähler auf 0 zurückgesetzt.

Tabelle 3.11 beschreibt den Aufbau einer Counter-Reset-Confirmation PDU.

3.2.5.6 Vergleich zur *IP/UDP/RTP Header-Compression for Low-Speed Serial Links*

In den vorangegangenen Abschnitten wurde mehrfach der in Entwicklung befindliche Internet-Standard *IP/UDP/RTP Header-Compression for Low-Speed Serial Links* [CJ97] referenziert. In diesem Abschnitt wird abschließend das dort dargestellte Verfahren mit dem in dieser Arbeit entwickelten Verfahren verglichen.

Der wichtigste Unterschied zwischen den Verfahren ist, daß die Header-Komprimierung nach [CJ97] für den Einsatz auf der Schicht des Link-Layers entworfen wurde, wohingegen das hier entwickelte Verfahren auf der Transportschicht operiert. Daraus ergibt sich eine Reihe von Unterschieden, die in Abschnitt 3.2.5.4, Seite 105, detailliert vorgestellt wurden.

Als Konsequenz daraus wird die Notwendigkeit zur Fehlerbehebung beim Verlust des Kontextes weitgehend vermieden. Dazu dienen die in Intervallen durchgeführte Synchronisierung des Kontextes und die Wiederherstellung der Sendereihenfolge beim Empfänger auf der Basis der Sequenznummern der Pakete. Bei dem in [CJ97] vorgeschlagenen Verfahren wird im Fehlerfall durch Aussenden eines Context-State-Requests die Synchronisierung des Kontextes explizit angefordert. Die Wiederherstellung der Sendereihenfolge ist nicht erforderlich. Der jeweilige Ansatz zur Fehlerbehebung bzw. Fehlervermeidung ist eine Optimierung mit Blick auf die unterschiedlichen Randbedingungen.

Ein weiterer wichtiger Unterschied zwischen den Ansätzen ist, daß das in [CJ97] entwickelte Verfahren keine Meta-Informationen über den Inhalt der jeweiligen Flows besitzt. Es muß für RTP- und allgemeine UDP-Flows gleichermaßen geeignet sein. Dem Mbone-Access-Gateway sind diese Meta-Informationen zugänglich und werden genutzt. Ein Ergebnis ist, daß RTP-Flows mit einer geringeren Zahl von Flow-ID-Bits übertragen werden können und somit auch in Sitzungen mit vielen Teilnehmern die Flows mit großem Datenvolumen optimal komprimiert werden können.

Neben den Unterschieden haben die Verfahren auch Gemeinsamkeiten. Dies ist zuerst einmal das Header-Komprimierungsverfahren an sich. Das hier entwickelte Verfahren orientiert sich dabei stark an dem in [CJ97] vorgeschlagenen Verfahren. Allerdings sind die Verfahren nicht kompatibel zueinander.

Die wichtigste Gemeinsamkeit ist jedoch die ganzheitliche Betrachtung der Protokoll-Stapel. Anders als im *Open Systems Interconnection*-Modell der ISO [Tan90, Seite 17ff.] werden die Schichten nicht mehr isoliert betrachtet, sondern die Komprimierung erfolgt über Schichten hinweg. Beim Verfahren nach [CJ97] werden die Schichten des Internet-Protocol, des User-Datagram-Protocol sowie des Real-Time-Transport-Protocol zusammengefaßt. Bei dem in dieser Arbeit entwickelten Verfahren sind es das User-Datagram-Protocol, das Real-Time-Transport-Protocol sowie die Payload des jeweiligen Transportprotokolls.

Die ganzheitliche Betrachtung von Protokoll-Stapeln zur optimierten Bearbeitung des Datenstroms wird als *Integrated Layer Processing* (ILP) bezeichnet und wurde von Clark und Tennenhouse in [CT90] zusammen mit dem *Application Layer Framing* (ALF), auf dem RTP basiert, vorgestellt. Ein große Zahl neuerer Konzepte basiert auf ILP, so z.B. auch IP-Switching [Cis97].

Neben den hier genannten Punkten gibt es eine Vielzahl individueller Merkmale, die von untergeordneter Relevanz sind und daher nicht genannt werden.

Letztlich ist darauf hinzuweisen, daß das hier entwickelte Verfahren auf der Verwendung des Internet-Protokolls in der Version 4 basiert. Die Nutzung des Internet-Protokolls in der Version 6 ist grundsätzlich möglich, die aktuelle Implementierung berücksichtigt es jedoch nicht. Das in [CJ97] vorgeschlagene Verfahren bezieht neben dem Internet-Protokoll in der Version 4 auch das Internet-Protokoll in der Version 6 explizit mit ein.

3.2.6 Encapsulation

3.2.6.1 Grundlegendes Konzept

Weitere Optimierungen lassen sich nur erreichen, wenn das Ziel, eine möglichst geringe Übertragungszeit pro PDUs zu erreichen, aufgegeben wird. Die Optimierung erfolgt also auf Kosten der Interaktivität. Dieser Ansatz scheint auf den ersten Blick fragwürdig, gründet sich aber auf die Beobachtung, daß ein großer Teil der interpersonellen Kommunikation auf die passive Verfolgung interaktiver Sitzungen verwendet wird. Als Beispiel können hier die im Analyse-Teil in Abschnitt 2.4.3, Seite 39ff, untersuchten MBone-Konferenzen dienen.

Wird eine MBone-Sitzung passiv verfolgt, ist es nicht entscheidend, ob die Echtzeitdatenströme einige Millisekunden eher oder später ausgespielt werden. Wichtiger ist dem Nutzer eine möglichst optimale Wiedergabequalität. Gleichzeitig soll es ihm möglich sein, ohne weitere Vorkehrungen oder Konfigurationsmaßnahmen aus der passiven Rolle in die aktive Rolle zu wechseln.

Das MBone-Access-Gateway unterstützt die passive Nutzung des Systems zur Verbesserung der Wiedergabequalität, indem für beide Hauptrichtungen, Up- und Down-Stream, ein Systemzustand eingeführt wird. Dieser Zustand entscheidet darüber, ob der Tunnel in der jeweiligen Richtung im Lecture-Mode oder im Interactive-Mode betrieben wird.

Befindet sich der Tunnel im Interactive-Mode, werden die eintreffenden Datagramme möglichst schnell weitergeleitet, d.h. es werden ausschließlich die vorstehend beschriebenden Kompressionsverfahren angewendet. Jede eingegangene PDU wird als eigene Tunnel-PDU weitergeleitet.

Wird der Tunnel im Lecture-Mode betrieben, werden die in den Warteschlangen zur Übertragung anstehenden PDUs zu größeren Tunnel-Paketen aggregiert und gemeinsam übertragen. Damit kann ein größerer Teil des Video-Datenstroms zum Empfänger übertragen werden. Die dafür notwendige Ersparnis ergibt sich, indem bei der aggregierten Übertragung von N PDUs $N - 1$ IP-Header und Teile des UDP-Headers eingespart werden können. Der Transport der aggregierten PDUs erfolgt unter der Nutzung von Tunnel-Encapsulation-PDUs, einer Klasse der Tunnel-Service-Packets.

Wie bereits eingangs angemerkt, ist die automatische Umschaltung der Tunnel von einem in den anderen Betriebsmodus wichtig. Typischerweise wird der Down-Stream-Tunnel[26] initial im Lecture-Mode betrieben, der Up-Stream-Tunnel[27] hingegen im Interactive-Mode. Solange der Benutzer keine RTP-Daten oder andere als RTCP-Daten über UDP versendet, bleibt dieser Status erhalten. Beginnt der Nutzer Daten zu übertragen, wird im Down-Stream-Tunnel der

[26] Unter dem Down-Stream-Tunnel wird der Tunnel vom Internet zum Heimarbeitsplatz verstanden.
[27] Unter dem Up-Stream-Tunnel wird der Tunnel vom Heimarbeitsplatz zum Internet verstanden.

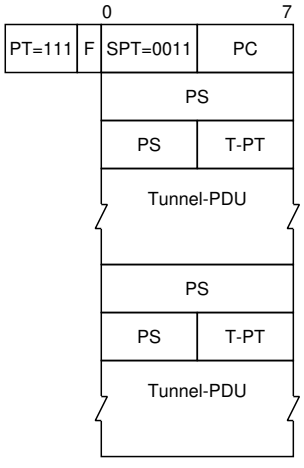

Abbildung 3.15: Struktur einer Encapsulation-PDU

Betriebsmodus vom Lecture-Mode auf den Interactive-Mode umgeschaltet. Diese Konfiguration bleibt unverändert, bis der Nutzer über einen längeren Zeitraum keine Daten mehr in Up-Stream-Richtung überträgt. Diese Zeitdauer ist ein Konfigurationsparameter und sollte in der Größenordnung mehrerer Minuten eingestellt werden, um häufige Laufzeitveränderungen zu verhindern.

3.2.6.2 Struktur der Encapsulation-PDUs

Wird der Tunnel im Lecture-Mode betrieben, wird überprüft, ob mehr als eine PDU zum Transport ansteht. Ist dies der Fall, wird überprüft, ob durch Aggregation eine Encapsulation-PDU entsteht, die ohne Fragmentierung transportiert werden kann. Ist auch diese Rahmenbedingung erfüllt, werden die zur Übertragung anstehenden PDUs wie in Abbildung 3.15 dargestellt zusammengefaßt.

Der Service-Packet-Type 0011_2 zeigt an, daß es sich um eine Encapsulation-PDU handelt. Darauf folgt ein 4 Bit breiter Packet-Counter, der die Anzahl der im weiteren Paket enthaltenen PDUs angibt. Er ermöglicht die Übertragung von 1-16 Paketen pro Encapsulation-PDU.

Jedem eingebetteten Paket geht dann ein 12 Bit breites Packet-Size-Feld (PS) voraus. Es gibt die Größe des Pakets in Bytes an und hat einen Wertebereich von 1-4096. Daran schließt sich der Tunnel-Packet-Type der eingebetteten PDU an. Konkrete Werte sind der Tabelle auf Seite 100 zu entnehmen. Letztlich folgt die eigentliche Payload des eingebetteten Pakets.

Weitere eingebettete PDUs folgen direkt am Ende dieser PDU, erneut mit einem Packet-Size-Feld.

3.2.7 Schleifenerkennung

Ziel der Schleifenerkennung ist es nicht, absichtlich erzeugte Schleifen zu verhindern. Dies kann und soll von diesem System nicht geleistet werden. Vielmehr geht es um die Unterbindung

versehentlich erzeugter Schleifen. Diese Schleifen entstehen zumeist dann, wenn Systeme durch Tunnel verbunden werden, die ohnedies schon über den MBone gekoppelt sind.

Zur Erkennung solcher Schleifen bieten sich drei Verfahren an: Das erste Verfahren basiert auf der Emission von Datagrammen auf festgelegten Multicast-Transportadressen am Multicast-Access-Server und seinem Gegenstück am Heimarbeitsplatz. Empfängt eine Instanz die emittierten Daten erneut, liegt eine Schleife vor. Dieses Verfahren verursacht zusätzliche Datenströme, die möglicherweise über den weltweiten MBone übertragen werden und ist daher ungeeignet.

Alternativ kann der MBone-Access-Server die RTP-Datenströme beobachten und die im RTP-RFC 1889 dargestellte Schleifenerkennung implementieren. Dieses sichere Verfahren erfordert jedoch die Implementierung des RTP und sollte daher möglichst vermieden werden.

Einfacher ist es, den Mixern und Transcodern für RTP-Datenströme die Schleifenerkennung zu überlassen und lediglich entsprechende Fehlermeldungen auszuwerten, die zur Einstellung der Datenvermittlung führen. Dieses Verfahren setzt jedoch die Implementierung der *Loop Detection* in Mixern und Transcodern voraus. Die hier eingesetzten Werkzeuge realisieren eine solche Loop-Detection. Daher wurde dieser Weg für die Implementierung gewählt.

3.2.8 Behandlung von SAP/SDP-Datenströmen

Die Vermittlung von Sitzungsankündigungen für MBone-Sitzungen unterscheidet sich von anderen Datenströmen in bezug auf die Behandlung des *Time to Live*-Feldes (TTL).

Das Time-to-Live-Feld kontrolliert bei IP-Multicast in Kooperation mit dem Multicast-Routing den Verbreitungsbereich der Datagramme. Eine zusammenfassende Erläuterung der zugrundeliegenden Mechanismen kann [Grü96] entnommen werden.

Während für einen einzelnen Audio- oder Video-Datenstrom die Belegung dieses Feldes beim Start des jeweiligen Tools festgelegt wird und für die Laufzeit des Programms konstant ist, gilt dies für SAP/SDP-Flows nicht.

Der Nutzer am Heimarbeitsplatz eine Sitzung legt bei der Erstellung einer Sitzungsankündigung fest, mit welchem TTL-Wert sie ausgesendet wird. Die TTL-Felder der Medien-Werkzeuge werden mit dem gleichen TTL-Wert belegt. So ist sichergestellt, daß die Empfänger der Sitzungsankündigung an der Sitzung partizipieren können. Alle Sitzungsankündigungen werden mit variierenden TTLs an eine Multicast-Transportadresse gesendet.[28] Der TTL-Wert der auf diese Multicast-Adresse gesendeten Pakete kann sich von Paket zu Paket ändern.

Das folgende Beispiel verdeutlicht die Problematik: Der Nutzer am Heimarbeitsplatz generiert zwei Sitzungsankündigungen. Die erste Sitzung dient der Besprechung mit Kollegen an der selben Hochschule. Daher wird ein TTL von 15 gewählt. Die zweite Ankündigung betrifft einen Seminarvortrag, der deutschlandweit zu empfangen sein soll und für den daher ein TTL-Wert von 48 festgelegt wird.[29] Beide Sitzungen werden fortan vom Announcement-Tool des Nutzers angekündigt, jede in einem Paket mit unterschiedlichem TTL.

[28] Die zugehörige IP Class-D Adresse lautet 224.2.127.254 (sap.mcast.net). Der Empfangsport ist der Port 9875.
[29] Für die Diskussion von TTL-Werten siehe [BFS97], Seite 18ff.

Konventionelle Multicast-Router, wie der DVMRP-Router *mrouted*, haben Zugriff auf das TTL-Feld des IP-Datagramms (vgl. Abbildung 3.11, Seite 99) und können die Pakete entsprechend weiterleiten oder verwerfen. Dem MBone-Access-Gateway ist das TTL-Feld des IP-Datagramms nicht zugänglich. Daher muß eine alternative Strategie zur Besetzung des TTL-Feldes bei der Weiterleitung der Multicast-Pakete gefunden werden. Das Verhalten des MAGW bei der Vermittlung üblicher Medienströme ist wie folgt:

Down-Stream
: Jede empfangene PDU wird vom MAS an das MAG weitergeleitet. Das MAG leitet die Pakete mit einen konfigurierbaren aber konstanten TTL weiter.

Up-Stream
: Der Nutzer hat beim Start des jeweiligen Medienwerkzeugs einen TTL-Wert festgelegt. Dieser wird bei der Anforderung des Datenstroms vom MAG an den MAS übermittelt. Erhält der MAS vom MAG diesem Datenstrom zugehörige Pakete, werden sie mit dem festgelegten TTL weiter gesendet.

Für Partner im regulären Internet entsteht somit der Eindruck, daß die Datenströme des Nutzers am Heimarbeitsplatz ihre Quelle am MAS hätten. Umgekehrt erhält der Nutzer am Heimarbeitsplatz alle Pakete eines Datenstroms, die vom MAS empfangen wurden.[30]

Zur korrekten Behandlung von SAP/SDP-Datenströmen ist dieses Verfahren nur in Down-Stream-Richtung anwendbar. Der Wert des TTL-Feldes für SAP/SDP-Pakete, die vom MAS im Auftrag des Nutzers am Heimarbeitsplatz emittiert werden, muß von Paket zu Paket durch den MAS bestimmt werden.

Der einfachste Weg hierzu besteht darin, daß der MAS jede in den regulären MBone zu übertragende SAP/SDP-PDU analysiert und den erforderlichen TTL-Wert bestimmt, indem das Maximum der TTL-Werte der in der Sitzung registrierten Medienströme bestimmt und die SAP/SDP-PDU mit diesem TTL-Wert in den MBone gesendet wird.

Letztlich hat dieses Verfahren eine Unzulänglichkeit. Sie betrifft das Announcement von Sitzungen, die auf SAP-Ebene verschlüsselt werden. Da der entsprechende Schlüssel ausschließlich dem Announcement-Tool beim Nutzer zugänglich ist, kann das Analyse-Modul im MAS die SDP-PDU nicht analysieren und daher auch nicht den erforderlichen TTL-Wert ermitteln. Für diesen Sonderfall wird zwischen MAG und MAS ein Standard-TTL-Wert vereinbart, mit dem PDUs dieser Klasse ausgesendet werden.

3.2.9 Vermittlung verschlüsselter Datenströme

3.2.9.1 Kontext der Behandlung verschlüsselter Datenströme

Dieser Abschnitt behandelt die Vermittlung verschlüsselter Datenströme durch das MBone-Access-Gateway. Die Relevanz dieses Aspekts leitet sich daraus ab, daß Vertaulichkeit bei

[30] Dies gilt unter dem Vorbehalt, daß die Daten nicht durch das Regelungssystem, einen Mixer oder einen Transcoder verworfen werden.

MBone-Konferenzen durch Verschlüsselung der Datenströme erzielt wird (vgl. Abschnitt 2.4, Seite 33ff.).

Aus Sicht des Nutzers besteht der Unterschied zwischen der Teilnahme an einer öffentlichen, unverschlüsselten Sitzung und einer vertraulichen, verschlüsselten Sitzung darin, daß als Parameter beim Start des jeweiligen Medien-Werkzeugs oder nach dem Programmstart in einem Konfigurationsdialog ein Schlüssel für diesen Medienstrom angegeben wird. Der Schlüssel ist für alle Teilnehmer gleich und wird im Vorfeld der Sitzung über eine verschlüsselte E-Mail, z.B. unter Nutzung des *Pretty Good Privacy*-Systems [Zim95], ausgetauscht.

Von besonderer Relevanz für das in dieser Arbeit zu entwickelnde System sind die technischen Aspekte der verschlüsselten Übertragung der Datenströme. Im Vordergrund stehen Datenströme, die als Transportprotokoll das Real-Time-Transfer-Protocol (RTP) nutzen. Ausgangspunkt der Betrachtungen ist der RTP-Standard [SCFJ96, Abschnitt 9]. Hier werden Paketformat und Verschlüsselungsverfahren für die Übertragung verschlüsselter Datenströme festgelegt. Dabei werden der RTP-Nutzdatenstrom sowie der RTCP-Datenstrom getrennt betrachtet. Als standardmäßiges Verschlüsselungsverfahren wird der *Data Encryption Standard*-Algorithmus (DES) im *Cypher Block Chaining Mode* (CBC) empfohlen [Bal93, Abschnitt 1.1], [Sch96a, Seite 309ff.].

Die Verschlüsselung des Nutzdatenstroms erfolgt, indem die aus Nutzdaten und RTP-Header aggregierte UDP-PDU vor der Netzwerkübertragung verschlüsselt wird. Der RTP-Header wird folglich mit verschlüsselt. Der UDP-Header wird unverschlüsselt übertragen. Die Verschlüsselung des zugehörigen RTCP-Datenstroms unterscheidet sich hiervon, indem dem RTCP-Paket eine 32-Bit Zufallszahl vorangestellt wird, um Plaintext-Attacken vorzubeugen. Erst danach erfolgt die Verschlüsselung der PDU.

3.2.9.2 Übertragungsverfahren für verschlüsselte Datenströme

Das MBone-Access-Gateway muß die Vermittlung verschlüsselter Datenströme unterstützen, andernfalls würde dem System eine große Gruppe von Einsatzszenarien verschlossen bleiben. Dazu müssen die Schlüssel für Audio- und Video-Datenströme den Komponenten des Systems zugänglich gemacht werden, die die verlustbehaftete Kompression der multimedialen Echtzeitdatenströme durchführen. Sonst würde die Datenratenanpassung für die ISDN-Strecke nachhaltig erschwert.

Dies ist ein grundsätzliches Problem des Einsatzes von Transcodern und Mixern bei der Vermittlung verschlüsselter Datenströme: Der Nutzer muß darauf vertrauen, daß die von ihm übermittelten Schlüssel nicht mißbraucht werden. Unter der Voraussetzung, daß bei der Initialisierung der Verkehrsbeziehung zwischen Nutzer und MBone-Access-Gateway sowie den verteilten Komponenten des MBone-Access-Gateways untereinander eine gegenseitige Authentifizierung stattfindet, ist dieses Risiko für viele Betriebsfälle akzeptabel.

Dennoch ist zu berücksichtigen, daß es Fälle gibt, in denen die Bekanntgabe der Schlüssel gegenüber dem Vermittlungssystem nicht hinnehmbar ist. In diesem Fall müssen die daraus resultierenden technischen Unzulänglichkeiten vom Nutzer akzeptiert werden. Allerdings muß das MBone-Access-Gateway diesen Betriebsfall unterstützen.

Für den Fall, daß der Nutzer die Übermittlung der Schlüssel an das Vermittlungssystem akzeptiert, stellt sich die Frage, wie die Verschlüsselung der Echtzeitdatenströme über den Tunnel

erfolgt. Eine Option ist, den jeweiligen Datenstrom nach der Datenratenanpassung durch einen Mixer oder Transcoder erneut dem RTP-Verschlüsselungsverfahren zu unterziehen und als UDP-Datenstrom über den Tunnel zu transportieren. Alternativ hierzu kann der Tunnel-Datenstrom zwischen MAS und MAG grundsätzlich verschlüsselt übertragen werden. Hier könnte auf die erneute Verschlüsselung des Medien-Datenstroms nach dem Durchlaufen eines Mixers oder Transcoders verzichtet werden. Ein Vorteil dieser Alternative ist, daß ein RTP-Datenstrom auch als RTP-Datenstrom über den Tunnel transportiert werden kann. Daraus ergeben sich verbesserte Möglichkeiten zur Header-Komprimierung. Nachteilig ist, daß alle Datenströme verschlüsselt würden. Dadurch entsteht eine zusätzliche Verzögerung, die als kritisch einzustufen ist.

Beide Verfahren haben demnach Vor- und Nachteile. Im Rahmen dieser Arbeit werden beide Wege unterstützt. Auf der Ebene der verlustbehafteten Kompression durch Mixer und Transcoder werden drei alternative Verfahren zur Behandlung verschlüsselter Datenströme angeboten:

Decrypt-Compress-Forward
Der Datenstrom wird den folgenden Bearbeitungsschritten unterworfen:

- Entschlüsselung des eingehenden Datenstroms.
- Verlustbehaftete Kompression.
- Unverschlüsselte Weiterleitung als RTP-Datenstrom.

Decrypt-Compress-Encrypt
Der Datenstrom wird den folgenden Bearbeitungsschritten unterworfen:

- Entschlüsselung des eingehenden Datenstroms.
- Verlustbehaftete Kompression.
- Verschlüsselung und Weiterleitung als UDP-Datenstroms.

Bypass
Die Daten werden ohne weitere Bearbeitung als UDP-Datenstrom weitergeleitet. Dieser Modus wird benutzt, wenn ein verschlüsselter Datenstrom ohne zwischenzeitliche Entschlüsselung im MBone-Access-Gateway weiterzuleiten ist.

Der Standard-Modus ist Decrypt-Compress-Forward, wobei die Daten unverschlüsselt über die Tunnel-Strecke übertragen werden. Dies ist in vielen Anwendungsfällen als wenig kritisch einzustufen, da dem Nutzer am Heimarbeitsplatz die Architektur des benutzten Netzwerks bekannt ist und er annehmen kann, daß das Risiko der Präsenz eines Zuhörers gering ist. Andernfalls kann er sich dazu entscheiden, den gesamten Datenverkehr, der über den Tunnel abgewickelt wird, zu verschlüsseln. Diese Verschlüsselung findet auf der Ebene des Tunnel-Protokolls zwischen Output-Modul des Senders und Input-Modul des Empfängers statt.

3.2.9.3 Authentifizierung und Schlüsselübertragung

Wenn der Datenstrom im Modus Decrypt-Compress-Forward oder Decrypt-Compress-Encrypt übermittelt wird, muß der Endnutzer dem MBone-Access-Gateway den jeweiligen Schlüssel bekanntgeben.

Dazu sollte sich der Dienst gegenüber dem Nutzer zuvor authentifizieren. Diese Authentifizierung ist Bestandteil des MAS-Kontrollprotokolls, daß in Abschnitt 3.2.10, Seite 119, vorgestellt wird.

Nachdem die Authentifizierung erfolgreich abgeschlossen wurde, können Datenströme angefordert werden. Handelt es sich um vertrauliche Datenströme und erfordert der Betriebsmodus die Übertragung des Schlüssels, so muß der Nutzer den Schlüssel gegenüber dem Vermittlungssystem bekanntgeben. Diese Phase ist kritisch, weil ein hierbei kompromittierter Schlüssel zum unbemerkten Abhören des Datenstroms genutzt werden kann.

Da der Schlüssel ebenfalls unter Nutzung des MAS-Kontrollprotokolls übertragen wird, muß dieses sicherstellen, daß die Kompromittierung nicht oder nur schwer möglich ist.

3.2.9.4 Management der Schlüssel

Bei konventioneller Anwendung der MBone-Werkzeuge werden Schlüssel für Medienströme entweder durch den Nutzer beim Programmstart des Medien-Werkzeugs als Parameter angegeben oder in einem entsprechenden Dialog während der Laufzeit konfiguriert. Für den vorliegenden Einsatzfall ist die interaktive Festlegung des Schlüssels problematisch, da bei der Anforderung des Datenstroms der Schlüssel festliegen muß.

Die Anforderung des Datenstroms erfolgt durch Wrapper-Anwendungen (vgl. Abschnitt 3.1.4, Seite 61), denen der Zugriff auf vom Nutzer im Medien-Werkzeug konfigurierten Schlüsseln nicht möglich ist. Daher muß der Nutzer der Wrapper-Anwendung vor dem Start des Medien-Werkzeugs den Schlüssel bekanntgeben. Dazu erscheint nach dem Start der Wrapper-Anwendung ein Dialog, der dem Nutzer die Angabe eines Schlüssels und eines der verfügbaren Übertragungsmodi erlaubt. Wurde bereits beim Start der Wrapper-Anwendung ein Schlüssel angegeben, wird dieser im Dialog eingetragen. Der gewählte Schlüssel und der erforderliche Übertragungsmodus werden, soweit erforderlich, per MASCP dem MBone-Access-Gateway übergeben. Schließlich wird das MBone-Werkzeug mit der erforderlichen Schlüssel-Einstellung gestartet. Änderungen, die durch den Nutzer im laufenden MBone-Werkzeug vorgenommen werden, können nicht berücksichtigt werden. Sollen Schlüssel oder Übertragungsmodus geändert werden, ist ein Neustart von Wrapper-Anwendung und MBone-Werkzeug erforderlich.

Abschließend ist anzumerken, daß die entwickelten Verfahren zur Übertragung verschlüsselter Datenströme als pragmatischer Weg zur Zusicherung der Vertraulichkeit der Kommunikation zu verstehen sind. Es bleibt ein Restrisiko und der Nutzer muß entscheiden, ob er dieses akzeptieren kann. Insgesamt zeigt sich, daß das Thema Vertraulichkeit im Kontext der MBone-Technologie ein wunder Punkt ist. Viele Fragen, wie das Management der Schlüssel und die Authentifizierung der Nutzer, bleiben weitgehend unbeantwortet. Es ist allerdings auch nicht das Anliegen dieser Arbeit, diese Schwierigkeiten zu beheben.

3.2.10 MBone-Access-Server Kontroll-Protokoll (MASCP)

3.2.10.1 Einordnung des MASCP

Abseits der zuvor untersuchten Problematiken bei der Vermittlung multimedialer Echzeitdatenströme ist ein Protokoll zur Realisierung administrativer Funktionen erforderlich. Hierzu dient das *MBone Access Server Control Protocol* (MASCP). Es stellt folgende Funktionen bereit:

Initialisierung des Systems
Bevor das MBone-Access-Gateway die Vermittlung von Nutzdaten übernehmen kann, ist der Aufbau einer Verkehrsbeziehung zwischen MAG und MAS erforderlich. Sie wird typischerweise dadurch initiiert, daß der Benutzer am Heimarbeitsplatz eine Wrapper-Applikation startet. Dieser Prozeß wendet sich entsprechend der Konfiguration des Systems an ein MAG um den gewünschten Datenstrom anzufordern. Dabei wird überprüft, ob bereits eine Verbindung zum konfigurierten MAS besteht. Wenn dies nicht der Fall ist, wird die Verbindung aufgebaut und entsprechend den Erfordernissen konfiguriert.

Anforderung von Datenströmen
Ein wesentliches Merkmal der Anwendungen des *Internet Conferencing Protocol Stack* ist die lose Koppelung der Medienströme untereinander. Zu jedem Zeitpunkt einer laufenden Konferenz können Medienwerkzeuge ergänzt oder entfernt werden. Das MAS-Kontroll-Protokoll verfügt über Anweisungen zur Unterstützung dieser Vorgänge.

Freigabe belegter Ressourcen
Die spontane Beendigung von Medien-Werkzeugen sowie der Austritt eines Konferenzteilnehmers aus der Konferenz am Heimarbeitsplatz erfordert die Freigabe belegter Ressourcen im MAG und MAS. Eine Komponente des Protokolls gewährleistet die Abwicklung der erforderlichen Aktivitäten.

Das MASCP wird ausschließlich zur Abwicklung administrativer Tätigkeiten genutzt. Alle Kontrollfunktionen, die mit der Vermittlung der Datenströme zusammenhängen, werden über die in den Datenstrom zwischen MAS und MAG eingefügten Service-PDUs realisiert. Das mit MASCP-Nachrichten verbundene Datenaufkommen ist im Vergleich zu den anderen Komponenten des Systems gering. Wichtig ist jedoch, daß die Nachrichten zuverlässig übermittelt werden. Daher wurde als Transportprotokoll das *Transmission Control Protocol* (TCP) gewählt.

3.2.10.2 Authentifizierung und Vertraulichkeit

Ein wichtiger Aspekt ist die Privatheit der Kommunikation zwischen MAS und MAG. Insbesondere gilt dies für das MASCP. Wie bereits in Abschnitt 3.2.9.2, Seite 116, dargestellt, wird die Vertraulichkeit der Kommunikation bei MBone-Konferenzen durch Verschlüsselung der Datenströme erzielt. Für die Vermittlung verschlüsselter Datenströme unter Nutzung von Mixern und Transcodern ist die Übertragung des entsprechenden Passworts vom MAG zum MAS erforderlich. Die unverschlüsselte Übertragung ist nicht akzeptabel. Beobachter auf dem Übertragungsweg können das Passwort ausspähen und unbemerkt der Kommunikation folgen.

Erforderlich ist daher die verschlüsselte Übertragung der Nachrichten zwischen MAG und MAS sowie eine Authentifizierung beim Verbindungsaufbau. Zur Realisierung dieser Funktionalität lassen sich verschiedene Wege beschreiten. Um die Komplexität des Systems nicht unnötig zu erhöhen und Diskussionen zur Sicherheit des angewendeten Verfahrens zu vermeiden, kommt ein etabliertes und allgemein anerkanntes System zur Wahrung der Vertraulichkeit zum Einsatz: Das von Netscape entwickelte *Secure Socket Layer Protocol* (SSL) [Fre96].

Der Einsatz von SSL dient hier zwei Zwecken:

- Authentifizierung des Servers gegenüber dem Client.
- Vertraulichkeit der Kommunikation zwischen Client und Server.

Diese Funktionen werden realisiert, indem der Server bei der Abarbeitung des SSL-Handshake-Protokolls ein Zertifikat an den Client sendet und ggf. ein Server-Key-Exchange folgt, eine Anfrage nach dem Zertifikat des Client erfolgt nicht. Damit bleibt der Nutzer anonym, der Server hat sich gegenüber dem Client jedoch authentifiziert [Fre96, Abschnitt 5.5]. Damit kann ein Session-Key für die weitere private Kommunikation über diese Verbindung sicher ausgetauscht werden.

Erst nachdem dieser Schritt abgeschlossen ist, authentifiziert sich das MAG gegenüber dem MAS, d.h. der Client gegenüber dem Server. Dabei werden *Username* und *Password* über die durch SSL geschützte Verbindung an den Server übertragen und dort verifiziert.

Wenngleich auch eine gegenseitige Authentifizierung über SSL möglich wäre, bietet dieses Vorgehen folgende Vorteile:

- Auch unter Verzicht des Einsatzes von SSL kann sich das MAG gegenüber dem MAS authentifizieren und, für den Einsatz des MASCP zur Kommunikation zwischen Wrapper-Anwendungen und MAG, die Wrapper-Anwendung gegenüber dem MAG.
- Für die Betreiber von MAGs ist kein verifizierbares Zertifikat oder ein Schlüsselpaar erforderlich.
- Verwaltungstechnisch ergibt sich der Vorteil, daß der Betreiber des MAS dem Nutzer ein Passwort zuweisen kann. Der heute nur in Ausnahmefällen vorhandene Public-Key eines Nutzers braucht nicht zertifiziert zu werden.
- Der weitgehend automatische Betrieb des Systems würde erfordern, daß der Private-Key des Nutzers in einer Datei des MAG-Rechners gespeichert würde. Sofern dieser Schlüssel für andere Zwecke genutzt wird, stellt die ungeschützte Speicherung dieses Schlüssels eine Kompromittierung des Nutzers dar. Die Speicherung einer Nutzerkennung und eines entsprechenden Passworts für einen spezifischen Dienst ist hingegen gängige Praxis.[31]

Generell wird durch dieses Vorgehen das Problem des Key-Managements nicht gelöst. Vorteilhaft ist, daß nur der MAS über ein zertifiziertes Schlüsselpaar verfügen muß oder der Public-Key

[31] Ein Beispiel ist die Authentifizierung des Nutzers gegenüber dem Access-Server unter Nutzung von CHAP [Sim96] oder PAP [LS92] während der Initialisierung des PPP-Kanals.

des MAS-Systems dem Nutzer auf vertrauliche Weise bereitgestellt wird; z.B. im Rahmen der Zulassung für die Systemnutzung kann dem Nutzer auf einer Diskette der Public-Key des Systems ausgeliefert werden. Das Verfahren ist Stand der Technik im Kontext der Nutzung sicherer WWW-Dienste und erscheint daher auch für den Einsatz im MBone-Access-Gateway als geeignet.

3.2.10.3 Protokollstruktur

Die generelle Protokollstruktur des MASCP lehnt sich eng an die Richtlinien in [New97] an. Die Ausprägung des MASCP vereinigt Elemente aus dem *File Transfer Protocol* (FTP) [PR85] und dem *Hyper Text Transfer Protocol* (HTTP) [Fie97].

Es handelt sich um ein striktes Client-Server-Protokoll. Im Betriebsfall der Kommunikation zwischen MAG und MAS übernimmt der MAG die Rolle des Client, der MAS fungiert als Server. Im Fall der Kommunikation zwischen Wrapper-Anwendungen und MAG ist jede Wrapper-Anwendung ein Client. Das MAG dient als Server.

Wie bereits dargestellt, wird TCP als Transportprotokoll verwendet. Die Verbindung zwischen Client und Server wird zur Zeit der Etablierung des Tunnels aufgebaut und bleibt bestehen, bis der Tunnel abgebaut wird. Es handelt sich folglich nicht um ein transaktionsorientiertes TCP-Protokoll wie etwa HTTP. Trotzdem werden Transaktionen über die Verbindung abgewickelt. Jede Anfrage des Client wird vom Server beantwortet. Bevor die Anfrage nicht endgültig beantwortet ist, kann keine weitere Anfrage gestellt werden. Wie beim HTTP wird der Inhalt der Antworten über die gleiche Datenverbindung gesendet, über die auch die Anfrage gestellt wurde.

Die Nachrichten zwischen Client und Server werden als ASCII-Text übertragen. Dies gestattet die einfache Analyse der Nachrichten und eine flexible Erweiterbarkeit des Protokolls. Zudem wird die Problematik der unterschiedlichen Interpretation von Zahlen umgangen. Wenngleich ein binäres Protokoll ein geringeres Datenvolumen generiert, ist hier einzuwenden, daß das durch das MASCP erzeugte Datenvolumen im Vergleich zu dem über den Tunnel zu übertragenden Datenvolumen vernachlässigbar klein und somit ohne praktische Relevanz ist.

Anfragen an den Server bestehen aus jeweils einer Zeile Text, die durch die *ASCII*-Zeichen *Carriage-Return,Line-Feed* (CR,LF) abgeschlossen werden. Die Antworten des Systems bestehen aus einer oder mehren Zeilen Text. Das erste Wort der Antwort ist jeweils ein Numerisches Literal eines Antwort-Codes. Darauf folgt, getrennt durch ein Leerzeichen, ein informeller Text zur Interpretation des Antwort-Codes durch den Nutzer. Handelt es sich um eine mehrzeilige Antwort, endet der Antwort-Code mit einem Trennstrich (−). Am Ende der mehrzeiligen Antwort wird der Antwort-Code ohne nachgestellten Trennstrich wiederholt.

In den folgenden Abschnitten wird detailliert auf die Komponenten des MASCP eingegangen.

3.2.10.4 MASCP-Anfragen

Login

Anweisung	Beschreibung
USER *username*	Name des Nutzers.
PASS *password*	Passwort des Nutzers.
QUIT	Beendigung der Verkehrsbeziehung.
PARAM *<parameter list>*	Client teilt Server Konfigurationsparameter mit.
GETPARAM *<list of requested parameters>*	Client fordert vom Server eine Parameterliste an.
PORT *address p0* (*p1 ... p63*)	Client teilt Server die Empfangsadresse und Port-Zuordnung mit.
GETPORT	Client fordert Empfangsadresse und Port-Zuordnung des Server an.
GETSTREAM *<stream description>*	Client fordert Vermittlung eines Datenstrom an.
DELSTREAM *<stream id>*	Client fordert Server zur Einstellung der Übertragung eines Datenstroms auf.
HELP	Client fragt der implementierten Anweisungen des Servers ab.
STATUS	Client fragt Status des Servers ab.

Tabelle 3.12: MASCP-Anfragen

Bevor der Client-Prozeß Datenströme vom Server-Prozeß anfordern kann, ist die Authentifizierung des Client gegenüber dem Server erforderlich. Die Authentifizierung erfolgt stets abschnittsweise, d.h. zwischen Wrapper-Anwendung und MAG bzw. zwischen MAG und MAS. Dadurch wird eine flexiblere Vergabe von Zugriffsrechten möglich, als dies mit einem Ende-zu-Ende Ansatz realisierbar wäre.

Das zur Authentifizierung gewählte Verfahren entspricht dem im FTP-Protokoll festgelegten Verfahren [PR85, Abschnitt 4.1.1]. Im ersten Schritt überträgt der Client-Prozeß den Namen des Nutzers mit einer USER-Anweisung. Der Server-Prozeß fordert darauf zur Übertragung des Passworts auf. Dieses wird mit der PASS-Anweisung übertragen. Der Erfolg oder Mißerfolg der Authentifizierung wird dem Client in der zugehörigen Antwort mitgeteilt.

Für den Fall, daß keine individuelle Authentifizierung erforderlich ist, ist als USER der Name *anonymous* anzugeben und als Passwort sollte die E-Mail-Adresse des Nutzers angegeben werden. Dieses Verfahren entspricht den im Internet üblichen Konventionen beim Zugriff auf Anonymous-FTP-Server.

Austausch von Konfigurationsparametern

Wird das MASCP zwischen MAG und MAS eingesetzt, ist für die Etablierung des Tunnels der Austausch von Konfigurationsparameter erforderlich. Dafür dienen die PARAM- und GETPARAM-Anweisungen. Üblicherweise beginnt diese Abstimmungsphase damit, daß der

Client seinen Parametersatz in einer PARAM-Anweisung an den Server überträgt. Die Tabelle 3.12 angegebene *<parameter list>* hat folgende Struktur.

```
<parameter list> := Parameter=Wert{ Parameter=Wert}
```

Dabei überträgt das MAG die in Tabelle 3.13 dargestellten Parameter.

Die Parameter des MAS werden vom MAG mit der GETPARAM-Anweisungen abgerufen. Die Tabelle 3.12 angegebene *<list of requested parameters>* hat folgende Struktur.

```
<list of requested parameters> := Parameter{ Parameter}
```

Vom MAS wird zur Zeit nur der in Tabelle 3.14 beschriebene Parameter unterstützt.

Die Übermittlung der Parameter-Werte vom Server zum Client erfolgt in einer Antwort, die aus mehreren Zeilen besteht. Wenngleich die Antworten des Servers erst in Abschnitt 3.2.10.5, Seite 129, detailliert behandelt werden, illustriert das folgende Beispiel die Kommunikation zwischen Client und Server bei der Abfrage der Server-Parameter und der Übertragung der Parameter Werte in der Antwort.

```
MAG> GETPARAM TSTOP
MAS> 225- MAS Parameter Description
MAS>  TSTOP=300
MAS> 225 End of MAS Parameter Description
```

In diesem Beispiel fordert das MAG den Parameter TSTOP vom MAS an und erhält in der Antwort mitgeteilt, daß der aktuelle Wert 300 Sekunden, d.h. 5 Minuten beträgt.

Etablierung des Tunnels

Eine wesentliche Anwendung des MASCP ist die Einrichtung eines Tunnels zwischen MAG und MAS. Zur Etablierung des Tunnels ist es wichtig, daß sich MAS und MAG darauf einigen, ob zur Nachrichtenübertragung UDP-Ports genutzt werden. Dazu dient der vom MAG zum MAS übertragene Parameter PORTTRANS. Wenn der Wert von PORTRANS 0 ist, wird nur ein Port beim MAG und MAS benutzt. Wenn PORTRANS den Wert 1 hat, werden auf jeder Seite 64 Ports benutzt.

In jedem Fall ist es zur Etablierung des Tunnels erforderlich, daß die zu verwendenden Ports und IP-Adressen für die Tunnel-Kommunikation der jeweiligen Partnerinstanz mitgeteilt werden. Dazu dienen die PORT- und GETPORT-Anweisungen.

Nachdem das MAG in Abhängigkeit des Parameters PORTRANS die entsprechende Zahl von Sockets geöffnet hat, überträgt es die Adresse des IP-Interfaces des Tunnel-Endpunktes zusammen mit der Liste der reservierten Ports an den MAS. Im zweiten Schritt stellt das MAG eine GETPORT-Anfrage ohne weitere Parameter an den Server, und erhält eine vergleichbare Parameterliste in der Antwort übertragen.

Parameter	Beschreibung
PORTTRANS	Flag, welches angibt, ob UDP-Ports zur Datenübertragung genutzt werden sollen. Mögliche Werte: $1 \wedge 0$
KEY	Die über den Tunnel transportierte Payload wird generell mit nach dem DES-CBC-Verfahren verschlüsselt. Der Parameterwert ist der einzusetzende Schlüssel. Siehe hierzu auch Abschnitt 3.2.9.2, Seite 116.
DEFTTL	Standard-TTL-Wert zur Ankündigung von SAP/SDP-Konferenzbeschreibungen. Siehe Abschnitt 3.2.8, Seite 114.
DTE	Dauer der Stand-By-Phase des Registersatzes zur Bestimmung von L_{min_i} und $P(L_{min_i})$ (Δt_E) in s. Siehe Abschnitt 3.2.2.1, Seite 78.
IMT	Interactive Mode Timeout in s. Siehe Abschnitt 3.2.6, Seite 112.
CINIT	Vom Nutzer abgeschätzte Tunnel-Übertragungskapazität in $kbps$. Siehe Abschnitt 3.2.2.4, Seite 86.
DTQMIN	Unterer Schwellwert für das Queuing-Delay $\Delta t_{Q_{min}}$ in ms. Siehe Abschnitt 3.2.2.4, Seite 86.
DTQMAX	Oberer Schwellwert für das Queuing-Delay $\Delta t_{Q_{min}}$ in ms. Siehe Abschnitt 3.2.2.4, Seite 86.
LRKRIT	Schwellwert für kritische Paketverlustrate in Prozent. Siehe Abschnitt 3.2.2.4, Seite 86.
MOB	Protokolloverhead in Bytes pro Paket. Üblicherweise beträgt er bei IP/PPP/ISDN 35 Byte (7 Bytes PPP, 20 Bytes IP, 8 Bytes UDP).
MTU	Maximum Path Transmission Unit. Bei IP/PPP/ISDN beträgt sie üblicherweise 1500 Bytes.

Tabelle 3.13: Durch das MAG beeinflußbare Konfigurationsparameter des MAS

Parameter	Beschreibung
TSTOP	Maximales Zeitintervall zwischen dem Empfang von zwei Tunnel-Reports in s. Siehe Abschnitt 3.2.2.4, Seite 90.

Tabelle 3.14: Durch das MAS an das MAG übertragene Konfigurationsparameter

Das folgende Beispiel illustriert diesen Protokoll-Schritt für einen Tunnel mit dem Konfigurationsparameter $PORTRANS = 0$.[32]

```
MAG> PORT 130.75.249.63 1063
MAS> 200 Command okay
MAG> GETPORT
MAG> 100 Command accepted, final answer follows
MAS> 226- MAS Port Description
MAS>  130.75.5.239 2780
MAS> 226 End of MAS Port Description
```

In diesem Beispiel teilt das MAG dem MAS in der ersten Anweisung mit, daß der Tunnel-Endpunkt das IP-Interface mit der Adresse `130.75.249.63` ist. Zudem wurde beim MAG der Port `1063` für den Empfang der Tunnel-Pakete vorbereitet. In der darauffolgenden Zeile teilt der Server dem Client mit, daß er diese Parameter akzeptiert hat.

In der dritten Zeile fordert der Client die Empfangsadresse und die Port-Zuordnung des Servers an. Die Antwort signalisiert, daß der Tunnel-Endpunkt beim Server das IP-Interface mit der Adresse `130.75.5.239` ist und Port `2780` für den Empfang der Daten vorbereitet wurde.

Anforderung von Datenströmen

Während die Protokoll-Schritte zum Austausch von Konfigurationsparametern sowie der Etablierung des Tunnels nur bei der Kommunikation zwischen MAG und MAS verwendet werden, ist die Anforderung von Datenströmen ein Protokoll-Element, welches auch für die Kommunikation zwischen Wrapper-Anwendungen und MAG erforderlich ist.

Die Anforderung von Datenströmen erfolgt unter Nutzung der Protokoll-Anweisung GETSTREAM. Sie erhält als Parameter die Beschreibung des anzufordernden Datenstroms in Form einer <*stream description*>, die aus folgenden Komponenten besteht.

```
<stream description> := <Medien-Typ> <Transportadresse> \
                        <Transportprotokoll> <Optionen>
```

Die Komponenten der Stream-Description werden durch Leerzeichen voneinander abgegrenzt. Die Komponenten sind wie folgt aufgebaut:

- Der *Medien-Typ* entscheidet darüber, mit welcher Priorität die Pakete des Datenstroms im Client und Server behandelt werden. Folgende Typen werden unterschieden:

[32] Für die Etablierung eines Tunnels über eine ISDN-Strecke wird der Wert von PORTRANS in der Regel 1 sein. Nur so kann die effizienteste Form der Datenvermittlung erreicht werden. Hier wurde aber besonderer Wert auf die Illustration des MASCP gelegt und daher zur besseren Lesbarkeit der Parameter $PORTRANS = 0$ gewählt.

Transportprotokoll	Beschreibung
RTP/AVP:*n*	RTP-V2 wird als Transportprotokoll benutzt. Die Zahl *n* gibt den *Payload-Type* gemäß [Sch96b] an und entscheidet damit über die Codierung des Datenstroms auf der Tunnel-Strecke.
SAP/SDP	Das Session-Announcement-Protocol [Han96a] dient als Transportprotokoll. Die Payload ist gemäß dem Session-Description-Protocol [HJ97] codiert.
UDP	Es handelt sich um einen nicht weiter spezifizierten UDP-Stream.

Tabelle 3.15: Transportprotokollbeschreibungen für Tunneldatenströme

Medien-Typ	Priorität	Beschreibung
audio	1	Audio-Datenströme, RTP und andere
shared-tools	2	Shared-Tools und Conference-Control
video	3	Video-Datenströme, RTP und andere
other	4	Andere Datenströme geringer Priorität, z.B. SAP/SDP

- Die *Transportadresse* legt fest, an welche Unicast- oder Multicast-Transportadresse die Daten des jeweiligen Streams zu vermitteln sind. Die Transportadresse besteht aus folgenden Komponenten, die durch Doppelpunkte gegeneinander abgegrenzt werden:

Komponente	Beschreibung
IP-Adresse	IP-Adreßteil der Transportadresse ggf. ergänzt um TTL
Port	Port-Teil der Transportadresse
Anzahl	Anzahl der sequentiell aufeinander folgenden Ports

- Das *Transportprotokoll* entscheidet über die Behandlung des Datenstroms von Server und Client, insbesondere über die anzuwendende Header-Komprimierung und die Konfiguration eventuell erforderlicher Mixer und Transcoder. Die in Tabelle 3.15 beschriebenen Alternativen sind zu unterscheiden.

- Es können eine oder mehrere *Optionen* angegeben werden, die zusätzliche Informationen zur Behandlung des Datenstroms bereitstellen. Die Optionen werden durch Leerzeichen gegeneinander abgegrenzt. Tabelle 3.16 zeigt die unterstützten Optionen.

Die folgende Transaktion zeigt die Anforderung eines Audio-Datenstroms von einer Wrapper-Anwendung an das MAG:

```
Wrapper> GETSTREAM audio 224.2.139.43/47:3456:2 RTP/AVP:3 \
         c=192.168.240.11:5330
    MAG> 100 Command accepted, final answer follows
    MAG> 227- MAG STREAM Description
    MAG>  f=7
    MAG> 227 End of MAG STREAM Description
```

Option	Beschreibung
recvonly	Dieser Datenstrom soll ausschließlich empfangen werden. Das Forwarding erfolgt nur in Down-Stream-Richtung. Bei RTP-Streams bleiben die RTCP-Nachrichten davon unberührt.
sendonly	Dieser Datenstrom soll ausschließlich gesendet werden. Das Forwarding erfolgt nur in Up-Stream-Richtung. Bei RTP-Streams bleiben die RTCP-Nachrichten davon unberührt.
k=*key*	Die Übertragung dieses Datenstroms erfolgt verschlüsselt. Diese Option gibt den entsprechenden Schlüssel für Mixer und Transcoder an.
e=*encryption-mode*	Diese Option ist erforderlich, wenn es sich um einen verschlüsselten RTP-Datenstrom handelt. Sie beschreibt das Verfahren zur Behandlung des Datenstroms gemäß Abschnitt 3.2.9.2, Seite 116. Mögliche Werte sind `decrypt-compress-forward`, `decrypt-compress-encrypt` und `bypass`.
b=*modifier:bandwidth*	Diese Option begrenzt die Datenrate für den jeweiligen Datenstrom in kbps. Der Modifier legt fest, ob die Datenratenbegrenzung pro Teilnehmer (UT) oder für die gesamte Konferenz (CT) gelten soll.
c=*transport-address*	Diese Option ist von Relevanz, wenn die Stream-Description von einer Wrapper-Applikation an das MAG gesendet wurde. Sie legt fest, an welche Adresse das Client den jeweiligen Datenstrom vermitteln soll. Die Syntax der Transportadresse entspricht den obigen Vorgaben. Die Anzahl der aufeinander folgenden Ports entfällt.
udpcompress	Die Übertragungs von UDP-Datenströmen, SAP/SDP-Datenströmen sowie RTCP-Daten soll unter Nutzung der verlustfreien UDP-Payload-Komprimierung erfolgen.
nortpcompress	RTP Medien-Datenstöme sollen nicht durch Mixer oder Transcoder einer verlustbehafteten Payload-Kompression unterzogen werden.

Tabelle 3.16: Optionen der Tunneldatenströme

Hier fordert die Wrapper-Anwendung einen Audio-Datenstrom an, der im MBone auf der Class-D-Adresse `224.2.139.43` und den Ports `3456` und `3457` transportiert wird. Vom Nutzer generierte Audio-Daten sollen vom MAS mit einem TTL-Wert von `47` ausgesendet werden. Die Transportprotokoll-Komponente gibt an, daß es sich um einen RTP-Datenstrom handelt, der über den Tunnel in GSM-Codierung übertragen werden soll. Die Option gibt an, daß das MAG den Datenstrom an das IP-Interface mit der Adresse `192.168.240.11` senden soll. Die Pakete sind an die Ports `5330` und `5331` zu adressieren.

Die Antwort-Code des MAG (`227`) signalisiert der Wrapper-Anwendung, daß die Anforderung akzeptiert wurde. Darauf folgt die Beschreibung der für den Client wichtigen Daten des Streams. Mit dem Parameter `f=7` wird dem Client mitgeteilt, daß dieser Stream fortan mit der Referenznummer `7` angesprochen wird. Dieser Wert ist in seiner Bedeutung mit einem File-Handle zu vergleichen.

Bevor das MAG der Wrapper-Anwendung die Antwort übermittelt, fordert es den Datenstrom beim MAS an. Die sich daraus ergebende Transaktion hat folgende Struktur:

```
MAG> GETSTREAM audio 224.2.139.43/47:3456:2 RTP/AVP:3
MAS> 227- MAS STREAM Description
MAS>  f=19
MAS> 227 End of MAS STREAM Description
```

Die Anfrage des MAG an den MAS ist der Anfrage der Wrapper-Anwendung an das MAG ähnlich. Es fehlt die in der Option angegebene Zieladresse des Streams, da dem MAG der Datenstrom über den Tunnel zwischen MAG und MAS vermittelt wird.

Beendigung der Vermittlung von Datenströmen

Wenn ein Nutzer am Heimarbeitsplatz ein Medien-Werkzeug beendet, ist es Aufgabe der Wrapper-Anwendung, den entsprechenden Datenstrom beim MAG abzubestellen. Dazu dient die DELSTREAM-Anweisung. Der einzige Parameter dieser Anweisung referenziert den abzubestellenden Datentstrom anhand seiner Stream-ID. Das folgende Beispiel illustriert die Kommunikation zwischen Wrapper-Anwendung und MAG.

```
Wrapper> DELSTREAM 7
    MAG> 100 Command accepted, final answer follows
    MAG> 200 Command okay
```

Bevor das MAG die positive Antwort übermittelt, wird eine entsprechende Transaktion zwischen MAG und MAS abgewickelt:

```
MAG> DELSTREAM 19
MAS> 200 Command okay
```

Code	Beschreibung
1yz	Positive Fortschrittsanzeige. Die Anweisung wurde akzeptiert, jedoch ist die eine weitere, endgültige Antwort abzuwarten.
2yz	Positiver Abschluß der Anfrage. Ein neues Kommando kann gesendet werden.
3yz	Positive zwischenzeitliche Antwort. Die Anweisung wurde akzeptiert, jedoch ist zum Abschluß der Transaktion die Übermittlung einer weiteren Anweisung erforderlich.
4yz	Transient negative Antwort. Die Anweisung konnte nicht ausgeführt werden, sollte später aber wiederholt werden.
5yz	Endgültig negative Antwort. Die Anweisung wurde nicht akzeptiert. Die Anfrage sollte nicht wiederholt werden.

Tabelle 3.17: Bedeutung der ersten Ziffer einer MASCP-Antwort

Beendigung der Verkehrsbeziehung

Bevor eine Wrapper-Anwendung sich beendet, löst sie die Verkehrsbeziehung zum MAG, indem sie eine QUIT-Nachricht an den MAG sendet. Danach werden ggf. abonnierte Streams abbestellt und die TCP-Verbindung geschlossen.

Wenn ein MAG selbst keine weiteren Clients hat, wird die Kommunikationsbeziehung zum MAS ebenfalls durch die Übermittlung einer QUIT-Nachricht beendet.

3.2.10.5 MASCP-Antworten

Die Antworten zu MASCP-Anweisungen orientieren sich in ihrer Struktur und Ausprägung am FTP-Protokoll [PR85, Abschnitt 4.3]. Jede Antwort besteht aus einem Antwort-Code, gefolgt von einem beschreibenden Text. Die Antwort-Codes bestehen aus drei Ziffern.

Die erste Ziffer zeigt an, ob die Antwort positiv, negativ oder unvollständig ist und ob im Fehlerfall durch eine wiederholte Anfrage eine positive Antwort zu erwarten ist. Eine Anwendung kann durch die Analyse der ersten Ziffer der Antwort erkennen, ob die Abarbeitung weiterer Protokollschritte zu positiven Ergebnissen führen kann. Tabelle 3.17 dokumentiert die verschiedenen Bedeutungen der ersten Ziffer der Antwort.

Die zweite Ziffer der Antwort beschreibt die Art des aufgetretenen Fehlers. Tabelle 3.18 beschreibt die Zuordnung zwischen Ziffern und Fehlerart.

Die dritte Ziffer dient der weiteren Unterscheidung der Antworten. Durch die Kombination der Ziffern entstehen eindeutige Antwort-Codes, die in Tabelle 3.19, geordnet nach den Codes, dargestellt sind. Die dritte Tabellenspalte gibt an, durch welche Anfragen die jeweiligen Antwort-Codes generiert werden können.

Die Implementierung des Systems überträgt mit jedem Antwort-Code eine textuelle Beschreibung zur Interpretation der Antwort durch einen Nutzer. Dabei handelt es sich um Übersetzun-

Code	Beschreibung
x0z	Syntax-Fehler. Die gestellte Anfrage beinhaltet einen Syntax-Fehler oder enthält eine nicht implementierte Anweisung.
x1z	Es handelt sich eine Antwort mit informellen Charakter. Beispiele sind positive Antworten sowie Antworten auf STATUS- und HELP-Anfragen.
x2z	Die Antwort betrifft den Status der Verbindung zwischen Client und Server oder die angeforderten Datenströme.
x3z	Die Antwort betrifft die Authentifizierung des Clients gegenüber dem Server.

Tabelle 3.18: Bedeutung der zweiten Ziffer einer MASCP-Antwort

gen der Beschreibung in englischer Sprache. Es ist zu beachten, das dieser informelle Text nicht Bestandteil des Protokolls ist.

Wie bereits einleitend angemerkt wurde, werden Antworten zur Übertragung von Informationen vom Server zum Client benutzt. Die Antwort besteht dabei stets aus mehreren Zeilen. Die erste Zeile beinhaltet den Antwort-Code sowie einen informellen Text, der die Antwort beschreibt. Durch ein dem Antwort-Code direkt nachgestelltes Minuszeichen (−) wird signalisiert, daß es sich um eine mehrzeilige Antwort handelt. Die für den Client vom Server übertragenen Informationen werden in den darauf folgenden Zeilen übertragen. Üblicherweise bestehen sie aus Zeilen der Form

```
Parameter=Wert
```

Dabei wird vor dem Antworttext stets ein Leerzeichen eingefügt. Das Ende der Information wird durch eine in der ersten Spalte beginnende Wiederholung des Antwort-Codes markiert. Ihr folgt optional eine textuelle Beschreibung.

Beispiele für den Transport von Informationen in MASCP-Antworten sind der Darstellung der Anfragen in Abschnitt 3.2.10.4, Seite 121ff., zu entnehmen.

3.3 Software-Architektur des MBone-Access-Gateways

In den folgenden Abschnitte wird die Software-Architektur des MBone-Access-Gateways (MAGW) ausgehend von der Betrachtung des Datenflußmodells dargestellt. Hier werden die in den vorangegangenen Kapiteln dargestellten System-Komponenten zu einem Gesamtsystem zusammengefaßt. Die Darstellung konzentriert sich auf die Betrachtung des MBone-Access-Gates (MAG) und des MBone-Access-Servers (MAS). Die Gestalt der Wrapper-Anwendungen lehnt sich in ihrer Architektur daran an; auf die eingehende Beschreibung wurde verzichtet.

Code	Beschreibung	Antwort auf
100	Anweisung wurde empfangen, endgültige Antwort folgt.	
200	Anweisung korrekt, wurde akzeptiert.	PARAM, PORT, DELSTREAM
211	Informationen über System-Status	STATUS
214	Liste der Server-Anweisungen	HELP
221	Verkehrsbeziehung wurde gelöst	QUIT
220	Dienst ist für neuen Nutzer bereit	Verbindungsaufbau
225	Antwort enthält angeforderte Konfigurationsparameter	GETPARAM
226	Antwort enthält Adreß- und Port-Konfiguration des Servers.	GETPORT
227	Angeforderter Datenstrom wird übertragen. Antwort enthält Weitere Informationen.	GETSTREAM
230	Login erfolgreich abgeschlossen	PASS
331	Nutzername korrekt, Passwort erforderlich.	USER
421	Verbindung wurde aufgrund eines Tunnel-Report-Timeouts gelöst.	
427	Der von einer Wrapper-Anwendung gewünschte LAN-Transport-Port ist beim MAG belegt. Neue Portwahl (c=...) erforderlich.	GETSTREAM
500	Syntax-Fehler in Anweisung	
501	Syntax-Fehler in Parametern	
502	Nicht implementierte Anweisung	
503	Falsche Reihenfolge der Anweisungen	
504	Merkmal wird nicht unterstützt	
521	Angegebene Stream-ID ist unbekannt	DELSTREAM
527	Ein Mixer für den angeforderten Datenstrom kann nicht gestartet werden oder ist nicht installiert	GETSTREAM
528	Bei der Anforderung des Datenstroms wurde ein nicht zuordbarer Fehler erkannt.	GETSTREAM
530	Nutzer nicht authentifiziert	PASS

Tabelle 3.19: Verzeichnis der MASCP-Antworten

3.3.1 Architektur des MAG

Die von den Funktionen umfangreichste Komponente des Gesamtsystems ist das MBone-Access-Gate. Daher wurde es als Ausgangspunkt der weiteren Betrachtungen gewählt.

Die für das Gesamtsystem wesentlichen Funktionen des MAG sind:

- Entgegennahme und Weiterleitung der Anforderungen zur Vermittlung multimedialer Echtzeitdatenströme von Wrapper-Anwendungen an den MAS unter Nutzung des MASCP.
- Aufbau und Management des Tunnels zwischen MAG und MAS, der zum Transport der Echtzeitdatenströme über einen "Slow-Speed-Serial-Link" dient.
- Payload-Kompression der Echtzeitdatenströme aus dem privaten Internet des Nutzers als Vorbereitung für die Übertragung in das reguläre Internet unter Nutzung von Transcodern und Mixern (RTP-AV-Datenströme) und Lempel-Ziv-Komprimierern (allgemeine UDP-Datenströme).
- Durchführung der Header-Komprimierung und PDU-Encapsulation als weitere Optimierung für die Datenübertragung über den Tunnel.
- Überwachung der Qualitätsmerkmale des Tunnels hinsichtlich Paketverzögerungen und -Verlusten nebst entsprechender Datenratenregulierung. In diesem Kontext ist auch die Bereitstellung eines Priority-Queuing-Systems erforderlich.
- Bereitstellung von Funktionen zur Ver- und Entschlüsselung der Datenströme zur Wahrung der Privatsphäre für die interpersonelle Kommunikation.

Ein in Funktionsblöcke untergliedertes Diagramm des MAG zeigt Abbildung 3.16. Das Bild zeigt die Konfiguration des MAG während der Vermittlung einer typischen MBone-Konferenz mit den Medien Audio, Video, Whiteboard als Shared-Tool sowie einem Session-Directory-Werkzeug. Auf der linken Seite ist das private Internet des Nutzers am Heimarbeitsplatz dargestellt.

Diesem Multicast-fähigen Netz entnehmen die Payload-Kompressions-Module sowie das Input-Modul die Medien-Datenströme für die Vermittlung in Up-Stream-Richtung. Die wesentliche Aufgabe der vorgelagerten Module ist die Komprimierung der RTP-Payload. Darüber hinaus ist auch die ggf. erforderliche Entschlüsselung vor der Komprimierung sowie die erneute Verschlüsselung nach der Komprimierung des jeweiligen Datenstroms ihre Aufgabe. Die Instantiierung und Parameterisierung dieser Module ist Aufgabe des Stream-Management-Moduls, das am linken Rand der mittleren Management-Ebene dargestellt ist.

Hiernach gelangen die Pakete der Datenströme in das Queuing-Modul. Die Zuordnung des Datenstroms zur Warteschlange wird durch das Queue-Management vorgegeben. Sie erfolgt auf der Basis von Informationen aus dem Stream-Management.

Die weitere Verarbeitung der Datenpakete in Up-Stream-Richtung wird durch das Tunnel-Output-Modul kontrolliert. Dieses Modul ist für die Datenratenregelung zuständig und steuert

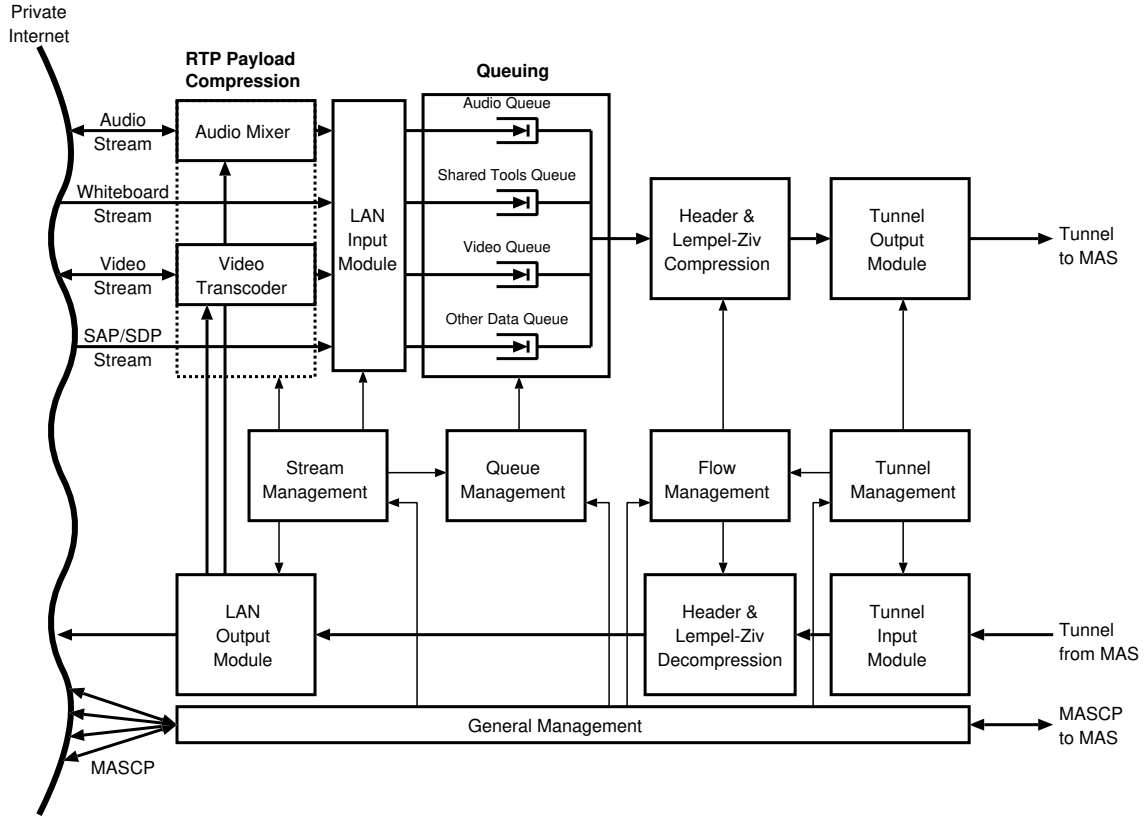

Abbildung 3.16: Funktionsblöcke des MBone-Access-Gate (MAG)

daher die Paketentnahme aus dem Queuing-Modul. Bevor die Pakete in das Output-Modul gelangen, werden sie der Header-Komprimierung unterworfen. Hier wird auch die Lempel-Ziv Komprimierung der UDP-Datenströme vorgenommen.

Die Header-Komprimierung arbeitet eng mit dem Flow-Management zusammen. Hier werden die Informationen über den Kontext und Status der Flows gehalten. Das Flow-Management steht wiederum mit dem Tunnel-Management in Beziehung, da vom Tunnel-Management-Modul die auf Seite 3.2.5.5 beschriebenen Context-State-Requests entgegen genommen werden. Die zentrale Aufgabe des Flow-Managements ist die Verwaltung der Flows und Flow-Kontexte. Die weitgehend autonome Verwaltung dieser Daten erfolgt auf Basis des Konzepts der mit dem RSVP eingeführten *Soft-States* [Bra97, Abschnitt 2.3].

Zur verlustfreien Komprimierung von UDP-PDUs werden der Deflate- und Inflate-Algorithmus benutzt (vgl. [DG96], [Deu96a], [Deu96b], [Roe97]). Dabei handelt es sich um patentfreie und frei verfügbare Verfahren, auf denen auch das weithin bekannte *gzip*-Werkzeug zur Komprimierung allgemeiner Dateien basiert.

Nachdem die Pakete im Tunnel-Output-Modul angelangt sind, werden sie ggf. zu Encapsulated-Packets (vgl. Abschnitt 3.2.6, Seite 112ff.) aggregiert und, wenn erforderlich, für die Übertragung über den Tunnel verschlüsselt. Der letzte Schritt der Bearbeitung im Tunnel-Output-Modul ist die Übergabe der entstandenen PDU an das Betriebssystem.

Weitere Funktionen des Tunnel-Output-Moduls in Kooperation mit dem Tunnel-Management-Modul sind die Einflechtung von Zeitstempeln und Tunnel-Reports in den Paketdatenstrom, sowie die Generierung von Counter-Reset-Requests und Counter-Reset-Confirmations (vgl. Abschnitt 3.2.3, Seite 93ff.).

Die zentrale Aufgabe des Tunnel-Managements sind die Überwachung der Qualitätsmerkmale des Tunnels zur Stauungserkennung (vgl. Abschnitt 3.2.2.1, Seite 71ff.) sowie die Datenratenregelung (vgl. Abschnitt 3.2.2.4, Seite 86ff.) zur optimalen Auslastung des Tunnels. Weiterhin werden hier auch die Zähler für empfangene und gesendete Pakete sowie empfangene und gesendete Octets geführt. Sie dienen, zusammen mit der Auswertung der Tunnel-Reports (vgl. Abschnitt 3.2.4, Seite 96), als Grundlage für die Erkennung von Paketverlusten (vgl. Abschnitt 3.2.3, Seite 93ff.), die wiederum Auswirkungen auf die Regelung des Systems haben. Damit kommt dem Tunnel-Management-Modul zentrale Bedeutung für die korrekte Funktion des MBone-Access-Gateways zu.

Aufgabe des Tunnel-Input-Moduls ist der Empfang von Tunnel-Paketen vom MAS. Wenn der Datenaustausch über den Tunnel verschlüsselt erfolgt, werden die Pakete zuerst entschlüsselt. Danach erfolgt, falls erforderlich, die Zerlegung der Pakete in PDUs. Enthält ein Paket bestimmte Service-PDUs (Zeitstempel, Tunnel-Reports und Counter-Reset-PDUs), werden sie aus dem Paketstrom entfernt und an das Tunnel-Management-Modul weitergeleitet.

Der so modifizierte Paketstrom wird an die Header-Dekomprimierung vermittelt. Dort werden Context-State-Requests aus dem Paketstrom entfernt und an das Flow-Management weitergeleitet. Der verbleibende Datenstrom wird der Header-Dekomprimierung unterzogen und, im Fall komprimierter UDP-Pakete, die Lempel-Ziv Dekomprimierung durchgeführt.

Nachdem auch dieser Schritt abgeschlossen ist, gelangen die Datenpakete in das LAN-Output-Modul. Hier wird mit Hilfe des Stream-Management-Moduls untersucht, an welche Zieladresse

die Daten zu vermitteln sind. Schließlich werden ggf. erforderliche Verschlüsselungen vorgenommen. Die so entstandene UDP-PDU wird an den Empfänger im privaten Internet gesendet. RTP-Datenströme werden über die RTP-Payload-Kompressoren in das lokale Netz weitergeleitet, um die Schleifenerkennung zu ermöglichen.

Die zentrale Systemsteuerung ist Aufgabe des General-Management-Moduls. Auf der Seite des privaten Internets unterhält es TCP-Steuerverbindungen zu den Wrapper-Anwendungen, die, wie in Abbildung 3.5 auf Seite 63 dargestellt, die Medien-Tools steuern. Als Verbindungsprotokoll zwischen MAG und Wrapper-Anwendungen wird das MASCP (vgl. Abschnitt 3.2.10, Seite 119ff.) eingesetzt. Über diese Verbindungen werden die Medienstrom-Anfragen von den Wrapper-Anwendungen übermittelt. Desweiteren unterhält das General-Management-Modul eine TCP-Verbindung zum MAS, über die die Anfragen von Medienströmen weiter an das MAG geleitet werden.

Die wesentliche Aufgabe des General-Management-Moduls ist die Steuerung der verbleibenden Management-Komponenten des MAG. Dazu gehört die Bereitstellung von Konfigurationsdaten und Schnittstellen für den Zugriff auf administrative Daten, die von mehreren Modulen genutzt werden.

Das gesamte MBone-Access-Gate (MAG) ist in einem Prozeß zusammengefaßt, der als Daemon-Prozeß abläuft. Lediglich bei den RTP-Payload-Kompressoren handelt es sich um externe Prozesse, die durch das MAG kontrolliert werden. In jedem privaten Internet sollte nur eine Instanz dieses Prozesses aktiv sein, da das Regelsystem andernfalls nicht optimal arbeiten kann.

3.3.2 Architektur des MAS

Der MBone-Access-Server (MAS) dient dem MBone-Access-Gate (MAG) als Partnerinstanz im regulären Internet. Seine wesentlichen Funktionen sind:

- Betrieb des Tunnels zwischen MAS und MAG, der zum Transport der Echtzeitdatenströme an den Nutzer am Heimarbeitsplatz genutzt wird.
- Entgegennahme von Datenströmen aus dem regulären Internet im Auftrag des Nutzers am Heimarbeitsplatz.
- Payload-Komprimierung der Echtzeitdatenströme aus dem regulären Internet als Vorbereitung für die Übertragung in das private Internet des Nutzers. Dazu werden Transcoder und Mixer (RTP-AV-Datenströme) sowie Lempel-Ziv-Komprimierer (allgemeine UDP-Datenströme, RTCP-Pakete und SAP/SDP-Datenströme) genutzt.
- Durchführung der Header-Komprimierung und PDU-Encapsulation als weitere Optimierung für die Datenübertragung über den Tunnel.
- Überwachung der Qualitätsmerkmale des Tunnels hinsichtlich Paketverzögerungen und -Verlusten nebst entsprechender Datenratenregelung. Ein Priority-Queuing-System sorgt dabei für den Verwurf von Daten, die aufgrund der begrenzten Tunnel-Übertragungskapazität nicht an den Nutzer am Heimarbeitsplatz weiterzuleiten sind.

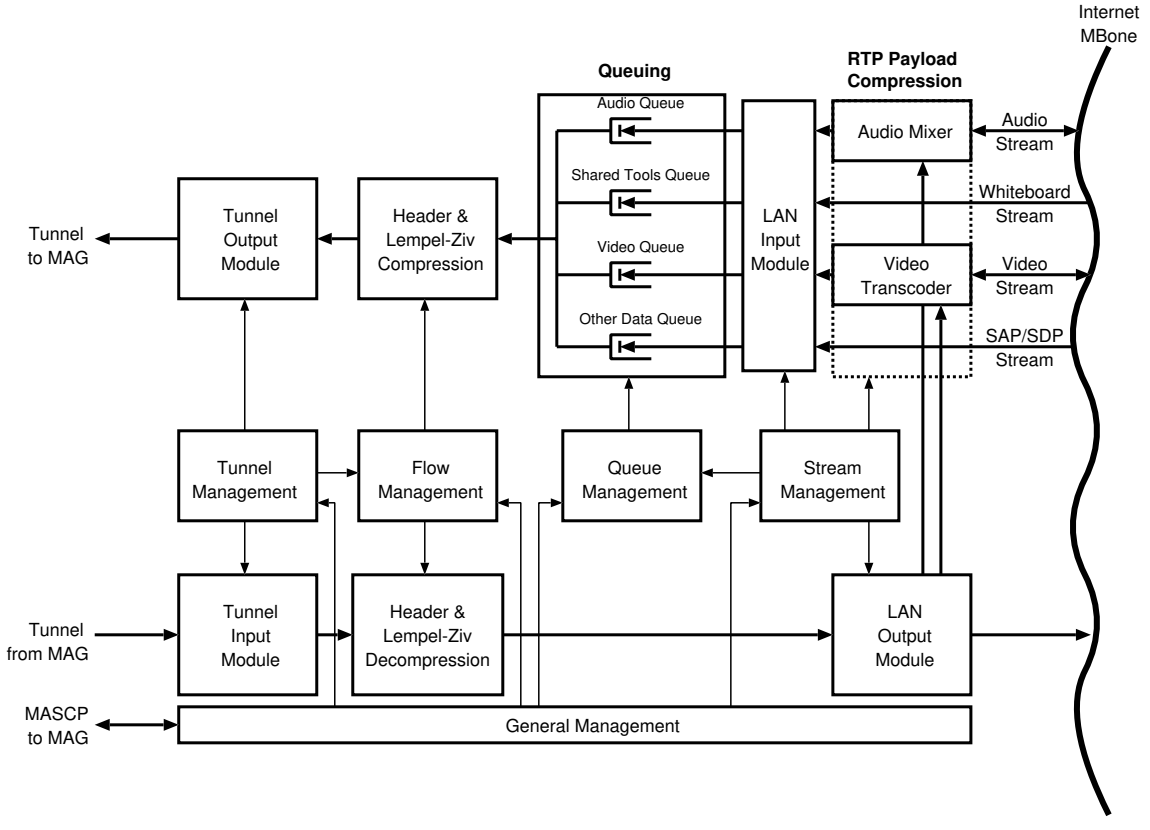

Abbildung 3.17: Funktionsblöcke des MBone-Access-Servers (MAS)

- Bereitstellung von Funktionen zur Ver- und Entschlüsselung der Datenströme zur Wahrung der Privatsphäre für die interpersonelle Kommunikation.

Die Architektur des MBone-Access-Servers (MAS) ähnelt der des MBone-Access-Gates (MAG). Abbildung 3.17 stellt die Funktionsblöcke des MAS dar. Ein offensichtlicher Unterschied zum in Abbildung 3.16 dargestellten MAG ist, daß das zentrale Steuerungsmodul (General Management) nur eine TCP-Verbindung zum MAG unterhält. Für jeden Tunnel wird eine MAS-Instanz betrieben. Die Server-Prozesse werden durch den Internet-Daemon des jeweiligen Rechners gestartet, sobald ein Verbindungswunsch von einem MAG den Server erreicht. Daraus folgt, daß auf einem Server-Rechner gleichzeitig mehrere MAS-Instanzen ablaufen können, von der jede einen Tunnel zu einem Nutzer an einem Heimarbeitsplatz unterhält.

Die verbleibenden Komponenten des MAS sind Spiegelbilder der entsprechenden MAG-Komponenten.

3.4 Zusammenfassung des Kapitels

Dieses Kapitel beschreibt den Entwurf eines Systems zur Vermittlung multimedialer Echtzeitdatenströme zwischen Nutzern an Heimarbeitsplätzen und Konferenzpartnern im interaktiven

Internet auf der Basis des Internet-Conferencing-Protocol-Stacks. Das System trägt den Namen MBone-Access-Gateway (MAGW). Die wesentlichen Komponenten des Systems sind das MBone-Access-Gate (MAG) im privaten Internet des Nutzers am Heimarbeitsplatz sowie der MBone-Access-Server (MAS) im regulären Internet. Beide Systemkomponenten werden durch einen Tunnel verbunden, der der Vermittlung von Echtzeitdatenströmen dient.

Das MBone-Access-Gateway verfügt über folgende Merkmale:

- Optimierung der Paketlaufzeit durch Header-Komprimierung und Payload-Komprimierung.
- Volle Ausnutzung der verfügbaren Tunnel-Übertragungskapazität.
- Elastische Reaktion auf konkurrierende Datenströme.
- Vermeidung von Queuing-Delays und Paketverlusten in IP-Routern auf dem Netzwerkpfad zwischen MAS und MAG.
- Bevorzugte Vermittlung von Audio-Datenströmen durch Priority-Queuing.
- Wahrung der Privatsphäre in verschlüsselten Konferenzen.
- Flexible Einsatzmöglichkeiten für den MBone-Zugriff von Einzel-Heimarbeitsplatzsystemen, privaten Internets und über Firewalls.
- Bestehende MBone-Tools können unverändert am Heimarbeitsplatz benutzt werden.

Die Payload-Komprimierung erfolgt für Audio- und Video-Datenströme verlustbehaftet durch Einsatz vorhandener Audio-Mixer und Video-Transcoder. Andere Datenströme werden verlustfrei nach dem Lempel-Ziv-Algorithmus komprimiert.

Das Verfahren zur Header-Komprimierung orientiert sich am in [CJ97] dargestellten Verfahren und reduziert den Protokoll-Overhead durch die Verwaltung eines Flow-Status an beiden Tunnel-Endpunkten. Nur Status-Änderungen werden übertragen.

Die Übertragungskapazität des Tunnels wird durch die Mitnutzung von UDP-Ports zur Nachrichtenübertragung erhöht. Ein Datenraten-Regelungssystem, welches Stauungen in Routern auf dem Netzwerkpfad zwischen MAS und MAG anhand der Variation der Paketlaufzeiten erkennt, ermöglicht zusammen mit dem Priority-Queuing die volle Ausnutzung der Tunnel-Übertragungskapazität zur Vermittlung von Echtzeitdatenströmen und ermöglicht die elastische Reaktion auf konkurrierende TCP- und UDP-Datenströme.

Das MAGW unterstützt die sichere, verschlüsselte Vermittlung von Nutzdatenströmen bis zum Heimarbeitsplatz mit einem abgestuften Sicherheitskonzept. Die Verschlüsselung der Nutzdatenströme erfolgt auf der Basis des *Data Encryption Standard*-Algorithmus (DES) im *Cypher Block Chaining Mode* (CBC). Die TCP-basierten Kontrolldatenströme werden durch das Secure-Socket-Layer-Verfahren gesichert.

Durch die Unterstützung von Adreßraumumsetzungen ist die flexible Nutzung des Systems in unterschiedlichen Einsatzszenarien möglich. Dazu gehören der MBone-Zugriff von Einzel-Heimarbeitsplätzen, von Arbeitsplätzen in privaten Internets sowie über Firewalls hinweg.

Der unmodifizierte Einsatz bestehender MBone-Werkzeuge wird durch die Bereitstellung von Wrapper-Anwendungen möglich, die die Anforderung von Datenströmen aus dem Mbone sowie die korrekte Parametrisierung der originären Werkzeuge realisieren.

Wenngleich das System für den MBone-Zugriff über 64 kbps ISDN-Wählverbindungen konzipiert wurde, ist es auch für den Einsatz über Fest- und Wählverbindungen mit kleinerer und größerer Übertragungskapazität geeignet. Durch die Ausrichtung auf die Nutzung von Standard-Transportprotokollen (TCP/UDP) über dem Internet-Protokoll (IP) ist es nicht erforderlich, daß auf dem "Slow-Speed-Serial-Link" PPP als Link-Layer-Protokoll eingesetzt wird.

In der Summe der Eigenschaften ist ein Gateway-System entstanden, das vergleichbaren, bestehenden Gateway-Systemen hinsichtlich Effizienz, Funktionalität, Sicherheit und Flexibilität überlegen ist.

Kapitel 4

Implementierung und Bewertung

4.1 Implementierung

Im Rahmen der vorliegenden Arbeit wurde eine exemplarische Implementierung des MBone-Access-Gateways (MAGW) realisiert. Ziel dieser Implementierung war in erster Linie die Überprüfung der Anwendbarkeit der entwickelten Konzepte. Vorrangig galt es die Frage zu beantworten, ob die dynamische Regelung der Datenrate auf Basis der Schätzung der Verzögerung von Paketen durch Verweilzeiten in Warteschlangen in einer realen Produktionsumgebung zuverlässig ist. Ebenfalls von großem Interesse war, ob die Pakete beim Durchlaufen des MBone-Access-Servers und des MBone-Access-Gates nicht so stark verzögert werden, daß dies die interaktive interpersonelle Kommunikation behindert. Daneben galt es eine Reihe weiterer Punkte zu klären, darunter auch die Bewertung der in Abschnitt 3.2.5 (vgl. Seite 97) beschriebenen Header- und Payload-Komprimierungsverfahren.

Die vorliegende Implementierung ist zur Untersuchung der offenen Fragen geeignet, hat aber keinen Produkt-Charakter. Hierfür sind ergänzende Arbeiten zur leichteren Konfiguration des Systems und die generelle Verbesserung der Nutzerfreundlichkeit zu leisten. Ein weiteres Manko ist, daß zur Zeit nur UNIX-Betriebssysteme unterstützt werden.

Die Wahl der Programmiersprache fiel auf C++ [Str91]. Sie bietet eine hohe Effizienz der Programme zur Laufzeit und unterstützt die objektorientierten Programmierung. Alternativ wurden die Sprachen C [KR90] und Java [Sun97] als Implementierungssprachen erwogen. Die Sprache C bietet den Vorteil der besseren Portierbarkeit, sie unterstützt jedoch nicht das objektorientierte Programmiermodell. Java [Sun97] zeigt sich in vielen Anwendungsfällen als eine gute Alternative zu C++, die für das zur Realisierung des Systems erforderliche Laufzeiteffizienz kann jedoch nicht sichergestellt werden. Zudem befand sich Java zum Zeitpunkt der Auswahlphase der Programmiersprache noch in einem zu frühen Stadium: Es war abzusehen, daß die Entwicklungsschritte der in Anfängen steckenden, neuen Programmiersprache von Version zu Version umfangreich sein und dementsprechende Modifikationen der jeweiligen Programm-Quelltexte nach sich ziehen würden. Zudem wurde IP-Multicast anfänglich nicht unterstützt.

Als Compiler wurde der GNU C++-Compiler in der Version 2.7.2 eingesetzt (vgl. [Fre97]). Neben der C++-Basisbibliothek und der C-Bibliothek des jeweiligen Betriebssystems wurde zur verlustfreien Komprimierung der UDP-Payload die frei verfügbare *zlib* in der Version 1.0.4

genutzt (vgl. [DG96], [Deu96a], [Deu96b], [Roe97]). Audio-Mixer und Video-Transcoder wurden ebenfalls nicht selbst implementiert, stattdessen wurden die frei verfügbaren Werkzeuge *rat* (vgl. [SHK+]) und *vgw* (vgl. [AMZ95], [Ami95], [AM]) eingesetzt. Außer den genannten Werkzeugen und Bibliotheken wurden keine weiteren Hilfsmittel für die Implementierung des MBone-Access-Gateways benutzt.

Für den Zugriff auf das Netz wurde die Berkeley-Socket-Schnittstelle [Ste92, Seite 396] benutzt. Alle relevanten UNIX- und Microsoft-Betriebssysteme unterstützen diese Schnittstelle als Kernel-Schnittstelle. Die einzige Ausnahme bilden ältere Sun-Solaris Versionen. Hier wurde bis zur Version 2.6 eine Emulation der Berkeley-Sockets über ein erweitertes Transport-Layer-Interface angeboten, welches sich in das Streams-Konzept des UNIX System V, Release 4, eingliedert.

Auf der Basis der Berkeley-Socket-Schnittstelle entstand eine von der spezifischen Anwendung unabhängige C++-Klassenbibliothek, die unter der Nutzung der System-Funktion *select* asynchrones I/O für alle Gerätegruppen (Dateien, Netzwerk-Sockets, Terminals) ermöglicht. Diese Klassenbibliothek wurde auch für die Implementierung der in Anhang A beschriebenen Meßwerkzeuge eingesetzt.

Die Implementierung des MAGW besteht aus einer Zahl von Klassen, bei deren Entwurf hohe Kohäsion innerhalb der Klassen und geringe Kopplung zwischen den Klassen angestrebt wurde (vgl. [FGV+94, Seite 123ff.]). Dabei entstand ein Rahmenwerk von Klassen, die innerhalb des Gesamtsystems eine gute Wiederverwendbarkeit sicherstellen und dadurch die Pflege und Weiterentwicklung des Systems erleichtern. Die Einzelanwendungen des MAGW, d.h. MAG, MAS und Wrapper, ererben die Grundeigenschaften (MASCP-Client, MASCP-Server) von abstrakten Basisklassen und erweitern diese um die für die jeweilige Anwendung spezifischen Funktionalitäten und Eigenschaften. Die verbleibenden Systemkomponenten wurden auf klassische Entwurfsmuster gemäß [Gam96] abgebildet und realisiert.

Ein häufig auftretendes Problem mit in C++ implementierten Systemen ist die mangelnde Portabilität. Sie ist vor allem auf unterschiedliche Entwicklungsstände der Compiler sowie der begleitenden Bibliotheken zurückzuführen. Diesem Problem wurde mit methodischer Restriktion begegnet, d.h. es wurde auf den Einsatz fortgeschrittener C++-Eigenschaften weitestgehend verzichtet. Die für das Management der Flows erforderlichen Map-Klassen wurden nicht der *Standard Template Library* (vgl. [MS96]) entnommen, sondern es wurden vergleichbare, eigene Klassen eingesetzt. Allerdings ist auch hierfür die Unterstützung von Templates durch den Compiler erforderlich.

Bei der Realisierung des Systems zeigte sich erneut, daß die IP-Multicast-Unterstützung der verschiedenen Betriebssysteme durchaus unterschiedlich ist. Obgleich der Zugriff über die Berkeley-Socket-Schnittstelle erfolgt, differiert die Semantik der Implementierungen, teilweise von einer Betriebssystem-Version zur anderen.

Die vorliegende Implementierung des MAGW entspricht, bis auf kleine Einschränkungen, dem in Kapitel 3 dargestellten Entwurf. Die Einschränkungen betreffen folgende Punkte:

- Die Authentifizierung des Servers gegenüber dem Client auf Basis des Secure-Socket-Layer (SSL) (vgl. [Fre96]) wurde bisher nicht realisiert.

- Die Übertragung verschlüsselter Datenströme gemäß Abschnitt 3.2.9.2 (vgl. Seite 116) wird nur eingeschränkt unterstützt.

- Die in Abschnitt 3.2.8 (Seite 114) beschriebene Sonderbehandlung von SAP/SDP-Datenströmen ist in der vorliegenden Implementierung noch nicht enthalten. Sie werden als normale UDP-Datenströme übertragen.

Die zur Installation und zum Betrieb des MAGW erforderliche Dokumentation ist Bestandteil der für die Distribution vorbereiteten Software-Pakete.

Zusammenfassend ist zu vermerken, daß das Ziel der Bereitstellung einer exemplarischen Implementierung des MAGW für die Überprüfung der Anwendbarkeit der entwickelten Konzepte gelungen ist. Darüber hinaus kann das vorliegende System für die Teilnahme an multimedialen Online-Konferenzen auf der Infrastruktur des Internets benutzt werden. Es hat jedoch, wie bereits erwähnt, noch keinen Produkt-Charakter. Hierfür ist eine Portierung auf die Microsoft-Betriebssysteme Windows-95 und Windows-NT erforderlich. Zudem ist die Nutzerfreundlichkeit zu verbessern. Durch die eingeschränkte Unterstützung für die Übertragung verschlüsselter Datenströme bleiben der vorliegenden Realisierung des MAGW einige der angestrebten Einsatzszenarien verschlossen.

4.2 Messungen

Die exemplarische Implementierung des MBone-Access-Gateways (MAGW) ermöglicht die Überprüfung des Systemverhaltens durch Messungen. In den folgenden Unterabschnitten wird die Meßumgebung beschrieben, in der die Eigenschaften des MAGW überprüft wurden. Darauf aufbauend werden Meßverfahren zur Ermittlung von Paketlaufzeiten, der Datendurchsatzrate des Systems sowie zur Untersuchung des Systemverhaltens bei Überlagerung des Tunnel-Datenstroms mit konkurrierenden TCP-Datenströmen dargestellt. Ergänzend werden die Ergebnisse entsprechender Messungen vorgestellt.

4.2.1 Meßumgebung

Die Messungen zur Überprüfung des Systemverhaltens des MAGW erfolgten im wesentlichen unter Nutzung von Komponenten im Datennetz der Universität Hannover sowie einem privaten Internet. Abbildung 4.1 zeigt die relevanten Komponenten der Meßumgebung.

Im oberen Bereich der Abbildung ist das private Internet zu erkennen, in dem zwei Rechnersysteme zur Durchführung der Messungen dienen. Der Rechner *jack2* fungiert als Arbeitsplatzrechner. Auf ihm kommen im normalen Betrieb die MBone-Werkzeuge wie *vic*, *vat* und *sdr* zusammen mit den entsprechenden Wrapper-Anwendungen zum Einsatz. Für die Messungen werden sie durch die Meßprogramme *udp-echo*, *udp-discard*, *rtest* und *tcpdump* ersetzt.

Der Rechner *jack* verbindet das private Internet über das ISDN-Netz mit dem Datennetz der Universität Hannover. Auf ihm ist das MBone-Access-Gate (MAG) installiert. Zudem fungiert

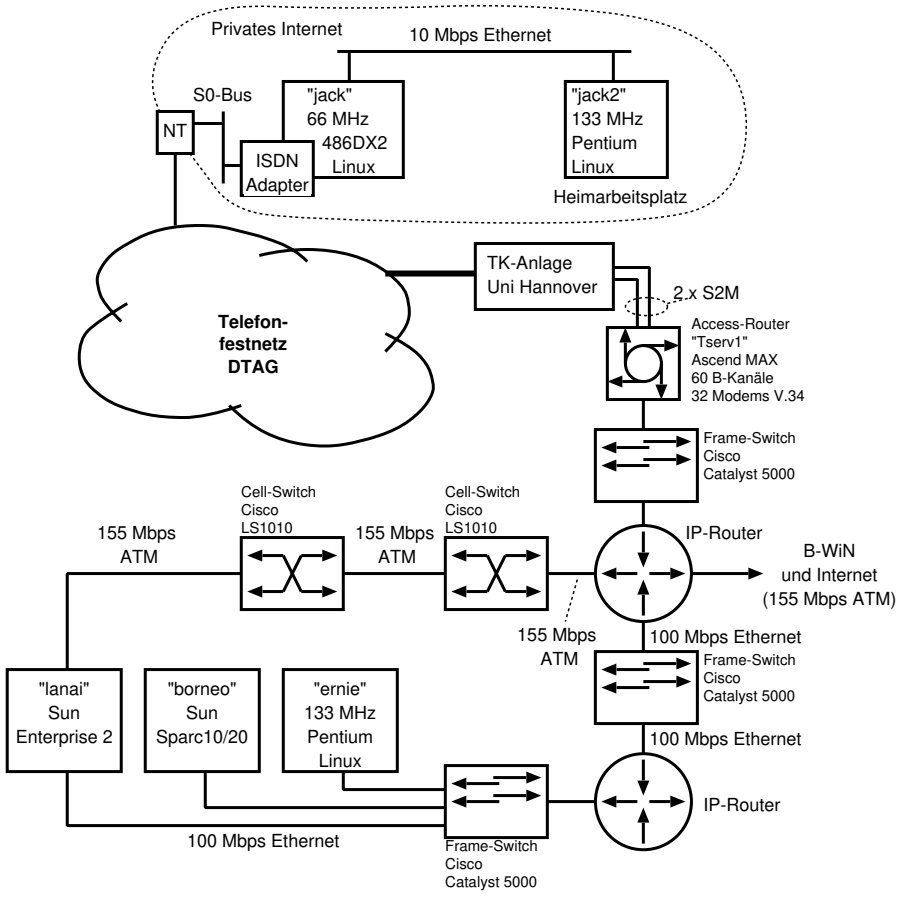

Abbildung 4.1: Meßumgebung zur Durchführung der Messungen

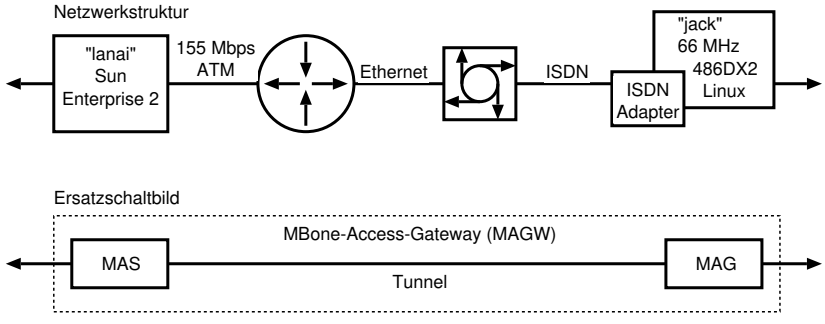

Abbildung 4.2: Struktur des MAGW während der Messungen

er als *Masquerading-Router*[1] für Unicast-Datenströme.

Im unteren Bereich der Abbildung 4.1 ist ein Ausschnitt des Institutsnetzes des Lehrgebiets Rechnernetze und Verteilte Systeme der Universität Hannover dargestellt. Es sind nur die für die Messungen relevanten Rechner dargestellt. Auf dem Sun Enterprise-Server-2 *lanai* ist der MBone-Access-Server (MAS) installiert. Für die Kommunikation zwischen dem MAG und dem MAS wird der ATM-Pfad über die Cell-Switches zum Router benutzt. Die Rechner *ernie* und *borneo* dienen vor allem als Sender für Testdatenströme.

Bei den anderen, in Abbildung 4.1 dargestellten Komponenten handelt es sich um Standard-Elemente in IP-Netzen zur Datenvermittlung. Verbindungen, die nicht anderweitig gekennzeichnet sind, sind in konventioneller 10 Mbps Ethernet-Technik ausgeführt.

Für die Messungen wird das MBone-Access-Gateway als Black-Box aufgefaßt. Abbildung 4.2 hebt den zugrundeliegenden Netzwerkpfad mit den beteiligten Komponenten auf IP-Ebene hervor. Die Rechner *lanai* und *jack* fungieren als MAS und MAG. Der Tunnel-Datenstrom zwischen MAS und MAG wird über einen IP-Router sowie einen Access-Router geleitet. Durch diese Komponenten vergrößert sich die Paketlaufzeit zwischen MAS und MAG gegenüber den Betrachtungen im Entwurf (vgl. Kapitel 3, Seite 55ff.). Diese Vergrößerung der Paketlaufzeit ist aufgrund der niedrigen Serialisierungszeit im ATM-Netz jedoch vernachlässigbar.

4.2.2 Paketverzögerung durch das MAGW

Bei der Vermittlung multimedialer Echtzeitdatenströme ist es von besonderem Interesse, daß die Paketlaufzeit eine obere Schranke nicht überschreitet. Zudem sollte diese Schranke möglichst klein sein. Dies wurde im Analyse-Kapitel ausgeführt (vgl. Abschnitt 2.3.2, Seite 23ff.). In Abschnitt 2.6 (Seite 52ff.) wurde auf der Basis von [Int93] gefordert, daß die Paketlaufzeit vom originären Sender im Breitband-Wissenschaftnetz (B-WiN) zu einem Empfänger an einem Heimarbeitsplatz 400 ms nicht übersteigen soll. Anzustreben ist eine maximale Ende-zu-Ende-Verzögerung von 150 ms. Ergänzende Stichproben-Messungen im B-WiN ergaben maximale

[1] Ein Masquerading-Router ersetzt bei IP-Datagrammen, die aus dem privaten Internet in das reguläre Internet fließen, die originäre Sender-Transportadresse durch eine eigene Transportadresse, wobei die eingesetzte IP-Adresse die des im regulären Internet befindlichen Interfaces des Systems ist. Antworten aus dem Internet werden der gleichen Umsetzung in umgekehrter Richtung unterzogen.

Paketlaufzeiten von 20-30 ms. Zur Erfüllung der gestellten Forderung darf das MAGW für die Übertragung der Pakete daher nicht mehr als 370 ms beanspruchen.

Ziel der Messungen ist, zu prüfen, ob das MAGW die gestellte Forderung erfüllt. Dazu werden die Ergebnisse folgender Messungen dargestellt:

- Messung der Round-Trip-Time zwischen zwei Endsystemen unter Nutzung des MAGW bei einer Datenrate unterhalb der Übertragungskapazität der ISDN-Strecke. Dabei wird die Paketgröße variiert. Dies entspricht dem normalen Betriebsfall des MAGW.
- Messung der Entwicklung der Round-Trip-Time zwischen zwei Endsystemen unter Nutzung des MAGW bei variierenden Bitraten und fester Paketgröße im Zeitverlauf. Diese Messung erlaubt Rückschlüsse auf das Verhalten des MAGW in Überlast-Situationen.

In den folgenden Unterabschnitten wird die Durchführung dieser Messungen beschrieben sowie Ergebnisse der Messungen dargestellt und interpretiert.

4.2.2.1 Round-Trip-Times beim Einsatz des MAGW

Für die Messung der Round-Trip-Times fungiert ein netztopologisch dem MAS oder MAG naher Rechner als Sender eines Meßdatenstroms, der mit dem in Anhang A (Seite 159ff.) beschriebenen Meßprogramm *rtest* generiert wird. Der Datenstrom wird über das MAGW geleitet. Auf der Gegenseite empfängt ein ebenfalls dem MAG oder MAS netztopologisch naher Rechner den durch das MAGW vermittelten Datenstrom und sendet ihn unter Nutzung des (gleichfalls in Anhang A beschriebenen) Programms *udp-echo* über das MAGW an den originären Sender zurück. Dieser registriert beim Empfang der Pakete den Empfangszeitpunkt. Durch den Vergleich des Sendezeitpunktes mit dem Empfangszeitpunkt der emittierten Pakete werden im Anschluß an die Messung die Paketlaufzeiten berechnet und einfache statistische Berechnungen durchgeführt und ausgegeben.

Ziel der Messung ist die Ermittlung der Round-Trip-Time für den Fall, daß die Datenrate des zu übertragenden Datenstroms kleiner ist, als die über den ISDN-Kanal maximal übertragbare Datenrate. Dies sollte für den realen Betriebsfall die Regel sein, da die zu übertragenden Datenströme vor der Übermittlung durch Mixer und Transcoder in ihrer Datenrate begrenzt werden.

Im Fall der Messungen, die zu den hier dargestellten Ergebnissen führten, fungierte der im privaten Internet befindliche Rechner *jack2* als Sender und der im universitären Datennetz befindliche Rechner *ernie* als Empfänger. Der Datentransport erfolgte über eine IP-Multicast-Gruppe. Jede Messung bestand aus einer Reihe von Einzelmessungen. Bei jeder Einzelmessung wurden 50 UDP-PDUs mit einer pro Einzelmessung konstanten Paketgröße emittiert. Die eingestellte Sendedatenrate betrug bei den Einzelmessungen 16 kbps, die Paketgröße variierte zwischen 16 UDP Payload-Bytes und 1400 UDP Payload-Bytes.

Der Tunnel zwischen MAG und MAS war bei allen Einzelmessungen so konfiguriert, daß die Ports nicht mit zur Datenübertragung genutzt wurden. Die UDP-Payload-Kompression (vgl. Abschnitt 3.2.5, Seite 97ff.) und die Zusammenfassung mehrerer originärer Paket zu einem Tunnel-Paket (vgl. Abschnitt 3.2.6, Seite 112ff.) wurden abgeschaltet. Die maximale Tunneldatenrate

UDP-Paketgröße (Payload) (Bytes)	RTT-Anteil Serialisierung ISDN (ms)	min. RTT jack-lanai (ms)	min. RTT jack2-ernie (ms)	Median RTT jack2-ernie (ms)
16	12.75	24.2	30.2	34.0
50	21.25	32.9	40.8	50.8
100	33.75	45.6	52.2	54.2
200	58.75	71.6	82.7	86.0
300	83.75	97.5	104.9	115.3
400	108.75	122.9	132.5	142.4
500	133.75	148.9	160.1	163.8
600	158.75	175.0	184.8	195.4
700	183.75	200.7	212.7	225.0
800	208.75	226.7	238.3	249.2
900	233.75	253.1	269.4	278.2
1000	258.75	277.9	291.0	299.2
1100	283.75	304.5	319.0	334.3
1200	308.75	328.7	343.9	355.1
1300	333.75	355.8	372.3	378.7
1400	358.75	381.1	397.2	406.1

Tabelle 4.1: UDP-RTTs zwischen *jack2* und *ernie* unter Nutzung des MAGW

war auf 61 kbps eingestellt. Die die dynamische Datenratenregelung beeinflussenden Parameter hatten folgende Werte[2]:

Parameter	Wert
dtqmin	150
dtqmax	500
dte	90

Tabelle 4.1 enthält die Ergebnisse der Messung, erweitert um die berechneten minimalen RTT-Anteile, die durch die Serialisierung auf dem ISDN-Kanal entstehen. Weiterhin wurde eine Vergleichsmessung über IP-Unicast zwischen den Rechnersystemen *jack* und *lanai* ohne Nutzung des MAGW, aber mit den gleichen Meßwerkzeugen durchgeführt. Als Referenz wurde die minimale RTT einer jeden Einzelmessung angegeben.

Bezüglich der Meßergebnisse zwischen den über das MAGW verbundenen Endsystemen *jack2* und *ernie* wurden aus der Vielzahl der von *rtest* ermittelten Ergebnisse nur die minimale RTT sowie der Median der RTT dargestellt. Dies sind aus den Erfahrungen mit vergleichbaren Messungen die robustesten Schätzwerte für die RTT.

Eine grafische Darstellung der Meßergebnisse enthält Abbildung 4.3. Die unterste Linie repräsentiert die Meßergebnisse über Unicast zwischen den Rechnern *jack* und *lanai*. Sie dient als Referenz der unter Nutzung des MAGW ermittelten minimale RTT und dem Median der RTT

[2]Die Bedeutung des Parameters *dte* ist in Abschnitt 3.2.2.1, Seite 78, erläutert. Für die Beschreibung der Parameter *dtqmin* und *dtqmax* siehe Abschnitt 3.2.2.4, Seite 86.

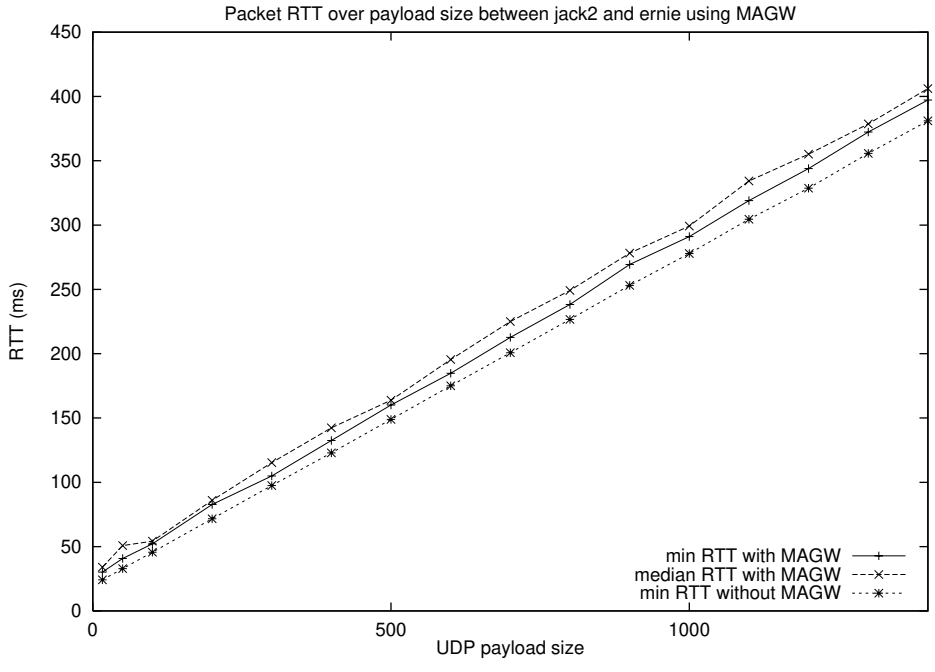

Abbildung 4.3: UDP Round-Trip-Times unter Nutzung des MAGW

zwischen *jack2* und *ernie*. Die mittlere der drei Linien beschreibt die minimale RTT unter Nutzung des MAGW in Abhängigkeit von der Paketgröße. Die obere Linie beschreibt den Verlauf des Median der RTT für den gleichen Fall.

Abbildung 4.3 zeigt, daß die RTTs unter Nutzung des MAGW stets geringfügig höher als ohne Nutzung des MAGW sind. Diese Differenz in der Größenordnung von bis zu 15 ms begründet sich durch Bearbeitungszeiten des MAGW sowie die mehrfache Serialisierung der Pakete auf Ethernet-Strecken beim Durchlaufen des MAGW.

Konzentrieren sich die Betrachtungen auf die ausschließliche Vermittlung von Audio-Datenströmen, ist gemäß Tabelle 2.7 (Seite 40) der Bereich von Paketgrößen bis zu 400 Byte UDP-Payload von Interesse. Hier beträgt der Median der RTT über das MAGW 142.4 ms. Wird unterstellt, daß die durchschnittliche Paketlaufzeit die Hälfte des Medians der RTT beträgt, ergibt sich hierfür ein Wert von 72 ms. Damit wird für reine Audio-Datenströme, deren Datenrate unterhalb der Übertragungskapazität des ISDN-Kanals liegen, die erweiterte Forderung nach einer maximalen Verzögerung durch das MAGW von 120 ms erfüllt.[3]

Auch bei großen Paketen wächst der Median der RTT nicht über 410 ms. Dies entspricht einer Paketlaufzeit in einer Richtung von etwa 205 ms. Folglich wird auch hier die obere Schranke von 370 ms nicht überschritten.

Aus Tabelle 4.1 geht hervor, daß für Pakete ab einer Größe von 100 Byte die RTT hauptsächlich durch die Serialisierung der Pakete auf dem ISDN-Kanal beeinflußt wird. Daran wird

[3]Der Wert von 120 ms ergibt sich gemäß [Int93] aus der Forderung, daß die Verzögerung von 150 ms im normalen Betriebsfall nicht überschritten werden sollte, gekoppelt mit der Erfahrung, daß Paketlaufzeiten im B-WiN in der Größenordnung von 20 ms liegen (vgl. Abschnitt 2.6, Seite 52f. und Abschnitt 4.2.2, Seite 143f.).

deutlich, daß eine signifikante Verringerung der RTT nur durch eine Erhöhung der Datenrate auf dem ISDN-Kanal möglich ist. Geringfügige Verbesserungen lassen sich durch die Header-Kompression bei RTP-Datenströmen sowie die Mitnutzung der Port-Felder von UDP-Datagrammen zur Nutzdatenübertragung erzielen. Da es sich bei dem Meßdatenstrom jedoch um einen UDP-Datenstrom handelte, konnte die Verringerung der RTP-Header-Kompression bei dieser Messung nicht dargestellt werden. Bei der Mitnutzung der UDP-Ports zur Nutzdatenübertragung verringert sich die RTT um 0.375 ms.

4.2.2.2 Entwicklung der Round-Trip-Time bei unterschiedlichen Datenraten

Bei dieser Messung werden dem MBone-Access-Gateway (MAGW) Datenströme variierender Datenrate bei konstanter Paketgröße zugeführt. Auf der Gegenseite wird der vom MAGW übertragene Datenstrom empfangen und an den originären Sender zurückübermittelt. Beim originären Sender werden die Round-Trip-Times je empfangenem Paket sowie die Paketverluste ermittelt. Die Ergebnisse werden für jedes emittierte Paket ausgegeben. Als Generator für den Meßdatenstrom dient erneut *rtest*. Für den Empfang und die Rückübermittlung der Daten kommt wiederum *udp-echo* zum Einsatz.

Die Datenrate der Meßdatenströme wird für die Teilmessungen wie folgt variiert:

- Der Meßdatenstrom weist eine Datenrate auf, die unterhalb der des ISDN-Kanals liegt. Dies entspricht dem normalen Betriebsfall.
- Der Meßdatenstrom weist eine Datenrate auf, die in der Größenordnung geringfügig über der des ISDN-Kanals liegt. Hierdurch würde ohne Einsatz des MAGW langsam eine Stauung im Access-Router aufgebaut.
- Der Meßdatenstrom weist eine Datenrate auf, die die Übertragungskapazität des ISDN-Kanals übersteigt. Hierbei wird unter Verzicht des MAGW schnell eine Stauung im Access-Router aufgebaut.

Ziel der Messung ist, zu untersuchen, wie sich das MAGW im Fall einer Überlastsituation verhält. Von besonderem Interesse sind dabei:

- Die Entwicklung der Round-Trip-Time im Zeitverlauf.
- Das Verhalten der RTT in Abhängigkeit von der Datenrate.
- Die Veränderung der Paketverluste über die Zeit.
- Die erzielten Datenübertragungsraten.
- Der Vergleich des Systemverhaltens des MAGW mit der Übertragung von Datenströmen ohne MAGW.

Die im folgenden dargestellten Meßergebnisse resultieren aus Messungen, die in der eingangs beschriebenen Umgebung durchgeführt wurden. Der Rechner *ernie* diente als Generator des Meßdatenstroms. Der Rechner *jack2* empfing den Meßdatenstrom und sendete ihn zurück.

Sende-datenrate (kbps)	min. RTT ernie-jack2 (ms)	Median RTT ernie-jack2 (ms)	Empfangs-datenrate (kbps)
40	102.9	115.2	40.0
55	119.1	521.1	51.2
70	210.8	530.5	51.6

Tabelle 4.2: UDP-RTTs und erzielte Datenrate unter Nutzung des MAGW bei unterschiedlichen Sendedatenraten

Das MAGW wurde mit Datenströmen von 40 kbps, 55 kbps und 70 kbps belastet. Die Belastung des MAGW mit einem Datenstrom von 40 kbps entspricht grundsätzlich dem normalen Betriebsfall, da die Brutto-Datenrate kleiner als die Übertragungsrate der ISDN-Strecke ist. Ein Datenstrom von 55 kbps führt zu einem Brutto-Datenstrom, der im Grenzbereich der Übertragungskapazität der ISDN-Strecke liegt (vgl. Abschnitt 2.3.2, Seite 23ff.). Aufgrund des durch das Tunnel-Protokoll verursachten Overheads und der Tatsache, daß die UDP-Payload-Kompression im Verlauf dieser Messungen ausgeschaltet war und damit der Overhead des Tunnel-Protokolls nicht vollständig ausgeglichen wird, überschreitet die Brutto-Datenrate Übertragungskapazität der ISDN-Strecke geringfügig (vgl. Tabelle 4.2).[4] Ein UDP-Datenstrom von 70 kbps führt zu einer Brutto-Datenrate, die die Übertragungskapazität der ISDN-Strecke deutlich übersteigt. Letztlich entsprechen die gewählten Übertragungsraten den in Abschnitt 2.3.5 (Seite 32ff.) benutzten Werten und erlauben somit den direkten Vergleich mit dort erzielten Meßergebnissen.

Als Zieltransportadresse wurde eine IP-Multicastgruppe benutzt. Zudem wurde eine Vergleichsmessung durchgeführt, bei der ein Unicast-Datenstrom mit 70 kbps vom Rechner *ernie* an den Rechner *jack* gesendet wurde, ohne das MAGW zu nutzen. Bei allen Teilmessungen wurde mit Paketen gearbeitet, die 300 Byte UDP-Payload enthielten. Während jeder Teilmessung wurden 1000 Pakete vom *rtest*-Programm auf dem Rechner *ernie* emittiert.

Der Tunnel zwischen MAG und MAS war bei allen Einzelmessungen in gleicher Weise wie bei der vorhergehenden Messung konfiguriert (vgl. Abschnitt 4.2.2.1, Seite 144ff.).

Die Messung führte zu den in Tabelle 4.2 dargestellten Ergebnissen hinsichtlich der RTT sowie der erzielten Empfangsdatenrate. Die Entwicklung der RTT über die Zeit gibt die in Abbildung 4.4 dargestellte Grafik wieder. Zur besseren Übersichtlichkeit werden hier nur die RTTs der ersten 500 Pakete wiedergegeben. Auf dem Round-Trip verworfene Pakete sind durch einen RTT-Wert von 0 markiert. Ergänzt wird die Grafik durch die Darstellung der RTT-Entwicklung auf der direkten Übertragungsstrecke zum Rechner *jack* ohne Nutzung des MAGW.

Die in Tabelle 4.2 enthaltenen Angaben zur minimalen RTT suggerieren, daß dieser Wert von der Sendedatenrate abhängig ist. Dieser Sachverhalt ist nicht gegeben. Im Fall der Messung mit 40 kbps wird der Übertragungskanal nicht ausgelastet. Das MBone-Access-Gateway kann eingehende Paket in der Regel unmittelbar übertragen. In den Input-Queues des MAGW entstehen keine Warteschlangen. Daher ist die Angabe des minimale RTT hier korrekt. Der Wert von 102.9 ms zeigt dies deutlich an. Er ist geringer als der bei der Ermittlung der RTT mit geringeren Datenraten ermittelte Wert. Bei den Messungen mit 55 und 70 kbps ist dies nicht der Fall. Hier

[4] Ergänzend ist anzumerken, daß während der durchgeführten Messung das MAGW die Tunnel-Übertragungsrate auf 61 kbps begrenzt hat (vgl. Abschnitt 4.2.2.1, Seite 144f.).

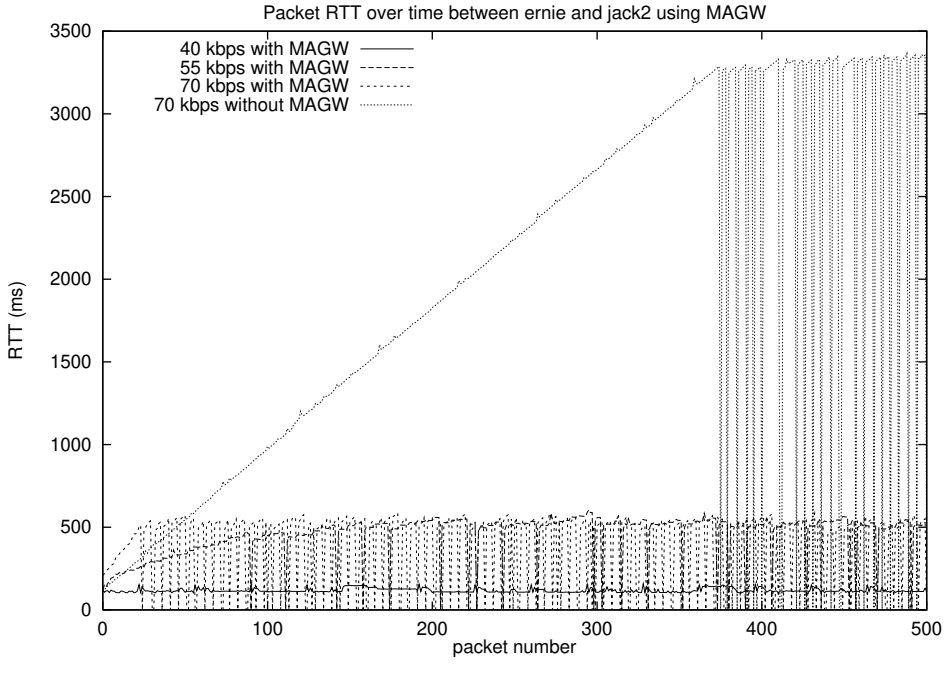

Abbildung 4.4: RTT Entwicklung unter Nutzung des MAGW über die Zeit

entspricht der minimale RTT-Wert der RTT des ersten übermittelten Pakets. In beiden Fällen liegt die Empfangsdatenrate unter der Sendedatenrate. Daher entsteht nach dem Empfang des ersten Pakets des Meßdatenstroms eine Warteschlange, die sich über die Dauer der Messung hinweg nicht abbaut.

Der Median der RTT für den 40 kbps Datenstrom unterscheidet sich nachhaltig vom Median des RTT für 55 bzw. 70 kbps Datenströme, die in der gleichen Größenordnung liegen. Grund hierfür sind die Input-Queues des MAGW. In der vorliegenden Implementierung stehen für alle Warteschlangen gemeinsam fünf Paketspeicher zur Verfügung. Im Fall der durchgeführten Messung hatten alle zu übertragenden Datenpakete die gleiche Priorität. Dabei entartet das Priority-Queuing-Verfahren zu einem FIFO-Queuing-Verfahren. Damit ist im Fall der Überlastung des Systems mit einem Datenstrom hoher Priorität der Median der RTT linear von der Anzahl der Paketspeicher der Queue abhängig.

In Abbildung 4.4 ist die Entwicklung der RTT über die Zeit deutlich zu erkennen. Beim 40 kbps Datenstrom treten keine Paketverluste auf und die RTT behält über den Zeitraum der Messung hinweg ihr initiales Niveau.

Beim 55 kbps Datenstrom entwickelt sich die RTT vom initialen Wert in der Größenordnung der RTT des 40 kbps Datenstroms auf einen Wert in der Größenordnung von 500 ms ohne zwischenzeitliche Paketverluste. Von diesem Punkt an treten in Abständen Paketverluste auf. Die RTT behält ihre Größenordnung.

Die RTTs des 70 kbps Datenstroms steigen schnell auf das Niveau von 500 ms. Über den weiteren Verlauf der Messung behalten sie in etwa diesen Wert. Die Paketverlustrate ist deutlich höher als beim 55 kbps Datenstrom. Der Paketverlust weist im Zeitverlauf ein regelmäßiges Muster auf.

Einen gänzlich anderen Charakter weist der Datenstrom auf, der ohne Nutzung des MAGW übertragen wurde. Die RTT steigt linear vom initialen Wert des 40 kbps Datenstroms auf einen Wert von über 3 Sekunden. Dieser Wert wird nach dem Empfang von etwa 350 Paketen erreicht. Von diesem Punkt an werden regelmäßig Pakete verworfen, wodurch das Niveau der RTT in dieser Darstellung gleich bleibt. Bei Messungen des Laufzeitunterschiedes beim Empfänger mit größeren Paketen (1000 Pakete, 1000 Byte UDP-Payload) wurden ohne die Nutzung des MAGW Laufzeitunterschiede von mehr als 8 Sekunden ermittelt.

Die Meßergebnisse zeigen deutlich, daß das MBone-Access-Gateway die Paketlaufzeit sicher begrenzt. Auch im ungünstigsten Betriebsfall, der dauerhaften Überlastung der Input-Queue des MAGW, wächst die Round-Trip-Time für Pakete typischer Audio-Datenströme nicht wesentlich über 500 ms. Die damit verbundene Paketlaufzeit in eine Richtung beträgt unter der Annahme unsmymmetrischen Paketlaufzeiten ca. 370 ms.[5]

4.2.3 Datenübertragungsrate des MAGW

Für die Konfiguration und den Betrieb des MBone-Access-Gateways (MAGW) ist seine maximale Datenübertragungsrate von Interesse. Sie wird bei der Nutzung über ISDN-Kanäle vor allem durch die Datenrate des ISDN-Kanals (64 kbps) begrenzt. Dieser Abschnitt beschreibt, welche Datenübertragungsrate unter Nutzung des MAGW für einen multimedialen Echtzeitdatenstrom erreicht wird und wie hoch die Auslastung des ISDN-Kanals dabei ist.

Dabei müssen zwei Ausprägungen von Datenströmen unterschieden werden:

- RTP-Datenströme.
- Andere UDP-Datenströme.

Der wesentliche Unterschied liegt in der Behandlung dieser Datenströme durch das MAGW. RTP-Datenströme werden einer UDP-Header-Kompression sowie einer RTP-Header-Kompression unterworfen, andere UDP-Datenströme jedoch nur der UDP-Header-Kompression. Dadurch ergeben sich bei der Betrachtung des MAGW als Black-Box unterschiedliche, maximale Datenübertragungsraten.

Die maximale Datenübertragungsrate von RTP-Datenströmen wird bestimmt, indem dem MAGW ein RTP-Datenstrom zugeführt wird, der die maximale Datenübertragungsrate des MAGW sicher übersteigt. Auf der Seite des Empfängers wird der vom MAGW vermittelte Datenstrom protokolliert und später ausgewertet. Zur Durchführung dieser Messung ist das im Anhang A beschriebene Programm *udp-discard* geeignet.

Die maximale Datenübertragungsrate von allgemeinen UDP-Datenströmen wird in gleicher Weise bestimmt. Auch hier kann ein RTP-Datenstrom vermittelt werden, das MAGW muß jedoch so eingestellt werden, daß er als UDP-Datenstrom übertragen wird. Günstiger ist es, einen

[5] Diese Abschätzung begründet sich auf der Annhame, daß im Fall dieser Messung die Verzögerung im wesentlichen durch die Queue des MAS und die Serialisierungszeiten der Pakete auf dem ISDN-Kanal entsteht. Wenn die Queue des MAG beim Eintreffen eines Pakets leer ist (weil das MAG die Daten maximal mit der maximalen Übertragungsrate des ISDN-Kanals empfangen kann), beträgt die Serialisierungszeit eines 1000-Byte großen UDP-Pakets incl. Protokoll-Overhead ca. 130 ms. Dieser Wert ist vom ermittelten RTT zu subtrahieren.

MAGW Vermittlungs-Modus	Durchschn. Paketgröße UDP Payload (Bytes)	Sende-datenrate (kbps)	Empfangs-datenrate UDP Payload (kbps)	Empfangs-datenrate ISDN (kbps)
UDP	300	55	54.2	60.9
UDP	300	70	54.3	60.9
UDP	324	70	54.7	61.0
UDP	1000	70	58.9	60.9
RTP/Audio	325.9	67	55.7	62.0
RTP/Video	732.8	505	58.6	61.6

Tabelle 4.3: Übertragungsleistung des MAGW-Tunnels

vom Programm *rtest* generierten UDP-Datenstrom zu nutzen, da dieser einfacher und genauer zu parameterisieren ist.

Die im folgenden dargestellten Meßergebnisse zu RTP-Datenströmen wurden gewonnen, indem dem MAGW der Audio- und Video-Datenstrom des MBone-Kanals *FAU-TV* (vgl. Abschnitt 2.4.3, Seite 39) zugeführt wurde und am Heimarbeitsplatz der empfangene Datenstrom mit *udp-discard* protokolliert wurde. Für die Ermittlung der maximalen UDP-Datenübertragungsrate fungierte ein auf dem Rechner *ernie* installiertes *rtest*-Programm.

Die Empfangsdatenrate wurde auf der UDP-Ebene nach dem in Anhang B (Seite 175ff.) dargestellten Verfahren bestimmt. Die ebenfalls angegebene Übertragungsrate auf dem ISDN-Medium wurde in gleicher Weise berechnet. Dabei wurden zur Paketgröße jedoch 2 Byte für das Tunnel-Protokoll zwischen MAG und MAS, 8 Byte UDP-Header, 20 Byte IP-Header sowie 7 Byte für die PPP/HDLC-Encapsulation addiert. Unberücksichtigt blieben auf der Ebene des Tunnel-Protokolls die Datenvolumina für unkomprimierte UDP-Header, Timestamps und Tunnel-Reports. Die Datenratenregelung des Tunnels hingegen berücksichtigt diese Daten. Ebenfalls unberücksichtigt blieb das Bit-Stuffing auf HDLC-Ebene.

Auch bei der Steuerung der Tunnel-Datenrate bleiben diese Daten unberücksichtigt. Aus diesem Grund wurde der Tunnel-Parameter *cinit* auf 61 kbps eingestellt. Dadurch wurde sichergestellt, daß unmittelbar nach dem Start der Datenübertragung über den Tunnel schnell die maximale Tunnelübertragungsrate erreicht wurde.

Die Meßergebnisse der UDP-Datenströme zeigen, daß sich die erzielbare Empfangsdatenrate umgekehrt proportional zur Größe der übertragenen Pakete verhält. Grund hierfür ist der Protokoll-Overhead für die UDP, IP und PPP/HDLC-Header (vgl. Abschnitt 2.3.2, Seite 23ff.). Die auf dem ISDN-Kanal erzielte Bitrate entspricht dem durch den entsprechenden Tunnel-Parameter eingestellten Wert von 61 kbps.

Bei der Übertragung von RTP-Datenströmen sind geringfügig höhere Übertragungsraten erzielt worden. Grund hierfür ist die RTP-Header-Kompression. Das über den Tunnel übertragene Paket ist dabei durchschnittlich kleiner, als das unkomprimierte Paket im LAN. Beim Audio-Datenstrom ermöglicht die Header-Kompression eine Anhebung der Datenrate um 1 kbps. Bei der Übertragung von Video-Datenströmen, deren Pakete durchschnittlich größer sind, ist die Auswirkung der RTP-Header-Kompression meßbar, jedoch betragsmäßig kleiner.

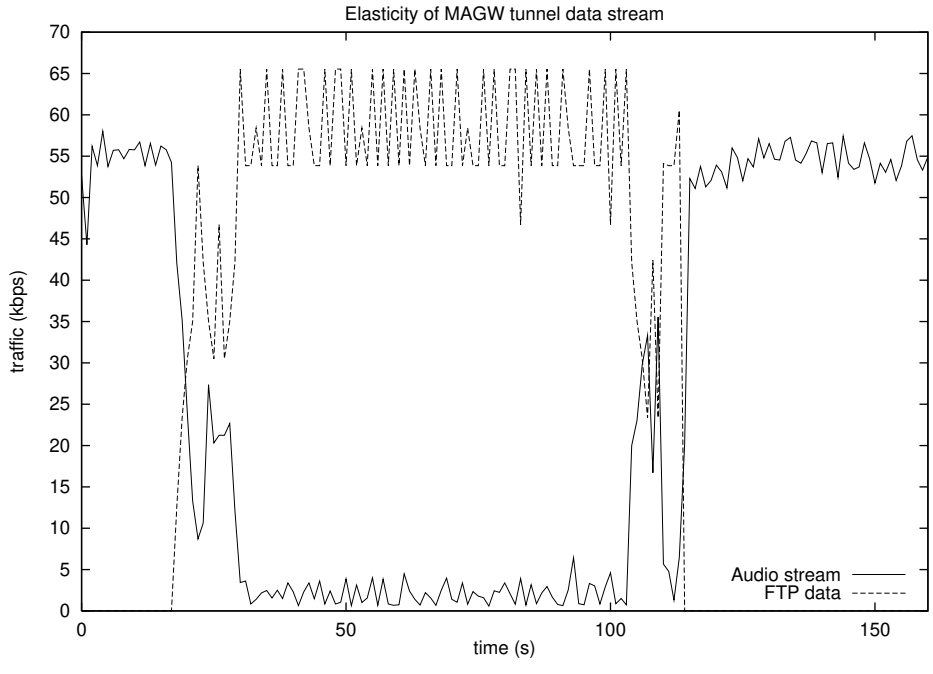

Abbildung 4.5: Elastizität des MAGW Tunnel-Datenstroms

4.2.4 Überlagerung unterschiedlicher Datenströme

Das MBone-Access-Gateway (MAGW) reagiert gemäß seiner Spezifikation elastisch auf konkurrierende Datenströme (vgl. Abschnitt 3.2.2.2, Seite 82ff.). Die nachfolgend dargestellte Messung dient der Überprüfung dieser Eigenschaft.

Über die Dauer der Messung hinweg wird dem MAGW ein Audio-Datenstrom zugeführt, der im Downstream aus dem regulären Internet an einen Heimarbeitsplatz zu vermitteln ist. Der Audio-Datenstrom wird so gewählt, daß er die Kapazität des Tunnels zwischen MBone-Access-Server (MAS) und MBone-Access-Gate (MAG) auslastet. Nach einer kurzen Wartezeit wird durch den Nutzer am Heimarbeitsplatz ein konkurrierender TCP-Datenstrom aus dem Internet an den Heimarbeitsplatzrechner initiiert, z.B. durch die Übertragung einer Datei mittels FTP oder durch den Abruf einer Seite aus dem World-Wide-Web. Mittels des Programms *tcpdump* werden die am Heimarbeitsplatzrechner empfangenen Pakete protokolliert.

Bei der im Anschluß erfolgenden Auswertung werden die relevanten Datenströme gefiltert und die Ergebnisse als Diagramm der Empfangsdatenraten über den Zeitverlauf der Messung dargestellt.

Die im folgenden dargestellten Meßergebnisse wurden gewonnen, indem der Audio-Datenstrom des MBone-Kanals *FAU-TV* (vgl. Abschnitt 2.4.3, Seite 39) über das MAGW zum Heimarbeitsplatzrechner *jack2* vermittelt wurde. Nach ca. 20 Sekunden wurde parallel eine ca. 640 KB große Datei von einem Rechner im universitären Datennetz auf den Heimarbeitsplatzrechner unter Nutzung des FTP-Protokolls übertragen. Nachdem die Dateiübertragung abgeschlossen war, wurde die Messung nach weiteren 30 Sekunden beendet.

Abbildung 4.5 zeigt die Ergebnisse einer solchen Messung. Die elastische Reaktion des MAGW

auf den konkurrierenden TCP-Datenfluß ist deutlich zu erkennen: Der nach ca. 20 Sekunden einsetzende FTP-Datenstrom drängt den Audio-Datenstrom stark zurück. Nachdem die Datei-Übertragung beendet ist, erkennt das MAGW die Entlastung der Queue des Access-Routers und hebt den Audio-Datenstrom schnell wieder auf das Niveau vor Beginn der FTP-Übertragung an. Bei dem während der FTP-Übertragung verbleibenden Datenstrom handelt es sich um RTCP-Pakete des RTP-Datenstroms (vgl. Abschnitt 2.4.2, Seite 37ff.).

4.3 Bewertung der Meßergebnisse

Die in den vorangegangenen Abschnitten dargestellten Meßergebnisse zeigen, daß die exemplarische Implementierung des MBone-Access-Gateways (MAGW) der gestellten Forderung – der Übertragung multimedialer Echzeitdatenströme über Internet-Zugangsnetze geringer Datenrate – gerecht wird.

Das Priority-Queuing-System und die dynamische Datenratenregulierung verhindern den Aufbau von Stauungen in den vermittelnden Komponenten entlang des Netzpfades zwischen den verteilten Komponenten des MAGW. Dadurch wird die Paketlaufzeit zwischen Sender und Empfänger begrenzt und ermöglicht die interaktive interpersonelle Kommunikation auf der Basis von Transportprotokollen, die auf dem User-Datagram-Protocol des TCP/IP-Protokoll-Stacks aufsetzen.

Die durchgeführten Messungen zur Ermittlung der maximalen Datenübertragungsrate zeigen, daß unter Nutzung des MAGW stets die maximal mögliche Datenübertragungsrate des ISDN-Kanals genutzt werden kann, ohne daß nachhaltige Paketverzögerungen entstehen. Datenströme geringer Priorität werden durch Datenströme hoher Priorität verdrängt.

Die UDP- und RTP-Headerkompression hat sich als geeignetes Mittel zur Reduktion der Datenrate auf der Tunnelstrecke zwischen MBone-Access-Server (MAS) und MBone-Access-Gate (MAG) erwiesen. Die Datenrate von RTP-Audio-Datenströmen kann um mehr als 1 kbps, abhängig von der Codierung, reduziert werden.

Die elastische Reaktion des MAGW auf konkurrierende Datenströme ermöglicht die Nutzung von Informationsdiensten in einer laufenden Konferenz über den gleichen Zugangskanal. Dabei verhindert die dynamische Datenratenregelung auf der Basis der Abschätzung des Queuing-Delays den Aufbau von Warteschlangen in den vermittelnden Komponenten entlang des Datenpfades bestmöglich.

Kapitel 5

Zusammenfassung und Ausblick

Der wesentliche Beitrag der vorliegenden Arbeit ist die Entwicklung eines neuen Konzepts für die Übertragung multimedialer Echtzeitdatenströme über Internet-Zugangsnetze geringer Datenrate. Es hebt sich von vergleichbaren Ansätzen ab, indem der Aufbau von Stauungen in Hochlastsituationen in den vermittelnden Komponenten des Netzpfades vermieden wird. Dies begrenzt die Paketlaufzeiten und Paketlaufzeitschwankungen nachhaltig.

Das Konzept basiert auf der Abschätzung der Verweilzeiten von Paketen in den Warteschlangen vermittelnder Komponenten durch den Nachrichten-Empfänger unter Berücksichtigung der von der Paketgröße abhängigen Serialisierungszeit der Pakete. Die so gewonnenen Informationen über den aktuellen Lastzustand des Netzwerkpfades dienen als Grundlage für eine dynamische Datenratenregelung beim Nachrichten-Sender.

Ziel der Arbeit ist die Bereitstellung eines Systems, das Nutzern über Internet-Zugangsnetze mit geringer Datenrate die interaktive interpersonelle Kommunikation mittels multimedialer Echtzeitdatenströme ermöglicht. Das Konzept konzentriert sich auf Anwender im wissenschaftlichen Umfeld, insbesondere Studierende. Die Forderung dieser Zielgruppe nach Kommunikation in Gruppen mit mehr als zwei Mitgliedern wird erfüllt, indem IP-Multicast als Standard-Adressierungsverfahren unterstützt wird. IP-Unicast wird als Spezialfall des Multicast behandelt.

Das das Konzept realisierende System führt den Namen *MBone-Access-Gateway* (MAGW). Es handelt sich um ein verteiltes System, bestehend aus den beiden Diensten *MBone-Access-Server* (MAS) und *MBone-Access-Gate* (MAG). Der MBone-Access-Server befindet sich auf einem Rechner im regulären Internet und vermittelt Datenströme aus dem Internet an ein zugeordnetes MBone-Access-Gate. Das MBone-Access-Gate nimmt Datenströme vom MBone-Access-Server entgegen und leitet diese an einen Einzelarbeitsplatz, in ein privates Internet oder Intranet weiter. Umgekehrt vermittelt das MAG Datenströme vom Nutzer über den MAS in das reguläre Internet. MAS und MAG stehen über einen UDP-Tunnel in Verbindung, der zur Übertragung der Nutzdaten sowie zum Austausch von Informationen zum aktuellen Lastzustands des Datenpfades dient. Zudem besteht eine TCP-Verbindung zwischen MAG und MAS, die der Authentifizierung und der Anforderung multimedialer Datenströme dient.

Bei der Übertragung der Echzeitdatenströme über den UDP-Tunnel werden die Pakete einer Header-Kompression sowie, abhängig vom Charakter des Datenstroms, optional einer verlust-

behafteten oder verlustfreien Kompression unterzogen. Dadurch wird die Datendurchsatzrate des Tunnels erhöht, die Verzögerung der Pakete durch das MAGW verringert und die aus der Kapselung der Nutzdatenpakete in UDP-Pakete resultierenden, nachteiligen Folgen ausgeglichen.

Für die Vermittlung von Datenströmen aus dem Internet in private Internets unterstützt das MAGW die Umsetzung von Adressen des privaten Internets in Adressen des regulären Internets. Ergänzend ist der Betrieb des Systems über Firewalls vorgesehen.

Der Anwender kommuniziert mit dem System über sogenannte Wrapper-Anwendungen. Sie umgeben die Standard-Anwendungen für die multimediale Echtzeit-Kommunikation und fordern benötigte Datenströme aus dem Internet an. Bei Beendigung des Medien-Werkzeugs signalisiert die Wrapper-Anwendung dem MAG, daß die Übertragung des jeweiligen Datenstroms eingestellt werden kann. Damit ist der Einsatz der etablierten Medien-Werkzeuge ohne Modifikation auch am Heimarbeitsplatz möglich.

Konzeptionell unterstützt das MAGW die vertrauliche Kommunikation durch Verschlüsselung der Datenströme mit einem abgestuften Verfahren. Der Endnutzer entscheidet von Fall zu Fall, welches Sicherungsverfahren angewendet wird. Die vorliegende Implementierung des MAGW unterstützt diese Option allerdings nur eingeschränkt.

Wenngleich das Augenmerk bei der Entwicklung des MAGW bei der Nutzung von ISDN-basierten Internet-Zugangsdiensten lag, ist das System auch über andere Zugangsnetze nutzbar. Wesentlichen Anteil an der damit verbundenen Flexibilität hat die Beschränkung auf die Nutzung der Internet-Transportprotokolle TCP und UDP.

Die vorliegende Arbeit untergliedert sich in fünf Kapitel. Das erste Kapitel führt in den Themenkomplex multimedialer Online-Konferenzen ein und beschreibt die Motivation für die Ausarbeitung dieser Arbeit.

Das zweite Kapitel, die Analyse, beschreibt die Struktur von Internet-Zugangsnetzen sowie die in diesem Kontext eingesetzten Protokolle. Detailliert wird der ISDN-Zugang zum Internet untersucht. Dazu gehören die Modellierung des Zugangssystems durch ein Warteschlangen-Modell, die Analyse von Paketlaufzeiten und maximalen Datenraten. Weiter werden die Überlagerung von Datenströmen unterschiedlicher Elastizität sowie das Verhalten des Zugangssystems in Hochlastsituationen beschrieben.

Im weiteren Verlauf der Analyse werden der Multicast-Backbone (MBone) und die in seinem Kontext eingesetzten Protokolle vorgestellt. Besondere Berücksichtigung findet dabei das Real-Time-Transport-Protocol (RTP). Ergänzend werden häufig im MBone eingesetzte Codecs beschrieben und die Eigenschaften typischer Datenströme auf dem MBone dargestellt. Letztlich wird eine Übersicht bestehender Systeme zum Anschluß von Heimarbeitsplätzen an den MBone gegeben. Das Analyse-Kapitel schließt mit der Zusammenfassung der Anforderungen an den Entwurf des MBone-Access-Gateways.

Kapitel 3 dokumentiert den Entwurf des MBone-Access-Gateways. Ausgangspunkt ist die Beschreibung der Systemstruktur. Sie geht auf die generelle Systemstruktur ein, stellt Einsatzszenarien dar, beschreibt die Tunnelstruktur zwischen MAS und MAG und behandelt die Architektur der Wrapper-Anwendungen. Darauf folgt die Beschreibung der grundlegenden Systembausteine des MAGW. Dies sind neben anderen das Queuing-Verfahren, Maßnahmen zur Vermeidung von

Stauungen, Erkennung von Paketverlusten, das Konzept der Tunnel-Reports sowie die Datenkomprimierung während der Tunnel-Übertragung. Der Entwurf wird mit der zusammenfassenden Darstellung der Software-Architektur des MAGW abgeschlossen.

Das vierte Kapitel, Implementierung und Bewertung, beschreibt die exemplarische Realisierung des MBone-Access-Gateways und untersucht das Systemverhalten anhand von Messungen mit synthetischen und realen Datenströmen. Schließlich werden die Meßergebnisse zusammenfassend dargestellt.

Das letzte Kapitel beinhaltet die Zusammenfassung und endet mit dem Ausblick auf die weiteren Entwicklungsmöglichkeiten des MAGW.

Ein Thema, welches in der aktuellen Diskussion um die Übertragung multimedialer Echtzeitdatenströme im Internet großen Stellenwert hat, ist das *Resource Reservation Protocol* (RSVP) [Bra97]. Erste Implementierungen für vermittelnde Komponenten und Endsysteme liegen vor und werden in Testbeds eingesetzt. Noch ist allerdings unklar, ob ein umfassender Einsatz des RSVP im Internet möglich sein wird. Insbesondere die Verwaltung der Status-Informationen für sehr große Flow-Zahlen in zentralen Backbone-Routern des Internets ist dabei ein wesentliches Problem.

Mit Blickrichtung auf diese Arbeit stellt sich die Frage, ob der Einsatz des Resource-Reservation-Protocol das MAGW überflüssig macht. Dies ist nicht der Fall. Das MAGW profitiert von der mit RSVP einhergehenden Einführung fairerer Queuing-Verfahren in IP-Routern. Zudem wird es möglich, den Aufbau von Stauungen für den Tunnel-Datenstrom zu verhindern. Damit einher geht, daß der Tunnel-Datenstrom unelastisch gegenüber konkurrierenden TCP-Datenströmen wird. Ob dies im Interesse des Nutzers liegt, bleibt zu untersuchen. Nutzer des MAGW profitieren von der Einführung des RSVP, indem das MAGW als Proxy für Quality-of-Service fungieren kann. Damit würden Anwendungen, die RSVP nicht explizit unterstützen, von seinen Vorteilen profitieren. Das MAGW-Signalisierungsprotokoll unterstützt bereits die dafür erforderlichen Parameter.

Besondere Vorteile bietet das MAGW im Zusammenhang mit der Einführung der nächsten Generation des Internet-Protokolls (IPv6) [DH95]. Durch die Vergrößerung des IP-Headers von 20 auf 40 Bytes verringert sich die Nutzdatenrate auf der Tunnel-Übertragungsstrecke deutlich. Dabei helfen die in der Arbeit entwickelten Verfahren zur Kompression der Protokoll-Header und der Zusammenfassung mehrerer Nutzdatenpakete zu einem Tunnel-Paket, den Verlust an Nutzdaten-Übertragungskapazität zu minimieren. Die aktuelle Implementierung der MAGW-Protokolle unterstützt IPv6 allerdings noch nicht.

Anhang A

Rtest – Emulation von Echtzeitdatenströmen

A.1 Motivation

Anwendungen zur multimedialen, interpersonellen Kommunikation erzeugen Datenströme mit grundlegend anderer Charakteristik als klassische Anwendungen in Rechnernetzen wie Datei-Übertragung und entfernter Rechnerzugriff. Damit stellen sie veränderte Anforderungen an Infrastruktur und Transportprotokolle.

Herausragendes Merkmal der Datenströme ist, daß sie, in Abhängigkeit vom jeweiligen Codec, eine nahezu konstante Datenrate haben.[1] Bitrate und Paketrate variieren in Abhängigkeit vom Medium (Audio, Video).

An die Infrastruktur wird die Forderung gestellt, die Datenströme mit geringer Verzögerung zu übertragen, die zudem möglichst konstant sein sollte. Nur so lassen sich auf der Ebene der Sprach- und Bild-Kommunikation erträgliche *Round Trip Times* (RTTs)[2] erzielen. Aktuelle Netzwerke sind in vielen Fällen diesen Anforderungen nur bedingt gewachsen. Besonders in Hochlastsituationen entstehen durch überfüllte Warteschlangen in vermittelnden Komponenten große Varianzen der Paketlaufzeit sowie Paketverluste.

Als Transportprotokoll auf Betriebssystem-Ebene kommt im Internet-Kontext in der Regel das *User Datagram Protocol* (UDP) zum Einsatz. Grund hierfür ist seine Eigenschaft, Paketverluste nicht durch wiederholende Übertragung zu kompensieren. Ergänzt wird dieses Protokoll durch ein Transportprotokoll der Anwendungsschicht, das *Real-Time Transport Protocol* (RTP) (vgl. 2.4.2, Seite 37). Seine wesentlichen Aufgaben sind die medienunabhängige Codierung generell erforderlicher Parameter wie Zeitstempel, Sequenz-Nummern, Quellenbezeichner und

[1] Als Beispiel hierfür dienen Audio-Datenströme, die häufig 8-bit μ-law codiert mit einer Abtastrate von 8kHz bei 20ms Sampling-Intervall übertragen werden (vgl. Tabelle 2.7, Seite 40). Ein solcher Datenstrom erzeugt eine Paketrate von 50 Paketen pro Sekunde und, bei Einsatz von RTP als Transportprotokoll, eine Datenrate auf der UDP-Ebene von 68.8 kbps. Die Konstantheit des Datenstroms wird durch das Audio-Device des Senders gewährleistet. Der sendende Prozeß liest die Daten vom Device im blockierenden Modus. Alle 20ms erhält er einen Datensatz.

[2] Unter der Round-Trip-Time wird das Zeitintervall verstanden, das ein Paket zur Übertragung vom Sender zum Empfänger und zurück benötigt.

Payload-Typen, sowie die Bereitstellung quantitativer Parameter des Paketdatenstroms, die eine Überwachung der Datenübertragungsqualität gestatten. Damit wird die medienunabhängige Abschätzung des Playout-Jitters beim Empfänger möglich. Mit dem Begriff Playout-Jitter wird diejenige Zeitspanne bezeichnet, die Pakete beim Empfänger maximal vor dem Ausspielen aufbewahrt werden, um Varianzen der Paketlaufzeiten vom Sender zum Empfänger auszugleichen. Sie bestimmt somit unter Umständen nachhaltig die Verzögerung des Gesamtsystems.[3]

Aus Sicht des Network-Engineerings ist es von besonderem Interesse zu untersuchen, wie sich ein komplexes Rechnernetz bei der Beaufschlagung mit multimedialen Datenströmen zur interaktiven interpersonellen Kommunikation verhält. Hierzu ist es in einem ersten Ansatz möglich, einen durch äußere Parameter vorgegebenen Netzwerkpfad testweise mit den Anforderungen entsprechenden Datenströmen zu belasten, um schließlich eine subjektive Bewertung durchzuführen. Dieser Ansatz wurde beispielsweise in [Fro95] verfolgt. Die Ergebnisse solcher Analysen sind in vielen Fällen für die grundsätzliche Bewertung, ob das Übertragungssystem für die Abhaltung multimedialer Online-Konferenzen geeignet ist, ausreichend.

Für eine tiefergehende Analyse der Eigenschaften des Übertragungssystems ist es erforderlich, möglichst viele Parameter des Datenstroms in feinen Schritten verändern zu können. Diesem Zweck dient das Meßwerkzeug *rtest*.

A.2 Eigenschaften und Einsatzmöglichkeiten von Rtest

rtest gestattet dem Nutzer, einen UDP-Datenstrom zu generieren, der hinsichtlich der emittierten Datenrate und der Paketgröße regulierbar ist. Die Payload besteht, neben einer Sequenznummer und einem Zeitstempel des Senders, aus zufälligen Daten. Die Ziel-Transportadresse sowie der Sende-Port sind durch den Nutzer einstellbar. Neben Unicast-Transportadressen können auch Multicast-Transportadressen gewählt werden.

Strenge Gleichmäßigkeit und Periodizität des emittierten Datenstroms (vgl. [Ste93, Seite 23f.]) hatten einen besonderen Stellenwert bei der Entwicklung von *rtest*. Diese Eigenschaften wurden realisiert, indem stets Pakete gleicher Größe während einer Messung generiert werden. Ergänzend ist es erforderlich, daß die Zeitintervalle zwischen der Emission von zwei Paketen konstant sind. Die Erfüllung dieser Forderung erwies sich bei der Entwicklung der Software als schwierig: Das übliche Verfahren der Realisierung eines Dispatchers unter Nutzung der *select*-Routine der Berkeley-Socket-Schnittstelle [Ste92, Seite 396] gerät dabei hinsichtlich der Timeout-Auflösung an seine Grenzen.[4]

[3] Die Varianzen der Paketlaufzeiten entstehen durch Paketstauungen in den Ausgabe-Warteschlangen der vermittelnden Komponenten entlang des Übertragungsweges (vgl. 3.2.2.1, Seite 73f.).

[4] Die *select*-Routine erhält beim Aufruf drei Datei-Deskriptoren-Mengen übergeben. Eine Menge enthält Datei-Deskriptoren bei denen auf Eingaben gewartet wird. Die zweite Menge enthält Deskriptoren, bei denen darauf gewartet wird, daß weitere Daten auf die zugeordnete Datei oder den Netzwerk-Socket geschrieben werden können. Die dritte Menge umfaßt Deskriptoren, bei denen auf Fehlersituationen in den zugeordneten Dateien und Netzwerkverbindungen zu achten ist. Von besonderem Interesse ist hier jedoch der letzte Parameter der *select*-Routine. Er spezifiziert einen Timeout-Wert, nach dem die *select*-Routine auch dann zurückkehren soll, wenn weder Daten zum Lesen oder zum Schreiben noch Fehler anliegen. Stevens schreibt dazu, daß die *select*-Routine dazu geeignet ist, Wartezeiten von weniger als einer Sekunde zu realisieren [Ste92, Seite 398].

Wenngleich der Timeout-Wert der *select*-Routine die Angabe von Wartezeiten in der Auflösung von Mikrosekunden gestattet, unterstützen aktuelle UNIX-Betriebssysteme diese Auflösung nicht. Messungen zeigen, daß die erzielbare Auflösung im Bereich von $10ms$ liegt. Mit dieser Auflösung läßt sich kein streng periodischer und gleichmäßiger Datenstrom erreichen. Alternativ kann ein Timer-Signal-Handler installiert werden und das Timer-Signal entsprechend gesetzt werden. Dies führt zu vergleichbaren Ergebnissen. Ein Audio-Device kann zur Synchronisierung nicht genutzt werden, da *rtest* auch dann einsetzbar sein soll, wenn ein Nutzer keinen Zugriff auf das Audio-Device hat. Zudem enthalten nicht alle Rechner Audio-Devices.

Daher wurde bei der Implementierung von *rtest* darauf verzichtet, einen Timeout unter Nutzung einer Betriebssystem-Funktion zu realisieren. Stattdessen werden in einer Warteschleife abwechselnd die *select*-Routine mit einer Wartezeit von Null sowie die Betriebssystem-Funktion *gettimeofday* [Ste92, Seite 155] aufgerufen. Mit *select* wird geprüft, ob Daten zum Lesen anliegen. Mit *gettimeofday* wird untersucht, ob der Zeitpunkt zum Versenden des nächsten Pakets erreicht ist. In diesem Fall wird das entsprechende Paket generiert und übertragen.

Mit diesem Verfahren ist es gelungen, eine den Qualitätsanforderungen entsprechende Quelle für gleichmäßige, periodische Datenströme zu realisieren. Nachteilig ist, daß der *rtest*-Prozeß eine kontinuierliche Last auf dem sendenden Rechner erzeugt. Auf einem anderweitig belasteten Rechnersystem kann es daher bei der Emission des Datenstroms zu Ungenauigkeiten in der Periodizität kommt. *rtest* gleicht diese Ungenauigkeiten aus, indem nach einer Periode mit zu geringer Paketrate die Emissionsrate erhöht wird, bis der Soll-Wert wieder erreicht ist.

Ein wichtiges Ziel bei der Entwicklung von *rtest* war, ein Werkzeug zu schaffen, welches die Ermittlung von Round-Trip-Times unter typischer Belastung des Netzes mit einem Nutzdatenstrom erlaubt. Im nächsten Abschnitt wird detaillierter auf die Durchführung solcher Messungen eingegangen. Für die Architektur von *rtest* ergibt sich daraus die Anforderung, nicht nur Datenströme senden zu können sondern auch entsprechende Echo-Antworten auf dem Sende-Port empfangen zu können. Dazu dient zum einen die bereits beschriebene Funktion zum Empfang von Paketen auf dem Sende-Port, zum anderen werden Antwortpakete beim Eintreffen mit Zeitstempeln versehen. Die Empfangszeitstempel werden nach Abschluß der Messung bei der durchzuführenden Aufbereitung der Testergebnisse ausgewertet.

Die bei einer Messung entstehenden Daten können, je nach gewählten Parametern, nennenswerten Umfang erreichen. Daher wurde ein Modul zur Auswertung der Meßergebnisse in das Programm integriert. Dabei werden sowohl der Sende- als auch der Empfangsdatenstrom untersucht. Die Analyse des Sendedatenstroms gestattet es zu prüfen, ob dieser hinsichtlich seiner Gleichmäßigkeit und Periodizität den gestellten Anforderungen genügt. Die Daten über den empfangenen Datenstrom dienen der Analyse der Eigenschaften des Netzwerkpfades zwischen Sender und Empfänger. Gesteuert durch Optionen lassen sich neben Ergebnissen der statistischen Berechnungen auch detaillierte Werte zu den Laufzeiten einzelner Pakete ausgeben. Die durch *rtest* generierten Ein- und Ausgabewerte sowie die Interpretation dieser Werte wird in den folgenden beiden Abschnitten behandelt.

Für die Messung von Round-Trip-Times kann auf der Seite des Empfängers in vielen Fällen auf Standard-Funktionen in heutigen UNIX-Betriebssystemen zurückgegriffen werden. Hervorzuheben sind der UDP-Echo-Port und der UDP-Discard-Port, die beide vom Internet-Daemon verwaltet werden.

Darüber hinaus gibt es Situationen, in denen eigenständige Programme für den Empfang und für

das Zurücksenden des Datenstroms an den originären Sender erforderlich sind. Dafür enthält das *rtest*-Paket neben dem eigentlichen Programm *rtest* noch die beiden Programme *udp-discard* und *udp-echo*. Ihre Aufgaben sind der Empfang und das Verwerfen von UDP-Datagrammen (*udp-discard*) sowie der Empfang und die unmittelbare Rückübermittlung der empfangenen Daten an den Sender per Unicast oder Multicast (*udp-echo*). Durch entsprechende Optionen können diese Programme für den jeweiligen Einsatzzweck konfiguriert werden und die Ausgabe von statistischen Daten zum empfangenen Datenstrom kontrolliert werden. Auf den Einsatz dieser Programme zur Durchführung von Messungen wird weiter unten eingegangen.

Das Programm *rtest* und die begleitendenen Programme *udp-discard* und *udp-echo* unterstützen jeweils zwei unterschiedliche Formen der Ausgabe von Ergebnissen. Im Verbose-Mode erhält der Nutzer Ausgaben, wie sie von einem interaktiven Programm für den spontanen Einsatz erwartet werden. Jede Zeile enthält einen Meßwert, dessen Interpretation durch begleitenden Text erleichtert wird. Im Batch-Mode faßt das Programm alle ermittelten Meßwerte in einem festen Format in einer Zeile zusammen. Dieser Modus ist besonders für die Durchführung umfangreicherer Meßreihen geeignet. Die Ergebnisse einer jeden Messung können leicht aufsetzenden Werkzeugen zur Visualisierung und Dokumentation zugeführt werden (*gnuplot* und *Perl*-Skripts).

Ein wichtiges Merkmal von *rtest* ist die Möglichkeit zur feinen Regelung der Datenrate des emittierten Datenstroms. Diese ist besonders hinsichtlich der Durchführung von Messungen wichtig, die der Analyse der Eigenschaften von Queues in Access-Routern dienen. Das dabei relevante Merkmal des Access-Routers ist, daß es sich um ein vermittelndes Element an der Grenze zwischen Netzwerk-Teilen mit stark unterschiedlichen Datenraten handelt.

rtest verfügt über keine integrierte Begrenzung der einstellbaren Datenrate. Dies birgt die Gefahr, daß Anwender das Werkzeug nutzen, um Datenströme hoher Datenrate zu generieren. An dieser Stelle sei vor der Generierung hoher UDP-Datenraten eindringlich gewarnt. Es besteht die unmittelbare Gefahr, daß Teile des Netzwerkpfades zwischen Sender und Empfänger sowie aktive Vermittlungskomponenten überflutet werden und die Netzwerkverbindungen anderer Nutzer durch derartige Experimente in Mitleidenschaft gezogen werden. Bis auf wenige Ausnahmen ist es nicht erforderlich oder gar hinderlich, Messungen mit Datenraten von mehr als wenigen kbps durchzuführen, insbesondere in Netzwerken, die weitgehend auf LAN-Technologie aufgebaut sind.

A.2.1 Rtest-Ausgaben bezüglich des gesendeten Datenstroms

Wie bereits dargestellt, werden an die Periodizität und Gleichmäßigkeit des von *rtest* ausgesendeten Datenstroms gegenüber üblichen Internet-Werkzeugen besondere Anforderungen gestellt. Zur Beurteilung der Qualität des emittierten Datenstroms ist es erforderlich, statistische Daten über Abweichungen des generierten Datenstroms vom erwarteten Datenstrom zu ermitteln. Dazu stellt *rtest* nach Abschluß einer Messung die folgenden Werte bereit:

- Anzahl der gesendeten Pakete.
- Größe der gesendeten Pakete.
- Gefordertes Zeitintervall zwischen dem Aussenden von zwei Paketen (in Mikrosekunden).

- Durchschnittlich erreichtes Zeitintervall zwischen dem Aussenden von zwei Paketen (in Mikrosekunden).

- Abweichung der tatsächlichen Sendezeitpunkte von den erwarteten Sendezeitpunkten, basierend auf dem geforderten Zeitintervall zwischen dem Aussenden von zwei Paketen. Aus den während der Messung entstandenen Daten werden durch statistische Berechnungen folgende Parameter ermittelt:

  - Minimal-Wert der Abweichung,
  - Maximal-Wert der Abweichung,
  - Durchschnitt (Streuung, Quadratwurzel der Varianz),
  - Median der Abweichung,
  - Standardabweichung bezüglich des Durchschnitts,
  - Standardabweichung bezüglich des Medians.

  (Alle Angaben in Mikrosekunden).

- Geforderte Sendedatenrate (in kbps) bezogen auf die UDP-Payload.

- Erreichte Sendedatenrate (in kbps) bezogen auf die UDP-Payload.

- Abweichung der erreichten Sendedatenrate von der geforderten Sendedatenrate (in kbps).

Einige der Parameter dienen vor allem der Dokumentation der Messung. Die Qualität des emittierten Datenstroms läßt sich im ersten Schritt an der Abweichung der erreichten Sendedatenrate von der geforderten Sendedatenrate erkennen. Weicht dieser Wert um mehrere kbps von Null ab, entspricht der emittierte Datenstrom nicht den Anforderungen. Ursache hierfür ist in der Regel eine Überlastung des sendenden Rechners. Die Ursachen hierfür können transient oder dauerhaft sein.

Genauere Aussagen über die Qualität des emittierten Datenstroms erhält der Nutzer aus den Berechnungen zur Abweichung der tatsächlichen Sendezeitpunkte von den erwarteten Sendezeitpunkten. Hier ist vor allem der Median sowie die Standardabweichung des Medians von Interesse. Erfahrungen mit dem Einsatz von *rtest* zeigen, daß der Median ein robusterer Schätzwert für den Erwartungswert ist als der Mittelwert der Abweichung. Grund hierfür ist, daß etwaige Verzögerungen bei der Emission von Paketen sich üblicherweise im Bereich von mehreren Millisekunden bewegen. Zum einen geht dieser relativ hohe Wert in die Mittelwert-Bildung ein, zum anderen wird durch die im folgenden beschleunigt gesendeten Datenpakete der Mittelwert weiter erhöht, da die Abweichungen als Betrag in die statistischen Berechnungen eingehen.

Ein Median der Abweichung in der Größenordnung von Millisekunden sollte in jedem Fall eine Betrachtung der Abweichungen der Sendezeitpunkte im Detail nach sich ziehen. Über eine Option kann bei *rtest* die Ausgabe dieser Werte veranlaßt werden.

A.2.2 Rtest-Ausgaben bezüglich des empfangenen Datenstroms

Alle Ausgaben des Programms basieren auf dem während der Messung emittierten Datenstrom. Auf der Basis von Sequenz-Nummern können die empfangenen Pakete den zuvor gesendeten Paketen gegenüber gestellt werden. Im einzelnen werden folgende Werte berechnet und ausgegeben:

- Anzahl der empfangenen Pakete.
- Anzahl der auf dem Übertragungsweg zerstörten Pakete.
- Paketverlust in Prozent.
- Anzahl und Prozentsatz der duplizierten Pakete.
- Anzahl der Pakete, die nicht in Sendereihenfolge empfangen wurden.
- Durchschnittliches Zeitintervall zwischen zwei aufeinanderfolgend empfangener Pakete in Mikrosekunden (Durchschnittliches Zwischenankunftszeitintervall).
- Abweichung der Paketlaufzeit zweier aufeinander eingetroffener Pakete. Durch statistische Berechnungen werden daraus die folgenden Werte berechnet:
  - Minimal-Wert der Abweichung,
  - Maximal-Wert der Abweichung,
  - Durchschnitt (Streuung, Quadratwurzel der Varianz),
  - Median der Abweichung,
  - Standardabweichung bezüglich des Durchschnitts,
  - Standardabweichung bezüglich des Medians.

  (Alle Angaben in Mikrosekunden).
- Erzielte Empfangsdatenrate (in kbps) bezogen auf die UDP-Payload.
- Minimale Round-Trip-Time (in Millisekunden).
- Maximale Round-Trip-Time (in Millisekunden).
- Durchschnittliche Round-Trip-Time (in Millisekunden).
- Median der Round-Trip-Time (in Millisekunden).
- Standardabweichung der Round-Trip-Time bezogen auf den Durchschnitt.

Die wesentlichen Größen für die Beurteilung des Netzwerkpfades sind die Round-Trip-Time und die Abweichung der Paketlaufzeit zwischen dem Empfang von zwei aufeinander eingetroffener Pakete.

Besondere Beachtung zur Abschätzung der Paketlaufzeit erfordert der Minimalwert der Round-Trip-Time. Wenn der Nutzer die Messung mit einer hinreichend großen Anzahl von Stichproben und geringer Sende-Datenrate durchgeführt hat, kann aus diesem Wert auf die minimale Paketlaufzeit über den Netzwerkpfad geschlossen werden. Dabei wird unterstellt, daß das Paket mit der kleinsten RTT alle vermittelnden Komponenten entlang des Netzwerkpfades durchlaufen hat, ohne in einer Wartschlange zwischengespeichert worden zu sein (vgl. Abschnitt 2.3.1, Seite 20f. und Abschnitt 3.2.2.1, Seite 73f.). Die minimale Paketlaufzeit in eine Richtung wird unter der Voraussetzung symmetrischer Netzwerkpfade für den Datenstrom vom Sender zum Empfänger und zurück in der Regel die Hälfte der minimalen RTT betragen.

Einen robusten Schätzwert für den Erwartungswert der RTT liefert der Median der RTT. Er ist frei von einzelnen Ausreißern der RTT, wie sie durch Hostname-Lookup und Path-MTU-Discovery bei der Übertragung des ersten Pakets entstehen. In einem Netzwerk mit MAN-Charakteristik, welches nicht an der Grenze seiner Übertragungskapazität betrieben wird, sollte der Median der RTT die gleiche Größenordnung wie der Minimalwert der RTT haben. Ist dies nicht der Fall, sollte der Netzwerkpfad einer weitergehenden Analyse unterzogen werden, denn es steht zu erwarten, daß mehr als die Hälfte aller emittierten PDUs auf dem Round-Trip durch das Netz in einer oder mehreren Queues vermittelnder Komponenten zwischengespeichert werden.

In vielen Fällen, besonders bei der Beurteilung von Netzwerkpfaden hinsichtlich ihrer Belastung durch konkurrierende Datenströme, ist es von Interesse abzuschätzen, wie häufig Pakete in den Warteschlangen vermittelnder Komponenten für welchen Zeitraum zwischengespeichert werden. Ein Indikator hierfür ist die von *rtest* ausgewiesene Abweichung der Paketlaufzeit zweier aufeinander eintreffender Pakete. Liegen bei hinreichend großer Stichprobenanzahl der Minimalwert, der Median sowie der Mittelwert eng beieinander und haben diese einen Wert in der Größenordnung von wenigen Millisekunden, so kann davon ausgegangen werden, daß der untersuchte Netzwerkpfad nicht überlastet ist. Wenn Minimalwert und Median eng zusammen liegen, der Mittelwert jedoch deutlich abweicht, ist davon auszugehen, daß gelegentliche Stauungen in den vermittelnden Komponenten des Netzes auftreten, eine weitergehende Analyse des verwendeten Pfades ist anzuraten. Wenn auch der Median signifikant vom Minimum abweicht, sollte der Netzwerkpfad unbedingt weiter untersucht werden.

Sichere Indikatoren von Netzwerkproblemen im MAN-Bereich sind Paketverluste, duplizierte Pakete sowie Pakete, die nicht in Sendereihenfolge empfangen werden. Paketverluste lassen darauf schließen, daß der untersuchte Netzwerkpfad oder das Partnersystem überlastet sind. Letzteres Problem ist durch kleine Sendedatenraten im Bereich weniger kbps praktisch auszuschließen. Zudem können Fehler an aktiven Netzwerkkomponenten wie Router- und Switch-Interfaces, Repeatern und Transceivern ihre Ursache haben. Gleiches gilt für duplizierte Pakete und Pakete, die nicht in Sendereihenfolge empfangen werden.[5]

Ein wichtiges Anwendungsfeld von *rtest* ist die Emulation von Datenströmen, wie sie bei der interaktiven interpersonellen Kommunikation in Online-Konferenzen entstehen. Der grundlegende Ansatz ist dabei die Aufnahme der Parameter des Datenstroms in einer entsprechenden

[5]Bei Messungen in Netzwerken, die auf Basis der LAN-Emulation über ATM betrieben wurden, konnten mit *rtest* regelmäßig duplizierte Pakete nachgewiesen werden. Nachträglich zeigte sich, daß dieser Fall immer dann auftrat, wenn ein *Switched Virtual Circuit* (SVC) zum Zielsystem aufgebaut werden mußte. Während des Übergangs des Kommunikationspfades vom Broadcast-and-Unknown-Server zum Zielsystem wurden Pakete dupliziert.

Online-Konferenz, möglichst in einer Labor-Umgebung. Relevante Daten sind die Datenrate und die Paketgröße. Die ermittelten Werte werden zur Parameterisierung von *rtest* benutzt. Im Produktionsnetz werden mit *rtest* Datenströme gleicher Charakteristik generiert. Die Ergebnisse der Messungen lassen Schlüsse über die zu erwartende Qualität der interaktiven interpersonellen Kommunikation in der Produktionsumgebung zu, basierend auf objektiven Meßergebnissen.

Zwei wichtige Größen bei derartigen Messungen sind die minimale RTT sowie der Mittelwert der Abweichung der Paketlaufzeit zweier aufeinander eintreffender Pakete. Der Minimalwert der Paketlaufzeit bestimmt die minimale Verzögerung der Sprachkommunikation im Netz. Der Mittelwert der Abweichung der Paketlaufzeit hat üblicherweise die gleiche Größenordnung wie der Erwartungswert des Playout-Jitters, zumindest bei Protokollen, die eine Playout-Jitter-Schätzung ermöglichen. Somit berechnet sich die durch das Netzwerk verursachte Verzögerung bei der Sprachkommunikation aus der Summe der minimalen Paketlaufzeit und dem Mittelwert der Paketlaufzeitabweichung.

A.2.3 Rtest im Vergleich zu anderen Netzwerk-Analyse-Werkzeugen

Nachdem in den vorangegangenen Abschnitten die Eigenschaften von *rtest* dargestellt sowie die Ausgabe-Werte vorgestellt und hinsichtlich ihrer Bedeutung für die Analyse von Netzwerken diskutiert wurden, wird in diesem Abschnitt *rtest* mit anderen Analyse-Werkzeugen verglichen. Dazu wird die Funktionalität der Werkzeuge vorgestellt und die Unterschiede zu *rtest* dargestellt.

ping

ping ist ein Programm, das auf fast jedem Rechner mit TCP/IP-Protokoll-Stack zu finden ist. Sein Name leitet sich von der sonaren Operation zur Lokalisierung von Objekten ab.

ping sendet ICMP-Echo-Requests an ein Zielsystem, das mit ICMP-Echo-Reply Nachrichten darauf antwortet. Die Antwort-Nachricht hat die gleiche Größe wie die Anfrage und die Payload entspricht der des Requests. Die Antwort wird in der Regel vom Betriebssystem-Kern des Zielsystems generiert. Anstelle einer Unicast-Zieladresse können auch eine Broadcast-Adresse oder ausgezeichnete Multicast-Gruppenadressen[6] angegeben werden. In diesem Fall antworten alle angesprochenen Systeme.

Die ausgesendeten ICMP-Echo-Requests enthalten eine Identifizierung des sendenden Prozesses (Prozeßnummer), eine Sequenz-Nummer sowie einen Zeitstempel. Dies ermöglicht beim Empfang der Antwort die Berechnung der Round-Trip-Time sowie die Erkennung von Paketverlusten.

ping wird für folgende Zwecke eingesetzt:

- Überprüfung, ob ein Ziel-Host auf IP-Ebene ansprechbar ist.
- Ermittlung der Round-Trip-Time zwischen zwei Systemen.

[6]Wird als Zieladresse die Multicast-Adresse 224.0.0.1 angegeben, antworten darauf alle Systeme im lokalen Subnetz. Auf eine Anfrage mit der Multicast-Adresse 224.0.0.2 antworten alle Router im lokalen Subnetz [Int97b].

Eine weitergehende Diskussion der Eigenschaften des *ping*-Programms ist [Ste94, Abschnitt 7] zu entnehmen.

Ein häufig vorgebrachtes Argument gegen die Ermittlung von Round-Trip-Times mit *ping* ist, daß die ICMP-Pakete in den Routern entlang des Pfades bevorzugt behandelt werden. Diese Aussage ist gemäß [Bak95] und [Ste94, Seite 35] falsch. ICMP-Echo-Nachrichten werden mit gleicher Priorität wie alle anderen IP-Pakete behandelt. Dennoch ergeben sich gelegentlich geringere RTTs, da ICMP-Echo-Requests vom Betriebssystem-Kern des Partnersystems beantwortet und nicht erst an einen Prozeß im User-Mode weitergeleitet werden müssen.

rtest ersetzt das *ping*-Programm nicht, sondern ergänzt es bei der Ermittlung von Round-Trip-Times. Hervorzuheben sind dabei die folgenden Eigenschaften:

- Messung der Round-Trip-Times auf der Ebene von Prozessen im User-Mode. Dies entspricht dem realen Einsatz von Netzwerk-Anwendungen.
- Generierung von Datenströmen gemäß der Charakteristik multimedialer Echtzeitdatenströme. *ping* erlaubt das Netz mit Requests zu fluten oder ein Zeitintervall zwischen dem Aussenden von zwei Datagrammen anzugeben, wobei der Minimalwert bei einer Sekunde liegt. Keine der Optionen ermöglicht die Modellierung typischer Echtzeitdatenströme.
- Flexible Spezifikation des Sende-Ports und der Ziel-Transportadresse. Diese Eigenschaft ist besonders bei der Untersuchung von Multicast-Routern und Gateways auf Anwendungsebene, wie MAGW, zwingend erforderlich.
- Ermittlung weiterer Daten bezüglich des Sende- und Empfangsdatenstroms.

traceroute

traceroute dient der Identifikation der Router entlang eines Netzwerkpfades. Es wurde 1988 von Van Jacobson entwickelt. Heute enthalten nahezu alle Betriebssystem mit TCP/IP-Stack eine *traceroute*-Implementierung. *traceroute* arbeitet nach dem folgenden Konzept:

- IP-Pakete enthalten ein Time-to-Live-Feld (TTL-Feld), daß durch den Sender einer Nachricht initialisiert wird und von den Routern entlang des Netzwerkpfades jeweils um den Wert eins verringert wird. Erreicht das TTL-Feld den Wert 0, wird das Datagramm vom Router verworfen und eine ICMP-Time-Exceeded-Nachricht an den Sender übermittelt. Die Quelladresse der ICMP-Nachricht identifiziert den Router, der das Paket verworfen hat.
- Werden Datenpakete mit der engültigen Zieladresse und von 1 wachsenden TTL-Werten versendet, identifizieren sich die Router entlang des Netzwerkpfades.

Eine weitergehende Beschreibung von *traceroute* ist [Ste94, Abschnitt 8] zu entnehmen.

rtest ist in einigen Fällen auf die Ergebnisse von *traceroute* angewiesen. Dies gilt immer dann, wenn *rtest*-Messungen von Hop zu Hop entlang eines Netzwerkpfades durchgeführt werden, um Netzwerkprobleme auf einzelne Teilnetze oder vermittelnde Komponenten einzukreisen.

tcpdump

Bei *tcpdump* handelt sich es um ein Programm zur Analyse des Netzwerk-Verkehrs. Im normalen Anwendungsfall schaltet *tcpdump* ein spezifiziertes Netzwerk-Interface in den Promiscuous-Mode und protokolliert die auf dem Netzwerk-Abschnitt sichtbaren Pakete. Die relevanten Daten der Paket-Header werden interpretiert und ausgegeben. Die Interpretation der Header kann, gesteuert durch den Nutzer, auf unterschiedlichen Schichten (von der Sicherungsschicht bis zur Anwendungsschicht) erfolgen. Ein Filter-Modul erlaubt die Begrenzung der Ausgaben auf zu spezifizierende Datenströme.

tcpdump kann zum einen interaktiv eingesetzt werden, um den aktuellen Netzwerkverkehr darzustellen. Zum anderen kann der Paketstrom zusammen mit Eingangszeitstempeln der Pakete in einer Datei gesichert werden, um später analysiert zu werden. Es handelt sich um ein universelles Werkzeug zur Analyse von Datenströmen, das seine Nützlichkeit in vielen Anwendungsfällen unter Beweis gestellt hat.

Beim passiven Verfolgen von MBone-Sitzungen stellt sich das Problem, daß ein Datenstrom erst dann an das jeweilige Endsystem geleitet wird, wenn die betreffende Multicast-Gruppe explizit angefordert wird. Entsprechende Meldungen werden von *tcpdump* nicht erzeugt. Für diesen Einsatzzweck ist daher ein nebenläufiges Werkzeug erforderlich, daß die betreffenden Multicast-Gruppen anfordert, ohne Meldungen des jeweiligen Transportprotokolls zu generieren. Diesem Zweck dient das zu *rtest* gehörende Werkzeug *udp-discard*. Die erste Version dieses Programms öffnete die erforderlichen UDP-Sockets und verwarf die empfangenen Pakete. Die Analyse des Verkehrsstroms wurde realisiert, indem parallel der Paketstrom mit *tcpdump* protokolliert und später analysiert wurde.

Für die längerfristige Protokollierung von MBone-Konferenzen zeigte diese Methoden aufgrund der Anwendungskomplexität Schwächen. Zudem sind zum Ausführen des Programms *tcpdump* Systemverwalter-Rechte erforderlich. So wurde die erforderliche Funktionalität zur Protokollierung der empfangenen Datenströme in das Programm *udp-discard* integriert.

Mit *udp-discard* steht ein einfach anzuwendendes Werkzeug zur Protokollierung der Datenströme in MBone-Konferenzen und zum Empfang und der Auswertung von *rtest* generierten Datenströmen bereit, welches den Einsatz von *tcpdump* häufig erübrigt.

rtest sowie die es begleitenden Programme *udp-discard* und *udp-echo* erweitern das Spektrum der vorhandenen Analysewerkzeuge im Internet zur detaillierten Untersuchung von Netzwerkpfaden. *rtest* vereinigt die Eigenschaften von Tools zur Messung von Round-Trip-Delays, indem Datenströme mit der Charakteristik von Echtzeitdatenströmen in Online-Konferenzen generiert werden und zusätzliche Parameter der Messung ermittelt werden. Die flexiblen Einstellmöglichkeiten bezüglich Sende-Port und Ziel-Transportadresse lassen den Einsatz in Szenarien zu, die existierenden Werkzeugen verschlossen bleiben. *udp-discard* und *udp-echo* vereinfachen die Durchführung von Messungen, indem die Merkmale der empfangenen Datenströme für die nachfolgende Analyse aufbereitet werden. Die flexible Einstellung von Empfangs-Ports und -Adressen ermöglicht die Messung von Round-Trip-Times und die Ermittlung des One-Way-Jitter auch bei der Verwendung von Gateways auf Anwendungsebene.

Nachdem in diesem Abschnitt detailliert auf die Eigenschaften von *rtest* eingegangen wurde, zeigt der folgende Abschnitt den Einsatz des Werkzeugs in verschiedenen Anwendungsszenari-

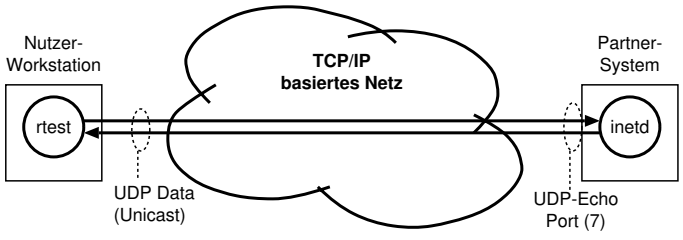

Abbildung A.1: Nutzung von *rtest* zur Ermittlung von Round-Trip-Times unter Nutzung von Unicast-UDP

en.

A.3 Einsatzszenarien

Wenngleich *rtest* als Testwerkzeug für die Entwicklung des MBone-Access-Gateways (MAGW) entwickelt wurde, hat sich im Zeitverlauf gezeigt, daß es auch in anderen Projekten nutzbringend eingesetzt werden konnte. Dazu zählten die Etablierung einer flächendeckenden Multicast-Infrastruktur an der Universität Hannover, die Analyse von generellen Netzwerkproblemen, die Ermittlung von Round-Trip-Delays im Campus-Netz der Universität Hannover sowie die Emulation von Verkehrslasten im Rahmen der Erprobung von Telefondiensten unter Nutzung des Internet-Protokolls.

Viele der im Rahmen dieser Entwicklungen durchgeführten Messungen weisen ähnliche Muster auf, die im folgenden beschrieben werden. Dies soll den Einsatz der Werkzeuge illustrieren und anderen Anwendern die Nutzung der Programme in vergleichbaren Situationen erleichtern.

A.3.1 Ermittlung von Unicast RTTs

Die Ermittlung von Round-Trip-Times (RTTs) unter Nutzung von Unicast-UDP und der Beaufschlagung der Übertragungsstrecke mit einer festen Datenrate war ein initiales Ziel der Entwicklung von *rtest*. Die Durchführung solcher Messungen ist in der Regel ohne die Installation eines Partner-Prozesses auf dem Zielsystem möglich. Abbildung A.1 illustriert daß Meßprinzip.

Der emittierte Datenstrom wird über vermittelnde Komponenten auf den UDP-Echo-Port eines Partner-Rechners oder Routers geleitet. Dieser Port wird vom Internet-Daemon, inetd, verwaltet. Eintreffende Datagramme werden ohne Veränderung des Inhalts unmittelbar an den Sender zurück gesendet.[7]

Die empfangenen Pakete werden beim Eintreffen mit einem Zeitstempel versehen. Nach Beendigung der Messung werden die beim Absenden der Pakete festgehaltenen Zeitstempel mit denen

[7]Hierzu ist einschränkend anzumerken, daß einige Betriebssysteme die Echo-Funktion so implementieren, daß Datagramme nur bis zu einer bestimmten Größe vollständig an den Sender zurückübertragen werden. So senden Sun-Systeme unter Solaris 2.5.1 nur bis zu 1024 Byte Payload zurück. Cisco´s IOS, IRIX 6.x sowie Linux beschränken die Größe der Payload nicht.

beim Eintreffen der Antwort genommen Zeitstempel in Relation gesetzt. Die Erkennung von Paketverlusten, duplizierten Paketen sowie Paketen, die nicht in Sendereihenfolge empfangen wurden, wird durch Sequenz-Nummern in den Paketen möglich.

Die Parameter für die Messung sollten so gewählt werden, daß das Netzwerk durch die Messung nicht geflutet wird, d.h. es sollte eine Datenrate von 32 kbps oder weniger eingestellt werden. Die Meßdauer sollte sich maximal im Bereich weniger Minuten bewegen. Alternativ kann die Anzahl der ausgesendeten Pakete eingestellt werden.

Die Meßergebnisse werden erst nach Abschluß der Messung ausgegeben. Sie bestehen aus den zuvor dargestellten Werten. Unmittelbar nach der Beendigung der Messung wartet *rtest* noch einige Sekunden auf eventuell verspätet eintreffende Pakete. Dieses Intervall ist auf 10 Sekunden eingestellt und sollte bei der Planung umfangreicher Messungen berücksichtigt werden.

A.3.2 Ermittlung von RTTs über IP-Multicast

Die Ermittlung von Round-Trip-Times über IP-Multicast ist dann von Interesse, wenn untersucht werden soll, ob Pakete, die unter Nutzung von IP-Multicast übertragen werden, eine größere Verzögerung auf dem Weg durch das Netz erfahren, als unter Nutzung von Unicast. Hintergrund dieser Untersuchungen ist, daß die Netzwerkpfade für Multicast-Datenströme häufig anders verlaufen als die für Unicast-Pakete. Weitere wichtige Anwendungen dieser Messung sind die Untersuchung der Eigenschaften von Multicast-Routern und Gateways auf Anwendungsebene, wie MAGW.

Die Ermittlung von RTTs über Multicast erfordert einen veränderten Versuchsaufbau, da hier nicht der UDP-Echo-Port des Zielrechners angesprochen werden kann. Seine Funktion wird von dem im *rtest*-Paket enthaltenen Programm *udp-echo* übernommen. Für diese Messung wird *udp-echo* so konfiguriert, daß es auf einer nicht anderweitig genutzten Multicast-Adresse auf Datagramme vom Sender lauscht und die Antworten über die gleiche Multicast-Adresse an den Sender zurücksendet. Dabei können die Kennwerte des empfangenen Datenstroms optional ausgegeben werden. Bei der Durchführung der Messungen ist darauf zu achten, daß die TTL-Werte für die Multicast-Übertragung so klein wie möglich gewählt werden, um andere Nutzer nicht mit einem für sie unnützen Datenstrom zu belästigen.

A.3.3 Ermittlung von RTTs über Application-Layer-Gateways

Bei Application-Layer-Gateways im Kontext multimedialer Online-Konferenzen wird häufig der Unicast als Spezialfall des Multicast angesehen. In beiden Fällen erfolgt die Kommunikation in der Weise, daß die Empfangs-Ports abschnittsweise gleich sein müssen. Abbildung A.2 illustriert diesen Sachverhalt.

In diesem Fall ist es nicht möglich, den UDP-Echo-Port des Partnersystems anzusprechen, da die Antworten an den UDP-Echo-Port des Gateways gesendet würden. Daher ist auch im Fall von Unicast-RTT-Messungen das *rtest* begleitende Programm *udp-echo* beim Partnersystem einzusetzen.

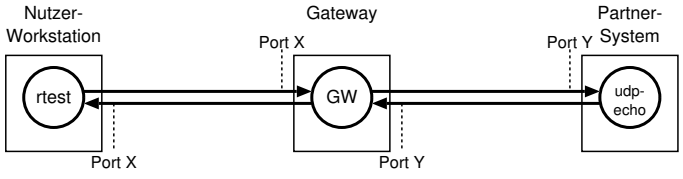

Abbildung A.2: Ermittlung von RTTs in Verbindung mit Application-Layer-Gateways

A.3.4 Analyse und Emulation von MBone-Echtzeitdatenströmen

Bei der Entwicklung von Vermittlungskomponenten für multimediale Online-Konferenzen auf der Infrastruktur des MBone stellt sich häufig die Frage nach dem Verkehrscharakter realer MBone-Konferenzen. Jede Konferenz hat ihre eigenen Parameter hinsichtlich der Zusammensetzung der Verkehrsströme, dem Datenvolumen der einzelnen Verkehrsströme sowie der Anzahl der Teilnehmer. Für den systematischen Test von in Entwicklung befindlichen Komponenten ist es daher erforderlich

1. reale Datenströme von Konferenzen auf dem MBone zu analysieren und
2. Werkzeuge bereitszustellen, die die Emulation der entsprechenden Verkehrsströme ermöglichen.

Zur Lösung dieser Aufgabenstellung bieten sich zwei Wege an:

- Quantitative Analyse der Datenströme, wobei die empfangenen Datenströme protokolliert und anschließend analysiert werden oder die Analyse bereits während des Empfangs erfolgt.

  Für den Test der in Entwicklung befindlichen Werkzeuge wird später ein Datenstrom-Generator als Datenquelle genutzt, der mit dem realen Szenario vergleichbare Datenströme generiert.

- Alternativ kann der reale Datenstrom zur Laufzeit der Konferenz vollständig aufgezeichnet werden und später, zum Test, wieder abgespielt werden.

Während die zweite Alternative den Vorteil bietet, den Test mit realen Audio- und Video-Datenströmen durchführen zu können, hat sie den Nachteil, daß ohne die Durchführung weitergehender Analysen keine quantitativen Daten über den Charakter der Verkehrsströme breitstehen und die für *Was-wäre-wenn*-Tests erforderliche Modifikation von Parametern nicht möglich ist. Dennoch ist die Durchführung solcher Tests sinnvoll und üblich. Zwei Werkzeuge stehen zur Durchführung entsprechender Tests zur Verfügung. Dies sind zum einen die von Henning Schulzrinne und anderen entwickelten *rtptools* [SSC]. Zum anderen ist der Einsatz des von Wieland Holfelder entwickelten *MBone-VCR* [Hol95] möglich.

Für das Testen von Vermittlungssystemen für *Low-Speed Serial Links* ist es wichtig, die Parameter der Datenströme modifizieren zu können. Deshalb steht bei der Entwicklung des MBone-Access-Gateways der erste Lösungsweg im Vordergrund. Dabei empfiehlt sich folgende Vorgehensweise:

- Analyse realer MBone-Datenströme mittels *udp-discard*, ggf. unter Mitnutzung von *tcp-dump*.
- Generierung vergleichbarer Datenströme mit *rtest* in einer Testumgebung.
- Analyse der beim Empfänger angelangten Datenströme mittels *udp-discard*.

Mit diesem Verfahren lassen sich in kurzer Zeit umfangreiche Tests durchführen, wobei unmittelbar quantifizierte Größen als Meßergebnisse vorliegen. Besonders letzteres ist für die Entwicklung eines Werkzeugs wie MAGW von Vorteil.

A.3.5 Analyse und Emulation von Echtzeitdatenströmen für allgemeine Multimedia-Anwendungen

Gelegentlich ist es erforderlich, wenig dokumentierte Datenströme, wie sie beispielsweise bei Anwendungen zur Realisierung von Telefondiensten unter Nutzung des Internet-Protokolls entstehen, zu generieren. Der wesentliche Unterschied gegenüber der Analyse von MBone-Konferenzen ist, daß der Verkehrsstrom nur dann zu beobachten ist, wenn ein das (unbekannte) Protokoll realisierender Empfänger vorhanden ist. Daher ist hier der Einsatz von *udp-discard* nicht möglich. Stattdessen wird der Verkehr auf einem im Subnetz des Empfängers installierten Rechner unter Nutzung von *tcpdump* protokolliert und nachträglich analysiert.

Aus der Analyse ergeben sich die relevanten Parameter des Datenstroms. Sie dienen in einem zweiten Schritt zur Emulation vergleichbarer Datenströme in einer Produktionsumgebung. Dabei wird der erforderliche Datenstrom von *rtest* generiert und mittels *udp-discard* empfangen. Alternativ werden ausschließlich Messungen mit *rtest* durchgeführt, wobei der UDP-Echo-Port eines Partnersystems im zu untersuchenden Netz angesprochen wird. Letztere Vorgehensweise hat den Vorteil, daß ein bidirektionaler Datenstrom entsteht, der dem realen Szenario in der Regel eher entspricht.

Der Vorteil dieser Vorgehenswiese liegt in der strikten Trennung der Analyse der Datenströme zur Ermittlung der charakteristischen Parameter und der Eignungsprüfung des jeweiligen Netzwerkpfades zur Nutzung des Dienstes. Die Eignungsprüfung kommt ohne menschlichen Eingriff aus und liefert quantitative Daten, die sich nach Abschluß der Messungen leicht weiterverarbeiten lassen. Damit wird es möglich, Eignungsprüfungen in großer Zahl und über den Zeitverlauf verteilt durchzuführen. Dabei werden auch unterschiedliche Lastsituationen im Netz erfaßt. Die Aussagekraft der Messungen kann ggf. durch parallele Protokollierung und Auswertung von Router-Statistiken erhöht werden.

A.3.6 Untersuchung des Queuing-Verhaltens von Access-Routern

Ebenfalls interessant ist das Queuing-Verhalten von vermittelnden Komponenten in Netzen. Von besonderem Interesse sind entsprechende Messungen immer dann, wenn sich die vermittelnde Komponente auf der einen Seite an ein Netz mit hoher Maximaldatenrate angeschlossen ist, das

Netz auf der anderen Seite jedoch über eine deutlich geringere maximale Datenrate verfügt. Beispiele hierfür sind vor allem Access-Router und Terminal-Server, aber auch Ethernet-Switches, die über 100 Mbps und 10 Mbps Anschlüsse verfügen.

Das grundlegende Konzept dieser Messungen ist, daß die vermittelnde Komponente aus dem Netz mit der höheren Übertragungskapazität heraus mit einem weiterzuleitenden Datenstrom belastet wird, der von seiner Datenrate geringfügig über der Übertragungskapazität des anschließenden Netzes liegt. Die Messung kann eine Einwegmessung sein, wobei auf der Seite des Empfängers *udp-discard* zum Empfang der Pakete eingesetzt wird, oder, im Fall von Netzen mit symmetrischen Übertragungsraten (z.B. ISDN), eine Round-Trip-Time-Messung.

Wichtig für die erfolgreiche Durchführung der Messung ist, daß der Sender mit hinreichender Genauigkeit den eingestellten Datenstrom generiert und der Empfänger nicht integrale Parameter des Datenstroms erfaßt, sondern die Round-Trip-Time bzw. den Zeitunterschied der Zeitstempel beim Absenden und beim Empfang pro Paket ermittelt. Anhand der Ergebnisdaten lassen sich detaillierte Aussagen über das Verhalten der jeweils untersuchten Komponente hinsichtlich Queuing-Verhalten und Paketverwurf im Fall von Stauungen treffen.

rtest und die Programme *udp-echo* sowie *udp-discard* können über eine Option so konfiguriert werden, daß die erforderlichen Daten als Meßergebnisse ausgegeben werden.

A.4 Zusammenfassung

rtest hat sich zusammen mit den Programmen *udp-echo* und *udp-discard* als hilfreiches Werkzeug während der Entwicklung des MBone-Access-Gateways (MAGW) erwiesen. Der erfolgreiche Einsatz in anderen Projekten untermauert diesen Eindruck. Besondere Merkmale von *rtest* sind die Möglichkeit, einen in weiten Bereichen einstellbaren, konstanten UDP-Datenstrom zu erzeugen und die Ausgabe umfangreicher Daten bezüglich der Qualität des emittierten und eingehenden Datenstroms. Das Einsatzspektrum von *rtest* wird durch andere, etablierte Werkzeuge wie *tcpdump*, *traceroute* und *rtptools* erweitert.

rtest unterstützt den Anwender bei der Durchführung von Messungen zur Analyse und zum Test von Netzen und vermittelnden Komponenten in Netzen. Insbesondere lassen sich mit *rtest* folgende Messungen durchführen:

- Ermittlung von Round-Trip-Times von UDP-Paketen in IP-Netzen (Unicast und Multicast).
- Ermittlung von Round-Trip-Times über Application-Layer-Gateways im MBone-Kontext.
- Analyse und Emulation von MBone- und allgemeinen Multimedia-Echtzeitdatenströmen.
- Untersuchung des Queuing-Verhaltens von Access-Routern.

In diesem Anhang wurde die Motivation zur Entwicklung von *rtest* dargestellt und ein Einblick in die Eigenschaften von *rtest* gegeben. Dazu gehörte auch der Vergleich mit anderen, weit verbreiteten Analysewerkzeugen. Schließlich wurde der Einsatz von *rtest* in typischen Einsatzszenarien an einigen Beispielen erläutert. Der Aufruf der Programme und die verfügbaren Optionen sind den die Software begleitenden Manual-Pages zu entnehmen.

Anhang B

Ermittlung der UDP-Bulk-Transferleistung mit Netperf

B.1 Einleitung

netperf [Hew96] ist ein etabliertes Programm zur Durchführung von Netzwerk-Performance Benchmarks. Dieser Anhang beschreibt die Überarbeitung von *netperf* zur zuverlässigen Ermittlung der UDP-Bulk-Transferleistung zwischen zwei Rechnersystemen, die über einen ISDN-B-Kanal miteinander verbunden sind. Entsprechende Messungen waren für den Analyseteil der vorliegenden Arbeit erforderlich. Ihre Ergebnisse sind in Abschnitt 2.3.2, Seite 23ff., dargestellt.

Die Überarbeitung wurde erforderlich, weil das originäre Programm die Durchführung von Messungen zur Bestimmung der UDP-Bulk-Transferleistung zwar prinzipiell unterstützt, die Implementierung in der eingesetzten Version 2.1.pl1 jedoch fehlerhaft ist. Obwohl dieser Fehler in einer LAN-Umgebung, die der typische Einsatzbereich von *netperf* ist, kaum offensichtlich wird, tritt er bei der Durchführung von Messungen über Low-Speed-Serial-Links deutlich zutage.

Abgesehen von Schwächen in der aktuellen Implementierung handelt es sich um einen prinzipiellen Meßfehler. Er resultiert aus der Übertragung des Konzepts zur Ermittlung von Performance-Messungen über TCP auf UDP. Dies ist einer der Gründe, warum die Überarbeitung des Programms hier dargestellt wird.

Einhergehend mit der zunehmenden Nutzung der Internet-Technologie zur Übertragung multimedialer Datenströme nimmt der Gebrauch des UDP-Protokolls zu, sodaß auch die Ermittlung der UDP-Bulk-Transferleistung höheren Stellenwert erlangt. Die Unterstützung von entsprechenden Messungen durch im Internet-Umfeld häufig angewendete Benchmarking-Programme ist häufig unzureichend (Beispiel: *TTCP*) oder fehlerhaft (Beispiel: *netperf* in der Standard-Distribution). Wird *netperf* in der hier beschriebenen Weise modifiziert, entsteht ein zuverlässiges Programm zur Ermittlung von UDP-Bulk-Transferleistungen, das auch in anderen Szenarien einsetzbar ist. Dies ist ein weiterer Grund für die Darstellung der Überarbeitung von *netperf*.

Im folgenden Abschnitt wird das Verfahren erläutert, nachdem ein unmodifiziertes *netperf*-Programm in der Version 2.1pl1 die UDP-Bulk-Transferleistung ermittelt. Im darauffolgenden Abschnitt wird die Überarbeitung des Verfahrens zur korrekten Ermittlung der Transferleistung vorgestellt. Dieser Teil des Anhangs schließt mit einer Zusammenfassung.

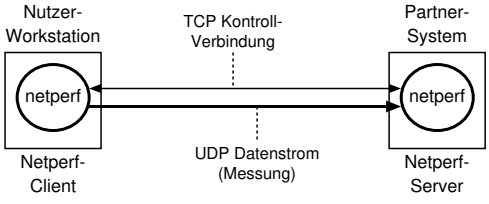

Abbildung B.1: Architektur des Benchmarking-Werkzeugs *netperf*

B.2 Meßverfahren von Netperf zur Ermittlung der UDP-Bulk-Transferleistung

Bei *netperf* handelt es sich um ein verteiltes System in klassischer Client-Server-Architektur. Jede Messung wird durch den Client initiiert und beginnt mit dem Aufbau einer TCP-Kontrollverbindung, über die die Messungen konfiguriert und die Ergebnisse ausgetauscht werden. Die Kommunikation über die Kontrollverbindung erfolgt über ein einfaches Anwendungsprotokoll, welches *netperf*-spezifisch ist. Die eigentliche Messung erfolgt über einen zweiten Datenstrom, der vom Client an den Server gesendet wird. Im betrachteten Fall handelt es sich um einen UDP-Datenstrom. Abbildung B.1 illustriert die *netperf*-Architektur.

Wenngleich *netperf* bei Messungen der UDP-Bulk-Transferleistung sowohl Sende- als auch Empfangsdatenrate ausgibt, ist nur die Betrachtung der letzteren relevant. Sie gibt Aufschluß über die Transferleistung des betrachteten Netzpfades.

Testmessungen mit einer unmodifizierten *netperf*-Version zur Ermittlung der UDP-Bulk-Transferleistung zwischen zwei über einen ISDN-B-Kanal miteinander gekoppelter UNIX-Systeme zeigten, daß die ermittelte Übertragungsrate in einigen Fällen die theoretisch mögliche überstieg.[1] Vergleichende Messungen, bei denen die beim Empfänger eingehenden Nachrichten mit dem Paket-Monitor *tcpdump* protokolliert und später analysiert wurden, lieferten den Beweis dafür, daß die von *netperf* ermittelten Ergebnisse falsch waren. Die Analyse des Programm-Quelltextes von *netperf* bestätigte dies.

Ursächlich für das Fehlverhalten ist die Methodik, der *netperf* zur Durchführung der Messung folgt. Der Client initiiert die Messung, indem über eine TCP-Kontrollverbindung ein *netperf*-eigener *DO_UDP_STREAM*-Request an den Server gesendet wird. Er beinhaltet die Dauer der Messung (T_{send}) sowie die Größe der zu übertragenen Nachrichten (UDP-Payload). Der Server antwortet mit einer *UDP_STREAM_RESPONSE*, die als Bestätigung dient und den zu nutzenden Server-Port beinhaltet. Der Server startet daraufhin einen Timer mit dem Wert $T_{recv} = T_{send} + 2s$, d.h. der Empfangs-Timer läuft 2 Sekunden länger als der Sende-Timer. Der Server startet den Timer unmittelbar und geht in die Empfangsschleife, die auf Daten vom Client wartet. Der Client verbindet sich nach dem Erhalt der Antwort mit dem gegebenen Port des Servers, startet seinen Sende-Timer und beginnt ohne Datenratenbegrenzung zu senden. Nach Ablauf des Timers beim Sender stellt dieser die Übertragung ein und wartet auf die Testergebnisse vom Server. Der Timer des Servers kann zu diesem Zeitpunkt bereits abgelaufen sein oder wird innerhalb von maximal

[1] Die theoretisch mögliche Übertragungsrate ergibt sich zu $\frac{(mesg.size)}{mesg.size+35} \cdot 64kbps$. Die Konstante im Nenner ergibt sich aus dem Protokoll-Overhead von 7 Byte für PPP, 20 Byte IP und 8 Byte UDP-Header.

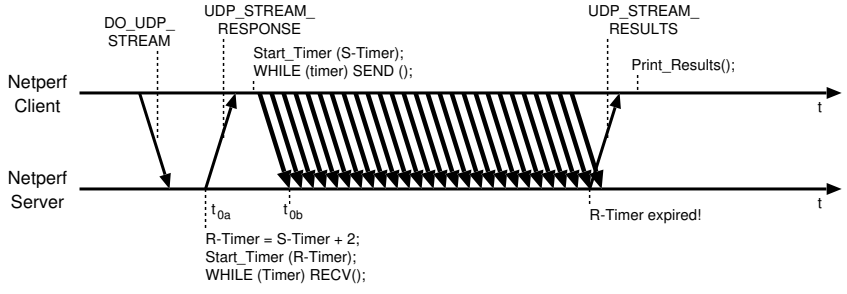

Abbildung B.2: Ermittlung der UDP-Bulk-Transferleistung mit *netperf*

2 Sekunden ablaufen. Nach Ablauf des Timers ermittelt der Server die Meßergebnisse, wobei die Zeitdauer der Messung mit T_{send} angenommen wird. Abbildung B.2 stellt die Vorgänge bei der Ermittlung der UDP-Bulk-Transferleistung dar.

Die Ergebnisse werden an den Client übertragen und dort ausgewertet. Ein Fehler der Implementierung ist, daß dabei nur die Anzahl der übertragenen PDUs benutzt wird. Die Summe der übertragenen Bytes wird aus dieser Zahl, multipliziert mit der Nachrichtengröße des Senders, ermittelt. Ein weiterer Implementierungsfehler ist, daß die Übertragungsrate aus der fehlerhaft ermittelten Zahl der übertragenen Bytes und der Meßdauer beim Client errechnet wird.

Der Meßfehler liegt nicht nur in den Schwächen der Implementierung, sondern auch im Konzept zur Ermittlung der tatsächlichen Zeit, die der Server zum Empfang der Daten benötigt hat: In das Meßergebnis geht als übertragene Datenmenge die Anzahl der empfangenen Pakete seit dem Start des Timers beim Empfänger und dem Ablauf des Timers beim Empfänger ein. Tatsächlich wurden in dem Zeitintervall $\Delta t_0 = t_{0b} - t_{0a}$ keine Daten empfangen.

B.3 Überarbeitung von *netperf* zur korrekten Ermittlung der UDP-Bulk-Transferleistung

Korrekte Werte ergeben sich, wenn der Empfänger mit der Zeitmessung nach dem Empfang der ersten PDU beginnt (t_0) und einen Zeitstempel der jeweils zuletzt empfangenen PDU (t_1) aufbewahrt. Unter der Randbedingung, daß die Größe der übertragenen PDUs (m) konstant ist, ergibt sich die Datenübertragungsrate beim Empfang von N PDUs zu:

$$C = \frac{(N-1) \cdot m}{t_1 - t_0}$$

Dabei bedeuten:

- C Übertragungskapazität des Netzwerkpfades zwischen Sender und Empfänger.
- N Anzahl der empfangenen Pakete.
- m Byte-Größe der empfangenen Pakete (UDP-Payload).
- t_0 Zeitpunkt des Empfangs des ersten Pakets.
- t_1 Zeitpunkt des Empfangs des letzten Pakets.

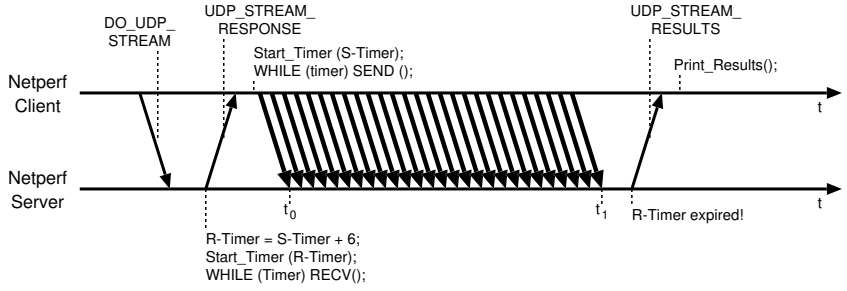

Abbildung B.3: Korrekte Ermittlung der UDP-Bulk-Transferleistung mit *netperf*

netperf wurde entsprechend des vorliegenden Modellansatzes korrigiert. Die vom Client übergebene Dauer des Meßzeitraumes wurde um 6 Sekunden vergrößert und dient nur noch dem Abbruch der Messung. Die in die Berechnung des Datenflusses eingehenden Zeiten t_0 und t_1 werden zusätzlich ermittelt und die Differenz als Zeitdauer der Messung an den Client übertragen. Die ebenfalls zu übergebende Anzahl von empfangenen PDUs wurde gemäß obigem Term berichtigt. Gleichfalls wird das übertragene Datenvolumen korrekt ermittelt und an den Client weitergegeben.

Abbildung B.3 zeigt den Ablauf einer Messung zur Ermittlung der UDP-Bulk-Transferleistung mit der korrigierten *netperf*-Version. Dabei wird besonders deutlich, daß der Beginn der Zeitmessung durch den Empfang des ersten Pakets des Meßdatenstroms festgelegt wird.

Der Client wurde so modifiziert, daß die vom Server übermittelten Werte für Datenvolumen, Anzahl der PDUs und die Zeitdauer für den Empfang der PDUs für die Berechnung des Informationsflusses herangezogen und die Ergebnisse korrekt ausgegeben werden.

Mit der korrigierten Version von *netperf* wurden erneut Testmessungen und Konvergenz-Tests unter der Nutzung von *tcpdump* durchgeführt. Die Ergebnisse zeigten, daß die ermittelten Datenflüsse nun unter den theoretisch möglichen Datenflüssen liegen und die *netperf*-Ergebnisse mit den *tcpdump*-Vergleichsmessungen übereinstimmen.

Letztlich stellt sich die Frage, warum die Messungen nicht unmittelbar unter Nutzung von *tcpdump* durchgeführt wurden. *tcpdump* erhält die Daten vom Data-Link-Layer und nicht von der UDP-Schicht, wie *netperf*. Zudem wird das Netzwerk-Interface in den *promiscuous mode* geschaltet. In Ethernet-Netzen werden in diesem Mode alle Daten, die über das Netz transportiert werden, empfangen. Außerdem muß jedes empfangene Paket während der Messung in eine Log-Datei auf der Festplatte des Servers geschrieben werden. Es war unklar, ob diese Randbedingungen die Meßergebnisse verfälschen. Daher wurden vergleichende Messungen durchgeführt. Das Ergebnis zeigt, daß Messungen mit *tcpdump* zumindest im betrachteten Bereich mit Datenflüssen von bis zu 64 kbps zuverlässig sind. Für die Aufnahme ganzer Meßreihen ist *netperf* aufgrund des höheren Nutzungskomforts *tcpdump* vorzuziehen.

B.4 Zusammenfassung

Am Beispiel des verbreiteten Netzwerk-Benchmarking-Programms *netperf* wurden Fehler bei der Ermittlung der UDP-Bulk-Transferleistung aufgezeigt und ein Verfahren zur Vermeidung dieser Fehler dargestellt. Die Fehler sind grundsätzlicher Natur, werden in LAN-Umgebungen jedoch kaum offensichtlich.

Die wesentliche Neuerung beim korrigierten Verfahren ist, daß der Anfangszeitpunkt der Messung durch den Empfang des ersten Pakets des Meßdatenstroms festgelegt wird. Das Verfahren ist auch auf die Ermittlung von Übertragungsleistungen von TCP-Verbindungen übertragbar. Das Programm *netperf* wurde so modifiziert, daß die zuverlässige Messung von UDP-Bulk-Transfer-Leistungen möglich wird.

Eine wichtiger Schluß aus den beschriebenen Erfahrungen ist, daß Meßverfahren zur Bestimmung von Durchsatzleistungen von Netzwerken auch in etablierten Werkzeugen fehlerhaft implementiert sein können und/oder konzeptionelle Fehler enthalten können, die nur durch außerordentlich kritische Analyse der Meßergebnisse sowie die Durchführung von Konvergenzmessungen zu erkennen sind.

Anhang C

Netz- und Rechnerkonfiguration

Die folgenden Abschnitte fassen noch einmal die technischen Daten der Komponenten zusammen, unter deren Nutzung die Entwicklung des *MBone Access Gateways* erfolgte. Dies soll die Reproduktion der Meßergebnisse ermöglichen. Der Anhang untergliedert sich in die Darstellung des Testnetzes und eine listenförmige Darstellung der Eigenschaften der beteiligten Rechnersysteme und aktiven Netzkomponenten.

C.1 Netzwerkkonfiguration

Abbildung C.1 zeigt die den Messungen zugrundeliegende Netzwerkkonfiguration. Deutlich hervorgehoben ist im oberen Bereich der Abbildung das private Internet. Es besteht aus den Rechnern *jack* und *jack2*, die über ein 10 Mbps Ethernet-Segment miteinander verbunden sind. Dieses Ethernet-Segment ist als klassiches Cheaper-Net ausgeführt. Während der Rechner *jack2* als Arbeitsplatzrechner fungiert, dient der Rechner *jack* als Host für das *MBone Access Gate* (MAG).

Gleichzeitig koppelt der Rechner *jack* das private Internet über einen ISDN-Adapter, den anschließenden S_0-Bus und das Telefon-Festnetz an das Datennetz der Universität Hannover. Den Aufpunkt für die IP-Kommunikation bildet hier der über zwei S_{2m}-Anschlüsse angekoppelte Access-Router *Tserv1*.

Der Access-Router *Tserv1* ist über zwei 10 Mbps Ethernet-Segmente und den Frame-Switch *switch-s5-0* mit dem zentralen Backbone-Router *BWINgate* verbunden.

Zentrale Bedeutung für die Messungen hat der Rechner *lanai*. Er dient für die Messungen als Host für den *MBone Access Server* (MAS). Zum einen ist er durch ein 155 Mbps ATM Netzsegment mit dem IP-Router *BWINgate* verbunden, zum anderen mit 100 Mbps Ethernet an den Frame-Switch *switch-s5-1* angeschlossen. Während die ATM Strecke unter Nutzung der LAN-Emulation zum Transport des Tunnel-Datenstroms genutzt wird, wird der *lanai* über den Ethernet-Anschluß der Multicast-Datenstrom aus dem deutschen MBone zugeführt.

Ebenfalls relevant für die dargestellten Meßergebnisse ist der Rechner *ernie*. Er diente als Partnerinstanz für Messungen vom Arbeitsplatzrechner *jack2*. Dieser Rechner ist mittels eines 10 Mbps Anschlusses an den Frame-Switch *switch-s5-1* angeschlossen.

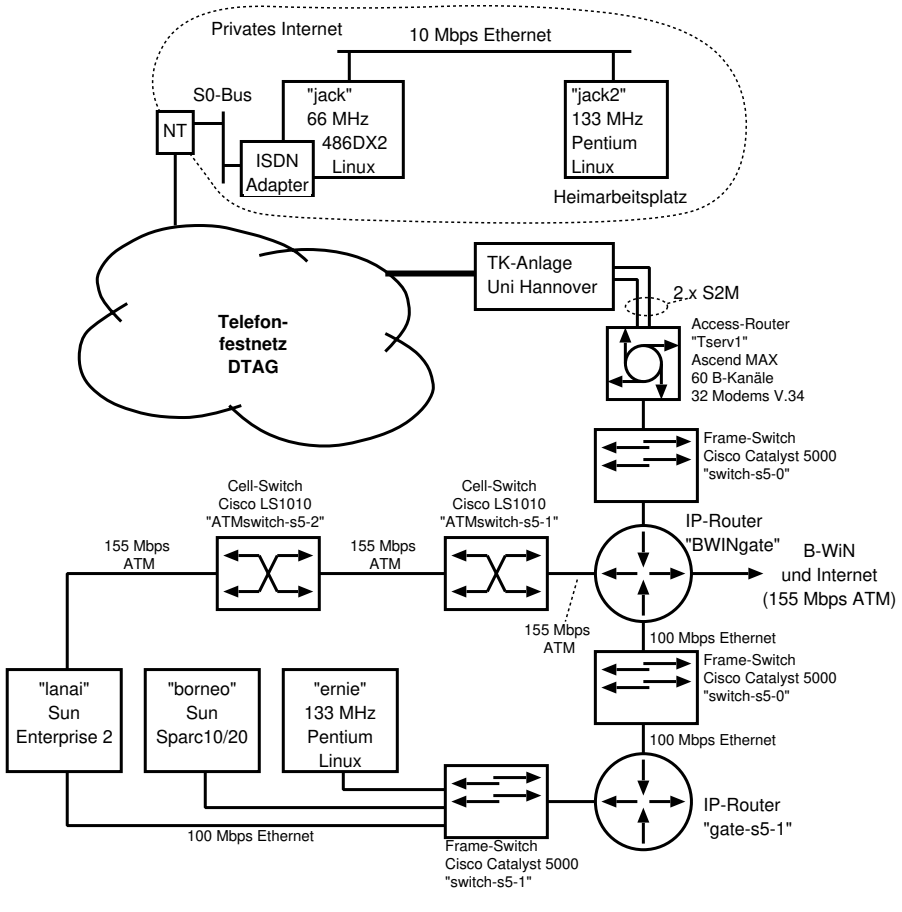

Abbildung C.1: Netzwerkkonfiguration

Der Rechner *borneo* wurde genutzt, um die Meßergebnisse zwischen *jack2* und *ernie* zu verifizieren. Auch dieser Rechner ist über 10 Mbps Ethernet an den Frame-Switch *switch-s5-1* angeschlossen.

C.2 Rechnerkonfiguration

In der folgenden Aufstellung werden die wesentlichen Merkmale der genutzten Rechner sowie der aktiven Netzwerkkomponenten genannt.

C.2.1 Rechner

jack2

Typ:	Personal Computer
CPU:	Intel Pentium 133 MHz
Arbeitspeicher:	32 MB
Betriebssystem:	Linux 2.0.24
Netzanschlüsse:	10 Mbps Ethernet (NE-2000)

jack

Typ:	Personal Computer
CPU:	Intel 486 DX 2 66 MHz
Arbeitspeicher:	16 MB
Betriebssystem:	Linux 2.0.29
Netzanschlüsse:	10 Mbps Ethernet (NE-2000)
	ISDN-Adapter (Teles PnP)

lanai

Typ:	Sun Enterprise Server 2
CPU:	2 UltraSPARC CPUs 200 MHz
Arbeitspeicher:	384 MB
Betriebssystem:	Solaris 2.5.1
Netzanschlüsse:	100 Mbps Ethernet (Sun Standard)
	155 Mbps ATM (Sun ATM-Adapter 2.1)

borneo

Typ:	Sun SPARCstation 10/20
CPU:	1 SuperSPARC CPU 33 MHz
Arbeitspeicher:	80 MB
Betriebssystem:	Solaris 2.4
Netzanschlüsse:	10 Mbps Ethernet (Sun Standard)

ernie

Typ:	Personal Computer
CPU:	Intel Pentium 133 MHz
Arbeitspeicher:	32 MB
Betriebssystem:	Linux 2.0.29
Netzanschlüsse:	10 Mbps Ethernet (NE-2000)

C.2.2 Aktive Netzwerkkomponenten

Tserv1

Typ:	Access Router
Modell:	Ascend MAX E1/PRI
Betriebssystem:	4.6Bp26

BWINgate

Typ:	IP Router
Modell:	Cisco 7507
Betriebssystem:	IOS RSP Version 11.2(7)

gates-s5-1

Typ:	IP Router
Modell:	Cisco 7507
Betriebssystem:	IOS RSP Version 11.2(7a)

switch-s5-0

Typ:	Frame Switch
Modell:	Cisco Catalyst 5000
Betriebssystem:	Catalyst Software 2.4(1)

switch-s5-1

Typ:	Frame Switch
Modell:	Cisco Catalyst 5000
Betriebssystem:	Catalyst Software 2.4(1)

ATMswitch-s5-1

Typ:	ATM Switch
Modell:	Cisco LightStream 1010
Betriebssystem:	IOS IISP 11.1(9)

ATMswitch-s5-2

Typ:	ATM Switch
Modell:	Cisco LightStream 1010
Betriebssystem:	IOS IISP 11.1(9)

Literaturverzeichnis

[AK97] G. Almes und S. Kalidindi. *Internet Draft: A One-way Delay Metric for IPPM*, March 1997. Work in progress.

[AM] E. Amir und S. McCanne. *RTPGW: An Application Level RTP Gateway*. `ftp://deadalus.cs.berkeley.edu/pub/rtpgw`.

[Ami95] E. Amir. *An Application Level Video Gateway*. Master's thesis, University of California, Berkeley, December 1995. `http://http.cs.berkeley.edu/~elan/papers/vgw-ms.ps`.

[AMZ95] E. Amir, S. McCanne und H. Zhang. An Application Level Video Gateway. In *ACM Multimedia '95 proceedings*, November 1995. `http://http.cs.berkeley.edu/~elan/papers/vgw.ps`.

[Asc96] Ascend Communications Inc. *MAX E1/PRI 4.6C Addendum*, September 1996.

[Bak95] F. Baker. *RFC 1812: Requirements for IP Version 4 Routers*, June 1995.

[Bal93] D. Balenson. *Privacy Enhancement for Internet Electronic Mail: Part III: Algorithms, Modes, and Identifiers*, February 1993.

[Ber94] A. von Berg. *Entwurf und Realisierung eines Vermittlungssystems für multimediale Online-Konferenzen*. Studienarbeit, Lehrgebiet Rechnernetze und Verteilte Systeme, Universität Hannover, Dezember 1994. `http://www.rvs.uni-hannover.de/arbeiten/studien/sa-avberg.html`.

[BFFM96] L. Berc, W. Fenner, R. Frederick und S. McCanne. *RFC 2035: RTP Payload Format for JPEG-compressed Video*, October 1996.

[BFGP96] B. Böker, C. Fricke, L. Grüneberg und H. Pralle. Entwicklung eines Management-Systems für multimediale Online-Konferenzen. In *Proceedings der Fachtagung SIWORK 96*, Mai 1996. `http://www.rvs.uni-hannover.de/reports/siwork.html`.

[BFS97] B. Böker, M. Fromme und H. Schulze. *Multimediale Zusammenarbeit in der Klimaforschung: Abschlußbericht RTB-Nord Projekt P6.1*, März 1997. `http://www.rtb-nord.uni-hannover.de/mmzusammenarbeit/p61.ps.gz`.

[Bor97a] C. Bormann. *Internet Draft: PPP in real-time oriented HDLC-like framing*, March 1997. draft-ietf-issll-isslow-rtf-00.txt, Work in progress.

[Bor97b] C. Bormann. *Internet Draft: Providing integrated services over low-bitrate links*, May 1997. draft-ietf-issll-isslow-02.txt, Work in progress.

[BP87] R. Braden und J. Postel. *RFC 1009: Requirements for Internet Gateways*, June 1987.

[Bra97] R. Braden et al. *RFC 2205: Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification*, September 1997.

[Bun95] Bundesministerium für Wirtschaft. *Die Informationsgesellschaft: Fakten, Analysen, Trends*, November 1995.

[CCC+91] D. Clark, L. Chapin, V. Cerf, R. Braden und R. Hobby. *RFC 1287: Towards the Future Internet Architecture*, December 1991.

[Cis96] Cisco Systems, Inc. *Cisco IOS Software Features for Differentiated Class of Service for Internetworks*, January 1996. `http://www.cisco.com/warp/public/732/General/cos_wp.htm`.

[Cis97] Cisco Systems, Inc. *Cisco Express Forwarding (CEF)*, May 1997. `http://www.cisco.com/warp/public/732/tag/cef_wp.htm`.

[CJ97] S. Casner und V. Jacobson. *Internet Draft: Compressing IP/UDP/RTP Headers for Low-Speed Serial Links*, July 1997. draft-ietf-avt-crtp-03.txt, Work in progress.

[CT90] D. Clark und D. Tennenhouse. Architectural Considerations for a New Generation of Protocols. In *SIGCOMM '90*, Seiten 200–208, Philadelphia, Sep 1990. ACM.

[CWHC96] J. Crowroft, I. Wakeman, M. Handley und S. Clayman. *Internetworking Multimedia*. UCL Press, 1996.

[Dee89] S. Deering. *RFC 1112: Host Extensions for IP Multicasting*, August 1989.

[Dee91] S. E. Deering. *Multicast Routing in a Datagram Internetwork*. PhD thesis, Department of Computer Science, Stanford University, December 1991.

[Dee97] S. Deering et al. *Internet Draft: Protocol Independent Multicast Version 2, Dense Mode Specification*, May 1997. draft-ietf-idmr-pim-dm-05.txt, Work in progress.

[Deu96a] P. Deutsch. *RFC 1951: DEFLATE Compressed Data Format Specification version 1.3*, May 1996.

[Deu96b] P. Deutsch. *RFC 1952: GZIP file format specification version 4.3*, May 1996.

[DFN] DFN-Verein and Rechenzentrum der Universität Stuttgart. *WiN-Shuttle*. `http://www.shuttle.de/`.

[DG96] P. Deutsch und J-L. Gailly. *RFC 1950: ZLIB Compressed Data Format Specification version 3.3*, May 1996.

[DH95] S. Deering und R. Hinden. *RFC 1883: Internet Protocol, Version 6 (IPv6) Specification*, December 1995.

[Dix93] T. Dixon. *RFC 1454: Comparison of Proposals for Next Version of IP*, May 1993.

[ECB97] M. Engan, S. Casner und C. Bormann. *Internet Draft: IP Header Compression over PPP*, April 1997. draft-engan-ip-compress-00.txt, Work in progress.

[Est97] D. Estrin et al. *RFC 2117: Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification*, June 1997.

[Eur96] Europäische Komission. *Bericht der Task Force "Multimedia und Lernprogramme"*, Juli 1996. `http://www.echo.lu`.

[Fen97] B. Fenner. *Internet Draft: Internet Group Management Protocol, Version 2*, January 1997. Work in progress.

[Fera] FernUniversität Hagen. *Studierende WS 1996/97 nach Hörerestatus und Fachbereichen*. `http://www.fernuni-hagen.de/FERNUNIINFO/stat1.gif`.

[Ferb] FernUniversität Hagen. *Virtuelle Universität/FernUniversität Online*. `http://vus.fernuni-hagen.de/`.

[FGV+94] M. Fromme, L. Grüneberg, J.-S. Vöckler, W. Determann und F. Patz. *Software-Engineering: Eine Einführung*, 1994. `http://www.rvs.uni-hannover.de/umdrucke/swl-se-ss94.ps`.

[Fie97] R. Fielding et al. *RFC 2068: Hypertext Transfer Protocol – HTTP/1.1*, January 1997.

[Fin93] C. Finseth. *RFC 1492: An Access Control Protocol, Sometimes Called TACACS*, July 1993.

[FJM+95] S. Floyd, V. Jacobson, S. McCanne, C.-G. Liu und L. Zhang. *A Reliable Multicast Framework for Light-Weight Sessions and Application Level Framing*, November 1995. `ftp://ftp.ee.lbl.gov/papers/wb.tech.ps.Z`.

[Fre96] A. O. Freier et al. *Internet Draft: The SSL Protocol Version 3.0*, November 1996. draft-freier-ssl-version3-02.txt, `http://www.netscape.com/newsref/std/SSL.html`, Work in progress.

[Fre97] Free Software Foundation. *GCC - The GNU C Compiler*, November 1997. `http://www.gnu.org/software/gcc/gcc.html`.

[Fri96] C. Fricke. *Confman: Entwicklung eines Online-Conferencing-Systems für nichtöffentliche Konferenzen in breitbandigen Netzen*. Studienarbeit, Lehrgebiet Rechnernetze und Verteilte Systeme, Universität Hannover, Juni 1996. `http://www.rvs.uni-hannover.de/arbeiten/studien/sa-cfricke.html`.

[Fro95] M. Fromme. *Multimedia-Konferenzen in der Wissenschaft: Szenarien, Technologie und Werkzeuge*. Studienarbeit, Lehrgebiet Rechnernetze und Verteilte Systeme, Universität Hannover, Juli 1995.

[Fro96] M. Fromme. *Virtual Meeting Control Center: Entwurf und Realisierung eines Systems zur Gesprächssteuerung und für Abstimmungen bei Online-Videokonferenzen*. Diplomarbeit, Lehrgebiet Rechnernetze und Verteilte Systeme, Universität Hannover, Dezember 1996.

[Gam96] E. Gamma et al. *Entwurfsmuster: Elemente wiederverwendbarer objektorientierter Software*. Addison-Wesley, Bonn, 1996.

[Grü96] L. Grüneberg. *Protokolle im Internet: Multimedia-Protokolle*, November 1996. `http://www.rvs.uni-hannover.de/people/gruen/vortraege/961113-MM-Seminar-Internet2/index.html`.

[Han96a] M. Handley. *Internet Draft: SAP: Session Announcement Protocol*, November 1996. Work in progress.

[Han96b] M. Handley. *The sdr Session Directory: An Mbone Conference Scheduling and Booking System*, April 1996. `http://ugwww.ucs.ed.ac.uk/mice/archive/sdr.html`.

[Han97] M. Handley. *On Scalable Internet Multimedia Conferencing Systems*. PhD thesis, University of London, August 1997. submitted to the University of London for the qualification of PhD (not yet examined), `http://north.east.isi.edu/~mjh/thesis.ps.gz`.

[HC92] S. Hailes und J. Crowcroft. *Mash*, January 1992. `ftp://cs.ucl.ac.uk/darpa/mash.1.Z` und `ftp://cs.ucl.ac.uk/darpa/mash.c.Z`.

[Hew96] Information Networks Division, Hewlett–Packard Company. *Netperf: A Network Performance Benchmark, Revision 2.1*, February 1996. `http://www.cup.hp.com/netperf/NetperfPage.html`.

[HFG96] D. Hoffman, G. Fernando und V. Goyal. *RFC 2038: RTP Payload Format for MPEG1/MPEG2 Video*, October 1996.

[HJ97] M. Handley und V. Jacobson. *Internet Draft: SDP: Session Description Protocol*, March 1997. Work in progress.

[Hol95] W. Holfelder. MBone VCR - Video Conference Recording on the MBone. In *Proceedings of the ACM Multimedia 95*, November 1995. `http://www.informatik.uni-mannheim.de/~whd/publications/acm-mm95/`.

[Int93] International Telecommunication Union. *ITU-T Recommendation G.114: One-Way Transmission Time*, March 1993.

[Int96a] International Telecommunication Union. *ITU-T Recommendation H.321: Visual Telephone Sytems to B-ISDN Environments*, March 1996.

[Int96b] International Telecommunication Union. *ITU-T Recommendation H.322: Visual Telephone Sytems and Terminal Equipment for Local Area Networks which provide a guaranteed Quality of Service*, November 1996.

[Int96c] International Telecommunication Union. *ITU-T Recommendation H.323: Visual Telephone Sytems and Equipment for Local Area Networks which provide a non-guaranteed Quality of Service*, November 1996.

[Int96d] International Telecommunication Union. *ITU-T Recommendation H.324: Terminal for low bit rate Multimedia Communication*, March 1996.

[Int97a] International Telecommunication Union. *ITU-T Recommendation H.320: Narrow-band Visual Telephone Systems and Terminal Equipment*, July 1997.

[Int97b] Internet Assigned Numbers Authority (IANA). *INTERNET MULTICAST ADDRESSES*, October 1997. `ftp://ftp.isi.edu/in-notes/iana/assignments/multicast-addresses`.

[Jac90] V. Jacobson. *RFC 1144: Compressing TCP/IP Headers for Low-Speed Serial Links*, February 1990.

[Jac97] V. Jacobson. *pathchar – a tool to infer characteristics of internet paths*, April 1997. `ftp://ftp.ee.lbl.gov/pathchar/msri-talk.ps.Z`.

[JM] V. Jacobson und S. McCanne. *Visual Audio Tool (VAT)*. `ftp://ftp.ee.lbl.gov/conferencing/vat`.

[JR86] R. Jain und S. A. Routhier. Packet Trains: Measurements and a New Model for Computer Traffic. *IEEE Journal on Selected Areas in Commonications*, 4(6):1162–1167, May 1986.

[JZ77] A. Lempel J. Ziv. A Universal Algorithm for Sequential Data Compression. *IEEE Transactions on Information Theory*, 23(3):337–343, May 1977.

[Kar93] F. Kardel. Verteilte Zeiten: Netzweite Synchronisation von Rechneruhren. *iX: Multiuser Multitasking Magazin*, Februar 1993.

[KB94] W. Kowalk und M. Burke. *Rechnernetze: Konzepte und Techniken der Datenübertragung*. B. G. Teubner, Stuttgart, 1994.

[KR90] B. W. Kernighan und D. M. Ritchie. *Programmieren in C, 2. Auflage*. Hanser, München, Wien; Prentice-Hall International, London, 1990.

[Kya96] O. Kyas. *Sicherheit im Internet: Risikoanalyse – Strategien – Firewalls*. Datacom Buchverlag, Bergheim, 1996.

[Lam78] Leslie Lamport. Time, Clocks, and the Ordering of Events in a Distributed System. *Communications of the ACM*, 21(7), July 1978.

[LGL+96] M. Leech, M. Ganis, Y. Lee, R. Kuris, D. Koblas und L. Jones. *RFC 1928: SOCKS Protocol Version 5*, March 1996.

[LS92] B. Lloyd und W. Simpson. *RFC 1334: PPP Authentication Protocols*, October 1992.

[McG92] G. McGregor. *RFC 1332: The PPP Internet Protocol Control Protocol (IPCP)*, May 1992.

[Mil92] D. Mills. *RFC 1305: Network Time Protocol (Version 3): Specification, Implementation and Analysis*, March 1992.

[MJ95] S. McCanne und V. Jacobson. vic: A Flexible Framework for Packet Video. In *ACM Multimedia '95 proceedings*, Seiten 511–522, November 1995.

[MS96] D.R. Musser und A. Saini. *STL Tutorial and Reference Guide*. Addison-Wesley, Reading, Massachusetts, 1996.

[Neg95] N. Negroponte. *Total Digital*. C. Bertelsmann Verlag GmbH, München, 1995.

[New97] C. Newman. *Internet Draft: Application Protocol Design Principles*, July 1997. draft-newman-protocol-design-01.txt, Work in progress.

[Ous94] J. K. Ousterhout. *Tcl and the Tk Toolkit*. Addison-Wesley, Reading, Massachusetts, 1994.

[Per96] C. Perkins. *RFC 2003: IP Encapsulation within IP*, October 1996.

[Per97] C. Perkins et al. *Internet Draft: RTP Payload for Redundant Audio Data*, June 1997. Work in progress.

[Pos80] J. Postel. *RFC 768: User Datagram Protocol*, August 1980.

[Pos81a] J. Postel. *RFC 791: Internet Protocol*, September 1981.

[Pos81b] J. Postel. *RFC 792: Internet Control Message Protocol*, September 1981.

[Pos81c] J. Postel. *RFC 793: Transmission Control Protocol*, September 1981.

[Pos83] J. Postel. *RFC 868: Time Protocol*, May 1983.

[PR85] J. Postel und J. Reynolds. *RFC 959: FILE TRANSFER PROTOCOL (FTP)*, October 1985.

[PSS97a] P. Parnes, K. Symnes und D. Schefström. *mTunnel: a multicast tunneling system with a user based Quality-of-Service model*, March 1997. `http://www.cdt.luth.se/~peppar/progs/mTunnel/IDMS97/mTunnel.ps`.

[PSS97b] P. Parnes, K. Symnes und D. Schefström. *mTunnel Version 0.1*, March 1997. `http://www.cdt.luth.se/~peppar/progs/mTunnel/mTunnel.0.1.tar.gz`.

[Pus97] T. Pusateri. *Internet Draft: Distance Vector Multicast Routing Protocol*, February 1997. draft-ietf-idmr-dvmrp-v3-04.txt, Work in progress.

[Rig97] C. Rigney et al. *RFC 2138: Remote Authentication Dial In User Service (RADIUS)*, April 1997.

[RMK+96] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot und E. Lear. *RFC 1918: Address Allocation for Private Internets*, February 1996.

[Roe97] G. Roelofs. *zlib: A Massively Spiffy Yet Delicately Unobtrusive Compression Library*, November 1997. `http://quest.jpl.nasa.gov/zlib/`.

[Rom88] J. Romkey. *RFC 1055: Nonstandard for transmission of IP datagrams over serial lines: SLIP*, June 1988.

[SCFJ96] H. Schulzrinne, S. Casner, R. Frederick und V. Jacobson. *RFC 1889: RTP: A Transport Protocol for Real-Time Applications*, January 1996.

[Sch96a] B. Schneier. *Angewandte Kryptographie: Protokolle, Algorithmen und Sourcecode in C*. Addison Wesley, Bonn (u.a.), 1996.

[Sch96b] H. Schulzrinne. *RFC 1890: RTP Profile for Audio and Video Conferences with Minimal Control*, January 1996.

[SH96] M. Speer und D. Hoffman. *RFC 2029: RTP Payload Format of Sun's CellB Video Encoding*, October 1996.

[SHK+] A. Sasse, V. Hardman, I. Kouvelas, C. Perkins, O. Hodson, A. Watson, M. Handley und J. Crowcroft. *Robust Audio Tool (RAT)*. `http://www-mice.cs.ucl.ac.uk/mice/rat/`.

[Sim93] W. Simpson et al. *RFC 1549: PPP in HDLC Framing*, December 1993.

[Sim94] W. Simpson et al. *RFC 1661: The Point-to-Point Protocol (PPP)*, July 1994.

[Sim96] W. Simpson. *RFC 1994: PPP Challenge Handshake Authentication Protocol (CHAP)*, August 1996.

[SSC] H. Schulzrinne, D. Sisalem und S. Casner. *Rtp-Tools (Rtpdump, Rtpsend, Rtpplay, Rtptrans)*. `ftp://ftp.cs.columbia.edu/pub/schulzrinne/rtptools`.

[Ste92] W. R. Stevens. *Advanced Programming in the UNIX Environment*. Addison-Wesley, Reading, Massachusetts, 1992.

[Ste93] R. Steinmetz. *Multimedia-Technologie: Einführung und Grundlagen*. Springer-Verlag, Berlin, Heidelberg, New York, 1993.

[Ste94] W. R. Stevens. *TCP/IP Illustrated, Volume 1: The Protocols*. Addison-Wesley, Reading, Massachusetts, 1994.

[Ste96] W. R. Stevens. *TCP/IP Illustrated, Volume 3: TCP for Transactions, HTTP, NNTP, and the UNIX Domain Protocols*. Addison-Wesley, Reading, Massachusetts, 1996.

[Str91] B. Stroustrup. *The C++ programming language, 2nd ed.* Addison Wesley, Reading, Massachusetts (u.a.), 1991.

[Sun97] Sun Microsystems. *JDK 1.1.4 Documentation*, October 1997. `http://www.javasoft.com/products/jdk/1.1/docs/index.html`.

[Tan90] A. S. Tanenbaum. *Computer-Netzwerke*. Wolfram's Fachverlag, 2. Auflage, 1990.

[TH96] T. Turletti und C. Huitema. *RFC 2032: RTP payload format for H.261 video streams*, October 1996.

[Ull93] R. Ullmann. *RFC 1475: TP/IX: The Next Internet*, June 1993.

[UNIa] UNIX-Manual-Page. *netdate - set date and time by ARPA Internet RFC 868.*

[UNIb] UNIX-Manual-Page. *rdate - get the date and time via the network.*

[Val97] A. Valencia et al. *Internet Draft: Layer Two Tunneling Protocol "L2TP"*, June 1997. Work in progress.

[Vöc97] J.-S. Vöckler. *Entwurf und Realisierung eines Vermittlungsagenten für multimediale Online-Konferenzen*. Diplomarbeit, Lehrgebiet Rechnernetze und Verteilte Systeme, Universität Hannover, Juni 1997. `http://www.rvs.uni-hannover.de/arbeiten/diplom/da-jvoeck.html`.

[Vog96] C. Vogt. *Erweiterung der Breitbandkabelnetze zur Realisierung interaktiver Dienste im Teilnehmeranschlußbereich.* Diplomarbeit, Lehrgebiet Rechnernetze und Verteilte Systeme, Universität Hannover, Dezember 1996.

[Wal96] L. Wall et al. *Programming Perl*. O'Reilly, Sebastopol, California, 2. Auflage, 1996.

[WC92] Z. Wang und J. Crowcroft. *RFC 1335: A Two-Tier Address Structure for the Internet: A Solution to the Problem of Address Space Exhaustion*, May 1992.

[WS95] G.R. Wright und W. R. Stevens. *TCP/IP Illustrated, Volume 2: The Implementation*. Addison-Wesley, Reading, Massachusetts, 1995.

[Yan97] J. Yang. *Accessiong MBone Sessions over ISDN*, June 1997. `http://www.cs.ucl.ac.uk/staff/J.Yang/doc/report2.ps.Z`.

[Zim95] P.R. Zimmermann. *The Official PGP Users´s Guide*. MIT Press, 1995.

Tabellarischer Lebenslauf

Lutz Grüneberg

Persönliche Daten

25.10.1963	geboren in Hannover als Sohn des Dipl.-Ing Horst Grüneberg und seiner Ehefrau Ingrid, geb. Liegmann.
seit Okt. 1990	verheiratet mit Ulrike Grüneberg, geb. Homes.
März 1991	Geburt des Sohnes Philipp.
Juni 1995	Geburt des Sohnes Oskar.

Schulbesuch und Ausbildung

1970 – 1974	Grundschule Hannover – Bemerode
1974 – 1983	Gymnasium Bismarckschule, Hannover. Abschluß mit dem Abitur.
1983 – 1984	Grundwehrdienst.
1984 – 1991	Universität Hannover, Studium der Elektrotechnik, Schwerpunkt Nachrichtenverarbeitung.
Mai 1991	Abschluß des Studiums mit der Diplomhauptprüfung.

Beruflicher Werdegang

1991 – 1995	Universität Hannover, Lehrgebiet Rechnernetze und Verteilte Systeme. Wissenschaftlicher Mitarbeiter, Betreuung von Vorlesungen und Übungen in den Bereichen Grundzüge der Informatik und Software-Engineering.
1995 – 1996	Universität Hannover, Lehrgebiet Rechnernetze und Verteilte Systeme. Gesamtprojektleiter für das Projekt “Regionales Testbed Nord im DFN”: 8 Teilprojekte in Bremerhaven, Hamburg, Hannover und Braunschweig, Erprobung von ATM-Weitverkehrsnetzen.
seit 1997	Universität Hannover, Regionales Rechenzentrum für Niedersachsen / Lehrgebiet Rechnernetze und Verteilte Systeme. Koordination für den Bereich F&E.