Universität Leipzig
Fakultät für Mathematik und Informatik
Institut für Informatik

Entwurf und Implementierung eines Sessionmanagement-Systems für das Multicast-Videokonferenzsystem „Visiphone“.

Diplomarbeit

Leipzig, April 1999

vorgelegt von
Horn, Sven

1 Einführung 1

2 Grundlagen Desktop-Videokonferenzing und Session-Control 3

2.1 Begriffsklärungen 3

2.2 Der H.320 Standard 4

3 Grundlagen Multimedia & Kommunikation 6

3.1 Das MJPEG Verfahren, Unterschiede zu MPEG 6

3.2 Das WAVE-Format (PCM) 8

3.3 Video-Overlays 9

3.4 Die PAL-Spezifikation 11

3.5 Das Windows Socket Interface (Winsock) 12

3.6 Die TCP/IP-Architektur 13

3.7 Realtime Transport Protokoll (RTP) 16

3.8 Multicast – Adressierung 17

3.9 Die Registry-Datenbank des Windows95-Betriebssystems 20

3.10 Begriffseinordnung „Frames“ 21

4 Eigenschaften und Parameter einer Auswahl vorhandener Videokonferenzsysteme 22

4.1 MBONE 22

4.1.1 Entstehung des MBONE 22

4.1.2 MBONE Video- und Audiotools 23

4.1.3 MBONE Whiteboards 24

4.1.4 Das MBONE Session Directory WindowsNT 25

4.1.5 MBONE Session Directory Version 2.2 UNIX-Portierung 27

4.1.6 Datenraten im MBONE 28

4.1.7 Der Time-To-Live (TTL) Parameter 29

4.1.8 Session Announcement Model und Session Description Protocol 30

4.1.9 MBONE-Session-Managements Übersicht 32

4.2 Microsoft Netmeeting und CUSeeMe 34

4.2.1 Einschränkungen der Systeme 34

4.2.2 CUSeeMe 34

4.2.3 Microsoft Netmeeting 36

4.3 VisitPhone 38

4.3.1 Ausgangspunkt 38

4.3.2 Systembeschreibung 38

4.3.3 Auftretende Datenraten 39

4.3.4 Untersuchung des VfW API‘s 40

4.3.5 Die verwendete Hardware 41

4.3.6 Das verwendete Multimedia Transportprotokoll 42

4.3.7 Der Aufbau von VisitPhone 43

4.3.8 Die Testanwendung unter Windows95 44

4.3.9 Spezielle Probleme im Hinblick auf Sessionmanagement Module ... 46

5 Konzeption des Sessionmanagements für Visitphone ... 49

5.1 Herausarbeitung allgemeiner Anforderungen ... 49

5.2 Prüfung der Verwendbarkeit von „sdr“ des MBONE als Sessionmanagement für Visitphone ... 50

5.3 Das Conference Control Channel Protocol (CCCP) als Grundlage des Visitphone - Sessionmanagements ... 52

5.4 Weitere Lösungsansätze ... 54

5.5 Die Realisierung einer Konferenzschaltung ... 55

5.6 MBONE Kompatibilität des Sessionmanagements ... 55

5.7 Einbindung einer Whiteboard-Anwendung ... 56

5.8 Kommunikation mit anderen Modulen und Shared-Applications ... 57

6 Implementierung des Sessionmanagementmoduls für Visitphone ... 59

6.1 Codierung des Sessionmanagementmoduls ... 59

6.2 Aufbau des Sessionmanagementmoduls ... 60

6.3 Die Kommandos des Befehlsinterfaces ... 63

6.4 Der Vorgang der Namensauflösung ... 63

6.5 Die Konferenzschaltung (Floor-Steuerung) ... 64

6.6 Der SDP-Dekoder ... 65

6.7 Der Konfigurationsteil ... 66

6.8 Eigenschaften und Testergebnisse ... 67

7 Zusammenfassung ... 69

8 Ausblick ... 70

Abbildungen und Tabellen ... 71

Abkürzungsverzeichnis ... 73

Literaturverzeichnis ... 74

Anhang ... 77

A1 Header des Audiodatenformats WAV ... 77

A2 Die MBONE Typologie Deutschlands ... 78

A3 Assigned Multicast Addresses ... 81

A4 Das Windows Socket Interface im Überblick ... 83

A5 Klassendefinition des Moduls „Sessionmm“ ... 84

A6 Spezielle Datenstrukturen ... 85

1 Einführung

Computer gehören heute bereits zum täglichen Leben wie Fernseh- oder Radiogeräte. Sie werden zur Verarbeitung verschiedenster Daten genutzt, wie zum Beispiel bei Einsatz von Textverarbeitungen und Tabellenkalkulationen, Steuerungen oder Archivierungen, um nur einige Anwendungen zu nennen.
Zunehmend tritt jedoch eine neuer Aspekt in den Vordergrund: Computer werden zur Kommunikation verwendet. Insbesondere durch die steigende Ausdehnung des Internets werden Dienste wie EMail, Newsgroups, oder Telefonieren über das Internet zunehmend populär.
Die logisch nächste Stufe zeichnet sich bereits ab: Eine Vernetzung von Arbeitsplätzen, die gleichzeitige Audio- und Videokommunikation direkt mit dem PC ermöglicht. Das wird häufig als (Desktop-) Videokonferenzing bezeichnet.
Die Verbesserung der höchstmöglich übertragbaren Bild- und Tonqualität dieser Videokonferenzsysteme ist ein wichtiges Ziel im Zuge dieser Entwicklung.
Die Qualität von Fernsehübertragungen stellt jedem Entwickler solcher Systeme einen einfachen Vergleichswert dar. Ein Ziel muß es daher sein, mindestens die Qualität dieser Übertragungen zu erreichen, besser noch diese zu übertreffen.
Unter diesem Aspekt wurde am Institut für Informatik der Universität Leipzig bereits 1996 mit Videoschnittkarten experimentiert. Durch die Kombination zweier dieser Karten war es am Institut erstmals möglich, Videosequenzen in Pal-Qualität vollduplex über ein ATM-Netz zu übertragen. Dieses Testsystem wurde der Öffentlichkeit unter dem Namen „Visitphone“ erstmals auf der Messe BIK'97 in Leipzig vorgestellt.
Dieser Hintergrund bildet den Ausgangspunkt der vorliegenden Arbeit. Im folgenden wird ihr Gegenstand erläutert:

Das Visitphone-System besteht aus einzelnen funktionalen Modulen (Netzwerkmodul, Videomodul, Audiomodul...). Zielstellung dieser Arbeit ist der Entwurf und die Implementierung eines Sessionmanagementsystems, welches die vorhandenen Module des Visitphone-Systems steuern kann. Dabei müssen die Einschränkungen und speziellen Parameter dieser Visitphone-Videoübertragungen beachtet werden, insbesondere die dabei auftretenden hohen Datenraten. Das System ist mit Erweiterungsmöglichkeiten auszustatten. Es ist zu ermitteln, ob das

Sessionmanagement des MBONE diese Anforderungen bereits erfüllt. Das System darf die Video- und Audioqualität der übertragenen Sequenzen nicht beeinflussen. Dazu werden bisher realisierte Lösungen, insbesondere das Sessionmanagement des MBONE, untersucht.

Das im Rahmen der Arbeit entwickelte und dargestellte Sessionmanagement-System wurde nach seiner Implementierung eingehend im Zusammenspiel mit vorhanden Visitphone-Komponenten getestet. Dazu wurde eine Testoberfläche zur Bedienung entworfen.
Das Sessionmanagement unterstützt IP-Multicast-Sendungen und ermöglicht Einzelverbindungen. MBONE-Kompatibilität wurde durch die Einhaltung vorhandener Spezifikationen erreicht.
Dazu wurden die im MBONE verwendeten Protokolle analysiert, ihre Standards ermittelt, und die verwendeten Mechanismen untersucht.
Die Arbeit ist wie folgt aufgebaut: Im Kapitel 2 sind allgemeine Informationen über Desktop-Videokonferenzsysteme dargestellt, insbesondere der H.320 Standard.
Darauf aufbauend wurden im Kapitel 3 für Videokonferenzing wichtige Multimedia-Grundlagen zusammengestellt. Des weiteren werden die Protokolle RTP, TCP und UDP und im MBONE verwendete Multicast-Mechanismen erläutert.
Kapitel 4 enthält Ergebnisse der Untersuchungen des MBONE und der im MBONE verwendeten Tools und Protokolle.
Als weitere Videokonferenzsysteme werden CuSeeMe und Netmeeting dargestellt. Darüber hinaus sind die vom Visitphone-System gegebenen Voraussetzungen, der Aufbau, auftretende Datenraten und Einschränkungen desselben enthalten.
Die Konzeption des entwickelten Sessionmanagementsystems legt Kapitel 5 dar. Ergebnisse der Implementierung, Aufbau und Funktionen sind im darauffolgenden Kapitel 6 zusammengestellt.
Abschließend wurden die Ergebnisse zusammengefaßt und ein Ausblick auf weitere Entwicklungen gegeben.

2 Grundlagen Desktop-Videokonferenzing und Session-Control

In diesem Kapitel sind grundlegende Informationen zusammengetragen, welche direkte Relevanz für ein Sessionmanagement-System besitzen. Die wichtigsten Begriffe werden definiert.

2.1 Begriffsklärungen

Verbindung

Als Verbindung wird der Zustand bezeichnet, in welchem mindestens zwei Endstationen bzw. Teilnehmer Daten austauschen. Dabei genügt es, wenn nur einer der Rechner sendet, und der andere nur empfängt. Ein Sonderfall sind sogenannte Multicastverbindungen (vergleiche Abschnitt 3.8). Multicastverbindungen können zwischen mehr als zwei Teilnehmern bestehen.

Vollduplex-Verbindung

In diesem Zustand ist es möglich, daß beide der Teilnehmer gleichzeitig senden. Beide Teilnehmer empfangen dann jeweils die Daten des anderen.

Verbindungsaufbau und Verbindungsende

Bei einem Verbindungsaufbau werden der Gegenstelle wichtige Parameter übermittelt, beispielsweise wer sendet, ob Videodaten gesendet werden, oder wurde ein Whiteboard gestartet. Das Verbindungsende wird als einfaches Signal übermittelt.

Session

Eine Session ist die Abstraktion einer aufrechterhaltenen Verbindung. Sie besteht vom Verbindungsaufbau bis zum Verbindungsende. Sie definiert Typ und Zweck der Verbindung. Die zum Verbindungsaufbau übermittelten Daten sind die Parameter der Session.

Sessionmanagement

Das Sessionmanagement ermöglicht das Starten einer Session (Verbindungsaufbau), deren Kontrolle und Aufrechterhaltung und die Steuerung des Ablaufs bis zum Verbindungsende.

Multicast-Sessionmanagement

Ein Multicast-Sessionmanagement hat zusätzlich zu den bei „Sessionmanagement“ genannten Punkten die Aufgabe, Verbindungen zwischen mehreren Teilnehmern (>2) zu ermöglichen, und dazu weiteren Teilnehmern ein Ein- und Austreten in die laufende Session zu ermöglichen.

Konferenz

Unter einer Konferenz wird eine Multicast-Session verstanden, in der der sendende Teilnehmer jeweils sein Recht zu sprechen an einen anderen Zuhörer, oder einen Konferenzmanager weitergeben kann. Alle Teilnehmer können die Session verfolgen.

2.2 Der H.320 Standard

Der H.320 Standard beschreibt die Richtwerte für Bildtelefonie bei Bandbreiten von p x 64 KBit/s, wobei 1 <= p <= 30 liegt. Das heißt, es werden vorwiegend Applikationen für schmalbandige Netze angesprochen.

Der Standard läßt verschiedene Qualitätsstufen zu. Systeme welche standardkonform sind, können also ganz unterschiedliche Qualitätsstufen aufweisen [KRAEMER].

Der Standard enthält im wesentlichen Verweise auf folgende Empfehlungen:

H.261	Videokompression
H.221	Kommunikation ("Frame structure for a 64 to 1920 KBit/s channel in audiovisual teleservices")
H.230	Steuerung ("Frame synchronous control and indication signals for audiovisual systems")
H.242	Verbindungsauf- und -abbau ("System for establishing communication between audiovisual terminals using digital channels up to 2MBit/s")
G.711	Audio in Telefonqualität, 48-64 KBit/s Narrowband
G.722	Audio in Stereoqualität, 48-64 KBit/s Wideband
G.728	16 Kbps Narrowband

Die für ein (Desktop-) Videokonferenzsystem ausschlaggebenden Qualitätsfaktoren sind:

- Bildauflösung
- Videoübertragungsrate (Frame Rate)
- Pre-/PostProcessing
- Motion Compensation
- Audiokodierung

Je nach unterstützen Features lassen sich für (Desktop-) Videokonferenzsysteme die folgenden drei Klassen identifizieren:

Klasse 1	minimale Anforderungen nach H.320 werden unterstützt
Klasse 2	wie 1. Klasse, dazu Unterstützung einiger optionaler H.320-Features
Klasse 3	wie 1. Klasse, zusätzlich Unterstützung aller optionalen H.320-Features

Nach H.320 müssen alle Hersteller zunächst nur die Anforderungen der Klasse 1 unterstützen. Hersteller, welche alle Features der Klasse 3 implementieren, stehen damit qualitativ über anderen Herstellern [KRAEMER].

Die dargelegten Empfehlungen sind unmodifiziert nur für Datenraten bis 2 MBit pro Sekunde und Bildauflösungen von 352 x 288 Punkten ausgelegt.

3 Grundlagen Multimedia & Kommunikation

Um ein Sessionmanagement für ein Videokonferenzsystem entwerfen zu können, ist es nötig, zunächst den Aufbau und die Funktionsweise der Kommunikationsapplikationen zu untersuchen.
Visitphone arbeitet ausschließlich mit dem Videostandard MJPEG, und dem Audioformat „Wave" (PCM). Diese und weitere Standards, wie PAL, Multicast oder RTP werden in diesem Kapitel erläutert.

3.1 Das MJPEG Verfahren, Unterschiede zu MPEG

Die JPEG Spezifikation wurde von der Joint Photographic Experts Group entwickelt. Im Unterschied zu GIF oder PGM ist JPEG ein verlustbehaftetes Verfahren.
MJPEG steht für Motion-JPEG, wobei JPEG einen Kompressionsstandard für Einzelbilder benennt, welcher auch von Zeichenprogrammen oder einem Webbrowser unterstützt wird. Der Standard MJPEG definiert eine Folge dieser Einzelbilder, es wird damit eine Komprimierung eines Videostromes realisiert [MILDE95].

Dagegen sind die in einem MPEG-Datenstrom vorliegenden (komprimierten) Bilder in drei verschiedene Typen einzuteilen:
1.: I-Frames (Intraframes): Dies sind Bilder, an denen ein Decoder mit einer Decodierung beginnen kann. I-Frames hängen von keinem anderen Bild ab (sie bringen die gesamte zur Decodierung nötige Information mit sich), sind jedoch schlechter komprimierbar als andere Frames.
2.: P-Frames (predicted frames): Dies sind Bilder, die für sich alleine nicht decodierbar sind, da sie als Bezugspunkt das zuletzt vorher im Datenstrom befindliche I-Frame benötigen, mit dessen Hilfe sie erzeugt wurden.
3.: B-Frames (bidirectionally predicted frames): Diese Frames sind, genau wie P-Frames, nicht eigenständig: Im Gegensatz zu P-Frames benötigen sie zwei Referenz-Bilder, ein vorhergehendes und ein nachfolgendes I- oder P-Frame. B-Frames liefern die beste Kompressionsrate.

Die Abfolge von I-, P- und B-Frames in einem MPEG-Datenstrom ist nicht in der MPEG-Norm festgelegt,
üblicherweise sind sie in folgendem Schema angeordnet: I B B B P B B B [NEUMANN96].

Günstig, im Gegensatz zu MPEG ist bei MJPEG der Umstand, daß zwischen den aneinandergereihten Einzelbildern keinerlei Abhängigkeiten bestehen. Dadurch wird bei Übertragungen erstens eine geringe Ende zu Ende Verzögerung erreichbar (es ist nicht nötig auf Frames zu warten), zweitens wurde es möglich, einfache Videoschnittkarten zu konzipieren, da der Aufwand an Elektronik durch das einfachere Verfahren sank.

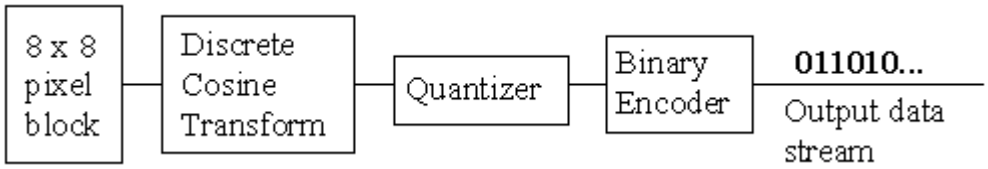


Abbildung 3.1.1: Darstellung des Datenflusses bei MJPEG

Die Schritte einer JPEG-Komprimierung zeigt Abbildung 3.1.1. Zunächst wird das Einzelbild in einzelne 8x8-Pixel-Blöcke zerlegt. Auf jeden dieser Blöcke wird eine zweidimensionale Diskrete Kosinus Transformation (DCT) angewandt. Darauf folgt eine Quantisierung der gewonnenen Daten. In diesem Schritt wird einerseits die Datenmenge auf bis zu 2% reduziert und andererseits treten Verluste (in Abhängigkeit von den gewählten Parametern) auf. Abschließend wird eine Huffmann-Codierung vorgenommen, womit noch einmal eine Reduzierung der Datenmenge erreicht wird. Bei der Huffmann-Codierung wird die Datenmenge nach Mustern durchsucht. Kurze Muster werden durch kurze Bitfolgen ersetzt, lange Muster durch längere Bitfolgen. Dieses Verfahren ist ähnlich dem bei Filekompressionen eingesetzten Zip-Verfahren. Die erreichbaren Kompressionsraten werden im Abschnitt 5.7 aufgezeigt [MILDE95].

3.2 Das WAVE-Format (PCM)

Ein wichtiges Audioformat, welches den MS-Windows-Betriebssystemen zugeordnet wird, ist das WAVE-Format. Ein Datenpaket digitalisierter Audiodaten besteht aus einem 40 Bytes langem Header worauf unmittelbar Daten folgen. Eine detaillierte Formatbeschreibung des Headers ist im Anhang A1 enthalten. Die Audiodaten werden nichtkomprimiert [NGUYEN98].
Die Digitalisierung erfolgt in Pulse-Code-Modulation (PCM). Beidem PCM-Verfahren wird die Amplitude des Audiosignals periodisch in einen 8-bit oder 16-bit Digitalwert umgewandelt.
Beim Vorgang der Digitalisierung wird das (stetige) analoge Eingangssignal abgetastet und auf einer Werteskala abgebildet (diskretisiert). Die Qualität der Digitalisierung von Schallhängtsomit von

- der Abtastfrequenz (Angabe in Hz) und
- vom Wertebereich der diskreten Skala (Angabe in Bit) ab.

Eine Erhöhung der Abtastfrequenz um mehr als 50% verringert einerseits den daraus resultierenden Abtastfehler, und ermöglicht es, noch höhere Frequenzen zu digitalisieren. Es entstehen keinerlei Informationsverluste, wenn die zu digitalisierende Funktion h(t) (das Audiosignal) mit einer Abtastfrequenz abgetastet wird, die mindestens doppelt so groß ist wie die höchste in h(t) enthaltene Frequenz (Theorem von Nyquist) [WAGEN97].
Somit ergeben sich verschiedene Qualitätsgruppierungen, welche in Tabelle 3.2.1 dargestellt sind.

Frequenz	Bits	Typ	Qualität
8000	8	Mono	Telefonqualität
11025	8/16	Mono/Stereo	Telefon-/Rundfunkqualität
22050	8/16	Mono/Stereo	Rundfunkqualität
44100	16	Stereo	CD – Qualität

Tabelle 3.2.1: Die verschiedenen Audioqualitäten bei Wave-Kodierungen

Das in Kapitel 4.3 vorgestellte Visiphone-System ist beispielsweise darauf ausgerichtet, Audiodaten in CD-Qualität zu übertragen. Dagegen verwenden Systeme wie CuSeeMe, welche für schmalbandige Verbindungen konzipiert wurden, für Audiodaten Telefonqualität [NGUYEN98].

3.3 Video-Overlays

Derzeit existieren für PC-Systeme drei Verfahren, einen Video Live-Datenstrom in einer Auflösung vom mehr als 320x200 Bildpunkten auf einen VGA Bildschirm aufzublenden: nachträgliches RGB-mischen (Overlay durch Y-Stecker), Videodatentransfer über FeatureConnectoren und PCI-Overlays [NGUYEN98].

Der Einsatz eines Y-Steckers sichert die Kompatibilität zu allen Grafikkarten und macht die Verwendung von IDE-Bus-Videokarten möglich.

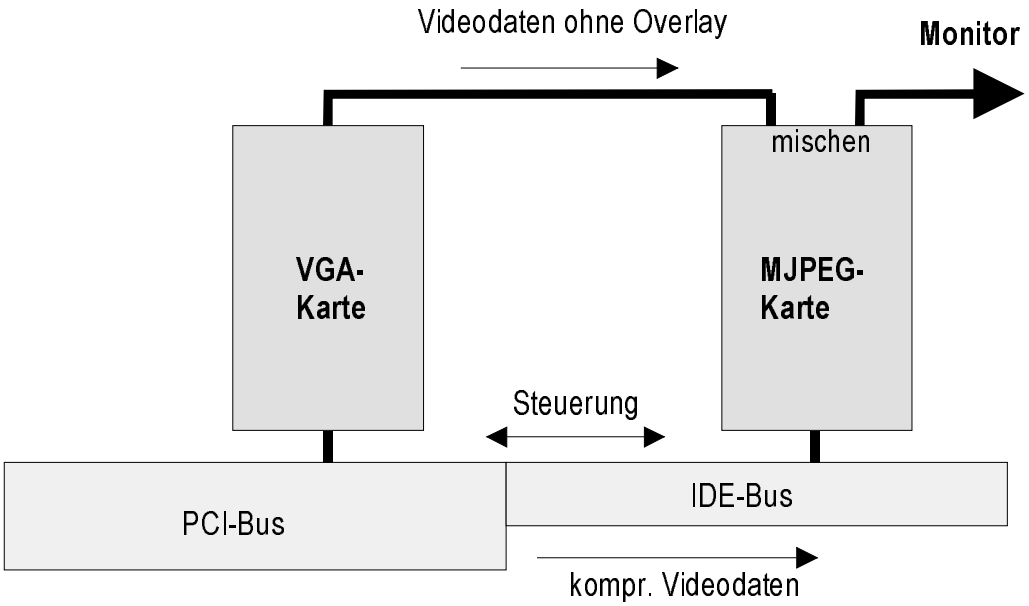


Abbildung 3.3.1: Konzeption des Verfahrens des nachträglichen RGB-mischens

Abbildung 3.3.1 zeigt die Konzeption dieses Verfahrens. Das Overlay wird durch Aufmischen des Live-Videos auf die RGB-Signale der VGA-Karte erreicht. Die Daten werden also nicht rechnerintern zur Grafikkarte übertragen, sondern direkt auf das Monitorsignal aufgeblendet.

Bei dieser Methode treten leichte Fehler in der Bildergonomie, insbesondere bei hohen Zeilenfrequenzen, auf. Denn durch die verlängerten Signalwege, zusätzlichen Bauteile usw. kommen Reflexionen, Übersprechen und andere Effekte zustande. Trotzdem werden noch heute, gerade im Bereich von 3Dfx-Karten, noch Y-Stecker-Overlays eingesetzt.

Die zweite Methode ist die Verwendung der FeatureConnectoren der Grafikkarten. Die Daten werden über ein zusätzliches Bandkabel, welches Grafikkarte und MJPEG-Karte verbindet (unabhängig vom Bussystem) übertragen. Diese hat sich aber aufgrund von ständigen Inkompatibilitäten und der Einführung des PCI-Bussystems nicht durchsetzen können.

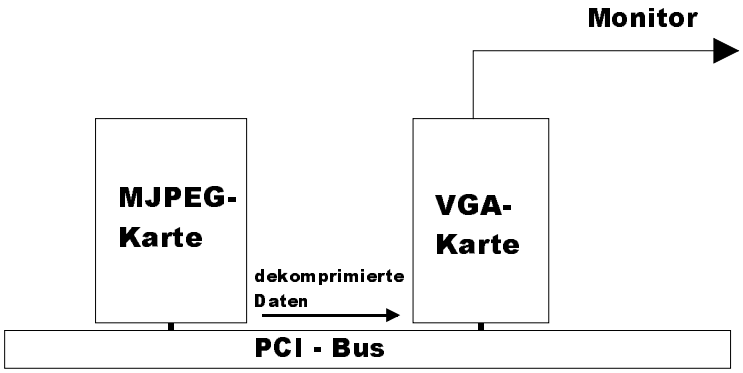


Abbildung 3.3.2: Datenfluß bei PCI Overlaytechnik

Die dritte Methode, das PCI-Overlay, verwendet den PCI-Bus zum Übertragen der dekomprimierten Videodaten. Dieser war als erstes Bussystem in der Lage, die hohen Datenraten bei der Übertragung eines Live-Videobildes zu ermöglichen (siehe auch Abschnitt 4.3.3). Als Schnittstelle wird die DirectDraw-Spezifikation von Microsoft verwendet. PCI-Overlay ist mit allen DirectDraw-kompatiblen Grafikkarten möglich.
Voraussetzung ist, daß Grafikkarte und MJPEG-Karte jeweils PCI-Bus Steckkarten sind.
Abbildung 3.3.2 zeigt die Anordnung der Karten und den Datenfluß der dekomprimierten Bilder.

3.4 Die PAL-Spezifikation

PAL (Phase Alternating Line) ist ein europäischer Videostandard mit dem Bildformat 4:3. Spezifiziert in CCIR-601 [CCIR]. Die Auflösung beträgt 625 Linien mit 833 Punkten. Sichtbar („active") sind davon 576 Linien mit je 720 Punkten. Die Bildwiederholrate beträgt 50 Hz. Abbildung 2.4.1 zeigt die verschiedenen Bildausschnitte im Vergleich.

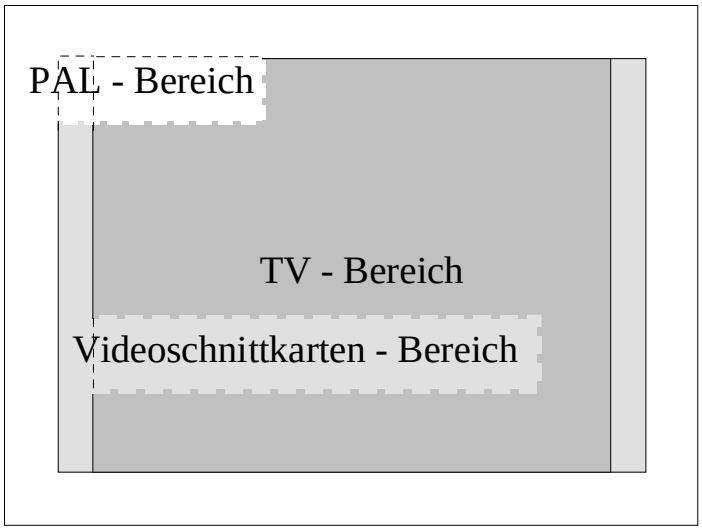


Abbildung 3.4.1: Die verschiedenen PAL-Bereiche

PAL nutzt eine Farbcodierung nach „YUV", ein Verfahren welches RGB Werte in einen Luminanz- und zwei Crominanzwerte wandelt:

$Y = 0.299R + 0.587G + 0.114B$ (Luminanzwert)

$U = 0.493 * (B - Y)$ (Crominanzwert)

$V = 0.877 * (R - Y)$ (Crominanzwert)

Videoschnittkarten digitalisieren einen Bereich von 768 x 576 Punkten. Die Bildwiederholrate beträgt 25 Hz. Dies entspricht qualitativ einem Fernsehbild, da dort ebenfalls nur 50 Halbbilder (= 25 Vollbilder) übertragen werden [CCIR].

Abgeleitet von der PAL-Spezifikation wurde die „PAL-Fernsehnorm" entwickelt.

Eingesetzt wird sie vorwiegend in Europa. Des weiteren existieren die Normen NTSC (US-Amerikanischer Bereich) und SECAM (östliches Europa). Tabelle 3.4.1 zeigt die Unterschiede dieser Standards [DIGVID97].

Parameter	NTSC	PAL	SECAM
Bildwiederholrate	60 Hz	50 Hz	50 Hz
Linien/Bild	525	625	625
Gamma-Wert	2.2	2.8	2.8
Audio Carrier	4.5 MHz	QAM	FM
Color Subcarrier	3.57 MHz	4.43 MHz	4.25 (+U), 4.4 (-V) MHz
Color Modulation	QAM	QAM	FM

Tabelle 3.4.1: Videonormen im Vergleich

3.5 Das Windows Socket Interface (Winsock)

Im Jahr 1981 wurden Sockets, das sind abstrahierte Kommunikationsendpunkte, als Schnittstelle für die Kommunikation lokaler Prozesse im 4.1c-BSD-System auf VAX-Anlagen eingeführt. 1986 erweiterte die Unix-Version 4.3 BSD die Socket-Schnittstelle für den Einsatz über TCP/IP und XNS-Protokollen. Diese Erweiterung der Berkeley Software Distribution ermöglicht Kommunikation zwischen verschiedenen Prozessen, egal ob sie lokal in einem Rechner laufen oder über das Netz verbunden sind. Dabei kommen alle Arten von physikalischen Datenwegen in Frage, z.B. Glasfaserkabel oder DFÜ-Verbindungen, völlig transparent für die Nutzer des Socketinterfaces.

Für Microsoft Windows wurde auf Grundlage dieses Berkley-Codes ein fast identisches Interface geschaffen, das Windows Socket Interface, kurz Winsock genannt.

Winsock liegt derzeit in der Version 2 vor. Vom orginalen Berkley-Code weichtes in der Kompatibilität nur bei einigen Funktionsbezeichnern ab.

Das Winsock-API bietet Zugriff auf viele verschiedene Netzwerkprotokolle: IP (TCP und UDP), IPX, AAL5 (ab Winsock 2), SNA, NetBios und NetBui. Alle Funktionen liegen gekapselt in der winsock.dll vor. Im Anhang (Teil 12.4) ist eine Übersichtszeichnung über diese Winsock-Zusammenhänge enthalten [NGUYEN98].

Während der Entwicklung von VisitPhone wurde zunächst mit dem Winsock 1.0 API programmiert. Dieses enthält jedoch einen gravierenden Fehler:
Das Empfangen von UDP-Frames, welche über eine Größe von 12 KByte hinausgehen, ist nicht möglich. Normalerweise sollten Frames bis 64 KByte problemlos zu empfangen sein. Dieser Effekt ist unabhängig vom verwendeten Rechnertyp oder der Netzkarte und deren Treibern. Im Winsock2-API ist dieser Fehler behoben worden.

Winsock2 besitzt eine Reihe wesentliche Neuerungen [FRANKE98]:

Protokollunabhängige Namensauflösung (Name Resolution): In Winsock2 wurden entsprechende Funktionen implementiert, um Abfragen zur Namensauflösung verschiedener Protokolle zu stellen.
Overlapped (Asynchronous) I/O-Operationen: Sockets können in einem Overlapped-Modus betrieben werden, d.h. empfangene Nachrichten werden sofort bei Erhalt in dem von der Anwendung bereitgestellten Puffer geschrieben, unter Umgehung von internen Pufferspeichern und Kopieroperationen. Die betreffenden I/O-Operationen kommunizieren dabei über den Aufruf entsprechender Ereignisroutinen.
Quality-of-Service: Winsock2 stellt Strukturen zur Verfügung, mit denen Anwendungen verschiedene Serviceparameter des Transportdienstes aushandeln können.
Protokollunabhängige Multicast- oder Multipoint-Operationen
siehe Abschnitt Multicast Adressierung (3.8)

Für die Entwicklung des Visitphone Sessionmanagements, ebenso für das Verständnis der MBONE Architektur, ist nur die Protokollfamilie TCP/IP (TCP und UDP) von Belang.

3.6 Die TCP/IP-Architektur

Unter dem Begriff TCP/IP faßt man die Protokolle UDP (User Datagram Protocol) und TCP (Transmission Control Protocol) zusammen, welche beide auf IP (Internet Protocol) aufsetzen.

Durch die Verwendung von IP war es möglich, ein globales Netzwerk, das Internet, mit Millionen von Rechnern über ein einheitliches Protokoll kommunizieren zu lassen. Anwendungen wie Webbrowser, Videokonferenzen, Dateiübertragungen nutzen IP. Zum Transport der Daten wird ein Transportprotokoll, entweder UDP oder TCP verwendet. Ihr Aufbau wird in folgenden erklärt.
Jeder Rechner in einem IP-Netz besitzt eine im Netz nur einmal vorkommende IP-Adresse. Eingeteilt werden sie in einzelne Klassen (Siehe Abschnitt 3.8). Abbildung 3.6.1 zeigt den Aufbau eines IP-Paketes. Eine IP-Adresse umfaßt jeweils 32 Bit.

Abbildung 3.6.1: Aufbau eines IP-Paketes

Geroutet wird IP mittels der Protokolle RIP, OSPF, EIGRP. Wenn ein Rechner oder Router eine Verbindung zu einem anderen Gerät aufbauen will, muß er dessen physikalische Adresse kennen. Diese kann er über das ARP (Adress Resolution Protocol) erfragen. Der umgekehrte Vorgang, das Ermitteln der IP Adresse nach einer physikalischen Adresse wird mit dem RARP (Reverse Adress Resolution Protocol) ermöglicht.

UDP ist ein verbindungsloses Protokoll, welches das Senden und Empfangen von Datenpaketen bis zu einer Größe von 64 KByte ermöglicht. Es ist kein sicheres Protokoll, dh. es kann vorkommen, daß Daten verbrengehen, ihre Reihenfolge vertauscht wird, oder sie dupliziert oder verfälscht werden. UDP-Pakete können als Multicastpakete übertragen werden, (siehe Abschnitt 3.8). Abbildung 3.6.2 zeigt den Aufbau eines UDP-Paketes.

0	1	2	3	Bytes
Absender-Port		Empfänger-Port		
Länge		UDP-Prüfsumme		
Daten				

Abbildung 3.6.2: Aufbau eines UDP-Paketes

TCP ist ein verbindungsorientierter Dienst, welcher ebenfalls Datenpakete bis zu einer Länge von 64 KByte überträgt. Es ist ein sicheres Protokoll. Daten werden stets korrekt übertragen. Intern wird dies über Antwortpakete realisiert. Im Fehlerfalle werden die Daten erneut übertragen. Es ist sichergestellt, daß die Reihenfolge der Pakete eingehalten wird, und keine Pakete dupliziert werden.

ISO Schicht	Bezeichnung	TCP/IP Protokolle	
7	Applikation	Applikationen Protokolle	*Telnet*
6	Präsentation		*FTP(Dateiübertragung)*
5	Sitzung		*RTP*
			HTTP(Browser)
4	Transport	Transport Protokolle	*TCP*
			UDP
3	Netzwerk	Internetwork Protokolle	*EGP, RIP*
			IP
			ARP
			RARP
2	logische Verbindung	Network Access Protokolle	*Ethernet*
			Tokenring
1	physikalische Verbindung		*FDDI*
			ATM

Abbildung 3.6.3: Die Zuordnung der Protokolle IP,UDP und TCP zu den Schichten des OSI-Modells

Die genannten Protokolle IP, TCP und UDP können in das ISO-OSI-Modell, welches 7 Schichten umfaßt, eingeordnet werden. Abbildung 3.6.3 zeigt diese Zuordnung [NGUYEN98].

3.7 Realtime Transport Protokoll (RTP)

Eines der wichtigsten Protokolle welches auf UDP aufsetzt, besonders in Bezug auf MBONE-Anwendungen, ist RTP. Es wird in einer der Arbeiten der IETF (Internet Engineering Task Force) spezifiziert [RFC1889].
Vorrangig wird es zusammen mit UDP genutzt, möglich ist ebenfalls die Verwendung von AAL5 oder anderen Stacktypen. RTP kann mit verbindungslosen und verbindungsorientierten Protokollen eingesetzt werden.
Konzipiert wurde es für Übertragungen von Video- und Audiodaten in Echtzeitanwendungen, beispielsweise Konferenzapplikationen. Inzwischen wird es vorwiegend im MBONE eingesetzt (siehe Kapitel 4.1).
RTP ist als Protokoll nicht im Systemkernel implementiert, sondern Teil der Applikation. Unterteilen kann man RTP in einen Datenteil und einen Kontrollteil. Bei Nutzung von UDP bedeutet dies die Aufteilung auf zwei verschiedene Ports.
RTP Datenpakete bestehen aus einem 12 Byte umfassenden Header und nachfolgend dem „Paybad", dh. den zu transportierenden Daten.
Der Header enthält [RFC1889]:

-Version: (2bit) (V)	steht für die verwendete RTP Version
-Padding: (1bit) (P)	Flag, gibt an ob dem Header noch zusätzliche Bytes hinzugefügt wurden
-Extension: (1bit) (X)	Flag, ähnlich Padding , beschreibt ob noch genau eine „Header-extension" angehängt wurde
-CSRC count: (2bit) (CC)	Anzahl der dem Header nachfolgenden „contributing source (CSRC) identifiers"
-Marker: (1bit) (M)	dient zum Markieren von Frameenden und anderen datenabhängigen Ereignissen
-Paybad type: (7bit)	identifiziert den Typ des Paybads zB. JPEG oder GSM
-Timestamp: (32bit)	ein Zeitstempel

-Sequence number: (16bit) Sequenznummer zur Sicherung der Paketreihenfolge bei gleichen Zeitstempeln oder Verlusten

-Synchronization source (SSRC) identifier: (32bit) im Netz einzigartiger, zufällig generierter Wert zur Identifkation des Senders.

In Abbildung 3.7.1 wird die Anordnung dieser Parameter noch einmal grafisch dargestellt.
Der Kontrollteil von RTP wird als RTCP (Realtime Control Protocol) bezeichnet. Er ermöglicht [RTP95]:

- Quality-of-Service Vereinbarungen zwischen den Hosts
- Syncronisation
- Identifkation
- und Session-Skalierung (zB. das Senden von „polling"-Paketen in Abhängigkeit zur bestehenden Netzlast)

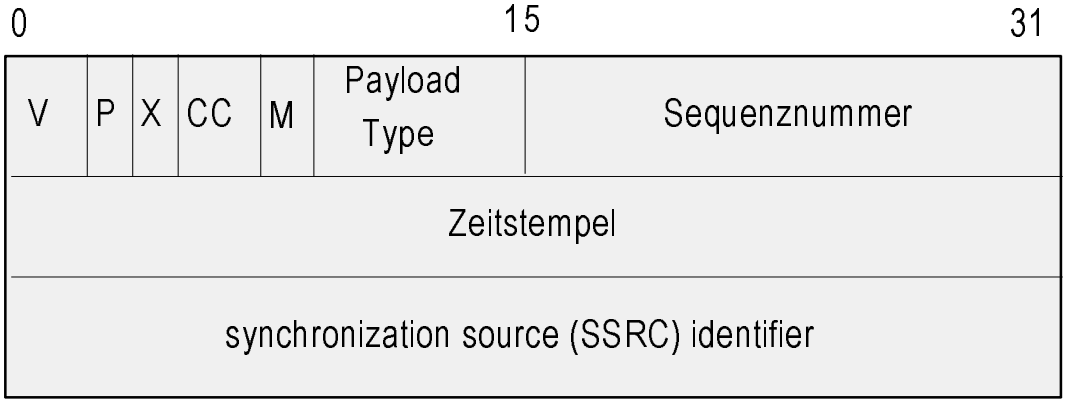


Abbildung 3.7.1: Aufbau eines RTP-Headers

3.8 Multicast – Adressierung

Abbildung 3.8.1 zeigt den unterschiedlichen Datenfluß bei Verwendung von Multicast im Vergleich zu Unicast.
Der Unterschied des Multicastroutings zum herkömmlichen Punkt zu Punkt – Verfahren besteht darin, daß die Pakete einmal abgesendet werden und von

Multicast-Routern zunächst dupliziert, und nachfolgend auf die weiteren Pfade zu den Endpunkten kopiert werden.
Geroutet werden diese Pakete mittels Internet Group Management Protocol (IGMP), Distance Vector Multicast Routing Protocol (DVMRP), Multicast Extension to Open Shortest Path First (MOSPF) und Protocol-Independent Multicast (PIM).

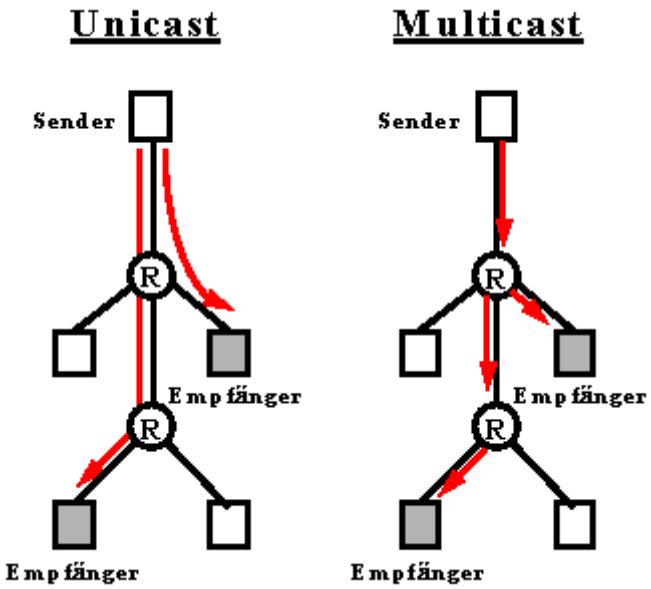


Abbildung 3.8.1: Schematische Darstellung des Datenflußes bei Unicast- und Multicastsendungen

Unicast Pakete müssen dagegen für jeden Empfänger einzeln abgeschickt werden. So kann es vorkommen, daß Pakete identischen Dateninhalts mehrfach über eine Verbindung gesendet werden [MBONE2].

Multicastadressen werden als Gruppen bezeichnet. Endpunkte bzw. Hosts können in eine dieser Gruppen eintreten und damit die Pakete dieser Gruppe empfangen. Dies ist mittels der, den herkömmlichen Socketfunktionen hinzugefügten, Funktionen JoinGroup(Adresse) und LeaveGroup(Adresse) auszuführen [NGUYEN98].

Dabei können jene Endhosts an diese oder an eine andere Gruppe senden. Dazu ist es nicht notwendig, Mitglied dieser Gruppe zu sein.
Es ist möglich, daß ein Host gleichzeitig Mitglied in verschiedenen Multicastgruppen ist, und deren Pakete empfängt.

Für Multicastgruppen wurde ein spezieller Adressbereich reserviert:

224.0.0.0 – 239.255.255.255. Diese Zuteilung nimmt die Internet Assigned Numbers Authority (IANA – Behörde für die Vergabe von Internet-Adressen) vor.
Multicastadressen sind Class D Adressen. Class D Adressen beginnen mit der Bitfolge 1110. Der Unterschied zu den Adressfamilien A,B,C wird in Abbildung 3.8.2 dargestellt [FAIL96].
Einige dieser Adressen sind reserviert, andere werden als ‚well known' bezeichnet. Die durch ihre Verwendung im MBONE reservierten Adressen sind in Tabelle 3.8.1 aufgeführt [MBONE2].

224.0.0.0 - reserviert
224.0.0.1 - 224.0.0.255 reserviert für Routingprotokolle
224.2.0.1 - Default VAT (audio) Addresse
224.2.1.1 - Default NV (video) Addresse
224.2.127.255, Port 9876 - Session Directory (SD) Addresse
224.8.8.8 - Default IVS (audio/video) Addresse
224.2.253.119 - Radio Free VAT

Tabelle 3.8.1: reservierte Multicastadressen

Eine komplette Liste aller Multicastadressen, welche zusätzlich einige nicht dem MBONE zugeteilten Adressen enthält, ist im Anhang A3 enthalten. Diese Adressfamilie (siehe Anhang) wird in RFC 1700 spezifiziert.

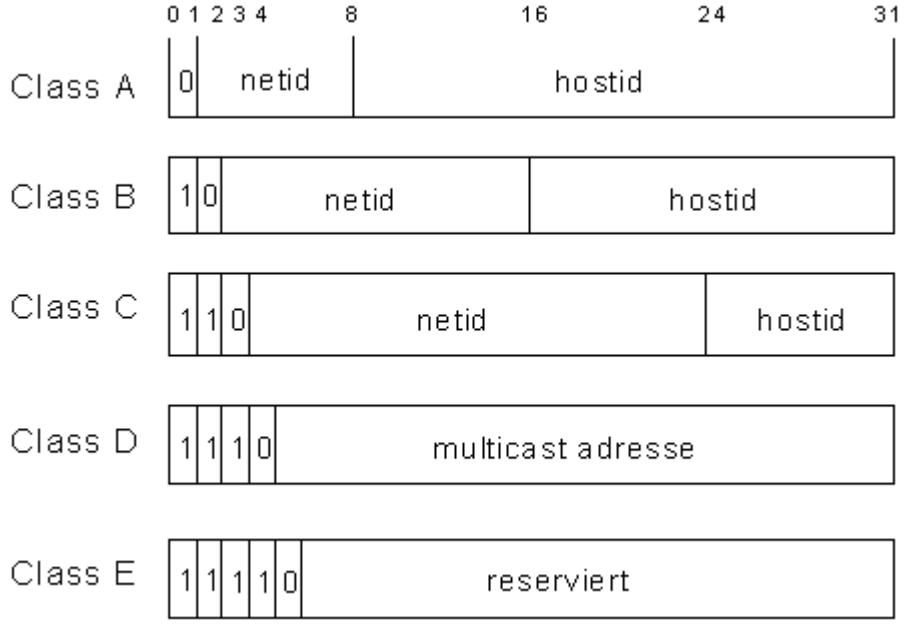


Abbildung 3.8.2: Darstellung der unterschiedlichen Internet Adressfamilien

Der Bereich 224.0.1.27 – 224.0.1.255 gilt als frei und kann von Anwendungsprogrammen genutzt werden.
Direkt auf diesen UDP-Stack wird bei MBONE-Tools das im Abschnitt 3.7 erläuterte RTP aufgesetzt [MBONE2].

3.9 Die Registry-Datenbank des Windows95-Betriebssystems

Die Registry-Datenbank unter Windows95 (oft kurz „Registry" genannt), dient der Speicherung aller system- und nutzerspezifischen Einstellungen der installierten Programme. Diese Datenbank hat die INI-Dateien von Windows 3.x abgelöst.
Applikationen können in ihr beliebige Informationen ablegen, welche auch nach einem Neustart des Systems der Anwendung noch zur Verfügung stehen sollen. Zugegriffen wird auf sie mittels folgender C++ Befehle:

```
RegSetValue( HKEY ,  //Hauptschlüsselbezeichnung
        subkeyname,  //Subschlüsselbezeichnung
        REG_SZ,  //Datentyp
        pbuffer,  //die zu schreibenden/lesenden Daten
        size      //ihre Länge );
```

und `RegQueryValue(HKEY, subkeyname, pbuffer, size );`

Die Daten der Registry sind bestimmten Schlüsseln zugeteilt. In der obersten Ebene sind das die Hauptschlüssel:

- HKEY_CLASSES_ROOT
- HKEY_CURRENT_USER
- HKEY_USERS
- HKEY_LOCAL_MACHINE
- HKEY_CURRENT_CONFIG
- HKEY_DYN_DATA
-

Diesen untergeordnet sind jeweils, von den Applikationen selbst festzulegende, Subschlüssel (Subkeys).

Visitphone verwendet HKEY_LOCAL_MACHINE mit dem Subkeys 'Visitphonexx“ (xx = 1..9).

3.10 Begriffseinordnung „Frames“

In Videokonferenzsystemen treten verschiedenste Arten von Datenpaketen auf. Oft werden diese als Frames bezeichnet.
Audioframes sind Datenpakete in unkomprimierter Form, welche von einem Wave-Device abgegriffen werden, und in dieser Form unverändert über das Transportprotokoll verschickt werden.
Videoframes werden in unkomprimierte und komprimierte Frames unterteilt. Unkomprimierte Videoframes treten zwischen Video- und VGA-Karte auf dem PCI-Bus auf. Komprimierte Videoframes werden dagegen im Speicher gehalten, kopiert und wie die Audiodaten über das Transportprotokoll übertragen.

4 Eigenschaften und Parameter einer Auswahl vorhandener Videokonferenzsysteme

4.1 MBONE

Der Begriff Multicast-Videokonferenzing wird oft mit dem Begriff des MBONE assoziert. Wo Gemeinsamkeiten liegen, oder auch Unterschiede und welche Konzepte für Session-Control existieren, beschreibt dieses Kapitel.

4.1.1 Entstehung des MBONE

Durch die zunehmende Nutzung des Internets als Kommunikationsmedium werden ständig neue Dienste populär. Durch immer größer werdende Bandbreiten wird der Austausch von Multimediadaten komfortabel und ohne speziellen Hardwareaufwand realisierbar. Heute zählen das Versenden von Audiodaten (Bsp: Radiosender, RealAudio) oder Videodaten zu den häufig genutzten Möglichkeiten. Durch die Entwicklung des MBONE wurde audiovisuelle Kommunikation unter Nutzung von Multicastverbindungen im Internet möglich.
Internet-Netzwerkanwendungen basierten vor der Einführung des MBONE auf Punkt zu Punkt Verbindungen, dh. jedes gesendete Paket hatte nur einen Empfänger, oder Broadcasts, dh. das Paket wird an alle empfangsbereiten Empfänger versendet. Das führte jedoch zu einer immensen Netzbelastung sobald viele Nutzer einen identischen Dienst nutzten. In einem solchen Fall wurden Datenpakete mit gleichem Inhalt mehrfach an verschiedene Endpunkte versendet.
Mit der Einführung des MBONE wurde eine neue Verfahrensweise angewandt. Pakete werden nur einmal an eine Multicast-Adresse verschickt, welche nicht mehr nur für einen einzelnen Endpunkt steht, sondern für eine Gruppe von Nutzern. Das Paket wird von den zwischenliegenden Routern kopiert (siehe Abschnitt 3.8 „Multicast Adressierung“). Dazu sind Router erforderlich, welche ein Multicast-Routingprotokoll beherrschen (Siehe Abschnitt 3.6). Historisch gesehen wurde diese Aufgabe zunächst von UNIX-Workstations übernommen, doch heute sind auch alle kommerziellerhältlichen Router wie zB. Cisco 7000 dazu fähig. Daraus entstand ein weltumspannendes Netz, das MBONE (Multicast Backbone). 1992 wurde es erstmalig zur Übertragung der IETF – Konferenz in Bild und Ton genutzt. Eine Übersichtskarte über die Typologie des heutigen MBONE (DE) ist im Anhang A2 enthalten [WEBMBONE].

4.1.2 MBONE Video- und Audiotools

Um Video und Audiodaten über das Mbone empfangen und anzeigen zu können, werden spezielle Programme benötigt. Häufig werden ‚vic' und ‚vat' genutzt. ‚vat' steht für Visual Audio Tool. Der Bezeichnung ‚vic' wird keine Langschrift zugeordnet, sie entstand in Anlehnung an ‚Video Conferencing'. Daneben existieren eine Reihe anderer Tools, zum Beispiel ‚rat'. Abbildung 4.1.1 und Abbildung 4.1.2 zeigen die Darstellungen der Tools „vic" und „vat".
Die Anwendungen sind in der Lage, Multicast-Pakete zu versenden. Multicast Pakete sind UDP-Pakete mit einer Multicast – Zieladresse (siehe auch Abschnitt 3.8). Zum Übertragen von Echtzeit-Multimediadaten ist ein zusätzliches Protokoll notwendig. Die genannten Tools nutzen das RTP (Realtime Transport Protocol) (RFC 1889). Es wurde im Abschnit 3.7 erläutert.

Abbildung 4.1.1: Bildschirmdarstellung des Videotools VIC

Das Videotool ‚vic' wird zum Senden und Empfangen von Videodaten genutzt. In der Version 2.8 unterstützt es die Videomodi: h261, jpg, celb, nv und nvdct. Es können ein bis dreißig Frames pro Sekunde verschickt werden.
Das Tool ‚vat' unterstützt die Audiomodi PCM, PCM2, PCM4, DVI, DVI2, DVI4, GSM und LPC4. Es kann zusätzlich Testtöne generieren, um Verbindungen zu überprüfen.

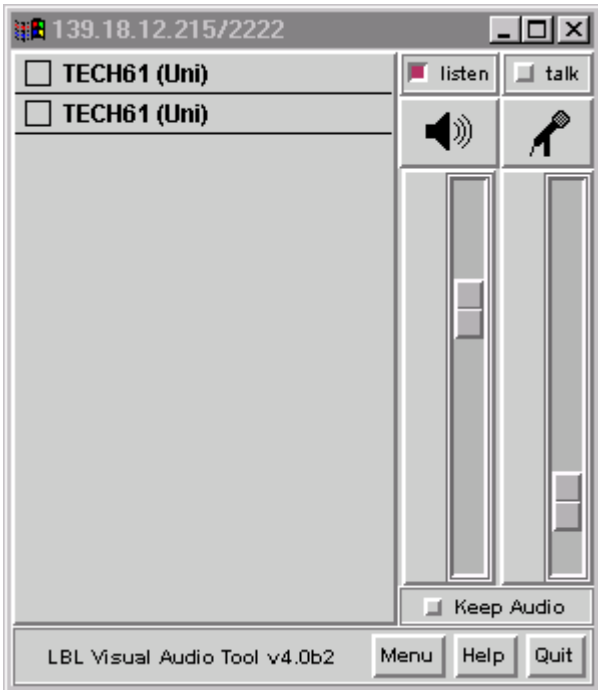


Abbildung 4.1.2: Bildschirmdarstellung des Audiotool s VAT

4.1.3 MBONE Whiteboards

Eine Whiteboard-Anwendung trägt zur Akzeptanz eines gesamten Videokonferenzsystems bei, beispielsweise in betriebswirtschaftlichen Anwendungsbereichen.
Dies ist ein Programm, welches mehreren Anwendern ermöglicht, gleichzeitig ein Grafikdokument zu bearbeiten. Im MBONE werden zB. die Tools wb, wbd, mashMB und MDesk eingesetzt. Für WindowsNT und Windows95 existieren Portierungen der Systeme wbd und mashMB. Die Programme sind jedoch untereinander und zur UNIX-Version „wb“ inkompatibel. Anwender welche miteinander an einem identischen Whiteboarddokument arbeiten möchten, müssen also stets identische Versionen von Whiteboards installiert haben.
Von einem Sessionmanagementsystem aus können alle genannten Whiteboardapplikationen (wb, mashMB, MDesk, mbd) durch einen Aufruf über die Kommandozeile, mit entsprechenden Parametern gestartet werden. Diese sind in der Angabe der IP Adresse und des Ports äquivalent zu den Tools vic und vat (siehe Syntax in Abschnitt 4.1.3) [MBONEDR].

4.1.4 Das MBONE Session Directory Windows NT

Das von Arlie Davis entwickelte Tool stellt die derzeit neuste Version des MBONE Session Directorys unter Windows NT, dar. Nach dem Programmstart werden im MBONE aktive Sessions erkannt, dekodiert (siehe Abschnitt 4.1.8) und aufgelistet. Abbildung 4.1.3 zeigt eine Auflistung dieser Sessions. Daten werden über die festgelegte Multicastadresse 224.2.127.254, Port 9875 empfangen. (Siehe Abschnitt „Multicast Adressierung 3.8).
Die Kodierung der Daten erfolgt nach dem im Abschnitt 4.1.8 erläuterten Protokollen SAP und SDP.

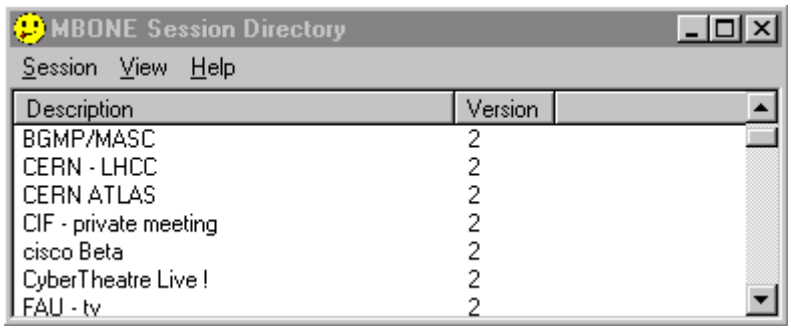


Abbildung 4.1.3: Bildschirmdarstellung des SDR-Tool s für Windows

Zur Registration und Identifikation des Benutzers ist die Angabe der eigenen Email-Adresse nötig, ebenso ein Benutzername. Diese Daten werden in der Registry-Datenbank des Windows-Betriebssystems zwischengespeichert. Dadurch sind diese Angaben nur einmal, beim ersten Programmstart notwendig. Video- und Audiotools werden über einen automatischen Aufruf auf der Kommandozeile gestartet. Je nach Art der Session ebenso das Whiteboard-Tool. Die Syntax des Aufrufs beschreibt folgendes Beispiel:
„vic -t30 139.18.12.214/1024 “

Erklärung: „139.18.12.214“ beschreibt die IP-Adresse des Zielrechners und „1024“ gibt den verwendeten Port an. „-t30“ setzt den Standard-TTL-wert der Sendung auf 30. Mit der beschriebenen Syntax können ebenfalls die Programme vic, rat und wb gestartet werden. Die genannte Vorgehensweise ermöglicht das einfache Austauschen von Tools, unhabhängig vom verwendeten Sessiondirectory.

Weiterhin werden vom Programm Email-Adresse, Telefonnummer und Homepage-Location des sendenden Teilnehmers erkannt und zwischengespeichert. Bei Bedarf stehen diese dem Nutzer zu Verfügung (zB: zum Start eines Webbrowsers). Es ist möglich, EMail und WWW-Applikationen aus dem Programm heraus zu starten.

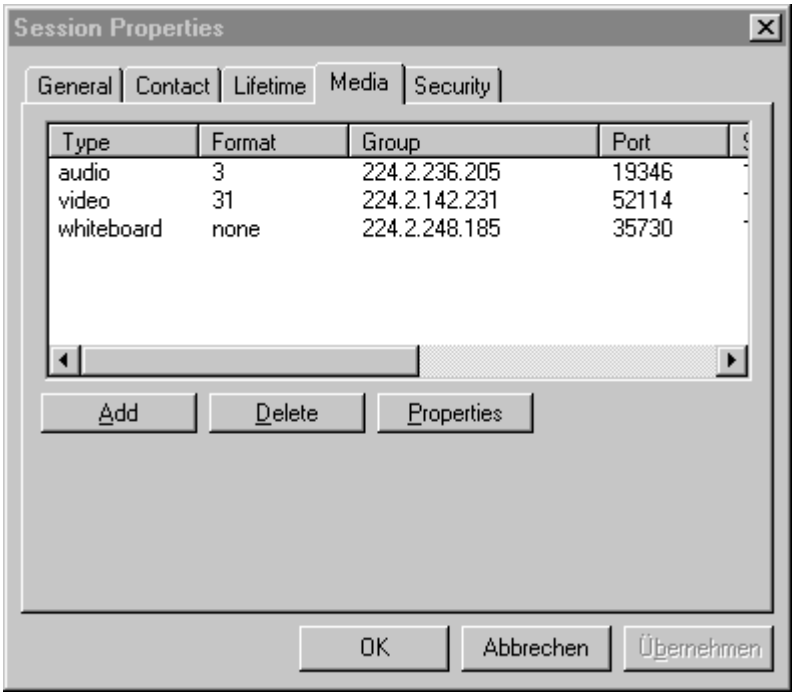


Abbildung 4.1.4: Bildschirmdarstellung der spezifischen Daten einer Session

Die einzelnen Datentypen wie Audio, Video oder Whiteboard werden jeweiligen Multicast-Adressen und Ports zugeordnet. Video- und Audiodaten können auf getrennten Adressen oder Ports versendet werden.
Jeder Session kann ein ‚Encryption-Key' zugeordnet werden, welcher das Empfangen und Entschlüsseln von privaten Sessions möglich macht.

Um Audio oder Videodaten zu empfangen oder zu senden, werden die Tools „rat", „tat", „vic" oä. gestartet. Diese sind vom Sessionmanagement unabhängig, also eigenständige Programme. Die Abbildungen 4.1.4 und 4.1.5 zeigen die Bildschirmdarstellungen der Einstellmöglichkeiten einer Session. Konfigurierbar sind Port, Adresse und Format der jeweiligen Datenströme. Der Parameter „Scope" gibt den TTL-Wert (siehe Kapitel 4.1.7) der Session an.

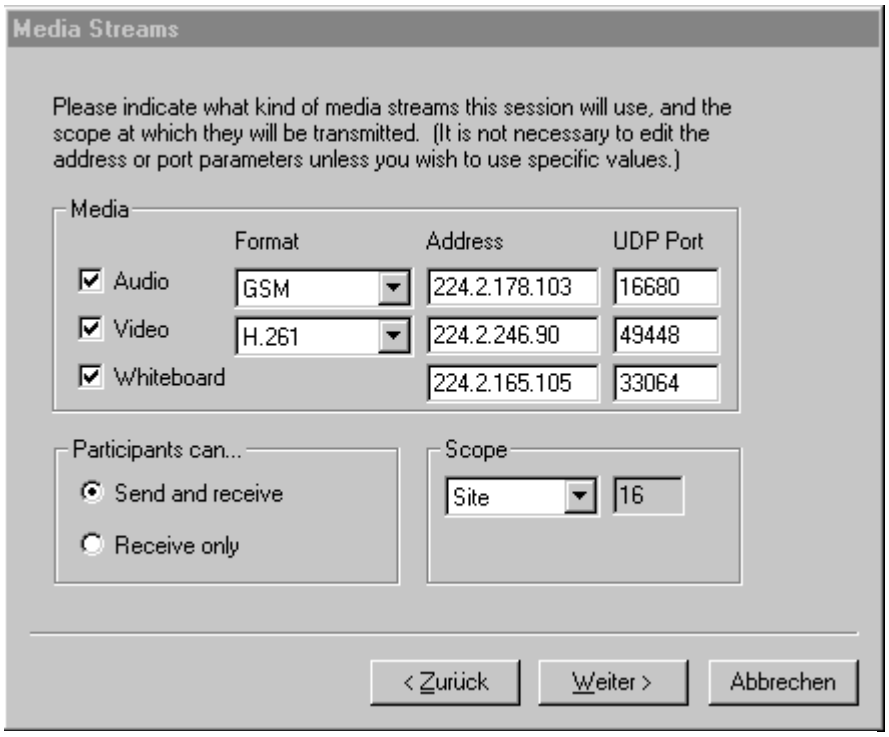


Abbildung 4.1.5: Die Bildschirmdarstellung der Stream Einstellungen

4.1.5 MBONE Session Directory Version 2.2 UNIX-Portierung

Abbildung 4.1.6 zeigt die Benutzerschnittstelle der UNIX Portierung des Session Directorys 2.2. Dieses Toolbietet die identische Funktionalität zu der im vorigen Abschnitt beschriebenen Applikation. Zusätzlich wurde ein Kalender implementiert.

Abbildung 4.1.6: Die UNIX Portierung

Unterschiede ergeben sich nur aus der Tatsache, daß diese Applikation auf dem ursprünglichen UNIX-Quellcode basiert, jedoch die WindowsNT-Version aus einer Umsetzung hervorging. Die Internen Strukturen unterscheiden sich daher nur in der Programmiersprache, nicht in Funktionsweise oder Aufbau.

4.1.6 Datenraten im MBONE

Über die Verbindungen des MBONE werden unterschiedlichste Daten verschickt. Audio-Daten werden überwiegend von 64 KBit/s, was unter Beachtung des Paketoverheads ca. 68-78 KBit/s entspricht, auf weniger als 36 KBit/s komprimiert:

- 36 KBit/s bei ADPCM (Adaptive Pulse Code Modulation)
- 17 KBit/s bei GSM (General Special Mobile)
- 9 KBit/s bei LPS (Linear Predictive Coding).

Zum Vergleich sind in Tabelle 4.1.1 einige andere Datenraten aufgeführt, welche zeigen wie gering die hier auftretenden Netzbelastungen, im Gegensatz zu beispielsweise Visitphone-Audiodaten sind [ZINKE].
Videodatenraten werden im allgemeinen durch Software-Kompression auf einen Wert unter 128 KBit/s reduziert. Höhere Datenraten sollten nur mit kleiner TTL (Siehe Abschnitt 4.6) in einem lokalen Netz versendet werden.
Zum Vergleich mit den Visitphone-Datenraten sind in Tabelle 4.1.1 und 4.1.2 die verschiedenen Werte aufgeführt.

Medium	Codeart	Datenrate
Visitphone PAL	MJPEG	45 MByte/s
Visitphone ¼ PAL	MJPEG	12 MByte/s
MBONE vic	MJPEG, H261, nv, cellb	10KBit/s – 3MBit/s
ISDN Bildtelefon	H.261 FCIF/QCIF, H.263	56KBit/s

Tabelle 4.1.1: Bitraten bei Videoübertragungen

Grafik und Textdaten, wie sie beispielsweise von Whiteboard-Applikationen erzeugt werden, bleiben unter 5 bis 6 KBit/s.

Medium	Codeart	Bandbreite	Datenrate
Musik-CD (Visitphone)	PCM	20kHz	2*16bit*44,1kHz=1,4MBit/s
DAT	Longplay PCM	15kHz	2*16bit*32kHz=1,024MBit/s
DCC	MPEG LayerI	15kHz	2*192 KBit/s
DAB	MPEG LayerII	20kHz	192 KBit/s
ISDN-Telefon	G711	3,4 kHz	1*8bit*8kHz=64 KBit/s
GSM-Mobilfunk	RELP	3,4kHz	13,2 KBit/s
US-FS-1016	CELP	3,4kHz	4,8 KBit/s
US-FS-1015	LPC-10E	3,4 kHz	2,4 KBit/s

Tabelle 4.1.2: Bitraten bei Audioübertragungen

Chat-Applicationen und ähnliche Anwendungen erreichen kaum 1 bis 2 KBit/s [KUMAR96].

4.1.7 Der Time-To-Live (TTL) Parameter

Jeder Session wird ein Time-To-Live (TTL) Parameter zugeordnet. Dieser Parameter beschreibt die Reichweite der jeweiligen Session.
Die Multimedia-Applikationen senden Datenpakete mit einem festen TTL-Wert. Jeder Router, welcher diese Pakete kopiert, erniedrigt diesen. Erreicht er Null, wird das Paket nicht mehr weitergeleitet.
Da das MBONE viele Subnetze enthält, welche durch Tunnel verbunden sind, muß man jedem existierenden Tunnel einen Wert zuordnen, der etwa der Distanz entspricht, welche er überbrückt. Somit hat man festgelegt:

Low Speed Tunnels: 128
Intercontinental: 64
International (in Europa): 48
zwischen Institutionen: 16-32
innerhalb von Institutionen: 1-16

Dieser Wert wird von den Tunnel-Rechnern bei jedem von innen transportierten Datenpaket vom TTL-Wert desselben abgezogen.
Diese Vorgehensweise ermöglicht es, räumlich begrenzte Sessions zu aktivieren. Beispielsweise ist es möglich, für den Raum Deutschland Bild und Ton eines Fernsehsenders zu übertragen, ohne die eventuell schlechter ausgebauten Netze der angrenzenden Staaten zu belasten [HANDLEY].

4.1.8 Session Announcement Mode und Session Description Protocol

Die Applikation sdr arbeitet unter der Verwendung des Session Directory Announcement Protocols. In seiner einfachsten Form bedeutet das, daß periodisch Pakete zur Beschreibung der laufenden Sessions verschickt, und erwartet werden. Durch das ständige „lauschen" auf einem fest definierten Port und einer festen Multicastadresse, ist das Programm in der Lage, Pakete, welche andere Sessions beschreiben, zu empfangen. Die Pakete sind nach dem Session Description Protocol codiert. Das Session Description Protocol basiert auf der Versendung von ASCII-Zeichenketten. Es ging mit der ersten Implementierung „sd session directory" der Lawrence Berkeley Labs einher. Jede Session wird durch folgende Parameter beschrieben [HANDLEY]:

- Name und Zweck der Session
- Zeit(en), wann die Session aktiv ist
- Medienströme, welche die Session enthält
- Informationen zum Empfang der Medien

Spezifiziert wird SDP in RFC 2327. Das SAP (Session Announcement Protocol) beschreibt UDP-Pakete, welche als Datenteil einen SAP-Header enthalten, und anschließend einen maximal 1024 Byte umfassenden ASCII-Textteil.
In diesem Textteil sind die Daten der angekündigten Session in SDP-Format abgelegt. Es werden jeweils durch den ASCII-Code 10 („\n") getrennte Strings der Form <Zeichen>=<String> codiert [RFC 2327].
Als Zeichen kommen in Frage:

v=	(protocol version)
o=	(owner/creator and session identifier).
s=	(session name)
i=*	(session information)
u=*	(URI of description)
e=*	(email address)
p=*	(phone number)
c=*	(connection information)

b=* (bandwidth information)

z=* (time zone adjustments)

k=* (encryption key)

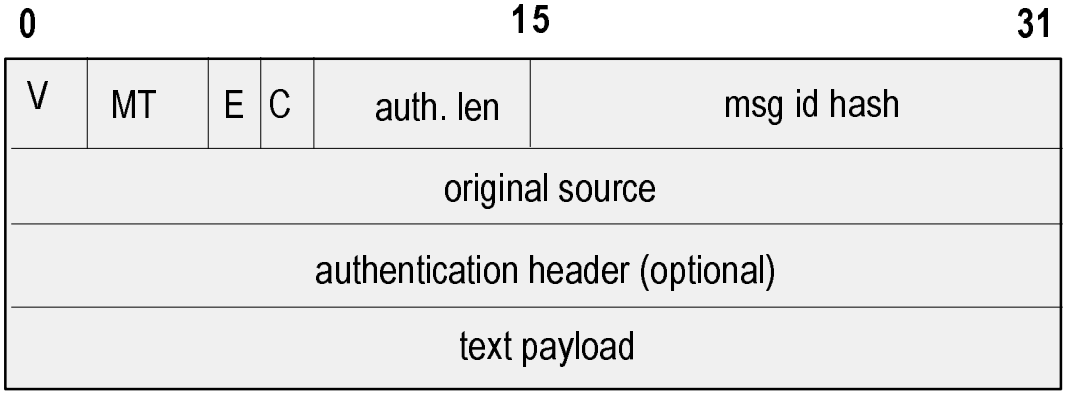


Abbildung 4.1.7: Darstellung des Aufbaus eines SAP Headers

Die Beschreibung des Aufbaus eines SAP Headers ist derzeit nur in einem Internet Draft dargelegt. Abbildung 4.1.7 zeigt den schematischen Aufbau des Headers [SAPDRAFT].
V beschreibt die Version des Protokolls (derzeit=1). MT steht für Message Type.
Es gibt zwei Arten von Messages:

-Session description announcement packet
-Session description deletion packet

Das Flag E steht für Encryption, C für Compression was sich jeweils auf den text paybad bezieht. Im Falle einer Kompression liegt der Text im ZIP-Format vor.
Der Wert „auth-len" gibt die Länge des „authentication headers" an. „msg id hash" ist ein im Netz einmalig vorkommender Wert zur Identifikation des Pakets, „orginal source" gibt die IP-Adresse des Absenders an.

Folgende Darstellung zeigt den Aufbau eines SDP-Paketes mit Zeilenumbrüchen:

```
v=0
o=mhandley 2890844526 2890842807 IN IP4 126.16.64.4
s=SDP Seminar
```

```
i=A Seminar on the session description protocol
u=http://www.cs.ucl.ac.uk/staff/M.Handley/sdp.03.ps
e=mjh@isi.edu (Mark Handley)
c=IN IP4 224.2.17.12/127
t=2873397496 2873404696
a=recvonly
m=audio 49170 RTP/AVP 0
m=video 51372 RTP/AVP 31
m=application 32416 udp wb
a=orient:portrait
```

Die in diesem Abschnitt dargelegten Informationen sind von Wichtigkeit für die Implementierung eines MBONE-kompatblen Sessionmanagements. SAP,SDP und der im Abschnitt 4.1.7 genannte Parameter TTL sind zum Verständnis der Kapitel 5 und 6 notwendig.

4.1.9 MBONE-Session-Managements Übersicht

MBONE SD(r) für win32-Plattform ermöglicht:

- pro Session ein Titel
- pro Session eine ausführliche Beschreibung
- drei Arten der Auflistung der Sessions
- Email, WWW-Adresse und Telefonnummer des Session-Senders
- Angabe einer Zeitspanne der Session
- Security über einen Encryption-Key
- Audio, Video, Whiteboard – Steuerung
- Angabe von Multicastadresse und Port für Video, Audio getrennt
- Formate GSM, DVI, LPC und PCM für Audio
- Formate H261, NV, JPG, CU See Me, CelB und Berkley Video für Videodaten
- Einsatz von RTP, SAP, SDP
- starten beliebiger Audio und Videotools über Automatische Nutzung der Kommandozeile

MBONE sdr UNIX-Portierung ermöglicht:

- zusätzlich einen Kalender

Probleme der vorgestellten Versionen:

Die UNIX Portierung wurde unter Verwendung des TCL/TK Quellcodes erstellt. Dieses bringt jedoch eine Reihe von Inkompatibilitäten mit sich. Bedienbar ist das Programm nur bei einer Bildschirmauflösung von mindestens 1024x768.

4.2 Microsoft Netmeeting und CUSeeMe

In diesem Kapitel werden zwei weitere Beispiele für Desktop-Videokonferenzsysteme angeführt. Ziel ist es, neue Ideen und Konzepte des Sessioncontrols abzuleiten.

4.2.1 Einschränkungen der Systeme

CUSeeMe und Microsoft Netmeeting sind keine echten Multicast-Konferenzsysteme, sie bieten jedoch ebenfalls einen Ausblick in moderne Mehrbenutzeranwendungen. Beide Systeme kann man über schmalbandige Netze (zB. Modemanschlüsse) betreiben, und richten sich damit eher an den Nutzerkreis, welcher über Telefonanschlüsse (ISDN oder analog) an das Internet angeschlossen ist.

4.2.2 CUSeeMe

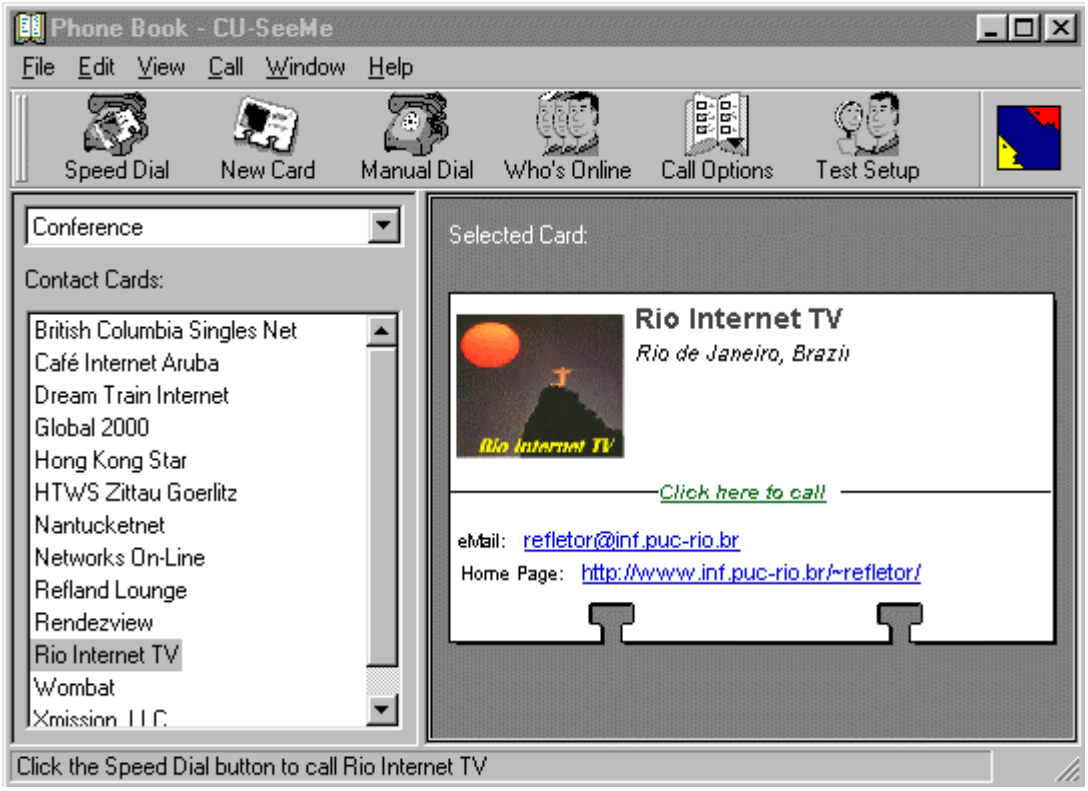


Abbildung 4.2.1: Benutzerinterface des CuSeeMe von White Pine

Dieses System wurde von der WhitePine Company entwickelt und liegt derzeit in der Version 3.1.1 vor. Abbildung 4.2.1 zeigt das Frontend des Systems.
Das CUSeeMe-System arbeitet mit sogenannten ‚Reflektoren'. Dies sind Server über die der Verbindungsaufbau stattfindet, und die Datenpakete an Unicast-Endhosts verteilen. Die Sessions fügen sich einzelnen Kategorien zu, zB.: Bussines oder Conference. Jeder dieser Server besitzt eine ‚Card' mit EMail Adresse, Homepagelocation und einem kleinen Bild. Der Nutzer des Systems entscheidet sich für einen dieser Server. Es ist möglich, mehrere Videoströme zu verfolgen, aus- und einzublenden, die Anzeigegrößen zu variieren und den Ton ausgewählter Stationen zu verfolgen.
Es ist möglich, selbst zu sprechen bzw. sein Bild einzuspielen.
Ein Chatsystem zum Austauschen von Textnachrichten kann von den jeweiligen Reflektoren angeboten werden. Mit dem Tool ‚WhitePine' steht den Anwendern zusätzlich ein Whiteboard zu Verfügung [CUSEEME].

Spezifische Eigenschaften CUSeeMe:

- Nutzer und Endhosts können sich an Reflektoren (Servern) anmelden.
- Server werden zu Gruppen zusammengefasst und gelistet
- Server besitzen ‚Cards' ähnlich den Karten von Funkstationen mit Bild
- Audio/Video oder Whiteboardsessions werden unterschieden und angeboten
- mehrere Videoströme sind gleichzeitig sichtbar, maximal 12
- Chat-System für Textnachrichten
- capture von 160x120x8bit Video, 24-bit True Color oder 4-bit Graustufen
- keine IP-Multicastunterstützung

4.2.3 Microsoft Netmeeting

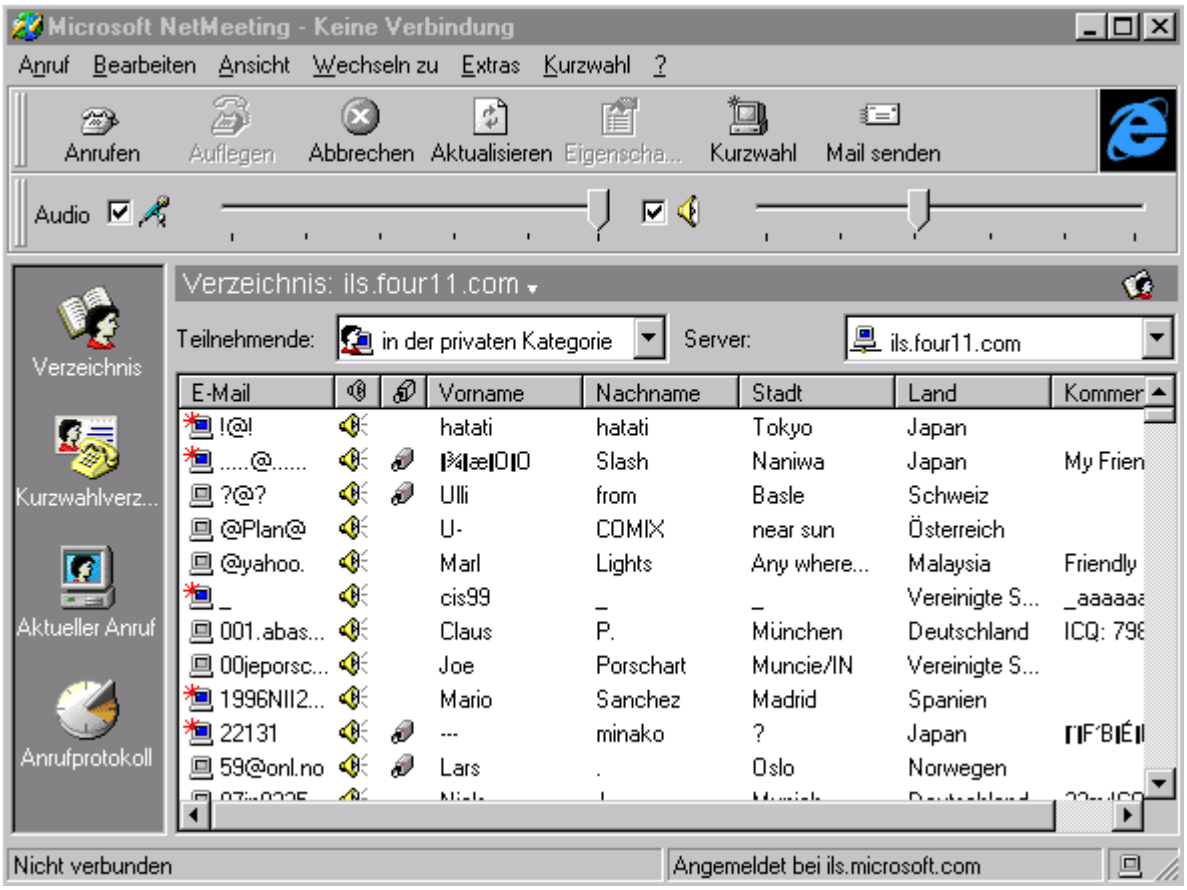


Abbildung 4.2.2: Benutzerinterface des Systems Microsoft Netmeeting

Abbildung 4.2.2 zeigt das Benutzerinterface der Microsoft Netmeeting Anwendung. Der Verbindungsaufbau erfolgt über Server und dem entsprechende Kategorien. Die Teilnehmer werden gelistet und können praktisch ‚angerufen' werden [NETMEET]. Microsoft Netmeeting arbeitet mit einem komplexen Protokollstapel, welcher die spätere Einbindung anderer Applikationen zuläßt (T120 Stapel).
Zur Übertragung der Audio- und Videodaten wird RTP verwendet. Zu beachten ist, daß der ‚User Location Service' (welcher dem Tool zum Verbindungsaufbau entspricht) ein eigenes, von Microsoft nicht veröffentlichtes Protokoll verwendet [TROSS97].

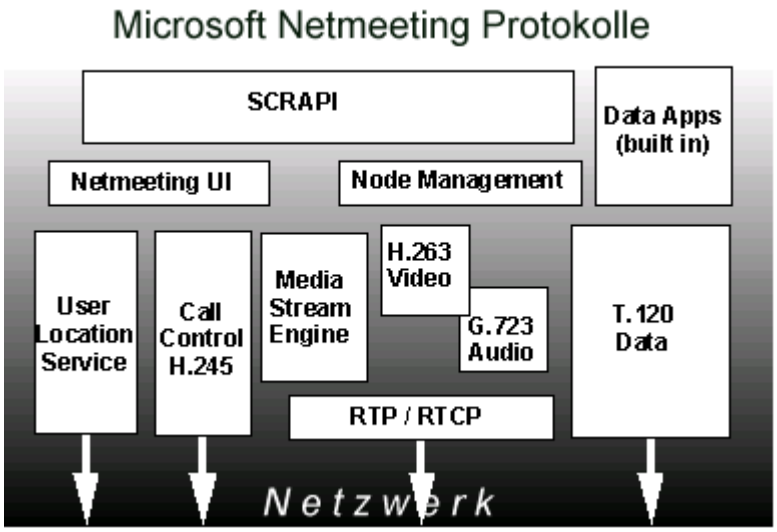


Abbildung 4.2.3: Der Protokollstapel von MS-Netmeeting

Abbildung 4.2.3 zeigt den komplexen Protokollstapel des Netmeeting Systems. Der T120 Stapel ermöglicht die Einbindung von „Shared Applications" also Applikationen welche gleichzeitig von mehreren Anwendern genutzt werden.
G.723 spezifiziert die Codierung der PCM-Audio-Daten. H.263 ist eine Erweiterung des im MBONE verwendeten H.261 Standards. Netmeeting Videodaten werden nach H.263 komprimiert. H.263 bietet nach [CODECS98] eine bessere Bild-Qualität (Verhältnis Farbtreue, Auflösung zur Bandbreite) als H.261 und wurde für geringe Bandbreiten konzipiert. Der User Location Service ist eine Eigenentwicklung von Microsoft. H.245 dient der standardisierten Aushandlung der Multimediaverbindung.

Spezifische Eigenschaften MS-Netmeeting:

- Unterstützung des H.263 Videostandards und G.723 Audiostandards
- Nutzung des H.323 Standards - Kompatibilität
- RTP Nutzung
- Servergruppierung und Kategorien
- Nutzer werden namentlich gelistet, mit Kommentar, Wohnort, Staat, und Email-Adresse
- Telefonverzeichnis
- keine Möglichkeit gleichzeitig mehrere Videoströme zu verfolgen
- keine IP-Multicast Unterstützung

4.3 VisitPhone

Dieses Kapitel beschreibt das Visitphone-System, erläutert seine Eigenschaften, zeigt seine Grenzen auf und versucht Konsequenzen für das zu entwerfende Sessionmanagement abzuleiten.

4.3.1 Ausgangspunkt

Das System Visitphone wurde unter der Zielsetzung entwickelt, zu untersuchen, ob Übertragungen in höchstmöglicher Video- und Audioqualität mit herkömmlichen PC-Systemen möglich sind. Zum Zeitpunkt des Entwicklungsbeginns wurden Intel-Pentium PC's mit 120 MHz Taktfrequenz eingesetzt. Angestrebt wurde PAL-Auflösung (Spezifikation: Abschnitt 3.4 Multimedia Grundlagen) und CD-Audioqualität.
Zu erreichen ist dies nur mit Erweiterungskarten, welche einerseits die Videodaten komprimieren (Details: Abschnitt 3.1 Multimedia MJPEG) und andererseits ein Video-Overlay ermöglichen. (Siehe Abschnitt 3.3 Multimedia: Overlaytechnik).
Beide Aufgaben übernehmen im VisitPhone-System handelsübliche Video-schnittkarten. Sie zeichnen sich dazu noch durch ein günstiges Preis/Leistungs-verhältnis aus.

4.3.2 Systembeschreibung

VisitPhone 2.0 ist ein experimentelles System, welches bereits eine Videoübertragung in PAL-Qualität mit handelsüblichen PCs ermöglicht. Full-Duplex-Betrieb wird unterstützt. Es werden Videoschnittkarten zur Kompression und Dekompression der Videodaten eingesetzt. Diese werden im Abschnitt 4.3.5 beschrieben.
Netzwerkseitig können für Übertragungen in PAL-Qualität Breitbandnetze eingesetzt werden (zB. ATM oder Fast-Ethernet). Über eine Standard-Ethernet-Verbindung (10MBit/s) kann unter Umständen bis 1/4 PAL-Auflösung übertragen werden (abhängig von der Netzlast). Das System setzt auf den Standard-IP-Stack auf. Es unterstützt Multicast-Sessions (Video Broadcasting ähnlich MBONE). Derzeit

existiert eine Anwendung für das Betriebssystem Windows95. Windows NT Unterstützung wird erst mit dem neuen DirectShow-API möglich werden.
Audiodaten können in Stereo CD-Qualität übertragen werden. Zusätzlich einstellbar sind alle anderen niedrigeren Audioqualitäten, welche im Abschnitt 2.2 aufgezeigt werden. 25 Video-Frames pro Sekunde können maximal angezeigt und übertragen werden.

4.3.3 Auftretende Datenraten

Die wichtigste Problematik bei der Entwicklung eines Videokonferenzsystems ist das Verarbeiten der Videodaten im Rechner.
Die folgende Rechnung zeigt, welche Datenraten auftreten:

Horizontalauflösung:	768 Pixel
Vertikalauflösung:	576 Pixel
Farbauflösung:	24 Bit

768x576x24 = 10.6 mill Bit = 1.3 MByte (pro Bild)
bei 25 Bildern/s entspricht das ca. 33.2 MByte/s, oder 266 MBit/s.

Diese Daten müssen unkomprimiert behandelt, und auf der Grafikkarte dargestellt werden. Das stößt an die Grenzen der verwendeten PC-Systeme, wie die folgende Rechnung zeigt:

Der PCI (Peripheral Component Interconnect) Bus besitzt einen maximalen theoretischen Durchsatz von 132 MB/s. Die tatsächlich erreichbare Durchsatzrate wird jedoch mit 80 bis 90 MB/s angegeben [SUND97]. Der Grund dafür ist, daß die Adressbytes übertragener Daten über selbigen Bus gesendet werden.
Die Darstellung zweier Overlays mit 33.2 Mb/s würde also das Bussystem bereits zu 66% auslasten. Es müssen jedoch zusätzlich Daten zwar geringerem Umfangs, jedoch ebenfals zeitkritisch, zu Netz- und Audiokarte gelangen, beispielsweise Steuersignale. Es wäre also technisch nicht möglich ein drittes Overlayfenster in PAL-Auflösung darzustellen, oder einen Dateitransfer in hoher Geschwindigkeit gleichzeitig ablaufen zu lassen.

Weiterhin treten bei einer Stereo-Ton-Übertragung in CD-Qualität noch zusätzlich auf:

Samplerate: 44kHz (44*1024Bit)
Auflösung: 16Bit (2bytes)
Kanäle: 2
resultierende Datenrate: 44*1024 Hz *2 Bytes *2 Kanäle = 176 KByte/s

Dies sind nur ca. 0.5% der gesamten Datenrate. Daher wurde auf eine Kompression dieser Daten verzichtet.

4.3.4 Untersuchung des VfW API's

Das VfW (Video for Windows) API läßt sich in vier Hauptbestandteile zerlegen:

1. AVIFile Functions and Macros
Dies ist ein HighLevelInterface zum Speichern und Abspielen von AVI-Files auf und von Festplatte. Es bietet keine Ansatzpunkte, um einzelne Videobilder abzugreifen oder zu zeichnen.

2. Video Compression Manager
VCM wird derzeit im Visitphone-System benutzt, um einzelne Videobilder zu dekomprimieren und sie per Overlay anzuzeigen.

3. Video Capture
VideoCapture bietet die Möglichkeit, einzelne Videobilder von der Videokarte bereits komprimiert zu erhalten.

4. DrawDib
DrawDib wurde während der Entwicklung von Visitphone zunächst als funktionierendes API zum Dekomprimieren und schnellen Zeichnen von Videobildern genutzt. DrawDib nutzt jedoch die GDI (Graphical Desktop Environment)-Funktionen

des Windows-Betriebssystems, und genügt daher den Performanceanforderungen nicht.
Im Test mit einem Pentium 120MHz, PCI-System wurde nur eine maximale Bildrate von 8 Bildern pro Sekunde erreicht.
In diesem Test wurde damit die Notwendigkeit der Verwendung von Video-Overlays bestätigt. Diese werden durch den Zugriff auf den VideoCompressionManager möglich.

4.3.5 Die verwendete Hardware

Da das Visitphone-System auf der Video for Windows Schnittstelle aufsetzt, ist theoretisch die Verwendung beliebiger MJPG-Karten möglich. Bedingung ist ein VfW-Treiber und eine einheitliches MJPG-Format der sendenden und der empfangenden Karte. Weiterhin müssen für den Vollduplexbetrieb zwei Karten in einem Rechner betrieben werden, denn die verwendeten Videoschnittkarten enthalten nur einen MJPG-IC. Ausschließlich ist das ein Fabrikat der Firma ZORAN, der nur einen Videostrom behandeln kann. Er kann entweder Frames komprimieren oder nur dekomprimieren. Es ist nicht möglich, in Echtzeit zwischen diesen zwei Betriebsmodi hin und herzuschalten. Die Umschaltzeit liegt (bei Verwendung der VfW-Treiber) etwa bei einer Sekunde.
Eine Ausnahme ist die Karte DigiMotion von Matrox. Sie ist in der Lage zwei MJPEG-Ströme gleichzeitig zu verarbeiten. Ihre Eignung für Videokonferenzsysteme wird derzeit im Rahmen einer anderen Arbeit getestet [FIEBIG 98].
In der Praxis treten daher oft Inkompatibilitäten und Probleme auf. Entweder arbeiten die Karten nicht paarweise zusammen, oder die Treiber unterstützen nur einige Betriebsmodi. In umfangreichen Tests und Recherchen wurden zwei Kartenpaare, welche zusammen in einem Rechner betrieben werden können, framekompatibel sind und zwei Overlays auf dem Bildschirm darstellen können, ermittelt. Dies sind einmal:
Miro DC30 und FAST FPS60, und als zweites Paar:
FAST AVMaster und DC30.
Tabelle 6.5.1 Zeigt die spezifischen Eigenschaften der Karten:

Karte	IC	decodieren	codieren	Auflösung	Bussystem
FastFPS 60	Zoran	ja	ja	½ PAL	ISA
Miro DC30	Zoran	nein	ja	PAL	PCI
FastAVMaster	Zoran	ja	ja	PAL	PCI

Tabelle 4.3.1: Videokarten im Vergleich

Weiterhin ist eine PCI-Grafikkarte mit DirectDraw-Treiber für die Overlayfunktionen notwendig. Getestet wurden die Miro Crystal VR4000 und Matrox Mystique 220. Beide Karten sind overlayfähig und können mehrere PCI-Overlays gleichzeitig darstellen. Die Karte von Matrox ist jedoch in der Lage ein korrekteres Clipping, das heißt ein Ausschneiden des Videooverlays bei darüberliegenden Fenstern darzustellen. Beim Verschieben von Fenstern kommt jedoch ein leichtes Flackern des Videobildes vor.
Die VR4000 stellt zeitweise (unregelmäßig) einige Clipping Fehler dar, liefert jedoch sonst einwandfreie Darstellungen. Insgesamt hat sich bei beiden Karten ein Heruntersetzen der Farbauflösung von 24Bit auf 16Bit Farbtiefe als performancesteigernd erwiesen, die Videodarstellung wird dann etwas „flüssiger". Beide Karten sind für den Einsatz im Visitphone-System geeignet.

4.3.6 Das verwendete Multimedia Transportprotokoll

Visitphone nutzt den User Datagram Protocol (UDP) -Stack des Windows-Betriebssystems. Aufgesetzt wurde ein Protokoll, welches Fehlererkennung und Fragmentation ermöglicht.
Ein Paket dieser Protokollschicht besteht aus einem Paketkopf und nachfolgend den Video- bzw. Audiodaten.
Der Paketkopf enthält:

- DgId bezeichnet die Identifikationsnummer eines UDP-Datenpakets. (2Byte)
- SeqSum Anzahl der Fragmente eines Frames (1Byte)
- SegId identifiziert einzelnes Fragment eines zerlegten Frames (1Byte)
- SeqCRC einfache Prüfsumme (4Byte)
- Time enthält einen Zeitstempel der lokalen Sendestation (64bit)

- DataLen UDP-Paketlänge (4Byte)
- PaketLen Länge des gesamten Pakets (4Byte)

Den Aufbau des Paketkopfes zeigt Abbildung 4.3.1 [NGUYEN98]:

DgId	SeqSum	SeqId
PaketLen	***DataLen***	***SeqCRC***
time		

Abbildung 4.3.1: Der Aufbau des Paketkopfes

4.3.7 Der Aufbau von VisitPhone

Visitphone besteht aus mehreren Modulen. Der Aufbau wurde im Klassenkonzept von C++ realisiert.
Im Rahmen der Arbeit vorangegangenen Entwicklungen, wurden die Module Main, Audio, Video, UDPSocket und diverse Fensterklassen (Siehe Abbildung 4.3.2) erstellt.
Auf der Ebene des IP-Winsocks wird direkt auf das UDP-Protokoll aufgesetzt. Ein in [NGUEN98] entwickeltes Netzwerkmodul übernimmt die Überwachung der Reihenfolge der Datenpakete, validiert diese (CRC - Check), und sorgt für das Versenden derselben bzw. benachrichtigt das Modul‚Main‘ über deren Eintreffen.

Das Videomodul besteht aus zwei Hauptbestandteilen: Wiedergabeteil und Capture. Das Capture besteht aus einer Callback – Funktion, welche jeweils nach erfolgter Kompression ein komprimiertes Frame an das Main-Modul schickt.
Der Wiedergabeteil nimmt dagegen ein komprimiertes Frame an, und stellt dieses per Overlay dar.
Diese Module setzen auf die Video for Windows (VfW) – Schnittstelle auf. Der VfW-Treiber übernimmt das Zeichnen des Videobildes per Overlay, indem er seinerseits auf den DirectDraw-Treiber der Grafikkarte zugreift.

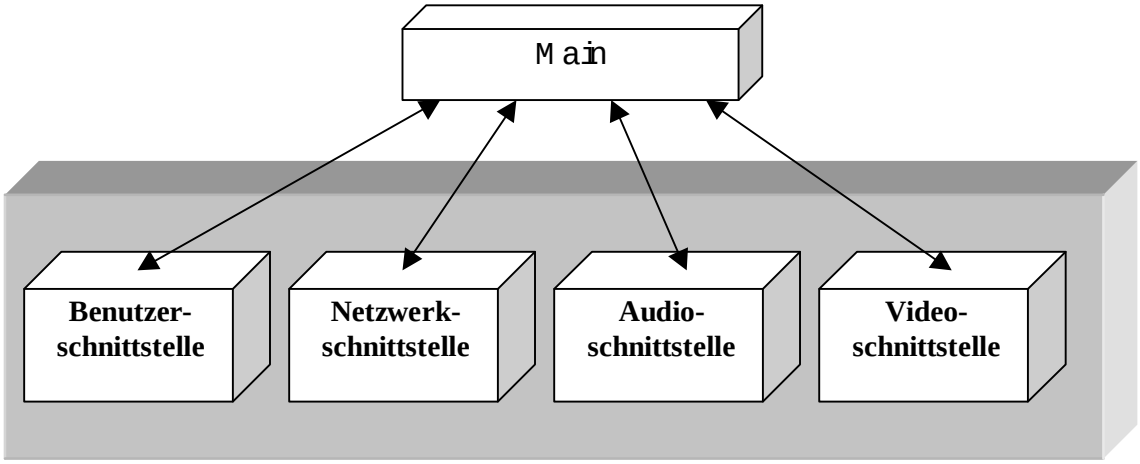


Abbildung 4.3.2: Die Module des Visitphone Systems

Audiodaten werden anders gehandhabt, das Wave-Device von MS-Windows fungiert als Ein- und Ausgabegerät. Diese Aufgaben regelt das Audiomodul. Dabei wird eine Callback Funktion verwendet.
Aufgrund der geringeren Datenrate der Audiodaten ist deren Verarbeitung im Rechner weniger kritisch (siehe Datenratenberechnung Abschnitt 4.3.3).
Das VisitPhone-System komprimiert Audiodaten aus Performance-Gründen nicht. Die dadurch eingesparte CPU-Zeit ist z.B. für die flüssige Overlay-Darstellung unbedingt notwendig. Die unwesentlich höhere Netzlast (vergleiche Abschnitt 4.3.3) ist vernachlässigbar.

4.3.8 Die Testanwendung unter Windows95

Ein Teil des Systems ist das Tool zum Senden von Video- und Audiodaten (Abbildung 4.3.3).
Es ermöglicht das Eingeben der Zieladresse, wobei Multicastadressen erlaubt sind. Diese Multicastadressierung wurde im Kapitel 3 über MBONE detailiert erklärt. Einstellungen über Netzwerkeigenschaften , Video- und Audioformate, deren Darstellung und Kompression können vorgenommen werden.

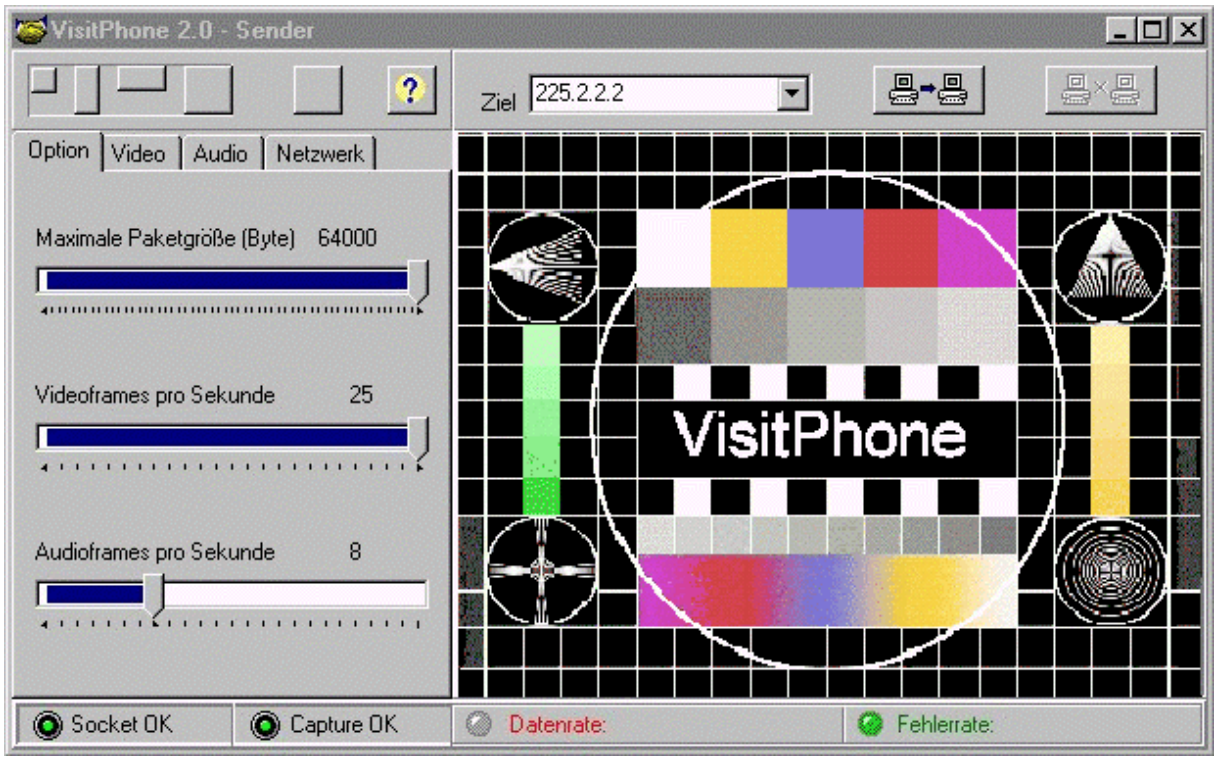


Abbildung 4.3.3: Das Benutzerinterface des Senders des Visitphone-Systems

Im laufenden Betrieb werden die Datenrate, Fehlerrate und Zustand des Capture- und Netzwerktreibers angezeigt. Diese Daten liefern wichtige Anhaltspunkte über den Zustand der Übertragung und der Videokarten.

Zweiter wichtiger Teil ist das Empfangstool. Es erwartet nach seinem Aufruf einkommende Video- und Audiodaten. Es „lauscht" an einem festdefinierten UDP-Port. In der derzeitigen Version ist dies der Port 2000. Es ist möglich, in eine Multicastgruppe einzutreten, um deren Datenstrom zu empfangen.

Identisch zum Sender des Systems, werden im laufenden Betrieb zusätzlich Datenrate, Fehlerrate und Informationen über die verwendeten Video- und Audioformate angezeigt.

Das Verändern der Auflösung des Overlaybildes ist nur statisch möglich. Folgende Auflösungen sind möglich: 768x576 , 768x288 , 384x576 und 384x288 Pixel.

Diese Beschränkung resultiert aus der Tatsache, daß die verwendeten Videokarten teilweise nur mit dem Faktor zwei oder einhalb skalieren können. Bei einer dynamischen Größenänderung könnten Verzerrungen oder Bildfehler auftreten.

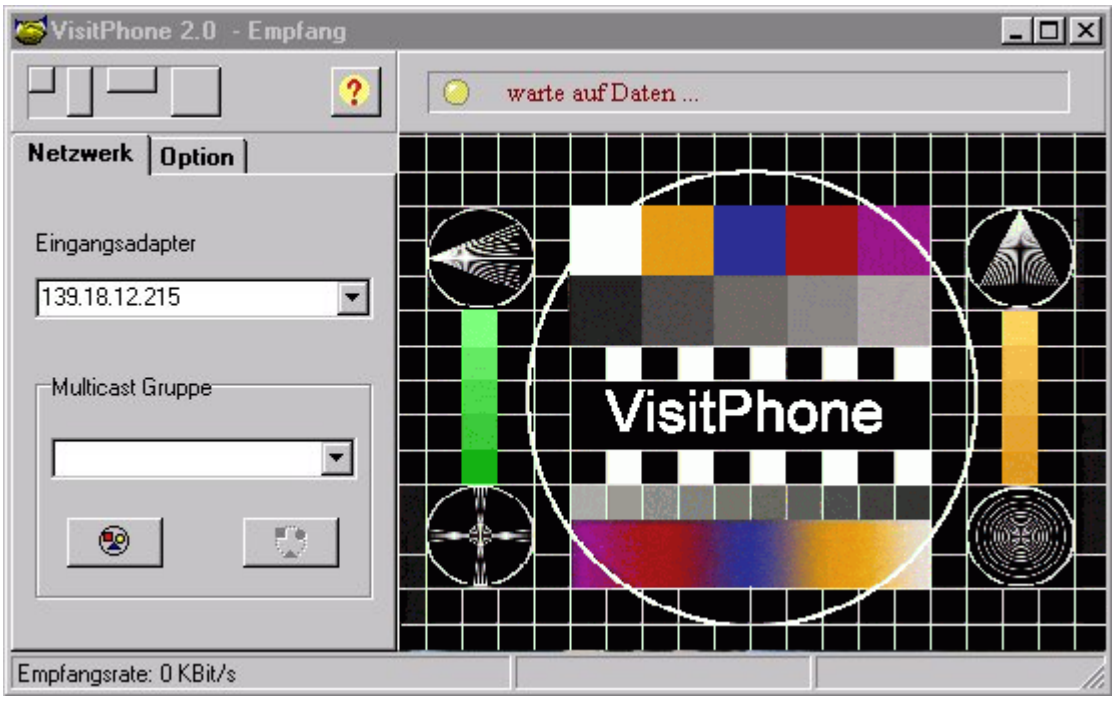


Abbildung 4.3.4: Das Benutzerinterface des Empfängers des Systems

4.3.9 Spezielle Probleme im Hinblick auf Sessionmanagement Module

Das Visitphone System wurde unter der Zielstellung entwickelt, Videosequenzen in PAL-Qualität zu übertragen. Notwendig und sinnvoll ist eine Kompression der Videodaten.

Dazu wird, wie schon beschrieben, MJPG eingesetzt. Die netzwerkseitig auftretenden Datenraten zeigt Abbildung 4.3.5. Als Maximum pro Videostrom (eine Richtung) kann man 5500 KByte/s (das entspricht ca. 45 MBit/s) annehmen.

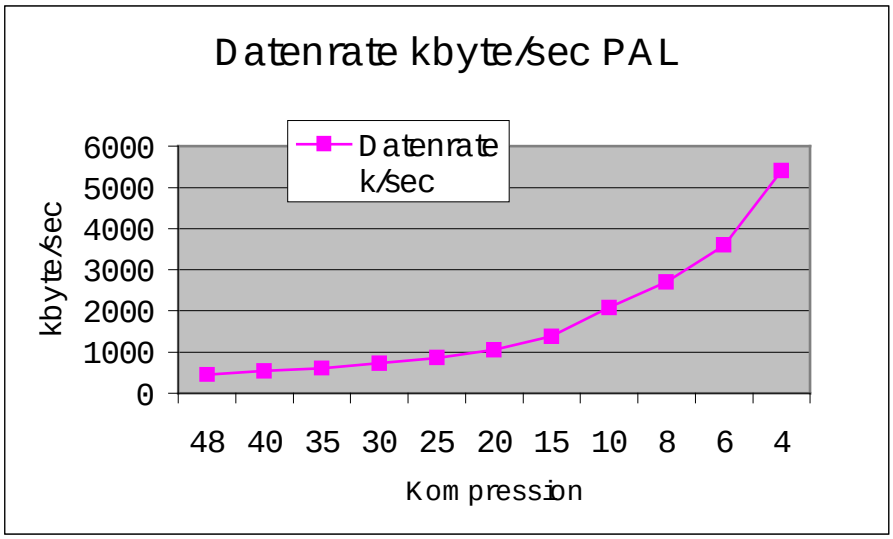


Abbildung 4.3.5: Datenrate in Abhängiggkeit zur Kompression

Bei Verwendung des Kompressionsparameters 4 treten keine mehr für das Auge sichtbaren Verzerrungen oder Farbfehler auf. Für Glasfasernetze (ATM) oder FastEthernet stellt diese Datenrate kein Problem dar.
Engpaß wird das im PC verwendete Bussystem, die CPU, oder das Timing der Interrupts der Netzkarten.

Ein weiterer wichtiger Punkt ist das Verhalten des Systems in Abhängigkeit zur installierten CPU und dem zugehörigen Motherboard.
Es wurden im Rahmen der Entwicklung drei verschiedene PCI-Motherboards von Soyo, Explorer und ASUS getestet. Intel Pentium-CPU's von 180 MHz und 233 MHz Taktfrequenz waren im Test eingeschlossen.

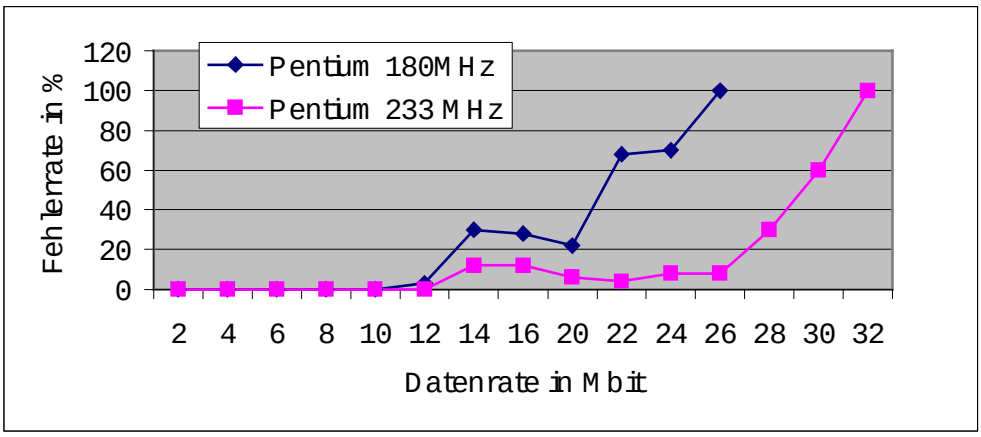


Abbildung 4.3.6: Fehlerraten in Abhängigkeit zur Datenrate bei Visitphone

Abbildung 4.3.6 zeigt, welche Datenraten über den klassischen UDP-Stack erreicht wurden. Diese Werte wurden mit 155 MBit ATM Karten von Fore und Madge ermittelt. Es wurde LAN-Emulation eingesetzt. Der minimalste erreichte Kompressionsfaktor war etwa 6 (entspricht 32 MBit).
Auf Empfängerseite ist die CPU-Taktrate von hoher Wichtigkeit. Mit Verdoppelung der CPU-Taktfrequenz (von 120MHz auf 233MHz) war eine maximale Datenraten-Steigerung um ca. 25 % erreichbar.
Auf Senderseite reichte eine 133MHz CPU stets aus. Eine Steigerung der CPU-Taktfrequenz brachte hier keine höhere Datenrate mit sich.

Die Prozessorbelastung liegt bei einer Halbduplexverbindung bei ca. 66%. (Testsystem: Pentium 233 MHz, 64MB RAM, Meßinstrument: NT-Taskmanager). Bei einer Vollduplexverbindung erreicht sie nahezu 90%.

Somit ergeben sich folgende Anforderungen an das Sessionmanagement-System:

- es sollte wenig Prozessorbelastung verursachen
- Kommunikation in Fehlerfällen ermöglichen
- die spezielle Overlayproblematik der Videokarten beachten
- keinen hohen ständigen zusätzlichen Datenverkehr verursachen.

Dies sind die Anforderungen, welche im speziellen durch VisitPhone impliziert werden. Darüber hinaus sind noch weitere Funktionen und Möglichkeiten zu implementieren. Dies wird in Kapitel 5 behandelt.

5 Konzeption des Sessionmanagements für Visitphone

Kapitel fünf stellt den Entwurf einer Sessionmanagementlösung für Visitphone dar. Es zeigt, benennt und bewertet einzelne Lösungsideen, welche in das Managementsystem einfließen.

5.1 Herausarbeitung allgemeiner Anforderungen

Das System Visitphone besitzt in der derzeitigen Version nur eine Möglichkeit, einen Verbindungsaufbau durchzuführen: Dem sendenden Teilnehmer muß die IP-Adresse der Gegenstation bekannt sein. Er gibt sie manuell der Applikation bekannt, und startet den Sendevorgang.
Es ist notwendig, eine Applikationschicht zu schaffen, welche Verbindungen einer abstrakteren Art ermöglicht. Das heißt, es sollten Verbindungen zwischen Teilnehmern existieren, nicht zwischen IP-Endgeräten. Diese Verbindungen müssen erzeugt und überwacht werden. Bei Rundsendungen (Multicast) sollten in dieser Applikationsschicht Ankündigungen verschickt werden (siehe auch Kapitel 4.1 MBONE). Bei Verbindungen sollte ihre Art unterschieden werden (Whiteboard, Bild, Ton oder Dateiübertragung).

Im Rahmen der Entwicklung von CCCP (siehe Absatz 5.3 „CCCP“) am University College London wurden eine Reihe wichtiger Punkte für Sessionmanagementsysteme herausgearbeitet [CCCP94]:

- Flexibilität insofern, daß alle Arten von Applikationen eingebunden werden können und jede Übertragungsart genutzt werden kann.
- Die Architektur sollte skalierbar sein, so daß eine sinnvolle Nutzung mit drei Personen in einem Raum genauso möglich ist wie mit Millionen von Personen, welche über Kontinente verteilt sind.
- Das schließt aus, daß ein zentralisierter Ansatz gewählt werden kann.

Durch Gespräche mit möglichen späteren Nutzern ergaben sich eine Reihe weiterer Anforderungen und Ideen. Es ist beispielsweise nur möglich, eine Akzeptanz bei

betriebswirtschaftlichen Anwendern zu erreichen, wenn eine Unterstützung für Whiteboards implementiert wurde. Folgende Aspekte wurden daher bereits aufgegriffen:

- Whiteboard Unterstützung
- Möglichkeiten des Dateitransfers
- Abruf von Video- und Audiodatenbanken
- Erweiterungs- und Einbettungsmöglichkeiten für shared Applications

Hinzu kommen die speziell durch Visitphone implizierten, aus Kapitel 4.3 abgeleiteten Anforderungen:

- es sollte wenig CPU-Zeit verbrauchen
- Kommunikation in Fehlerfällen ermöglichen
- die spezielle Overlayproblematik der Videokarten beachten
- keinen ständigen zusätzlichen Datenverkehr verursachen

Sekundär, jedoch selbstverständlich sind eine Reihe von Bedienungskriterien:

- einfache Bedienung, durchaus von unkundigen Personen
- schneller, unkomplizierter Verbindungsaufbau
- einfache Assoziationsmöglichkeiten Teilnehmer/Synonym

5.2 Prüfung der Verwendbarkeit von „sdr“ des MBONE als Sessionmanagement für Visitphone

Das Sessionmanagement des MBONE in der Implementierung „sdr“ unterstützt als einziges der getesteten Systeme Multicastverbindungen. Weiterhin ist es als offenes Tool ausgelegt, so daß man andere Video und Audiotools einbinden kann. Im Versuch wurde „sdr“ mit „Receiver“ und „Sender“ des Visitphone-Systems gekoppelt. Durch die Einbindung werden von „sdr“ die Visitphone-Tools über einen Aufruf auf der Kommandozeile gestartet.
Die Syntax des Aufrufes, Beispiel „vic“ lautet:

vic -t [ttl] [multicastaddress or IP address]/[port] (Siehe Auch Abschnitt 4.1.2)
Es genügt, diese Kommandosyntax für die Visitphone-Tools einzuhalten, um ein dem MBONE-sdr kompatibles Video- und Audiotool zu erhalten.
Neu implementiert werden mußte jedoch die Auswertung der Parameter ttl, Ip, Port in den Tools „Receiver" und „Sender".

Zu ändern war:
-die Verwertung des Parameters ttl und dessen Verarbeitung im Netzwerkmodul
-die Auswertung das Parameters „IP-Adresse" im Tool
-die Auswertung des Parameters „Port" im Netzwerkmodul

Nach den genannten Änderungen war es möglich, „sdr" als Sessionmanagement für Visitphone einzusetzen.

Es ergeben sich aus dieser Konstellation jedoch eine Reihe von Beschränkungen:
- Das Sessionmanagement „sdr" des MBONE überprüft nicht, welche Anzahl von Videokarten vorhanden ist, und ob weitere Overlays darstellbar sind. Dadurch kommt es beim Start unter Umständen zu Fehlern. Der Grund dafür ist die dem Visitphone eigene Beschränkung der Anzahl der Videoströme auf die Anzahl der Videokompressionskarten. Jede Karte kann derzeit genau einen Videostrom bearbeiten.
- Dieses Problem beschränkt sich nicht nur auf die Videodaten. Die Audiodaten können jeweils Full-Duplex oder einseitig bearbeitet werden. Das Fehlen eines Audiodaten-Mixers im Visitphonesystem wirkt sich an dieser Stelle aus. Es ist nicht möglich, gleichzeitig Audiodaten von mehr als einem Teilnehmer zu empfangen.
-Weiterhin wurde das Problem der Nutzerverwaltung im MBONE nicht einheitlich gelöst. Die Audio und Videotools bieten zwar die Möglichkeit der Einzelverbindung zweier Teilnehmer, jedoch unterstützt das Sessionmanagement keine Namenszuordnungen oder Teilnehmerlisten.

Nach der Implementierung und einem Testbetrieb unter Windows95, stellten sich eine Reihe weiterer Probleme heraus:

- Eine MBONE Session kann mehrere sendende Teilnehmer haben. Tritt dies bei einer Visitphone-Session auf, kann man trotzdem nur einen der Videoströme empfangen und darstellen.
Somit muß man bei jedem Start im sdr der Session explizit den Typ „nur empfangen“ vergeben.

- Das Tool „sdr“ des MBONEs kann nur eine Netzwerkkarte pro Rechner behandeln. Betreibt man beispielsweise eine ATM-Karte mit der ID=0 und eine Ethernetkarte mit der ID=1, und besteht die MBONE Verbindung nur über Ethernet kommt es zum Absturz des Programms „sdr“. Das deutet offensichtlich auf einen Programmfehler hin. Der zugehörige Quellcode steht nicht zu freier Verfügung.

- Mit Visitphone-Tools gesendete Sessions können wiederum nur mit Visitphone-Tools empfangen werden. Startet ein anderer Teilnehmer sein herkömmliches, unmodifiziertes vic & vat, so kann er keine Daten empfangen. Grund dafür ist das von Visitphone verwendete Transportprotokoll (Abschnitt 4.3.6). Es ist zu keinen der „vic“ oder „vat“ – Versionen kompatibel.

- Das System ist in dieser Form nicht erweiterbar. D.h. es können keine weiteren Tools wie zB. shared Applications eingebunden werden.

Deshalb wurde versucht, auf der Grundlage des im Abschnitt 5.3 vorgestellten CCCP ein speziell für Visitphone zugeschnittenes Sessionmanagement zu entwerfen.

5.3 Das Conference Control Channel Protocol (CCCP) als Grundlage des Visitphone - Sessionmanagements

Das Conference Control Channel Protocol (CCCP) wurde am University College London entwickelt. Es stellt ein Konzept vor, welches kleine Meetings (beispielsweise zwei Teilnehmer) von eng gebundenen Partnern genauso ermöglicht wie sehr große, lose gebundene Konferenzen.
Die Grundanforderungen wurden bereits im vorigen Abschnitt 5.2 aufgeführt: Besonders wichtig sind Flexibilität, Skalierbarkeit und Dezentralität.

Das Modell diente als Grundlage des entwickelten Sessionmanagementmoduls. Der Conference Control Channel, auf welchen das CCCP aufgesetzt wird, wird als Broadcast-Bus beschrieben. Die Empfänger filtern die für sie relevanten Nachrichten heraus. Die Rolle des CCCP-Broadcastbuses übernimmt analog zum MBONE ein Multicast-Bus, welcher die Nachrichten bereits nur an relevante Teilnehmer weiterleitet. Nachfolgend werden sie dann weiter gefiltert. Abbildung 5.3.1. zeigt die Konzeption dieses Multicast-Bussystems. Der Conference Control Channel wird im Visitphone System durch eine gemeinsame Multicastadresse inclusive festegelegtem Port realisiert, identisch zum MBONE-System. Davon unabhängig werden die Audio- und Videodaten ausgetauscht, teilweise auch über Punkt zu Punkt Verbindungen (Media Data). Der Verbindungsaufbau, -abbruch und die Verbindungskontrolle erfolgen über den Conference Control Channel (schmale Pfeile).

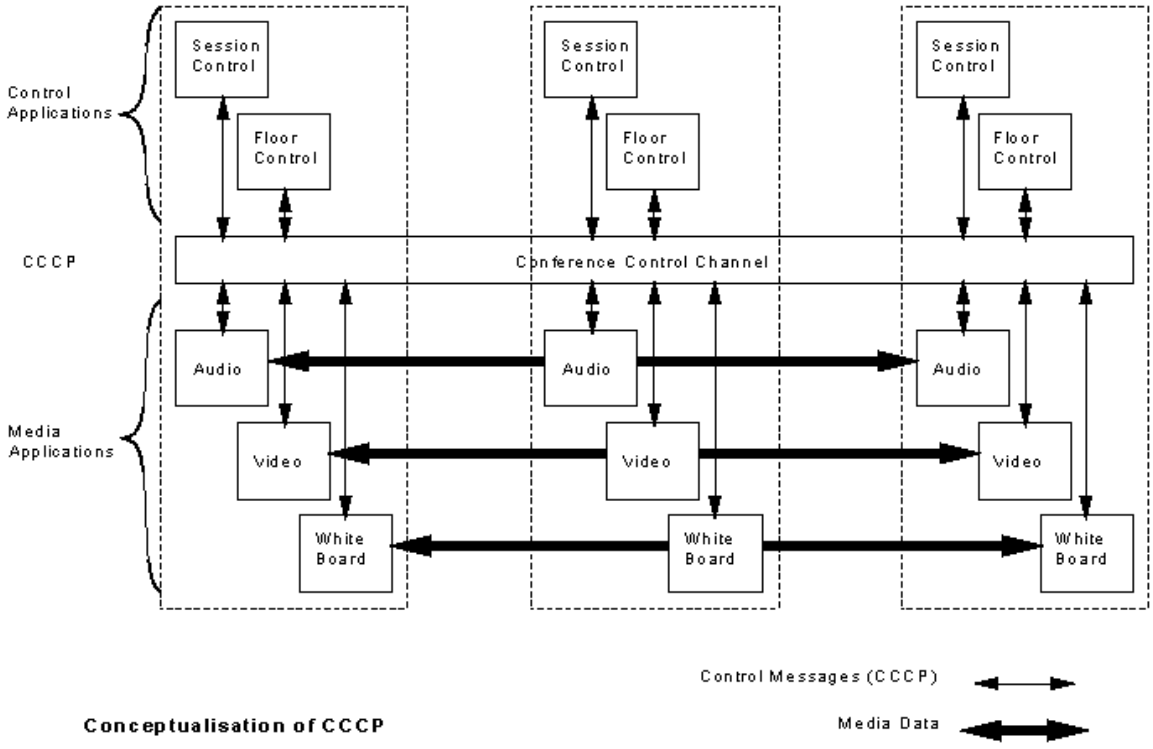


Abbildung 5.3.1: Die CCCP Konzeption

Addressiert werden Nutzer und Applikationen über ein Adressfeld: username@hostname. Diese Art der Adressierung hat sich bereits im Umfeld der E-Mail bewährt und wird im Visitphone-System eingesetzt werden.

CCCP wurde ausgelegt, vier Modi der „end-to-end Reliability", dh. der Zuverlässigkeit der Verbindung und der gesendeten Daten, zu unterstützen [CCCP94]:

- „send and forget"	keine Unterstützung
- „At least one"	es wird abgesichert, daß zumindest einer der Teilnehmer die Sendung empfängt
- „n out of m"	zumindest n Teilnehmer der Zielhosts empfangen
- „all"	es ist sichergestellt das alle Teilnehmer korrekte Daten erhalten.

Das Visitphone-System verfolgt nach Einbindung des entwickelten Session-managements zunächst die Strategie „send and forget", dh. es enthält keine Flußkontrolle.
In einer Multicastumgebung gestaltet sich eine solche Flußkontrolle extrem aufwendig, ihre Entwicklung würde deutlich den Rahmen dieser Arbeit überschreiten, denn jeder Teilnehmer der in eine Gruppe eintritt oder selbige verläßt, wird von den Multicast-Routern verwaltet. Auf der Senderseite liegen keine Informationen darüber vor, wer alles Empfänger der Daten ist.
Sämtliche Teilnehmer sind verbindungslos (vergleiche UDP Abschnitt 3.6) angeschlossen.

5.4 Weitere Lösungsansätze

Nutzerverwaltung: Um eine dezentrale und global einsetzbare Konzeptionierung der Nutzerverwaltung zu entwerfen, muß zunächst festgelegt werden, welche Synonyme man den Nutzern zuordnet.
Dazu kann man zwei Verfahren anführen, welche sich bereits in der Praxis bewährt haben: Man kann jeder Person eine Nummer zuordnen (Einsatz: Telefonnetz), oder einen eindeutigen Textstring pro Peron vergeben. Letzteres wird bei dem Versenden von Email angewandt.
Ausgewählt wurde für die Visitphone-Nutzerverwaltung eine Zuordnung der Form user@hostname .
Um einem dieser Teilnehmer eine Verbindung anbieten zu können, muß eine Adressauflösung stattfinden. Realisiert wird diese, indem man einen „IP-Request per

Name“ (Siehe Abschnitt 6.3) über den Multicast-Bus schickt. Der Teilnehmer mit dem erwünschten Namen wird seine IP-Adresse über selbigen Kanal zurücksenden.
Danach kann eine direkte Verbindung aufgebaut werden.
Dieses dezentral angelegte Vorgehen sichert die Skalierbarkeit des Systems nach [CCCP94]. Es werden dazu keine Server benötigt, welche dann bei einer hohen Nutzerzahl überlastet werden könnten.
Sessions werden zunächst aller 6 Sekunden angekündigt. Das geschieht durch das Versenden eines SDP-Paketes auf dem Multicast-Bus (vergleiche auch Abschnitt 5.6). Dadurch wird die Prozessor- und Netzbelastung gering gehalten.

5.5 Die Realisierung einer Konferenzschaltung

Grundsätzlich wird davon ausgegangen, daß alle Nutzer gleichberechtigt sind. Um eine Konferenzschaltung (siehe Abschnitt 2.1) zu realisieren, kann man so vorgehen, daß der sendende Teilnehmer darüber informiert wird, welche der Zuhörer zu sprechen wünschen. Er kann dann darüber entscheiden, welchem der zuhörenden Teilnehmer er das Wort erteilt, und wann das geschieht. Wie dieser Mechanismus umgesetzt wird, und welche Kommandos des Befehlsinterpreters dazu benutzt werden, beschreibt Abschnitt 6.5.

5.6 MBONE Kompatibilität des Sessionmanagements

Da das Sessionmanagement-Konzept auf dem gleichen Multicast-Adressraum basiert, welchen die bisherigen MBONE-Tools verwenden, ist eine MBONE-Kompatibilität durchaus sinnvoll. Im Abschnitt 3.6 wurden die Protokolle SDP und SAP beschrieben. Die Einhaltung dieser Protokolle genügt, um die MBONE-Sessions empfangen und dekodieren zu können. Identisch zu den MBONE Sessionmanagements „lauscht“ das Visiphone-System auf der Adresse 224.1.127.254 Port 9875. Die ankommenden SDP-Pakete werden dekodiert und gepuffert. So ist das Programm beim Starten einer Visiphone-Session in der Lage, eine im MBONE unbenutzte Adresse herauszufiltern, und der neuen Session zuzuweisen. Diese Funktionalität genügt, um mit geringem Aufwand die Erweiterungen zum Auflisten der Sessions und zum Starten der Tools „vic“ und „vat“ zu implementieren.

Dazu werden die Informationen der SDP-Pakete dekodiert und die entsprechenden Parameter für vic & vat generiert.
In die Implementation wurden die MBONE-Tools vic Version 2.8 und vat in der Version 4.02b eingebunden.
Umgekehrt ist es ebenso sinnvoll den MBONE-Routern die belegten Adressen bekanntzugeben. Das geschieht durch eine periodische Generierung von SAP und SDP-Paketen, welche ebenfalls auf der Adresse 224.1.127.254 Port 9875 gesendet werden.
Wichtig ist jedoch eine Unterscheidungsmöglichkeit zwischen Visitphone- und MBONE-Sessions. Denn eine standardisierte MBONE-sdr Installation kann die Daten einer Visitphone-Session nicht empfangen, obwohl es die zugehörigen Sessioninformationen erhält. Grund dafür ist das nicht RTP-kompatible Übertragungsprotokoll von Visitphone.

5.7 Einbindung einer Whiteboard-Anwendung

An der Berkeley Universität wurde als Ergänzung zu den Tools vic und vat ein Whiteboard unter der Bezeichnung mash-MB entwickelt. Es wird von [MBONEDR] für den Einsatz unter Windows95 und WindowsNT empfohlen. Gestartet wird es über die Kommandozeile mit den Parametern

-sa Adresse/Port/TTL	Bsp.: -sa 224.12.12.12/1234/32
-C Sessionname	Bsp.: -C Informationsaustausch

Das Programm „mb.exe“ muß sich als ausführbare Anwendung im Visitphone-Verzeichnis befinden. Ebenso die zugehörige Mash-Console („mash.exe“). Dann kann es vom Sessionmanagement durch einen WinExec-Aufruf vom Sessionmanagement aus gestartet werden.
Als Multicastadresse wird beim Aufruf die Adresse der Session angegeben, über die Audio- und Videodaten gesendet werden. Als Port können alle freien UDP-Ports außer 2000 und 2001 (Audio- und Videoport) verwendet werden.
Abbildung 5.7.1 zeigt die Bedienoberfläche der beschriebenen Applikation.

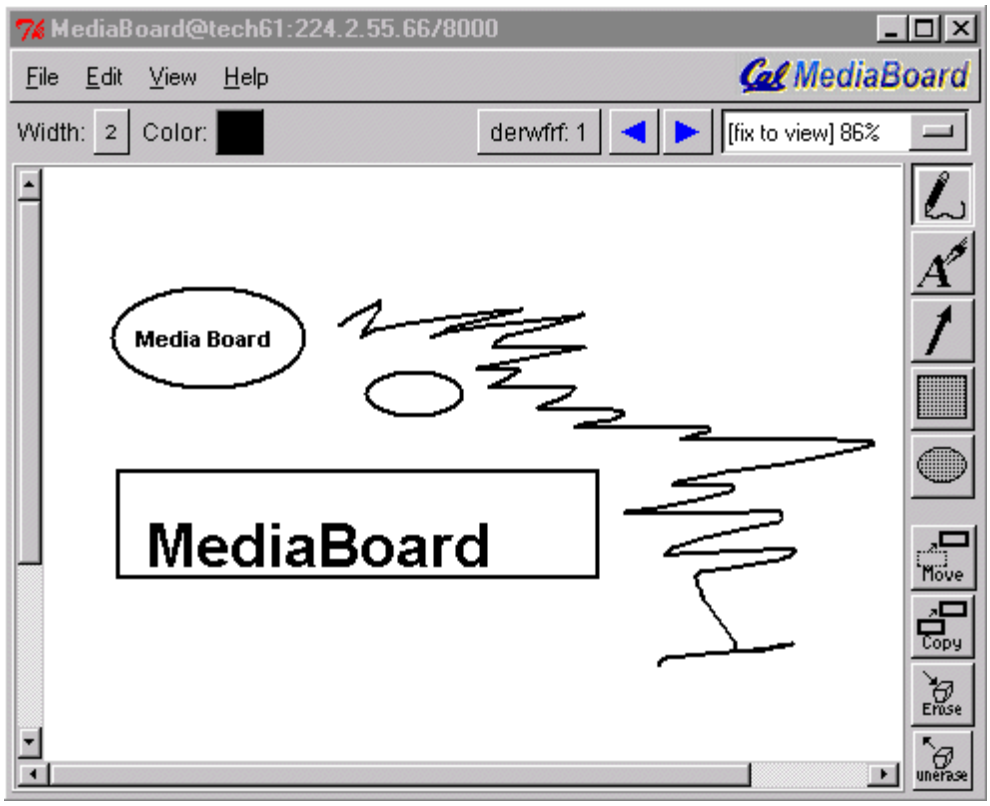


Abbildung 5.7.1: Das Benutzerinterface des Mash –Mediaboards

5.8 Kommunikation mit anderen Modulen und Shared-Applications

Das Sessionmanagementmodul wurde als offenes, erweiterbares System konzipiert. Durch das in Kapitel 6 exakt definierte Interface, insbesondere durch das Befehlsinterface ist es leicht möglich weitere Module zu ergänzen. Sie können dann die Parameter der bereits bestehenden Verbindung zwischen den aktuellen Teilnehmern abfragen, weitere Sessiondaten und -parameter über den Multicastbus austauschen, und so eine Verbindung ihrerseits aushandeln.

Die Datenübertragung zwischen den hinzugefügten Modulen (das können ganze Shared-Applications, also gesamte Anwendungen sein.), erfolgt dann über ein separates Socket. Programmiertechnisch kann hier das Socketinterface von [NGUYEN98] genutzt werden, logisch stellt es jedoch eine jeweils eigene Verbindung dar.

Auch das Whiteboardboardtool „mash-MB“ nutzt für die Übertragung der Whiteboarddaten ein seperates Socket. Standardmäßig verwendet es den UDP-Port 1235. Die Ports 2000 und 2001 sind von Video- und Audiotools belegt. Alle weiteren UDP-Ports stehen weiteren Anwendungen und Modulen zu Verfügung.

Ein Dateitransferprogramm könnte sich der bestehenden Audio- und Videoverbindung aufschalten, aber ein eigenes Bedienfenster öffnen und seine Daten (die zu transferierende Datei) über Port 9845 übertragen.

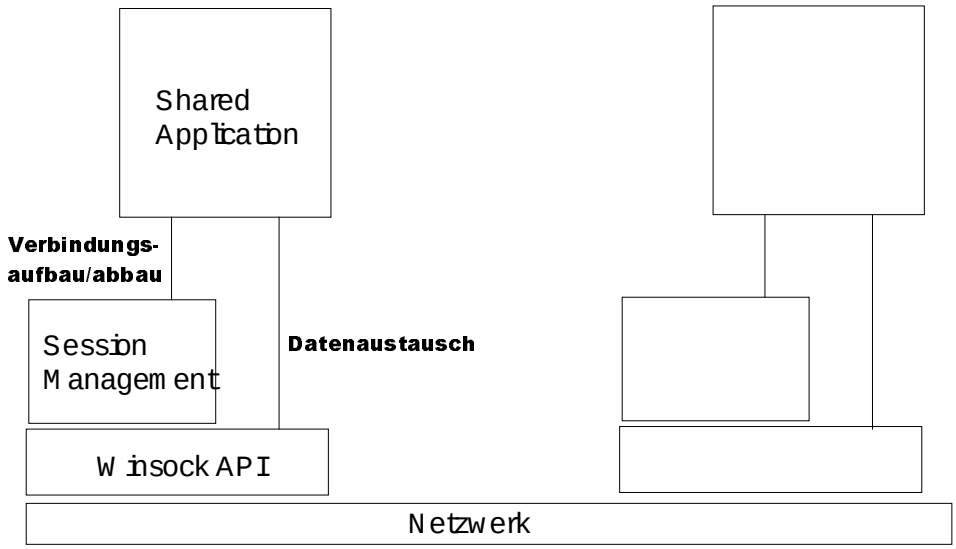


Abbildung 5.8.1: Schematisch: Angliederung von Shared Applications

Abbildung 5.8.1 zeigt den schematischen Aufbau dieser Konstellation. Der Verbindungsaufbau und -abbau wird vom Sessionmanagement übernommen. Die eigentlichen Daten der Session werden über ein seperates Socket gesendet.

6 Implementierung des Sessionmanagementmoduls für Visitphone

Dieses Kapitel beschreibt, in welcher Art das in Kapitel fünf dargelegte Konzept konkret in einem Sessionmanagementsystem für Visitphone umgesetzt wurde.

6.1 Codierung des Sessionmanagementmoduls

Implementiert wurde ein Modul im C++ Klassenkonzept, welches in der Lage ist, die Anforderungen von Visitphone zu bearbeiten. Als Entwicklungstool wurde der Borland C++ Builder 1 eingesetzt. Auf die Verwendung von mitgelieferten Klassenbibliotheken wurde verzichtet. Das sichert die Verwendbarkeit, falls in der weiteren Entwicklung andere Entwicklungswerkzeuge eingesetzt werden. Dieses Konzept hat sich bereits in der Entwicklung des Netzwerk- und des Videointerfaces bewährt. Die Umsetzung des Quellcodes von Microsoft Visual C++ auf Borland C++ Builder war problemlos möglich.
Desweiteren war es nötig, das Netzwerkinterface [NGUYEN98] um einige spezielle Funktionen zu erweitern. Dazu wurde die Technik der Vererbung von C++-Klassen verwendet. Die entstandene Klassenhirarchie ist in Abbildung 6.1.1. dargestellt.

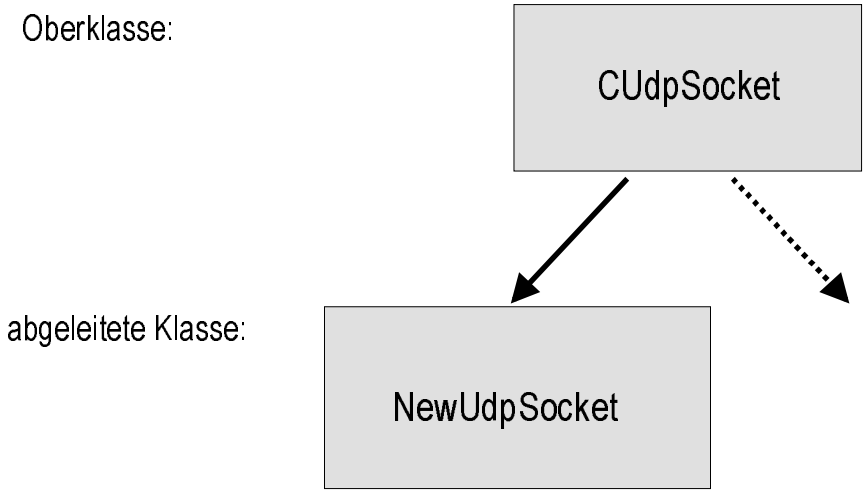


Abbildung 6.1.1: Die Klassenhirarchie der Netzwerkmodule

In der abgeleiteten Klasse wurden einige Funktionen und Datenstrukturen überschrieben, und damit erweitert. Das betrifft die Funktionen: RecvPollThread(struct RecvStructT *) und InitSARRecv(). Die Datenstruktur TRecvStructSess wurde ebenfalls überschrieben, und dabei um einige Parameter für SDP-Pakete ergänzt.

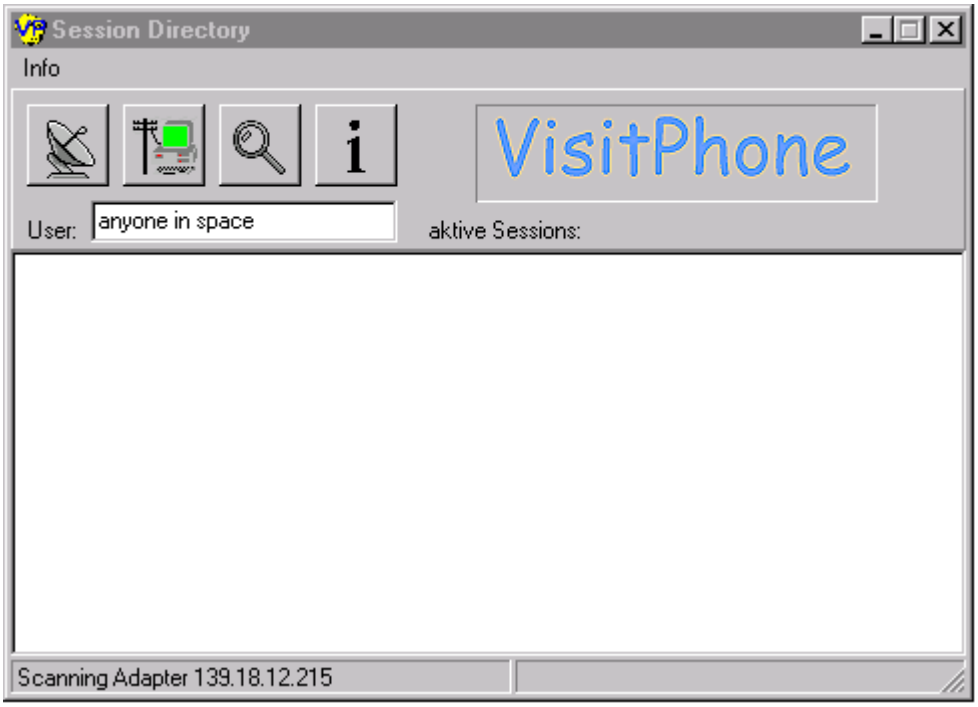


Abbildung 6.1.2: Die Darstellung der Sessionmanagement Testanwendung(Windows95)

Ergänzt wurde ein Thread: RecvPollThread(struct RecvStructT *). In dieser Konstellation ist das Modul in der Lage, die speziellen Poll-Pakete (Aufbau siehe nächster Abschnitt) zu empfangen. Vervollständigt wurde das Modul mit einem Testrumpf inklusive einer Oberfläche, um Praxistests durchführen zu können. Abbildung 6.1.2 zeigt die Oberfläche der beschriebenen Testanwendung unter dem Betriebssystem Windows95. Die Klassendefinition mit den dazugehörenden Datenstrukturen ist im Anhang A.5 und A.6 enthalten.

6.2 Aufbau des Sessionmanagementmoduls

Die Schnittstelle des modular aufgebauten Moduls läßt sich in vier Teile unterteilen: Multicast-Adressverwaltungs-Interface (im folgenden MAI), Befehlsinterface,

MBONE-Interface und dem Hauptinterface. Das MAI dient zur dynamischen Verwaltung von Multicastadressen. Folgende Funktionen ermöglichen dies:

Das Multicast-Interface (Auszug aus dem Anhang A5):

LPSTR GetFreeMulticastAddr();	- liefert eine freie Multicastadresse
int UnFreeMulticastAddr(LPSTR);	-markierte eine Multicastadresse als belegt
int FreeMulticastAddr(LPSTR);	-gibt eine Multicastdresse frei
BOOL IsFreeMulticastAddr(LPSTR);	-testet ob die betreffende Multicastadresse bereits in Verwendung ist

Das Befehlsinterface sichert die Erweiterungsmöglichkeit. Es dient zum Senden und Empfangen verschiedener Befehle über Multicast. Diese Befehle bestehen jeweils aus einem Kommando und zugehörigen Parametern. Detailliert beschrieben werden sie im nachfolgenden Absatz 6.3.
Das Hauptinterface bietet die elementaren Funktionen wie Initialisierung, Starten von Polling-Threads und Empfangsthreads.

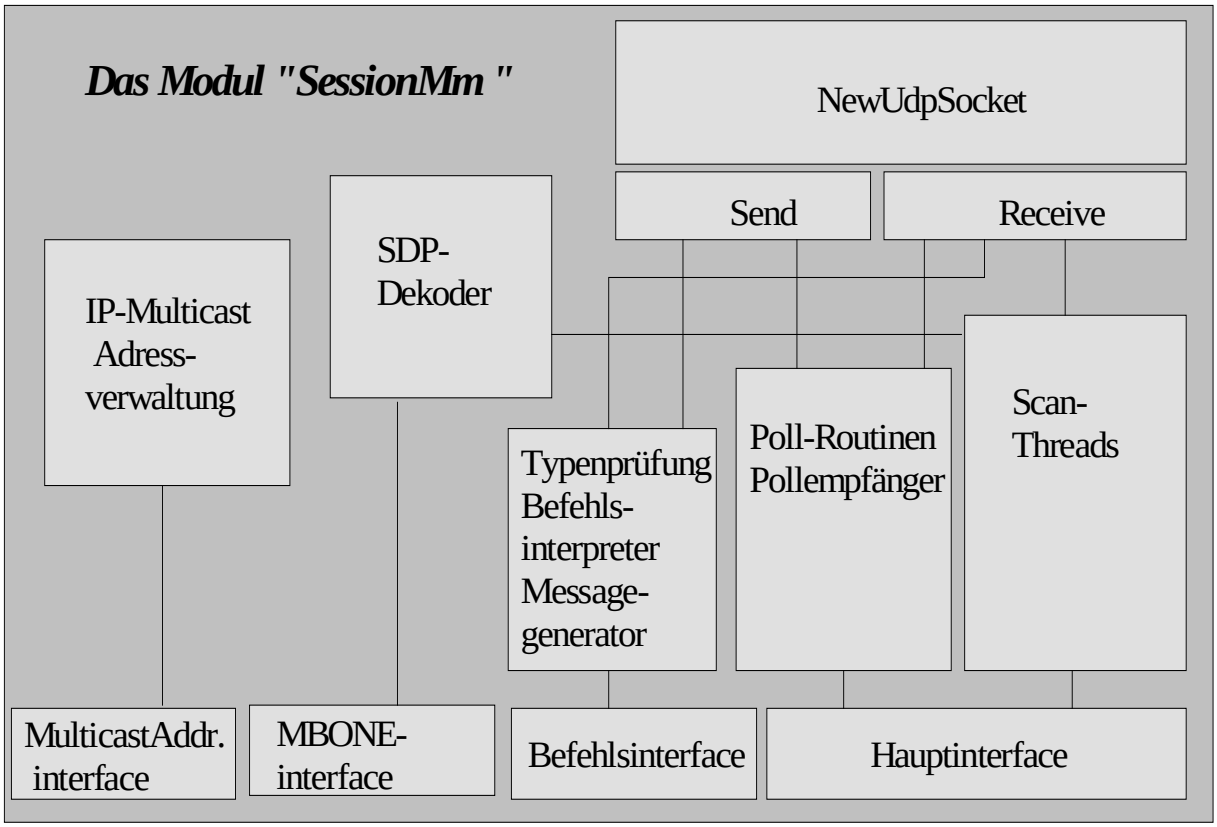


Abbildung 6.2.1: Der schematische Aufbau des Moduls „SessionMm“

Intern besitzt das Sessionmanagementmodul mehrere Instanzen der Klasse NewUdpSocket. Es realisiert die Anbindung an das darunterliegende Netzwerkinterface, hier: UDP.
Das MBONE-Interface verwaltet eine Liste der erkannten MBONE-Sessions. In jede dieser Sessions kann man eintreten (join). Dabei werden die Tools vic & vat gestartet. Das MBONE-Sessionpaket muß nach SDP dekodiert werden, um dann die Parameter für den Aufruf der Tools zu erzeugen.
Zusätzlich dazu kann man sich die Textparameter einer Session anzeigen lassen (info), die Liste komplett leeren (delete), oder sich Parameter des SDP-Strings anzeigen lassen.

Die Funktionen des MBONE Interfaces (Auszug aus dem Anhang A5):

```
int AddMboneSession(LPSTR);          -registriert Session
int JoinMboneSession(LPSTR);         -startet vic & vat
int InfoMboneSession(LPSTR);         -Info über die Session
int DeleteMboneSessions();           -Liste leeren
int SearchString(LPSTR buf,LPSTR arg,int a1,int a2,int opt); -SDP Dekoder
```

Die Funktion des Befehlsinterfaces wird im nachfolgenden Kapitel 6.3 genau beschrieben. Folgende Funktionen werden bereitgestellt:

Die Funktionen des Befehlsinterfaces (Auszug aus dem Anhang A5):

```
int SendMulticastCommand(LPSTR command,LPSTR parameter);
                                                  -sendet Kommando
int GetCommandParameter(int nr, LPSTR str);       -liefert Parameter
LPSTR GetCommand();                               -liefert Kommando
```

Werden Befehle von außen empfangen, generiert das Modul eine Message, welche an das ModulMain geschickt wird. Dieses kann dann mit dem Aufruf der Funktion GetCommand() ermitteln, welches Kommando empfangen wurde. Mit GetCommandParameter() kann es die Parameter desselben abfragen.
Eine schematische Gesamtdarstellung des Moduaufbaus zeigt Abbildung 6.2.1.

6.3 Die Kommandos des Befehlsinterfaces

Kommandos des Befehlsinterfaces können eine maximale Länge von 80 Zeichen ohne abschließende Null besitzen. Für die nachfolgenden Parameter gilt die gleiche Beschränkung. Es dürfen beliebig viele Parameter angehängt werden, jedoch muß ihre Anzahl fix und vorher bekannt sein. Die maximale Gesamtlänge beträgt ca. 64000 Zeichen. Bisher wurden folgende Kommandos festgelegt:

Kommando	Parameter1	Parameter2	Parameter3	Parameter4	Parameter5
GetIpByName	nameremote	ipbcal	namebcal	sessionname	sessiontyp
Sendip	nameremote	ip			
Rejectname	name	ip			
PollName	name	ip			
ReqFail	name				
SpeakRequest	name	ipbcal	ip		
Speak	name				
ChatString	string				

Tabelle 6.3.1: Die bereits verwendeten Kommandos des Befehlsinterfaces

Diesen Festlegungen können durch später eingegliederte Module des Systems weitere Kommandos ergänzt werden.
In diesem Rahmen bietet das Modul eine universalverwendbare Schnittstelle für shared Applications an. Shared Applications sind Anwenderprogramme, die gleichzeitig von mehreren Arbeitsplätzen bzw. Sitzungen aus, das Bearbeiten identischer Dokumente ermöglichen.
Neu eingegliederte Applikationen können beliebige Befehle über den Multicastbus senden und empfangen. Sie können ebenfalls die Adressauflösung der Nutzer und die bereits implementierten Mechanismen nutzen.

6.4 Der Vorgang der Namensauflösung

Jeder der Teilnehmer des Visiphone-Systems besitzt ein im Netzwerk eindeutiges Synonym der Form user@hostname. Dieser String wird in der Registry-Datenbank des Windows-Betriebssystems zwischengespeichert (vergleiche Abschnitt 3.9), und

bei jedem Programmstart von dort ausgelesen. Jede Änderung, jeder Programmstart und ebenfalls das erste Eingeben des Synonyms bewirken eine Anfrage des Typs „PollName". Dabei wird jedem anderen Sessionmanagemant der Name bekanntgegeben (unter Nutzung der Multicastmechanismen). Existieren andere Anwender mit gleichem Bezeichner, so senden sie eine Meldung des Typs „RejectName". Der neu hinzugekommene Teilnehmer wird dadurch aufgefordert, seinen Bezeichner zu ändern.
Durch diesen Mechanismus wird zunächst sichergestellt, daß jeder Name nur einmal im Netzwerk vorkommt.
Wählt nun einer der Teilnehmer den Start einer Einzelverbindung zu einem anderen Teilnehmer, so wird zunächst ein „IPRequestByName" generiert, und auf dem Multicastbus gesendet. Der Teilnehmer meldet sich mit „SendIp", und teilt dem „Anrufer" seine IP-Adresse mit. Nun ist dieser in der Lage seine Audio- und Videodaten über eine Unicastverbindung abzusenden. So wird für diese Einzelverbindung keine Multicastadresse verschwendet.

6.5 Die Konferenzschaltung (Floor-Steuerung)

Wärend einer laufenden Multicastsession können die empfangenden Teilnehmer dem Sender eine Anfrage schicken, bzw. ihm mitteilen, daß sie zu spechen wünschen. Dazu wird das Kommando „SpeakRequest" über den Multicastbus gesendet. Als Parameter enthält es den Namen des Teilnehmers, welcher zu sprechen wünscht, seine IP-Adresse und die IP-Multicastadresse der Session auf welche sich die Anfrage bezieht (siehe auch Abschnitt 6.3).
Bei dem sendenden Teilnehmer werden diese Anfragen in einer Warteschlange gespeichert. Vergibt er das Sprachrecht, müssen sich die übrigen Teilnehmer an den neuen Sprecher wenden.
Der sendende Teilnehmer verweigert entweder die Zuteilung des Sprachrechts oder schickt dem Anfragenden ein Kommando „Speak" (Parameter: Name des Sprechers, siehe auch Abschnitt 6.3).
Daraufhin wird das Senden des bisher sprechenden Teilnehmers unterbrochen, und beim Anfragenden startet automatisch die Applikation zum Senden, welche um Daten auszutauschen, die identische Multicastadresse wie bisher verwendet.

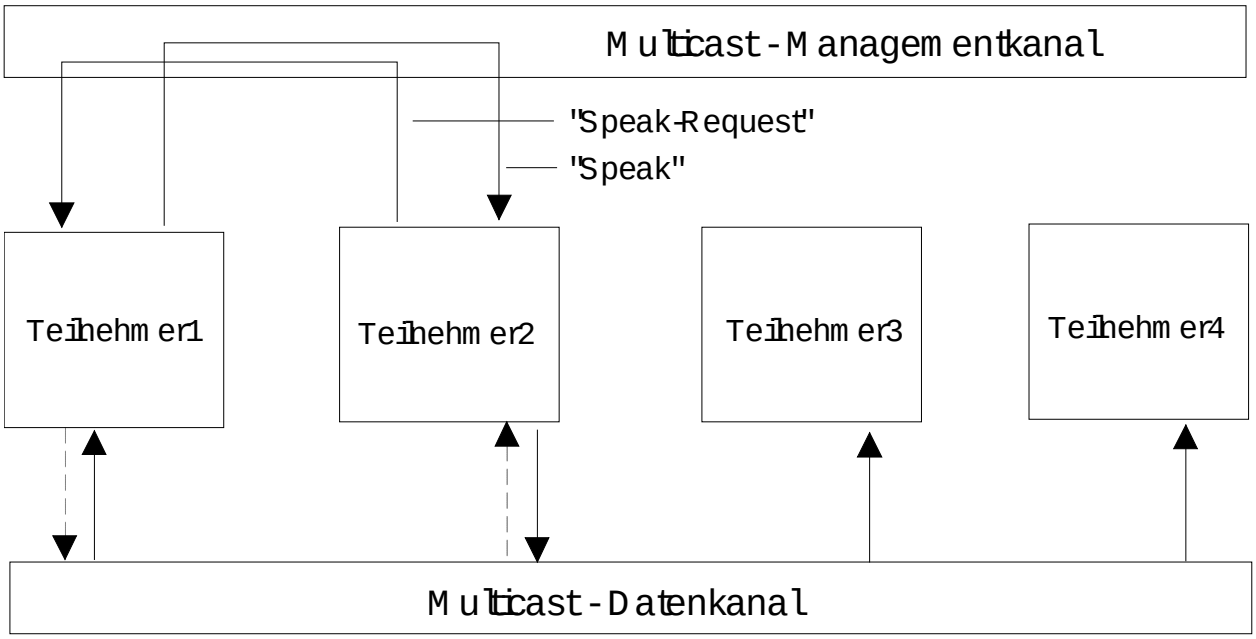


Abbildung 6.5.1: Der schematische Ablauf einer Übergabe des Sprachrechts bei Multicastsessions

Abbildung 6.5.1 zeigt diesen Vorgang in schematischer Darstellung. Auch hierbei ist der Managementkanal unabhängig vom Datenkanal angelegt.

6.6 Der SDP-Dekoder

Auf der Grundlage des im RFC2327 beschriebenen Standards werden die ankommenden MBONE-Pakete untersucht, und wesentliche Informationen herausgefiltert. Dieser Abschnitt erklärt das Vorgehen dabei. Der Aufbau der Pakete wurde im Abschnitt 4.1.8 beschrieben. Zunächst wird der SAP-Header abgeschnitten. Dann wird zuerst nach dem Namen der Session gesucht (String: „s=“), und dieser extrahiert. Nachfolgend wird aus dem mit „o=“ beginnendem Stringteil der TTL-Wert und die Standardadresse der Session ermittelt.
Nach diesem Vorgehen wird nach den Media-Daten, welche „m=“ folgen gesucht. Beispielsweise beginnt die Ankündigung von Audiodaten mit „m=audio...“. Dieser String enthält den Port des jeweiligen Mediums. Nachfolgend wird noch überprüft, ob im Mediateil eine weitere IP-Adresse angegeben wurde. Ist das der Fall, so überschreibt sie die vorher ermittelte, jedoch nur für dieses Medium.

Dieses Vorgehen wird für „m=video“ und „m=audio“ durchgeführt. Die ermittelten Werte werden in einem weiteren String für das spätere starten von vic und vat zwischengespeichert.
Wählt der Nutzer eine Anfrage des Typs „Info über Session“, wird nach dem Text „i=“ gesucht, und (wenn vorhanden) der dortige Textteil, welcher detaillierte Informationen der Session enthält, ausgegeben.

6.7 Der Konfigurationsteil

Einige der Daten und Parameter des Sessionmanagement-Moduls dürfen nicht statisch definiert sein, sondern müssen konfigurierbar gehalten werden. Zu diesem Zweck wurde ein Konfigurationsteil implementiert, welcher in der Lage ist, Parameter in der Registry-Datenbank des Windows-Betriebsystems zu speichern und selbige wieder zu lesen.
Dem Nutzer wird ermöglicht, durch ein Dialogfenster diese Werte zu ändern.

Abbildung 6.7.1: Die Konfigurationsmöglichkeiten als Nutzerinterface

Konkret dargestellt sind die änderbaren Parameter in Abbildung 6.7.1. Es sind Session Directory IP-Adresse (Multicast): 224.2.127.254, Session Directory IP-Port: 9875, MBONE Videotool: vic, MBONE Audiotool: vat, Whiteboard: mb, Whiteboard Datenport (mash-MediaBoard): 1235 und Visitphone IP-Datenport 2000.
Wichtig ist die Möglichkeit, die Adresse des Sessionmanagements und dessen Port freiwählen zu können. Voreingestellt sind hier die Werte des MBONE.

Durch das Nutzen anderer Adressen wird es möglich, für einzelne Abteilungen oder Nutzergruppen jeweils diskrete Auflistungen zu erhalten. Beispielsweise kann in einem firmeninternen Netz, welches keinen MBONE Anschluss benötigt, eine andere Adresse spezifiziert werden. Durch das Ändern der Portnummer kann man die Auflistung der MBONE-Sessions verhindern.

6.8 Eigenschaften und Testergebnisse

Die Eigenschaften des erstellten Systems wurden unter Verwendung der Betriebssysteme Windows95 und WindowsNT untersucht. Das System, inclusive der zur Bedienung notwendigen Testoberfläche, läuft unter beiden Betriebssystemen stabil.
Visitphone Videoübertragungen sind jedoch nur unter Windows95 möglich. Ursache dafür sind die Einschränkungen der VfW-Treiber unter WindowsNT (kein PCI-Overlay möglich).
MBONE Sessions können unter beiden Betriebssystemen verfolgt werden. Hier ist audiovisuelle Kommunikation innerhalb mehrerer Sessions, gleichzeitig möglich. (Durch die H.261 Softwaredekoder kann man mehrere Videoströme in begrenzter Auflösung verfolgen).
Die vom Sessionmanagement verursachte Prozessorlast liegt dauerhaft unter 1% (Testsystem: Pentium 233 Mhz, 64MB RAM, Meßinstrument: NT-Taskmanager).
Es sind somit keine vom Sessionmanagement gegebenen Einschränkungen für zusätzliche Applikationen vorhanden. Zu beachten ist lediglich, daß bei einer gestarteten Videoübertragung die Prozessorlast erheblich ansteigt, bei obigem Testsystem ca. auf 66% (Vergleiche Abschnitt 4.3.9).
Für die Übertragung von Audiodaten bei vollduplex-Verbindungen ist es notwendig einen vollduplex-Soundtreiber zu installieren. Für neuere Soundkarten (zB.: Creative-AWE32, Creative-PCI64) liegt dieser der Karte bei, für ältere Soundkarten (Creative Soundblaster16) ist er im Internet erhältlich (zB.: www.creativelabs.com).
Es vergehen ca. 3 Minuten bis die Mehrzahl der MBONE-Sessions erkannt wurden. Das resultiert aus dem (vorwiegend in MBONE verwendeten) Poll-Delay von 6 Minuten. Das heißt es wird alle 6 Minuten ein SAP/SDP-Paket gesendet.
Das entwickelte Sessionmanagement-System versendet seine Poll-Pakete

periodisch aller 6 Sekunden. Für das Erkennen aller im Netz vorkommenden Visitphone-Sessions ist also maximal eine Zeit von 6 Sekunden einzuplanen.

7 Zusammenfassung

Unter Bezugnahme auf die Eigenschaften der Systeme MBone-SDR, Microsoft Netmeeting und CuSeeMe wurde ein flexibel einsetzbares Sessionmanagementmodul entwickelt, welches speziell im Einsatz mit dem Testsystem Visitphone angewendet wurde. Es unterstützt eine dem MBONE kompatible Sessionverwaltung. Video- und Audiosessions des MBONE werden korrekt erkannt, und können zur Kommunikation verwendet werden. Darüber hinausgehend werden Visitphone-Sessions dem MBONE bekanntgegeben und anderen Visitphone-Nutzern angezeigt. Diese werden als spezielle Sessions geführt, welche ebenfalls nur mit einem Visitphone-Sessionmanagement zur Kommunikation verwendet werden können.
Eingesetzt werden die Protokolle SDP und SAP des MBONE.
Es besteht die Möglichkeit der Einzelverbindung zwischen zwei Nutzern, welche unter Pseudonymen der Form user@hostname den Systemen bekannt sind.
Unter den Bedingungen einer Multicastsession besteht die Möglichkeit einer Konferenzschaltung.
Die Erweiterbarkeit des Moduls wurde durch die Implementierung einer Befehlsschnittstelle gesichert. Auf sie können beliebige Erweiterungsmodule oder Anwendungen aufsetzen, und die bereits realisierten Kommunikationsmechanismen nutzen. Implementiert wurde dies in einfachem C++, ohne Verwendung von fertigen Klassenbibliotheken (zB. MFC). So werden spätere Probleme vermieden, welche durch Veränderung oder Austausch der Entwicklungswerkzeuge und Enwicklungsumgebungen auftreten können.
Getestet wurde das Modul durch eine parallel dazu entwickelte Oberfläche, welche in Verbindung mit den bestehenden Visitphone-Kommunikationsteilen und dem Sessionmanagementmodul eine komplette Videokonferenzanwendung bereitstellt.
Eingefügt wurden die zur Kommunikation mit dem MBONE notwendigen Tools „vic“ und „vat“. Die Whiteboardanwendung „mash-MB“ stellt eine Ergänzung zur Kommunikation in Text und Grafikform dar.
Das System erlaubt Videoverbindungen in Pal-Qualität und Audioverbindungen in CD-Qualität. Es können Rundsendungen unter Nutzung der Multicastmechanismen stattfinden, oder Einzelverbindungen zwischen den Teilnehmern geschaltet werden. Parallel dazu sind Whiteboardverbindungen möglich.

8 Ausblick

Jegliche Softwaresysteme sind ständigen Änderungen unterworfen. Fehlerkorrekturen, Verbesserungen, Anpassungen an sich verändernde Hardware und Betriebsysteme oder neue Anforderungen der Nutzer bedingen weitere Entwicklungsarbeiten.
Für das Visitphone-Videokonferenzsystem, inclusive seinem Sessionmanagement, kann man bereits eine Reihe von Fakten nennen, welche sich bereits als notwendige weitere Entwicklungen abzeichnen. Kurz gefaßt sind das:

- Anpassung des Systems an die Nutzung von native ATM
- Verhandlung von Quality of Service (QoS)–Vereinbarungen
- Integration einer Komponente für schmalbandige Verbindungen (ISDN)
- Entwicklung von geeigneten Verschlüsselungsalgorithmen für Video und Audio
- Implementierung eines Mixermoduls der Audiodaten, welches beim mehr als einer empfangenen Session nötig wird
- Einbindungen für Video on Demand Services und Dateitransfer
- Anpassung an den Einsatz unter WindowsNT, z.B.: durch die Verwendung des in [FIEBIG 98] entwickelten DirectShow Interfaces

Dazu gibt es einige derzeit ungelöste Probleme zu nennen:
- Bisher existiert keine Hardwareplattform welche Multicastverbindungen direkt über AAL5 ermöglicht. Dazu ist das Signalisierungsprotokoll UNI4 nötig, welches die bisher verwendeten ATM-Karten nicht unterstützen. Denkbar wäre eine Emulation der Multicastmechanismen mit einer transparenten Serverlösung.
- Um eine Anbindung über ISDN-Netze zu ermöglichen, ist eine Reduzierung der Datenrate auf weniger als 64 KBit/s pro Kanal notwendig. Dafür existiert derzeit keine Lösung im Visitphone-System. Die verwendeten MJPG-Kompressionskarten erreichen minimal eine Datenrate von etwa 500 KByte/s.
- Eine Verschlüsselung der Video und Audiodaten stellt ein extrem zeitkritisches Problem dar. Angedacht wurde eine nicht vollständige Verschlüsselung der Daten, bereits das Verschlüsseln wichtiger Headerinformationen würde als Schutz ausreichen.

Abbildungen und Tabellen

Abbildung 3.1.1: Darstellung des Datenflusses bei MJPEG 7
Tabelle 3.2.1: Die verschiedenen Audioqualitäten bei Wave-Kodierungen 8
Abbildung 3.3.1: Konzeption des Verfahrens des nachträglichen RGB-mischens 9
Abbildung 3.3.2: Datenfluß bei PCIO verbaytechnik 10
Abbildung 3.4.1: Die verschiedenen PAL-Bereiche 12
Tabelle 3.4.1: Videonormen im Vergleich 13
Abbildung 3.6.1: Aufbau eines IP-Paketes 14
Abbildung 3.6.2: Aufbau eines UDP-Paketes 15
Abbildung 3.6.3: Die Zuordnung der Protokolle IP, UDP und TCP zu den Schichten des OSI-Modells 15
Abbildung 3.7.1: Aufbau eines RTP Headers 17
Abbildung 3.8.1: Schematische Darstellung des Datenflußes bei Unicast- und Multicastsendungen 18
Tabelle 3.8.1: reservierte Multicastadressen 19
Abbildung 3.8.2: Darstellung der unterschiedlichen Internet Adressfamilien 19
Abbildung 4.1.1: Bildschirmdarstellung des Videotools VIC 23
Abbildung 4.1.2: Bildschirmdarstellung des Audiotools VAT 24
Abbildung 4.1.3: Bildschirmdarstellung des SDR-Tools für Windows 25
Abbildung 4.1.4: Bildschirmdarstellung der spezifischen Daten einer Session 26
Abbildung 4.1.5: Die Bildschirmdarstellung der Stream Einstellungen 26
Abbildung 4.1.6: Die UNIX Portierung 26
Tabelle 4.1.1: Bitraten bei Videoübertragungen 28
Tabelle 4.1.2: Bitraten bei Audioübertragungen 29
Abbildung 4.1.7: Darstellung des Aufbaus eines SAP Headers 31
Abbildung 4.2.1: Benutzerinterface des CuSeeMe von White Pine 36
Abbildung 4.2.2: Benutzerinterface des Systems Microsoft Netmeeting 36
Abbildung 4.2.3: Der Protokollstapel von MS-Netmeeting 37
Tabelle 4.3.1: Videokarten im Vergleich 42
Abbildung 4.3.1: Der Aufbau des Paketkopfes 43
Abbildung 4.3.2: Die Module des Visitphone Systems 44
Abbildung 4.3.3: Das Benutzerinterface des Senders des Visitphone-Systems 45
Abbildung 4.3.4: Das Benutzerinterface des Empfängers des Systems 46

Abbildung 4.3.5: Datenrate in Abhängigkeit zur Kompression 46
Abbildung 4.3.6: Fehlerraten in Abhängigkeit zur Datenrate bei Visitphone 47
Abbildung 5.3.1: Die CCCP Konzeption 53
Abbildung 5.7.1: Das Benutzerinterface des Mash-Mediaboards 57
Abbildung 5.8.1: Schematisch: Angliederung von Shared Applications 58
Abbildung 6.1.1: Die Klassenhierarchie der Netzwerkmodule 59
Abbildung 6.1.2: Die Darstellung der Sessionmanagement Testanwendung(Windows95) 60
Abbildung 6.2.1: Der schematische Aufbau des Moduls "SessionMm" 61
Tabelle 6.3.1: Die bereits verwendeten Kommandos des Befehlsinterfaces 63
Abbildung 6.5.1: Der schematische Ablauf einer Übergabe des Sprachrechts bei Multicastsessions 65
Abbildung 6.7.1: Die Konfigurationsmöglichkeiten als Nutzerinterface 66

Abkürzungsverzeichnis

AAL5	ATM Adaption Layer 5
API	Application Programming Interface
ARP	Adress Resolution Protocol
ATM	Asynchronous Transfer Mode
CCCP	Conference Control Channel Protocol
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
DCT	Diskrete Kosinus Transformation
GDI	Graphical Desktop Environment
HTDV	High Definition Television
IP	Internet Protocol
ISDN	Integrated Service Digital Network
JPEG	Joint Photographic Expert Group
LAN	Local Area Network
MBONE	Multicast Backbone
MFC	Microsoft Foundation Classes
MJPEG	Motion-Joint Photographic Expert Group
MPEG	Motion Picture Expert Group
PAL	Phase Alternation Line
PCI	Peripheral Component Interconnect
PCM	Pulse Code Modulation
RGB	Rot-Grün-Blau
RFC	Request For Comments
RTP	Realtime Transport Protocol
SAP	Session Announcement Protocol
SDP	Session Description Protocol
TTL	Time To Live
UDP	User datagram Protocol
VfW	Video for Windows
VGA	Video Graphics Adapter

Literaturverzeichnis

[CCCP94] M. Handley, I. Wakeman, „Conference Control Channel Protocol", University College London, 1994

[DIGVID97] B. G. Haskell, A. Puri, and A. N. Netravali, „Digital Video an Introduction to MPEG-2", 1997

[FAIL96] P. Fail „MBONE , Multimedia-Konferenzen über Internet IP-Multicast, Software Tools", RUS Universität Stuttgart, 1996

[FIEBIG98] M. Fiebig, Diplomarbeit in Bearbeitung, „Untersuchung der Eignung der Matrox DigiMotion für Videoconferencing", Universität Leipzig, 1998

[FRANKE98] M. Franke, Diplomarbeit in Bearbeitung, „Entwicklung einer Native-ATM Schnittstelle für Visiphone", Universität Leipzig, 1998

[KECK97] M. Keck, Vortrag „Arbeitsgruppe Videokommunikation" Universität Leipzig, 1997

[KUMAR96] V. Kumar, "MBone Interactive Multimedia on the Internet", 1996, USA

[MILDE95] T. Milde, „Videokompressionsverfahren im Vergleich", 1995 dpunkt-Verlag

[NEUMANN96] F. Neumann, Uni Oldanburg, Vortrag, „Bildkompression mit MJPEG und MPEG", 1996

[NGUYEN98] G. Nguyen, Diplomarbeit, „Entwicklung eines Videotransportprotokolls über UDP", Universität Leipzig, 1998

[OPENNET96] P. Fail, Vortrag „MBONE IP-Multicast", 1996

[RTP95] Busse, Deffner, Schulzrinne, „Dynamic QoS Control of Multimedia Applications based on RTP", GMD Fokus, 1995

[TROSS97] D. Trossen, T. Helbig RWTH Aachen, Vortrag, „Die ITU-T Standard-Familie T.120 als Basis für verteilte Mehrbenutzer-anwendungen", ITG/GI Fachtagung 1997

[WAGEN97] Wagenknecht, Uni Zittau, Vortrag, „Diskrete FOURIER-Transformation (DFT)", 1997

[CODECS98] Terran Interactive, Inc, 1995-98, http://www.terran-int.com/CodecCentral/Codecs/

[JPGTUT] W. Ray, „JPG-Tutorial", http://dynamo.ecn.purdue.edu/~ace/jpeg-tut/jpegtut1.html

[MBONE2] P. Dykstra, Informationssammlung MBONE, http://info.arl.mil/ARL-Directorates/ASHPC/HPCD/MBONE/

[NETMEET] Netmeeting, Funktionen und Features (Microsoft), http://www.microsoft.com/products/prodref/113_ov.htm

[CCIR] MPEG Iformations Collection, L. Filippini, http://www.crs4.it/~luigi/MPEG/ccir601.html

[CUSEEME] White Pine – Produktspezifikation /CU-SeeMe Homepage, http://www.wpine.com/Products/CU-SeeMe/

[HANDLEY] Mark Handley, Department of Computer Science, University College London, http://mice.ed.ac.uk/mice/archive/sdr.html

[KRAEMER] Diplomarbeit „Desktop Videokonferenzing“, http://www.rhrk.uni-kl.de/~mkraemer/mkraemer/diplom-/videoconf/index.html

[MBONEDR] „Mbone-Software-Sammlung“, Technische Universität Dresden, http://www-mm.urz.tu-dresden.de/mbone/software.html

[SUND97] Sundance Multiprocessor Technology Ltd., E&OE, 1997, http://www.sundance.com/pci.htm

[WEBMBONE] Deutsche Hompepage des MBONE, http://www.mbone.de/

[ZINKE] „Bitraten bei Audio“, http://www.fh-friedberg.de/fachbereiche/e2/telekom-labor/zinke/mmk/mmk110.htm

[SAPDRAFT] „draft-ietf-mmusic-sap-00.ps“ Internet Draft über SAP, IETF, Mark Handley 1996

[RFC1700] Assigned Numbers, Request For Comments der IETF (Internet Engineering Task Force)

[RFC1889] Realtime Transport Protokoll (RTP), Request For Comments der IETF (Internet Engineering Task Force)

[RFC2327] Session Description Protocol (SDP), Request For Comments der IETF (Internet Engineering Task Force)

Anhang

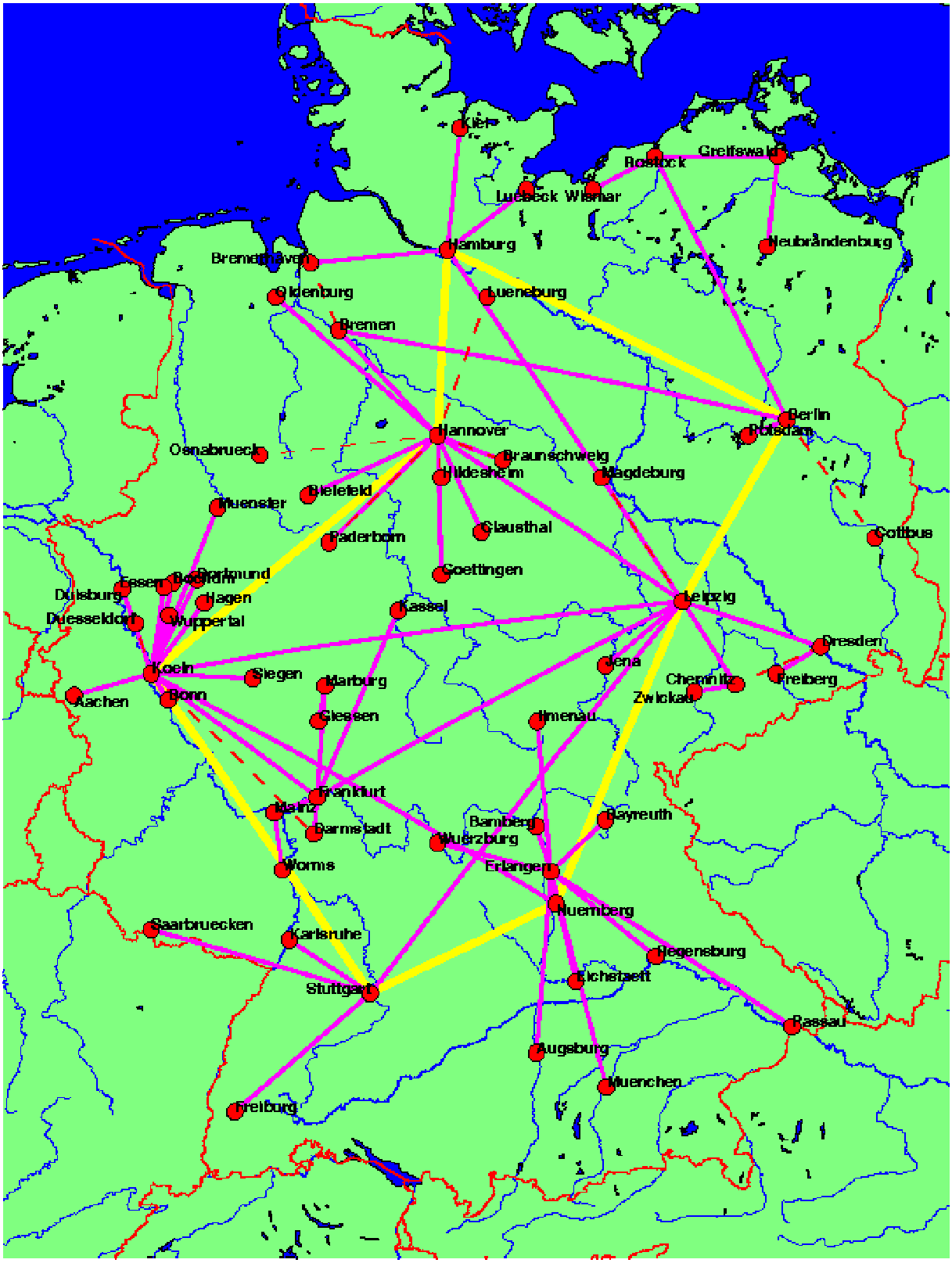
A1 Header des Audiodatenformats WAV

Offset Hex	Länge Byte	Festwert Hex	Bedeutung	Bemerkung
00-03	4		RIFF	WAV-Kennung
04-07	4		Dateilänge	
08-0E	7		WAVEfmt	WAV-Kennung
0F	1	20	Leerzeichen	
10-13	4		Länge des nächsten Blocks	
14	1	01	Formattyp	LSB
15	1	00	Formattyp	MSB
16	1		Anzahl der Kanäle	LSB
17	1		Anzahl der Kanäle	MSB
18-1B	4		Einfache Sampleclock	
1C-1F	4		Gesamt-Sampleclock[1]	
20	1		Gesamtdatenbyte/Sample	LSB
21	1		Gesamtdatenbyte/Sample	MSB
22	1		Auflösung/Sample	LSB
23	1		Auflösung/Sample	MSB
24-27	4	‚data'	Kennung für Datenanfang	
28-2B	4		Gesamtanz. Datenbytes	
2C			Erstes Datenbyte	LSB bei 16-Bit-Auflösung

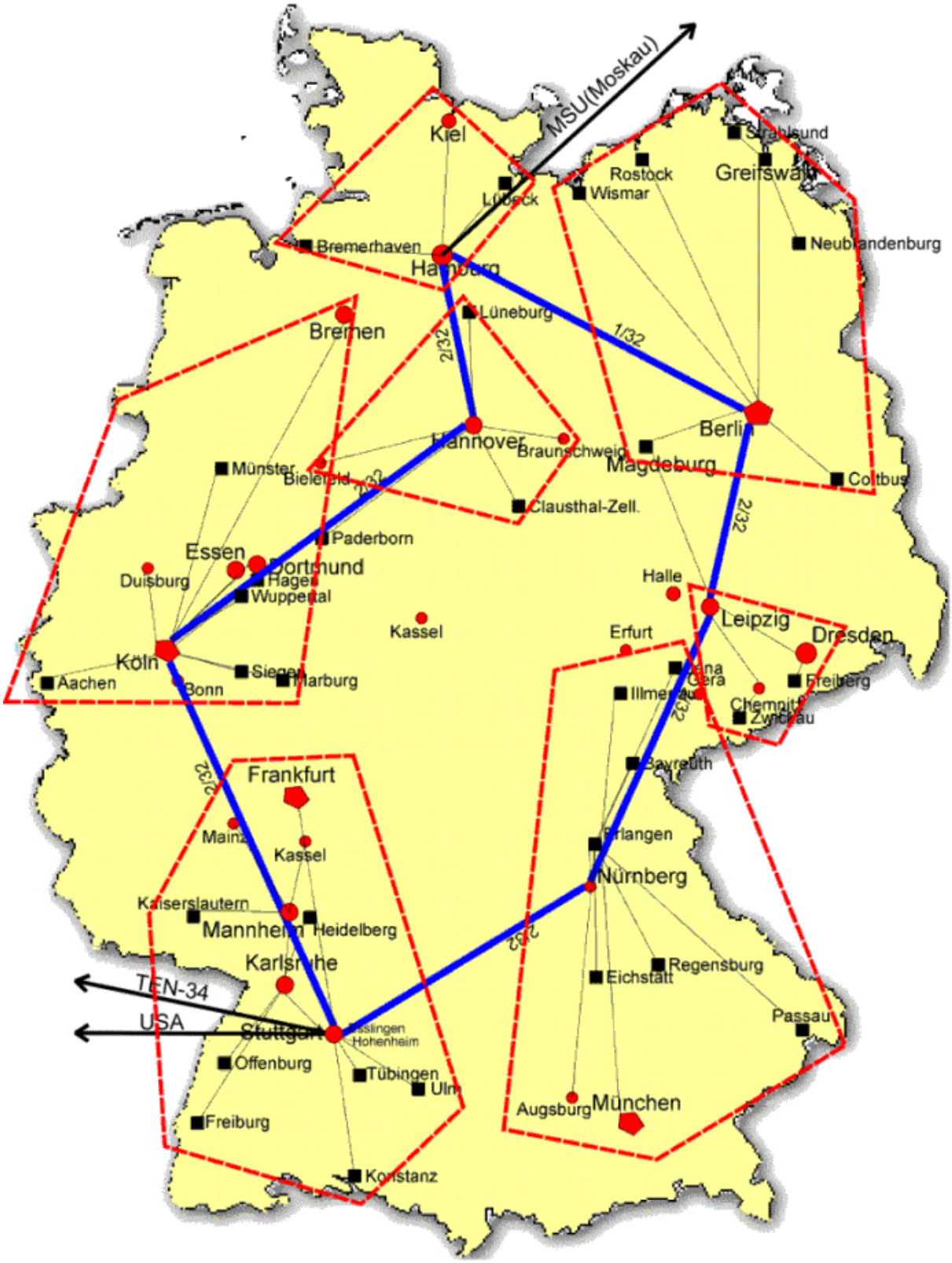
[NGUYEN98]

[1] Einfache Sampleclock*Auflösung/Sample*Anzahl Kanäle

A2 Die MBONE Typologie Deutschlands

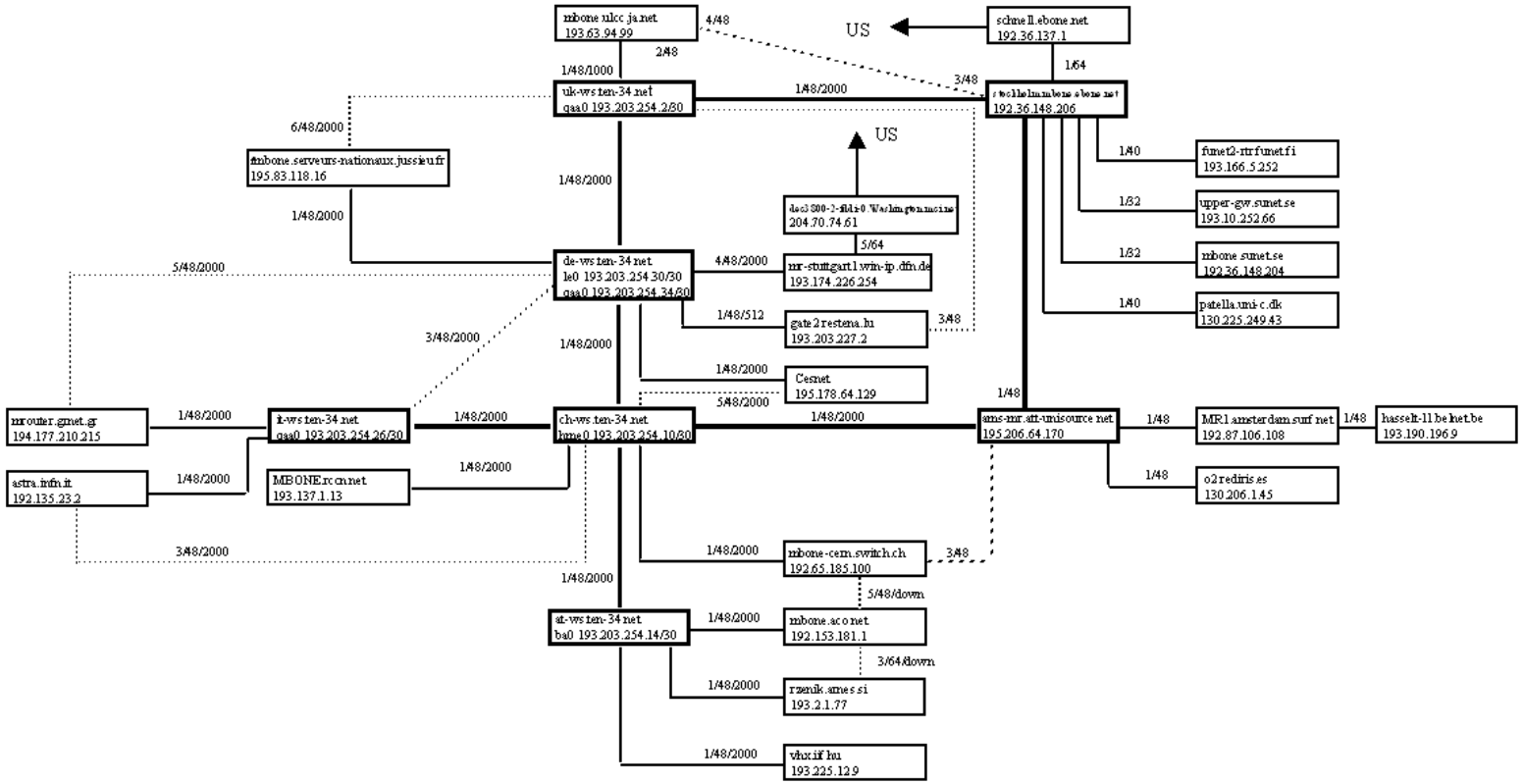


Mbone-DE
Administrative Adressbereiche
(B-WIN Region / 239.193.0.0)



TEN-34 MBONE Topology

25 June 1998



Jan Novak
mail changes to: rumc@dante.org.uk

A3 Assigned Multicast Addresses

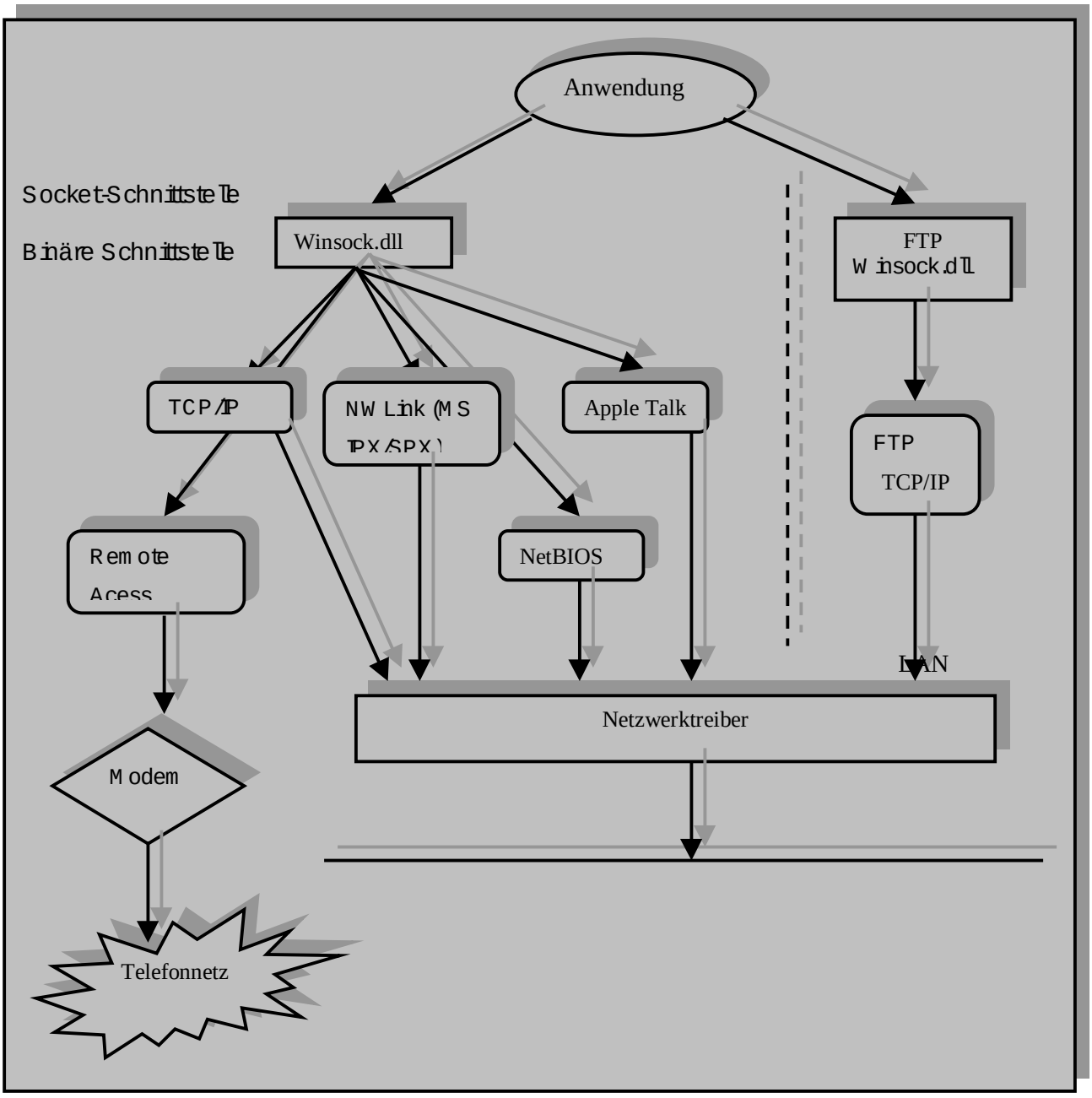
From RFC 1700, Assigned Numbers, October 1994.

224.0.0.0 Base Address (Reserved)	[RFC1112,JBP]
224.0.0.1 All Systems on this Subnet	[RFC1112,JBP]
224.0.0.2 All Routers on this Subnet	[JBP]
224.0.0.3 Unassigned	[JBP]
224.0.0.4 DVMRP Routers	[RFC1075,JBP]
224.0.0.5 OSPFIGP OSPFIGP All Routers	[RFC1583,JXM1]
224.0.0.6 OSPFIGP OSPFIGP Designated Routers	[RFC1583,JXM1]
224.0.0.7 ST Routers	[RFC1190,KS14]
224.0.0.8 ST Hosts	[RFC1190,KS14]
224.0.0.9 RIP2 Routers	[GSM11]
224.0.0.10 IGRP Routers	[Dino Farinacci]
224.0.0.11 Mobile-Agents	[Bill Simpson]
224.0.0.12-224.0.0.255 Unassigned	[JBP]
224.0.1.0 VMTP Managers Group	[RFC1045,DRC3]
224.0.1.1 NTP Network Time Protocol	[RFC1119,DLM1]
224.0.1.2 SGI-Dogfight	[AXC]
224.0.1.3 Rwhod	[SXD]
224.0.1.4 VNP	[DRC3]
224.0.1.5 Artificial Horizons - Aviator	[BXF]
224.0.1.6 NSS - Name Service Server	[BXS2]
224.0.1.7 AUDIONEWS - Audio News Multicast	[MXF2]
224.0.1.8 SUN NIS+ Information Service	[CXM3]
224.0.1.9 MTP Multicast Transport Protocol	[SXA]
224.0.1.10 IETF-1-LOW-AUDIO	[SC3]
224.0.1.11 IETF-1-AUDIO	[SC3]
224.0.1.12 IETF-1-VIDEO	[SC3]
224.0.1.13 IETF-2-LOW-AUDIO	[SC3]
224.0.1.14 IETF-2-AUDIO	[SC3]
224.0.1.15 IETF-2-VIDEO	[SC3]

224.0.1.16 MUSIC-SERVICE	[Guido van Rossum]
224.0.1.17 SEANET-TELEMETRY	[Andrew Maffei]
224.0.1.18 SEANET-IMAGE	[Andrew Maffei]
224.0.1.19 MLOADD	[Braden]
224.0.1.20 any private experiment	[JBP]
224.0.1.21 DVMRP on MOSPF	[John Moy]
224.0.1.22 SVRLOC	
224.0.1.23 XINGTV	
224.0.1.24 microsoft-ds	
224.0.1.25 nbc-pro	
224.0.1.26 nbc-pfn	
224.0.1.27-224.0.1.255 Unassigned	[JBP]
224.0.2.1 "rwho" Group (BSD) (unofficial)	[JBP]
224.0.2.2 SUN RPC PMAPPROC_CALLIT	[BXE1]
224.0.3.000-224.0.3.255 RFE Generic Service	[DXS3]
224.0.4.000-224.0.4.255 RFE Individual Conferences	[DXS3]
224.0.5.000-224.0.5.127 CDPD Groups	[Bob Brenner]
224.0.5.128-224.0.5.255 Unassigned	[IANA]
224.0.6.000-224.0.6.127 Cornell ISIS Project	[Tim Clark]
224.0.6.128-224.0.6.255 Unassigned	[IANA]
224.1.0.0-224.1.255.255 ST Multicast Groups	[RFC1190,KS14]
224.2.0.0-224.2.255.255 Multimedia Conference Calls	[SC3]
224.252.0.0-224.255.255.255 DIS transient groups	[Joel Snyder]
232.0.0.0-232.255.255.255 VMTP transient groups	[RFC1045,DRC3]

These addresses are listed in the Domain Name Service under MCAST.NET and 224.IN-ADDR.ARPA.

A4 Das Windows Socket Interface im Überblick



A5 Klassendefinition des Moduls „Sessionmm“

```
class SessionManagement
{
private:
     LPSTR SendBuff;                 // free mem
    LPSTR Command;              // Commandstring
    LPSTR Parameter[20];      // Parameterstrings
     unsigned long ipAddr;     // associated IP
    UdpSockSess *DirSocket;    // Socket for scanning
    UdpSockSess *DirSocketPoll;     // Socket for polling
    int adapter;           // adapter of this session
    int sockport;          // port of this session
    Mip *MulticastAddr;    // inital pointer of Mip-list
    MboneSession *MboneSessions;   // inital Pointer of MBONE
Sessions

Send(int plen);   // internal routine for packet-sending

public:
   LPSTR ScanBuffer ;      // Buffer for scan-data

SessionManagement();
~SessionManagement();

int StartScan(LPSTR adapter, LPSTR ipmulticast,int port,int
adapternr,HWND hwnd,unsigned long msg,int (__stdcall *
routine)(WPARAM,LPARAM));
   // starts thread waiting on Port for Session-Packets ,
inits Callback routine
int StartPoll(LPSTR adapter, LPSTR ipmulticast,int port,int
adapternr);
   // inits Socket for Polling

LPSTR GetFreeMulticastAddr();
   // returns the next free Multicast-IP
int   UnFreeMulticastAddr(LPSTR);
   // mark IP as used
int   FreeMulticastAddr(LPSTR);
   // mark IP as unused
BOOL  IsFreeMulticastAddr(LPSTR);
   // returns TRUE if IP unused


int   AddMboneSession(LPSTR);
     // add Session string
int   JoinMboneSession(LPSTR);
   // start vic & vat  LPSTR is sessionname
int   InfoMboneSession(LPSTR);
   // shows informations about the session
int   DeleteMboneSessions();
```

```
   // delete all MBONE Sessions
LPSTR SearchString(LPSTR buf,LPSTR arg,int a1,int a2,int opt);
   // eliminates strings from SDP, writes in buf, returns
position


int  SendMulticastCommand(LPSTR command,LPSTR parameter);
  //  sends a command to all users over the network
int GetCommandParameter(int nr, LPSTR str);
  //  returns the parameter of the incoming command
LPSTR GetCommand();
  // returns the actual Command string


int  Poll(Msession *sess);
   // sends a poll-packet
int  PollEnd(Msession *sess);
   // sends a poll-packet that Session ends

};
```

A6 Spezielle Datenstrukturen

```
struct Msession {   // each Session has name typ ip ...
LPSTR Name;
LPSTR Typ;
LPSTR Ip;
LPSTR LocalIp;
};

struct Ssession {   // for Single - Sessions
LPSTR Name;
LPSTR UserLocal;
LPSTR UserRemote;
LPSTR Typ;
LPSTR Ip;
LPSTR LocalIp;
};

struct Mip {          // for a list of multicast ip’s
LPSTR Ip;
Mip *next;
};

struct MboneSession {          // for a list of MBONE sessions
LPSTR session;
MboneSession *next;
};
```

Erklärung

Ich versichere, daß ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Leipzig, März 1999 .....