

**UNIVERSITÄT
BREMEN**



Entwurf und Implementierung eines
H.323-Gatekeepers zur Ressourcenverwaltung
und Zugangsregelung für IP-Telefonie-Dienste

Stefan Prella¹
Fachbereich 3 Informatik
Universität Bremen

28. Oktober 1999

¹prelle@tzi.org

Inhalt

1	Einleitung	1
1.1	Telefonie	1
1.1.1	Die Erfindung des Telefons	1
1.1.2	Vermittlungstechnik	1
1.2	Das Internet	4
1.2.1	Paketbasierte Datenübertragung	4
1.2.2	Netzkopplung und IP	5
1.2.3	Transportprotokolle	6
1.2.4	Traditionelle Dienste im Internet	6
1.2.5	Multimedia und Echtzeit in IP-Netzen	7
1.3	IP-Telefonie	7
1.3.1	Architektur eines IP-Telefoniesystems	8
1.3.2	Möglichkeiten der IP-Telefonie	11
1.3.3	Stand der IP-Telefonie	11
1.4	Internationale Gremien	13
1.4.1	ITU-T	14
1.4.2	IETF	14
1.5	Die Arbeitsgruppe Rechnernetze	14
2	Relevante Standards	17
2.1	ITU-T-Standards/Empfehlungen	18
2.1.1	H.323 v2	18
2.1.2	H.225.0	24
2.1.3	H.245	26
2.1.4	Q.931	26
2.2	Empfehlungen und Entwürfe der IETF	27
2.2.1	IETF-Draft SIP	27
2.2.2	IETF-Draft TBGP	28
2.2.3	IETF-Draft GLP	30
2.2.4	IETF-Draft PGRP	30
2.2.5	IETF-Draft Call Processing Language (CPL)	31
2.2.6	IETF-Drafts zum Message Bus	33
2.3	Zusammenfassung	37
3	Funktionalität des Gatekeepers	39
3.1	Zugangskontrolle	39
3.2	Adreßumsetzung	40
3.2.1	Adressen, die Personen kennzeichnen	42

3.2.2	Adressen, die Funktionen kennzeichnen	44
3.2.3	Ablauf der Adreßumsetzung	45
3.3	Call-Routing	46
3.4	Ressourcenverwaltung	47
4	Die Architektur des Gatekeepers	49
4.1	Die Module	49
4.1.1	Das H323-Modul	50
4.1.2	Die Policy-Module	50
4.1.3	Das Datenbank-Modul	51
4.1.4	Das API/GUI-Modul	54
4.2	Mögliche Erweiterungen	55
4.2.1	Protokollierung und Abrechnung	55
4.2.2	Anrufbearbeitung	55
4.2.3	Externe User-Location	56
4.2.4	Gateway Location	56
4.2.5	Mehrwertdienste	56
4.2.6	Andere Administrationsschnittstelle	58
4.3	Interne Abläufe und Entscheidungen innerhalb des H.323-Moduls	59
4.3.1	Gatekeeper-Discovery	59
4.3.2	Registrierung eines Endpunktes	59
4.3.3	Abmelden beim Gatekeeper	61
4.3.4	Erbitten der Anruferlaubnis	62
4.3.5	Bandbreite eines Gesprächs ändern	63
4.3.6	Gesprächsende signalisieren	64
4.3.7	Adresse auflösen	66
4.3.8	Status melden	66
4.4	Interne Abläufe und Entscheidungen in den Policy-Modulen . . .	67
4.4.1	Zugangsberechtigung für einen Endpunkt prüfen	67
4.4.2	Zugangsberechtigung für einen Benutzer prüfen	68
4.4.3	Prüfen, ob der Anruf erlaubt ist	68
4.4.4	Bandbreitenforderung prüfen	70
5	Implementierung	73
5.1	Programmiersprachen	73
5.2	Das Datenbanksystem	73
5.2.1	Bekannte Endpunkte/Netze - endpoints	73
5.2.2	Kostenlose Adressen - freenumbers	74
5.2.3	Funktionsadressen - functions	74
5.2.4	Gruppendefinitionen - privdef	74
5.2.5	Nutzerdaten - user	75
5.3	Das H323-Modul	76
5.4	Die Benutzungsschnittstelle	77
5.4.1	Allgemeine Einstellungen	77
5.4.2	IP-basierte Zugangskontrolle	78
5.4.3	Gruppenverwaltung	79
5.4.4	Benutzerverwaltung	80
5.4.5	Funktionsadressen-Verwaltung	81
5.4.6	Registrierte Benutzer und aktive Gespräche	82
5.5	Der virtuelle MBus	83

6	Verwendung des Gatekeepers	85
6.1	Systemanforderungen	85
6.2	Installation	85
6.3	Fehlerbehebung	86
6.4	Andere Datenbanken	86
7	Zusammenfassung und Ausblick	89
7.1	Performance	89
7.2	Interoperabilität	90
7.3	Naheliegende Erweiterungen	90
7.4	Ausblick	92
A	Konfigurationsdateien	95
A.1	System-Konfiguration	95
A.2	MBus-Konfiguration	96
B	Verwendete MBus-Nachrichten	97
B.1	Generelle Anmerkungen	98
B.2	Das H323-Modul	100
B.2.1	Steckbrief	100
B.2.2	An-/Abmelden von Endpunkten	101
B.2.3	Anrufanfang und -ende signalisieren	101
B.2.4	Adreßauflösung	102
B.2.5	Sonstiges	102
B.3	Datenbank-Modul	103
B.3.1	Steckbrief	103
B.3.2	mdb-Kommandos	103
B.3.3	Spezielle Kommandos	106
B.4	Zugangspolicy-Modul (Access-Modul)	107
B.4.1	Steckbrief	107
B.4.2	Kommandos	107
B.5	ResourceManager-Modul	108
B.5.1	Steckbrief	108
B.5.2	Kommandos	108
B.6	API-Modul (GUI-Modul)	110
B.6.1	Steckbrief	110
B.6.2	Kommandos	110
	Glossar und Abkürzungsverzeichnis	111

Abbildungsverzeichnis

1.1	Telefon aus dem Jahre 1880	1
1.2	Impulswähler	2
1.3	Beispielarchitektur	9
1.4	Einsatzmöglichkeiten von Gateways	10
2.1	H323 Protokollarchitektur	18
2.2	Interoperabilität von H.323-Endpunkten [16]	19
2.3	Beispiel eines Gatekeeper-routed-Calls mit einem Gatekeeper	20
2.4	Beispiel eines Direct-Calls mit einem Gatekeeper	21
2.5	Protokollphasen bei H.323	23
2.6	H.225.0-Anteil in einem Endpunkt, entnommen aus [15]	25
2.7	SIP Protokollarchitektur	28
2.8	Interaktion mit den TBGP-Sprechern [11]	29
3.1	Abbildung von H.323-Adressen auf Benutzer	42
3.2	Mehrere Benutzer teilen sich einen Endpunkt (1)	42
3.3	Mehrere Benutzer teilen sich einen Endpunkt (2)	43
3.4	Mehrere Benutzer mit Rufumleitung	43
3.5	Ablauf der Adreßumsetzung	46
4.1	Komponenten am MBus	49
4.2	Nachrichtenaustausch bei Gatekeeper-Discovery	60
4.3	Nachrichtenaustausch bei der Endpunkt-Registrierung	61
4.4	Nachrichtenaustausch bei der Endpunkt-Abmeldung	62
4.5	Nachrichtenaustausch beim Einholen einer Anruferlaubnis	64
4.6	Nachrichtenaustausch bei einer Bandbreitenänderung	65
4.7	Nachrichtenaustausch beim Beenden eines Anrufes	65
4.8	Policy-Entscheidungsablauf bei <code>isLocalZone</code>	67
4.9	Policy-Entscheidungsablauf bei <code>mayRegister</code>	68
4.10	Entscheidungsablauf bei Ressourcenanfrage	70
5.1	Interner Aufbau des H323-Moduls	76
5.2	Generelle Aufbau von MBus-Modulen	83
5.3	Aufbau des virtuellen MBus	84

Tabellenverzeichnis

2.1	Kommandos zur Anmeldung von Endpunkten	36
4.1	Voting, um Zonenzugehörigkeit von Endpunkten festzustellen . .	60
4.2	Voting, um Zonenzugehörigkeit von Endpunkten festzustellen . .	61
4.3	Kommando zur Anmeldung von Endpunkten	61
4.4	Kommando zur Abmeldung von Endpunkten	62
4.5	MBus-Kommandos für Adreßauflösung und Bandbreitenbestimmung	64
4.6	Voting zur Zugangserlaubnis anhand von Nutzer und IP-Adresse	65
4.7	MBus-Kommandos für Bandbreitenbestimmung- und -änderung .	66
4.8	MBus-Nachricht bei Endpunkt-Abmeldung	66
B.1	Übersicht über alle verwendeten MBus-Kommandos	98

Vorwort

Diese Arbeit handelt von IP-Telefonie, einer Technik, die sich anschickt, das bisherige Telefonsystem abzulösen. Dieser Übergang wird fließend sein und nur in dem Maße voranschreiten können, wie die Vernetzung aller Haushalte voranschreitet — also noch ein weiter Weg.

Eine zentrale Rolle in der IP-Telefonie spielen die Verwaltungskomponenten, sogenannte Gatekeeper. Gatekeeper sorgen für die Vermittlung von Anrufen und können Mehrwertdienste bereitstellen. Ein solcher Gatekeeper, der kontrollierten Zugang zur IP-Telefonie und die Verwaltung der Netzressourcen ermöglicht, ist Gegenstand dieser Arbeit. Es handelt sich zwar nicht um den ersten Gatekeeper, der jemals erstellt wurde, aber um den ersten, der auf die Bedürfnisse des universitären Bereichs zurechtgeschnitten wurde. Dies bedeutet vor allem Offenheit, sowohl in der Bereitstellung von Programmierschnittstellen für eigene Erweiterungen, als auch in der Bereitstellung des Quelltextes.

IP-Telefonie ist eine recht junge Technik. Viele Fragen sind noch ungeklärt, jedoch sorgt reges Interesse der Industrie dafür, daß diese nach und nach in den Standardisierungsgremien geklärt werden. An einigen Stellen in dieser Arbeit wird es daher Verweise auf offene Fragen und mögliche Lösungen geben.

Die offenen Fragen sind ein weiterer Grund, warum die Offenheit des zu erstellenden Gatekeepers so wichtig ist: Sobald die Fragen geklärt und die Lösungen standardisiert sind, können sie als Erweiterungen zu diesem Gatekeeper hinzugefügt werden. Dies ermöglicht dem System, mit dem Stand der Technik zu wachsen.

Diese Arbeit kann leider kaum genauer auf bereits existierende Gatekeeper oder Literatur dazu eingehen, bzw. tut dies nicht, da alle Gatekeeper, die dem internationalen Standard H.323 folgen, sich bis zu einem gewissen Grad identisch verhalten sollten. Genauere Analysen erfordern die kostenintensive Anschaffung von Gatekeeper-Lösungen, was weit außerhalb des für mich finanzierbaren gelegen hätte. Die für mich wichtigsten Quellen waren daher die Standard-Dokumente als Literatur bzw. freie H.323-Endpunkt-Lösungen als Testumgebung.

Gliederung

- **Kapitel 1** beschäftigt sich mit den beiden Welten, die IP-Telefonie verbindet. Es wird ein Blick auf die Geschichte und Technik der Telefonie und des Internets geworfen. Anschließend werden Konzepte der IP-Telefonie und das Umfeld, einschließlich in diesem Konext erwähnenswerter inter-

nationaler Gremien, vorgestellt.

- **Kapitel 2** zeigt auf, welche Standards die Entwicklung eines Gatekeepers bestimmen, und was sie beschreiben. Das Kapitel enthält auch eine Beschreibung des *Message Bus* — ein System zum Nachrichtenaustausch zwischen einzelnen Teilen einer Anwendung, welches für die Implementierung verwendet wurde.
Dieses Kapitel ist bereits deutlich technischer, da hier erstmals die internen Abläufe Gegenstand der Betrachtung sind.
- **Kapitel 3** befaßt sich erstmals direkt mit dem erstellten Gatekeeper und enthält Beschreibungen der umgesetzten Funktionalität.
- **Kapitel 4** beschreibt den Entwurf des Gatekeepers. Es wird auf die Modulstruktur eingegangen, auf mögliche Erweiterungen und auf den Nachrichtenaustausch auf dem Message Bus.
- **Kapitel 5** beinhaltet Details zur Implementierung des zuvor vorgestellten Entwurfes. Zusammen mit Kapitel 5 und 6 sollte er genügend Informationen enthalten, um auf dieser Basis spätere Erweiterungen zu entwerfen.
- **Kapitel 6** hat die Anwendung des Gatekeepers zum Thema. Die Konfiguration des Programmes wird hier ebenso vorgestellt, wie der Umgang mit dem Administrationstool.
- **Kapitel 7** schließt die Arbeit ab und geht nochmals auf naheliegende nächste Erweiterungsschritte ein.
- **Die Anhänge** enthalten detaillierte Informationen zu den Konfigurationsdateien und den Nachrichten auf dem Message-Bus.

Kapitel 1

Einleitung

1.1 Telefonie

1.1.1 Die Erfindung des Telefons

Die Telefonie, wie wir sie heute kennen, hat ihre Wurzeln in der zweiten Hälfte des 19. Jahrhunderts. Im März 1876 meldete der in Amerika lebende Schotte Alexander Graham Bell das Patent für sein Telefon an. Bells Erfindung war die erste, die nicht auf einfachen Stromkreisunterbrechungen basierte, sondern den Stromfluß im Rhythmus der Schallwellen schwingen ließ. Die somit entstehende bessere Übertragungsqualität überzeugte derart, daß 46 Jahre später bereits 14.374.000 Fernsprecher in den USA in Betrieb waren.

In Deutschland begann die Einführung der Telefone im Jahre 1877, und zum Ende des 19. Jahrhundert gab es ca. 530 Orte mit Telefonanlagen und ca. 144.000 Sprechstellen (vergl. Abb. 1.1).

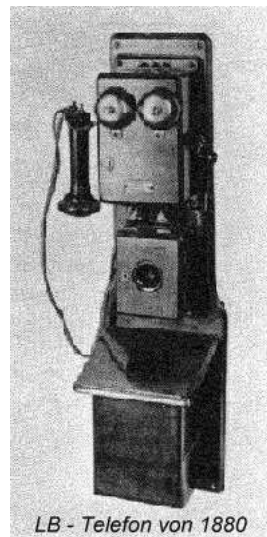


Abbildung 1.1: Telefon aus dem Jahre 1880

1.1.2 Vermittlungstechnik

Die manuelle Vermittlung

Änderten sich die technischen Details der Fernsprecher in den Jahrzehnten nach der Erfindung nicht mehr wesentlich, so tat sich doch einiges in den Vermittlungsämtern. Zunächst gab es nur handvermittelte Verbindungen, d.h. in großen Schränken in den Vermittlungsstellen wurden die Verbindungen durch das manuelle Erstellen von Kabenverbinden hergestellt.

Dieses Verfahren wurde mit steigender Anzahl an Teilnehmern aber zunehmend unübersichtlicher, da mehrere Vermittlungsschränke nebeneinander aufgestellt wurden und meist dazwischen verbunden werden mußte. Zwar schafften hier

technische Entwicklungen Abhilfe, indem sie es ermöglichten, das Kabelwirrwar zu minimieren, aber die Anfälligkeit durch den Faktor Mensch in der Bedienung blieb erhalten.

Beginn der automatisierten Vermittlung

Der folglich nächste Schritt war die Automatisierung der Vermittlungstechnik. 1889 wurde der „Hubdrehwähler“ erfunden, mit dem 100 Anschlüsse angewählt werden konnten. Wo mehr als 100 Teilnehmer vorhanden waren, mußte der Wählvorgang in Stufen erfolgen: Ein „Gruppenwähler“ schaltete zu Hundertergruppen durch und ein „Leitungswähler“ baute die Verbindung zum gerufenen Teilnehmer auf. Weitere Voraussetzung für den Hubdrehwähler waren Wahlimpulse vom Telefon, was zu der Einführung der Wählscheibe beim Telefon führte.

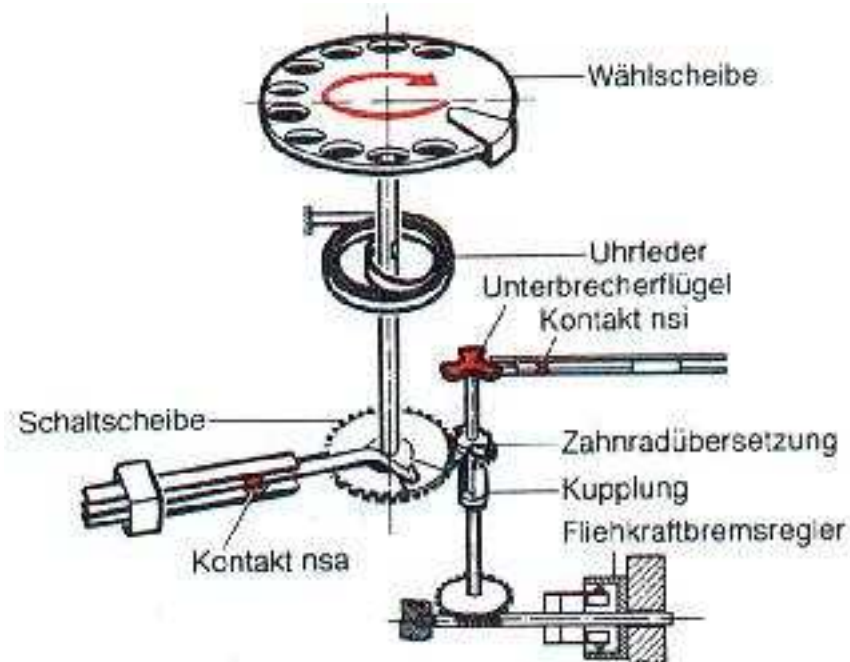


Abbildung 1.2: Impulswähler

Vom ersten deutschen Vermittlungsamt mit „Hubdrehwähler“ im Jahre 1908 ging die Entwicklung etwa 50 Jahre in kleinen Schritten weiter und fand ihren vorläufigen Höhepunkt im Jahre 1954 mit der Entwicklung des „Edelmetall-Motordrehwählers“.

Die elektronische Vermittlung

Durch den Siegeszug der Transistoren und integrierten Schaltungen wurde die elektromechanische Vermittlung nach und nach durch die elektronische Vermittlung abgelöst. So ging 1978 ein rechnergesteuertes Wählsystem von Siemens in

Serie, das eine größere Vermittlungsgeschwindigkeit als bisherige Systeme und eine Kurzwahleinrichtung aufwies.

Im Jahre 1979 fällt die Deutsche Bundespost eine Grundsatzentscheidung zur „Digitalisierung des Fernsprechnetzes“ von der Deutschen Bundespost. Ab 1982 wurden daher bei Neuinstallationen im Fernnetz und später auch in den Ortsnetzen ausschließlich digitale Übertragungs- und Vermittlungssysteme eingesetzt. Mit der digitalen Ortsvermittlung standen den Telefonkunden erstmals die Tonwahl und ein nahezu verzögerungsfreier Verbindungsaufbau zur Verfügung.

ISDN

Mit der zunehmenden Digitalisierung der Vermittlungsstellen und der Teilnehmeranschlußleitungen und dem Aufbau eines leistungsfähigen Fernnetzes mit Glasfaserkabeln waren die Grundlagen für ein System geschaffen, das die bisher parallel existierenden Dienste für Sprache, Text, Bild und Daten in sich vereinte. 1988 wurde in mehreren deutschen Großstädten der Betrieb des „Integrated Services Digital Network (ISDN)“ aufgenommen.

Ein ISDN-Basisanschluß (Basic User Network Interface) stellt drei Kanäle zur Verfügung: zwei leitungsvermittelte Kanäle à 64 kBit/s für Nutzdaten (B-Kanäle) und einen paket-orientierten Kanal mit 16 kBit/s für Steuerinformationen (D-Kanal). Die einzelnen Kanäle werden mittels eines speziellen Rahmenformates über eine physikalische Leitung übertragen.

Über einen ISDN-Anschluß können z.B. zwei Nutzer gleichzeitig mit unterschiedlichen Zielen telefonieren. In diesem Fall bekommt jeder Nutzer einen B-Kanal zugeordnet.

Die Digitalisierung des Netzes brachte aber auch die Möglichkeit zu einigen Mehrwertdiensten, die zwar technisch nicht unbedingt ISDN voraussetzen, aber von der Telekom meist nur im Zusammenhang mit ISDN angeboten wurden. Einige bekannte Beispiele sind im folgenden genannt:

- **Rufnummeranzeige (Caller Line Identification Presentation Restriction)**
Bei ISDN ist es möglich, daß die Telefonnummer des Anrufers mit übermittelt wird. Der Angerufene kann sich die übermittelte Nummer dann anzeigen lassen (CLIP). Jeder ISDN-Teilnehmer kann jedoch entscheiden, daß seine Rufnummer nicht übermittelt werden soll (CLIR).
- **Rückruf bei besetzt**
Wenn der angerufene Gesprächspartner gerade in ein anderes Gespräch verwickelt ist, wird das Telefon des Anrufers angerufen, sobald der Gesprächspartner sein Gespräch beendet hat. Nimmt der Anrufer diesen automatischen Anruf ab, wird erneut eine Verbindung zum Ziel aufgebaut.
- **Anruf abweisen (Call Deflection)**
Bei einem eingehenden Anruf und evtl. Anzeige der Nummer des Anrufers kann der Angerufene entscheiden, den Anruf abzuweisen, ohne daß der Anrufer dies erfährt.
- **Anrufweiterleitung (Call Forwarding)**
Es ist möglich, daß System so zu konfigurieren, daß eingehende Anrufe an eine andere Nummer weitergeleitet werden.

- **Gespräch parken**
Ein laufendes Gespräch kann geparkt werden, um von einem Telefon zu einem anderen zu wechseln und es dort weiterzuführen.
- **Anklopfen**
Wenn ein Anruf eintrifft, während man gerade telefoniert, hört man im Hörer einen Anklopftön. Den neuen Anruf kann man annehmen oder ablehnen. Wird angenommen, so wird der erste Anruf gehalten.
- **Anruf halten/Makeln (Call Hold/Retrieve)**
Wenn z.B. nach einem angenommenen Anklopfen ein Gespräch gehalten wird, während ein anderes geführt wird, so kann zwischen diesen beiden Anrufen hin- und hergeschaltet werden (Makeln).
- **3er-Konferenzen (Ad hoc Conference)**
Aus zwei einzelnen Gesprächen (einem aktiven und einem gehaltenen) kann eine 3er-Konferenz aufgebaut werden. Die Konferenz kann bei Bedarf auch wieder in zwei Einzelgespräche geteilt werden.
- **Böswillige Anrufe feststellen (Malicious Caller Identification)**
Der Netzbetreiber stellt die Möglichkeit zur Verfügung, Anruferdaten wie Datum, Uhrzeit und beteiligte Rufnummern aufzuzeichnen und zu speichern, um sie ggf. Strafverfolgungsbehörden zur Verfügung zu stellen.

Die Liste der möglichen Mehrwehrtedienste (Supplementary Services) wächst durch Initiative der einzelnen Hersteller von Telefonie-Produkten ständig. Jedoch werden in der Regel längst nicht alle Dienste von allen Netzbetreibern unterstützt.

Neben vielen neuen Mehrwertediensten bot ISDN erstmals die Möglichkeit zur Bildtelefonie auf einer breiteren Basis. Das Datenaufkommen bei Bildtelefonie war bis vor kurzem so groß, daß für ein Gespräch beide B-Kanäle benötigt wurden. Die somit entstehenden doppelten Kosten und die teureren Telefone hat aber bisher eine große Verbreitung verhindert. Jüngste Entwicklungen erlauben es zwar mittlerweile, Bildtelefonie mit nur einem B-Kanal zu betreiben, jedoch ist dies nur bedingt kompatibel mit dem älteren Standard.

1.2 Das Internet

Im Laufe der zweiten Hälfte des 20. Jahrhunderts, nach der Verbreitung der ersten Computer, begannen Bemühungen, die Rechner zwecks Datenaustausch über eine größere Entfernung miteinander zu verbinden. In den 60er Jahren startete die *Advance Research Project Agency* (ARPA) des US-Verteidigungsministeriums ein Forschungsvorhaben, das Mittel und Wege suchen sollte, die kostspieligen Computer-Einrichtungen miteinander zu vernetzen und so die Ressourcen besser zu nutzen.

1.2.1 Paketbasierte Datenübertragung

Die Forschung der ARPA ergab unter anderem, daß die verwendeten leitungsvermittelten Datenübertragungen nicht optimal waren. Bei der Leitungsvermitt-

lung wird immer eine feste Größe von Ressourcen (z.B. eine Leitung) für eine Verbindung verwendet, unabhängig davon, ob die Leitung die gesamte Zeit und immer in gleichem Maße benötigt wird. Günstiger erweisen sich Verfahren, mehrere Datenströme über eine Leitung zu versenden (Multiplexing), ohne daß feste Kapazitäten reserviert werden mußten. Damit dies erreicht werden konnte, mußten die Informationsströme vom Sender in Pakete bestimmter Größe unterteilt und vom Empfänger wieder zusammengesetzt werden.

Eine weitere Neuerung, die mit der Paketvermittlung einherging, war die Abkehr von den traditionellen verbindungsorientierten Datenübertragungen. Jedes Paket erhielt nun eine Zieladresse und wurde anhand dieser übertragen. Die Konsequenz war allerdings eine geringere Dienstgüte, da z.B. keine Kontrolle mehr bestand, ob der Empfänger die Daten überhaupt korrekt und in der richtigen Reihenfolge erhalten hat. Wurden bessere Dienstgüten benötigt, so mußten diese bei Bedarf von höheren Protokollschichten geboten werden (s.u.).

Paket-Vermittlung stellt grundlegend andere Anforderungen an die Knoten im Netz, über die die Daten geleitet werden. Dies sind z.B. die Größe der Pakete, aber auch deren Ziel. So muß z.B. jedem Paket Informationen über sein Ziel mitgegeben werden, damit dieses korrekt weitergeleitet werden kann.

1.2.2 Netzkopplung und IP

Anfänglich waren die Rechnernetze übersichtlich und oft lokal begrenzt. In diesen vernetzten Inseln existierten oft eigene Lösungsansätze für Probleme wie Netztopologie, Zugangsverfahren, Übertragungsmedien und eingesetzter Protokolle. Als versucht wurde die unterschiedlichen Netze miteinander zu verbinden, wurde folglich Protokollumsetzung und netzübergreifende eine Adressierung benötigt. Zur Lösung dieser Probleme wurde das *Internet Protocol* (IP) entwickelt, das bis heute noch als Grundlage für alle Datenübertragungen im Internet dient.

IP ermöglicht die Kopplung von Netzen und die netzübergreifende Adressierung. Jeder an das Internet angeschlossene Rechner ist über eine IP-Adresse eindeutig identifizierbar. Eine IP-Adresse ist dabei eine 32-Bit-Zahl, die in zwei Komponenten zerfällt: die Netzkennziffer und die Nummer des Rechners in dem Subnetz. IP sah dabei zunächst verschiedene Netzgrößen vor, abhängig davon wieviele Bits für die Netzkennziffer reserviert sind. In einem Class A-Netz sind dies z.B. 7 Bit, in einem Class C-Netz 21. Dazu kommt, daß zunehmend Realzeitkommunikation auf Basis von IP erfolgt, hierfür aber auf weitere Protokolle (RTSP) zurückgegriffen werden muß, da IP selbst nicht in dem Sinne realzeit-tauglich ist, daß es erlaubt, Bandbreite zu reservieren und zu garantieren.

Aus diesem Grund wird seit einiger Zeit an einer neuen Variante von IP, namens IPv6, gearbeitet, die 128 Bit lange Adressen vorsieht, also 2^{128} Rechner adressieren kann, Bandbreitenreservierungen und andere Mechanismen zur Realzeitkommunikation vorsieht, und die von unnötigen Optionen befreit wurde.

IP arbeitet verbindungslos und sieht nur den Versand von Paketen (sogenannten Datagrammen) vor. Dabei ist IP nicht sequenzerhaltend, d.h. es ist nicht garantiert, daß die Pakete in der gleichen Reihenfolge ankommen, in der sie gesendet wurden.

Ein Problem, das man sich durch die Kopplung von Netzen einhandelte, war

die Wegewahl zwischen den Netzen. Das Internet ist, wie bereits erwähnt, ein gewachsenes Gebilde aus unterschiedlichen Netzen, in dem die einzelnen Netze mit ihren „Nachbarnetzen“ verbunden sind. Der Weg zum Ziel kann also mitunter über viele Zwischenschritte führen. Des weiteren muß bei einem Netz, das an mehrere andere Netze angeschlossen ist, entschieden werden, in welches andere Netz das Paket gesendet werden soll. Die Anzahl der möglichen Wege zum Ziel ist nahezu beliebig groß¹, weswegen Mechanismen gefunden werden müssen, Wege zum Ziel zu finden bzw. den geeignetesten Weg zu wählen (Routing). Diese Aufgabe übernehmen spezielle Netzknoten (Router), die somit ähnliche Aufgaben wie Vermittlungsstellen im Telefonnetz erfüllen.

Eine besondere von IP zur Verfügung gestellte Art, Empfänger von Daten zu adressieren, ist *Multicasting*. Hierbei ist es möglich, Pakete an eine bestimmte Gruppe zugleich zu versenden, ohne das der Sender weiß, wer alles die Pakete empfängt, d.h. wer alles zu dieser Gruppe gehört.

Um dies zu ermöglichen, wurde ein spezielles Class D-Netz reserviert, dessen 28 Bit Rechneradresse als Gruppenbezeichner dient. Ein Empfänger teilt dabei den IP-Routern mit, für welche Gruppen er Pakete empfangen möchte. Der Sender wiederum adressiert seine Pakete einfach an eine bestimmte Gruppe. Eine Reihe von Multicast-Routing-Protokollen sorgen für die Verbreitung der nötigen Informationen.

1.2.3 Transportprotokolle

Die Dienstgüte der durch IP gegebenen Vermittlungsschicht ist abhängig von den konkreten unterliegenden Netzen. Es ist aber wünschenswert, daß ein Anwendungsprogramm unabhängig von den Netzen eine bestimmte Dienstgüte (*Quality of Service* (QoS)) anfordern kann. Eng mit IP verbunden ist daher das *Transmission Control Protocol* (TCP), das zur Erlangung und Gewährleistung einer höheren Dienstgüte verwendet wird. TCP arbeitet als verbindungsorientiertes Verfahren und stellt eine Abbildung zwischen Byteströme und den in IP verwendeten Datagramme zur Verfügung. Dabei sorgt es selbst für die erneute Versendung verlorener Datagramme, Sequenzerhaltung und für Flußkontrolle, um vor Überlastungen zu schützen.

Dort, wo nicht alle Leistungen von TCP benötigt werden, steht mit dem *User Datagram Protocol* (UDP) ein schlankeres Protokoll zur Verfügung. Wie auch TCP bietet es Anwendungsadressierung und End-zu-End-Fehlererkennung, jedoch arbeitet es verbindungslos und ohne Sequenzerhaltung.

1.2.4 Traditionelle Dienste im Internet

Auf Basis dieser oben angegebenen Protokolle entwickelten sich unterschiedliche Dienste, die das Internet nutzten. Bedingt durch die hohen Kosten waren es zunächst militärische Einrichtungen, große Unternehmen oder Bildungs- und Forschungseinrichtungen, die am Internet teilhatten. Der Hauptnutzen beschränkte sich auf Dateitransfer (FTP), eMail, Arbeiten auf entfernten Rechnern (Telnet), Newsgroups oder auf textbasierte Informationssysteme (Gopher).

¹ Alle Wege führen nach Rom.

Mit der Einführung des *Hypertext Transfer Protocols* (HTTP) und der *Hypertext Markup Language* (HTML) begann die Popularität enorm zu wachsen. Informationen konnten jetzt dank HTML optisch ansprechender gestaltet werden, und in späteren Versionen von HTML gesellte sich zu weiteren Gestaltungsmöglichkeiten auch die Unterstützung für Multimediadaten dazu. Heute ist das auf HTTP und HTML basierende *World Wide Web* (WWW) neben eMail der bekannteste Dienst des Internets, dessen Nutzung leicht erlernbar ist, so daß man ihn auch ohne Fachkenntnisse nutzen kann.

1.2.5 Multimedia und Echtzeit in IP-Netzen

Mit zunehmender Popularität des Internets wurde Ende der 80er Jahre auch über andere Arten der Nutzung nachgedacht. Die zunehmende Popularität des Internets sorgte dafür, daß durch steigende Investitionen bald genug Bandbreite zur Verfügung stand, um über die Übertragung von kontinuierlichen Audio- und Videoströme zu erlauben. Zusammen mit der Möglichkeit der weltweiten Verteilung dieser Ströme wurde das Internet zu einem Medium, über das auch Funk, Fernsehen und Telefongespräche übertragen werden konnten. Die bedeutendste Infrastruktur, die sich so herausbildete, war der *Multicast Backbone* (Mbone), der lose gekoppelte Konferenzen auf Basis der Protokolle der *Internet Engineering Task Force* (siehe auch 1.4.2 und 2.2) ermöglichte.

Nach und nach tauchten Radio- und TV-Übertragungen im Internet auf und es gab Anwendungen, die es zwei oder mehr Teilnehmern ermöglichen sollten, miteinander zu sprechen. Die Qualität war jedoch nie so gut, daß sie an die der etablierten Medien heranreichte.

Aufgrund der Vielfalt der Lösungsansätze und Protokolle, darunter auch viele proprietäre Lösungen, waren längst nicht alle Anwendungen, die in Realzeit Medien übertragen, zueinander kompatibel. Es gab zwar mit dem *Realtime Transport Protocol* (RTP) und dem *Real-time Streaming Protocol* (RTSP) durchaus anerkannte Standards, die die Übertragung von Medienströmen in Echtzeit unterstützten, doch fehlte die Einigung auf ein einheitliches Signalisierungsprotokoll

H.323 ist eine Art Meta-Standard, der sowohl die Komponenten des Telefonsystems beschreibt, als auch festlegt, welche anderen Standards und Empfehlungen im Rahmen dieses Telefonsystems verwendet werden sollen. Eine der in H.323 definierten Komponenten, der *Gatekeeper*, ist Gegenstand dieser Arbeit.

1.3 IP-Telefonie

IP-Telefonie ist das Telefonieren unter Nutzung der Internet-Technologien. Dabei bedeutet eine Umstellung auf diese Technologie nicht einfach nur, daß sich das Telefonsystem intern ändert, ohne daß es weitere Effekte hätte: Eine Umstellung von gegenwärtigen Vermittlungstechnik auf IP-basierte Vermittlung bedeutet vor allem eine Vereinfachung und Kostenreduzierung des Telefonsystems.

In einer normalen Vermittlungsstelle wird für ein Gespräch eine „Leitung“ vermittelt. Diese Leitung steht für die Dauer des Gesprächs exklusiv zur Verfügung — gleichgültig, ob gerade gesprochen wird oder nicht. Die Ausnutzung der Leitungsressourcen ist somit nicht optimal.

Im Internet wird wegen Paketvermittlung (siehe 1.2.2) eine physikalische Leitung für mehrere Teilnehmer genutzt. Dies gelingt, indem die Daten in Pakete zerlegt werden und gemeinsam mit den Paketen anderer Teilnehmer über eine Leitung geschickt werden. Dies führt zur effektiveren Ausnutzung von Leitungskapazitäten und ist damit erheblich billiger.

Die im Internet verwendete Vermittlungstechnik von IP ist auch erheblich einfacher als die komplexen und teuren Telefonvermittlungen. Dies reduziert den Wartungsaufwand und somit auch die Wartungskosten. Dazu kommt, daß mit IP im Prinzip schon Routing-Mechanismen zur Verfügung stehen, die einen, unter bestimmten Gesichtspunkten (z.B. Kosten) optimalen, Weg wählen können — ein Merkmal, das in herkömmlichen Telefonvermittlungen nur mit hohem Aufwand international realisierbar wäre. Hier kommt somit zur Kostenreduzierung auch noch die Vereinfachung des Systems — es können immerhin heute schon verwendete Router eingesetzt werden — dazu.

Anfänglich wird Telefonieren über IP sich auf Intranetze beschränken und in den jeweiligen Einrichtungen Haustelefonssysteme ablösen. Solche IP-Telefonie-Inseln sind über ein entsprechendes Gateway mit dem restlichen Telefonsystem verbunden. Oft verfügen solche Institutionen/Firmen über eine permanente Verbindung ins Internet, so daß sich für sie ein Preisvorteil ergäbe, wenn die Gespräche über die sowieso schon angemietete Standleitung geführt werden können, da die normalen Telefonkosten wegfallen.

Aber selbst wenn nicht beide Gesprächsteilnehmer über die Möglichkeit zur IP-Telefonie verfügen, können sie miteinander reden und Kosten sparen. Gateways ermöglichen z.B. den Übergang zwischen dem normalen Telefonnetz und dem Internet. Ein Anruf von einem IP-Telefon in Deutschland könnte z.B. durch das Internet zu einem Gateway in den USA gelangen und erst dort wird eine „normale“ Telefonverbindung zum Gesprächspartner aufgebaut. Die Kosten für den Auslandsanruf entfallen, und im besten Fall zahlt der Anrufer nur den Ortstarif² und die Kosten für seinen Internet-Provider.

1.3.1 Architektur eines IP-Telefoniesystems

Ein IP-Telefoniesystem (siehe Abb. 1.3) besteht aus vielen Komponenten mit unterschiedlichen Aufgaben. Da wäre zunächst ein Telefonie-*Endpunkt*, d.h. ein IP-Telefon oder einen Computer mit der nötigen Hardware, um Telefonie zu betreiben. Mindestanforderung an einen Computer wäre die Möglichkeit, Audiodaten gleichzeitig zu senden und zu empfangen (Full-Duplex).

Theoretisch könnten zwei Endpunkte ohne Zuhilfenahme weiterer Komponenten miteinander kommunizieren, vorausgesetzt der Anrufer findet eine Möglichkeit, zu ermitteln, wo sich der Anzurufende gerade aufhält, d.h. auf welcher Transport-Adresse er empfangsbereit ist.

²Und bald evtl. sogar nur noch eine Flatrate, d.h. eine Grundgebühr, mit der alle Ortsgespräche abgegolten werden.

Dieses Verfahren ist nicht besonders zufriedenstellend, da sich a) man nur Leute anrufen kann, deren Daten man bereits kennt und b) IP-Adressen und Ports im Laufe der Zeit ändern können (z.B. wenn der Nutzer den Rechner wechselt oder in ein anderes Büro zieht).

Um nun diesem Problem beizukommen, braucht man eine weitere Komponente, die Namen zu Adressen auflöst (sog. *Location Service*). Ein bekanntes Beispiel für die Kombination von Endpunkten und *Location Server* ist *Microsoft Netmeeting* — das hier verwendete Protokoll zur Kommunikation mit dem Location Server ist allerdings eine Eigenentwicklung von Microsoft. Der für die IP-Telefonie maßgebende internationale Standard H.323 definiert für diesen Zweck eine zentrale Komponente *Gatekeeper*, deren Funktionalität im Abschnitt 2.1.1 näher erläutert wird.

Da es unter anderem aus Gründen der Auslastung und der Robustheit nicht sinnvoll ist, nur eine einzige zentrale Komponente für alle Teilnehmer weltweit zu haben, werden mehrere Endpunkte oder Personen zu einer Zone zusammengefaßt, für die dann eine eigene Verwaltungseinheit zur Verfügung steht. Eine solche Zone hat meist eine logische Grenze, wie z.B. ein Firmen-Intranet, ein Fachbereich einer Universität oder Kunden eines Internet-Providers in einer Stadt.

Dadurch, daß für unterschiedliche Zonen andere Verwaltungskomponenten verantwortlich sind, entsteht der Bedarf an einer Kommunikation dieser Komponenten untereinander, damit auch von einer Zone in eine andere telefoniert werden kann.

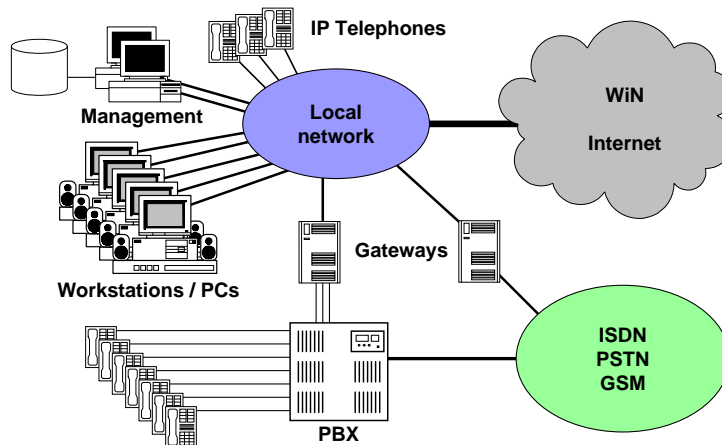


Abbildung 1.3: Beispielarchitektur

Die einer Verwaltungskomponente zugehörige Zone — meist ein lokales Netz — beinhaltet mehrere IP-Telefone, bzw. Computer mit IP-Telefonie-Software (siehe Abb. 1.3). Es ist eine direkte Kommunikation mit anderen Endpunkten über das Internet möglich — um jedoch Gespräche mit Telefonen aus anderen Netzen zu führen, bedarf es eines Gateways, daß für die Umsetzung der Daten zwischen dem LAN und Telefonnetz sorgt.

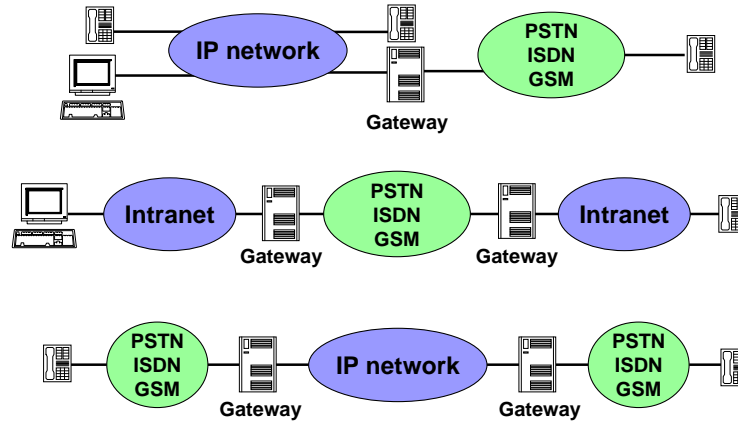


Abbildung 1.4: Einsatzmöglichkeiten von Gateways

Solche Gateways können für verschiedene Szenarien eingesetzt werden (siehe Abb. 1.4): Der einfachste Fall ist die Umsetzung zwischen IP-Netz und bestehenden Telefonnetzen, wie sie z.B. der Fall wäre, wenn eine Firma ihr hausinternes Telefonsystem auf IP-Telefonie umstellt und dann über die internen Telefone „nach draußen“ telefonieren möchte.

Wollen zwei Einrichtungen mit IP-Telefonie im Intranet und einem Gateway nach draußen miteinander telefonieren, hat man den Fall, daß für einen mittleren Übertragungsabschnitt das Telefonnetz wird und auf beiden Seiten Gateways die Umsetzung übernehmen.

Es werden ebenfalls zwei Gateways gebraucht, wenn z.B. das Internet als Übertragungsmedium verwendet wird, dies aber für bestehende Telefonsysteme transparent bleiben soll. In diesem Fall tritt ein Gateway als Vermittlungsstelle auf, die die Gespräche über das Internet statt über das verwendete Telefonnetz zum nächsten Gateway weiterleitet.

Mehrwertdienste

Im normalen Telefonnetz werden Mehrwertdienste entweder von einer Vermittlungsstelle oder bei größeren Organisationen von einer privaten Telefonanlage

(*Private Branch Exchange* PBX) zur Verfügung gestellt. Für IP-Telefonie ist hierfür der Gatekeeper in Verbindung mit weiteren Komponenten geeignet. So ist z.B. eine *Multipoint Control Unit* (MCU) für Konferenzschaltungen verantwortlich, indem sie die Audio- und Videoströme mischt und verteilt. Zur Realisierung von Anrufbeantwortern könnte ein eigener Rechner abgestellt werden, der als *Mediensever* fungiert und Daten aufzeichnen und wiedergeben kann. Diese Komponenten können in den Gatekeeper integriert werden — sie können aber auch eigenständig auf anderen Rechnern laufen.

1.3.2 Möglichkeiten der IP-Telefonie

IP-Telefonie wird das Bild der Telefonie in naher Zukunft stark ändern. Der heutige Trend zu immer breitbandigeren privaten Anschlüssen (ISDN, xDSL) wird aller Voraussicht nach dazu führen, daß ein Internet-Anschluß in privaten Haushalten so selbstverständlich wird wie ein Strom- oder Antennenanschluß. Die nötigen Endgeräte, die auch jetzt schon auf dem Markt sind, brauchen nicht komplizierter zu werden als heutige Telefone, die Zuordnung von IP-Adressen zu Telefonen kann mittels des *Dynamic Host Configuration Protocol* (DHCP) automatisch geschehen und somit vor den Anwendern versteckt werden. Man steckt das Telefon in eine entsprechende Buchse, und es funktioniert.

Das System der Telefonnummern wird sehr wahrscheinlich einer Wandlung unterzogen werden. Wahrscheinlich bekommt jeder Mensch eine Art eindeutigen globalen Kommunikationsbezeichner, unter dem er weltweit erreichbar ist. Der Zwang, sich Nummern zu merken oder zu überlegen, wo sich die gesuchte Person gerade aufhält, entfällt.

Aber auch andere Perspektiven rücken in greifbare Nähe. Bisher war es z.B. nicht ohne weiteres möglich, Gespräche abhörsicher zu machen. Der heutige Stand der Technik bietet aber bereits sichere Verschlüsselungsverfahren, die Abhörsicherheit garantieren würden. IP-Telefonie könnte demnach abhörsichere Kommunikation ermöglichen — eine Möglichkeit, die nicht jedem wünschenswert erscheint. Kritiker sehen in wirklich abhörsicheren Telefonaten eine Beschränkung der Mittel der Justiz, gegen Staatsfeinde und Verbrecher vorzugehen. Andererseits ist Abhörsicherheit ein Element der Privatsphäre und das Problem der Kontrolle nicht nur darauf beschränkt, daß Behörden überhaupt abhören/mitlesen können, sondern eher, daß sie es mittlerweile in großem Stil tun können.

Hier bedarf es einer internationalen Regelung oder zumindest eines definierten Umgangs von Gesprächen zwischen Staaten, mit unterschiedlicher Handhabung in Bezug auf abhörsichere Kommunikation.

1.3.3 Stand der IP-Telefonie

Bis zur Ablösung des heutigen Telefonsystems durch IP-Telefonie ist es aber noch ein weiter Weg, da sich wichtige Elemente immer noch in der Standardisierung befinden. Mit dem im Februar 1998 in seiner zweiten Version endgültig verabschiedeten Standard ITU-T H.323 [16] existiert eine stabile Basis für Telefonie und Multimediakonferenzen über IP. H.323 baut dabei auf bekannte und eingesetzte Standards auf, so daß die Erstellung von Endgeräten, die diesem

Standard entsprechen, keine kompletten Neuentwicklungen erfordert. Reges Interesse und Beteiligung seitens der Industrie sorgt für kontinuierliche Weiterentwicklung und wohl auch für zunehmende Verbreitung. Gegenwärtig gibt es mehrere Software- und einige Hardware-Lösungen, z.B. von Cisco, Microsoft, VocalTec, Elemedia und VoxWare, die dank H.323 in der Lage sind, miteinander zu kommunizieren.³

Neben H.323 gibt es mit dem von der IETF entwickelten *Session Initiation Protocol* (SIP) ein weiteres Protokoll, daß IP-Telefonie ermöglicht. Auch hier besteht Interesse seitens der Industrie, jedoch ist dies bislang nicht so groß, wie an H.323 - zumindest gemessen an den bisher verfügbaren SIP-Lösungen.

Heutige IP-Telefonie-Produkte

Wie schon von Telefonanlagen gewohnt, unterscheiden sich diese Lösungen in ihrer Funktionalität. Während die Basisfunktionalität von allen Lösungen geboten wird, funktionieren herstellerspezifische Erweiterungen aufgrund fehlender Standardisierung meist nicht mit anderen Systemen.

Ein großer Teil der Funktionalität der IP-Telefonie liegt bei den *Gatekeepern*, die als Verwaltungskomponenten bestimmter Zonen fungieren und somit in ihrer Funktion etwa den Vermittlungs- und Telefonanlagen entsprechen. In die Gatekeeper werden demnach alle Mehrwertdienste integriert, die das Telefonsystem beherrschen soll. Deshalb soll an dieser Stelle aufgelistet werden, was die bisher entwickelten Komponenten zu leisten in der Lage sind⁴.

- **Konferenzsteuerung**
Die Möglichkeit mehrere Teilnehmer, evtl. spontan, zu einer Konferenz zusammenzuschalten [5].
- **Mehrwertdienste**
Hierzu gehören z.B. Anklopfen, Rufumleitung, Wiederwahl, Halten[29]
- **Kontrollierbares Routing**
Das Routing der Anrufe kann in Abhängigkeit von Berechtigungen und Tageszeit erfolgen [40][39].
- **Lastenausgleich bei der Verwendung der Gateways**
Sorgt für eine gleichmäßige Verteilung der Last auf mehrere Gateways [40].
- **Authentifizierung**
Nutzer werden über ID und Paßwort authentifiziert. Verwendung von Zugangstokens nach H.235 [40][5][29][28][23].
- **Anrufprotokollierung**
Die Daten der vom System verwalteten Anrufe (*Call-Detail-Records*) (z.B. Startzeit, Endzeit, Anrufer, Ziel, ...) werden protokolliert, um Statistiken und Abrechnungen zu ermöglichen [40][5][28][39].

³Zumindest sollten sie theoretisch dazu in der Lage sein. Praktisch treten aber immer wieder Kompatibilitätsprobleme durch abweichende Auslegungen des Standards oder proprietäre Erweiterungen auf. Microsofts Netmeeting beispielsweise behauptet zwar H.323-konform zu sein, führt aber keine Suche nach Gatekeepern durch und reagiert auch nicht auf Bandbreitenzuteilungen des Gatekeepers.

⁴Wobei kein Anspruch auf Vollständigkeit erhoben wird.

- **Abrechnungsmöglichkeit**
Anhand der Call-Detail-Records (s.o.) können den Nutzern Rechnungen gestellt werden [40].
- **Gatekeeper-Hierarchien**
Unterstützung für ein hierarchisches System von Gatekeepern zum Auffinden von Benutzern in anderen Zonen [40].
- **Web-basierte Konfigurationstools**
Die Konfiguration des Systems kann über eine Anzahl von WWW-Seiten vorgenommen werden. Somit ist natürlich auch Fernwartung möglich [39].
- **SNMP**
Die Unterstützung des *Simple Network Management Protocol* zur Administration des Gatekeepers [40][23].
- **Ausfallsicherheit**
Eine Konfiguration als „Hot Standby-System“ ermöglicht bei Ausfall des primären Gatekeepers den sofortigen Umstieg auf einen sekundären Gatekeeper.
- **Programmierschnittstellen**
Offene Schnittstellen, die anderen Anbietern erlauben, Anwendungen für Protokollierungs- und Abrechnungsdaten, Call-Center und Quality-of-Service bereitzustellen [40][5][29].

Die Aussagen, daß ein Produkt ein bestimmtes Feature unterstützt, sind mit Vorsicht zu betrachten. So gibt es zwar schon Überlegungen, wie eine Hierarchie von Gatekeepern aufzubauen ist, aber definitiv beschlossen ist noch nichts. Ein Produkt, welches für sich in Anspruch nimmt, eine Hierarchie von Gatekeepern zu unterstützen, ist also wahrscheinlich (noch) nicht konform zu dem, was einmal Standard werden wird.

Ebenso könnte es sich mit der SNMP-Unterstützung verhalten. Zwar gibt es mittlerweile einen RFC der die MIB für H.323 definiert, jedoch ist nicht gesagt, daß ein Produkt mit SNMP-Unterstützung dies anhand dieser MIB macht.

Frei existierende Lösungen

Neben den industriellen Lösungen existieren seit Ende 1998 auch Ansätze im Open Source-Bereich zur Implementierung eines H.323-Protokollstacks. Informationen darüber finden sich unter [6].

1.4 Internationale Gremien

Kommunikation in weltweiten Rechnernetzen kann nur dann erfolgreich funktionieren, wenn alle Beteiligten dieselben Protokolle verwenden und die Protokolle allgemein zugänglich sind. Aus diesem Grund haben internationale Standardisierungsgremien für Netze, sei es das Internet oder das Telefonnetz, erhebliche Bedeutung.

An dieser Stelle folgt daher ein kurzer Überblick über die für diese Arbeit wichtigen Organisationen.

1.4.1 ITU-T

Maßgebend für Standards der eng gekoppelten Telefoniewelt ist der *Telecommunication Standardization Sector of ITU* (ITU-T). Die ITU-T ist innerhalb der *International Telecommunication Union* (ITU) für Telefonie und jene Aspekte der Informationstechnik zuständig, die die Aufgaben der ITU betreffen. Innerhalb der ITU-T existieren mehrere *Study Groups* (SG) zu verschiedenen Themen, wie z.B. ISDN oder Telematik-Dienste. Diese Study Groups sind wiederum in *Working Parties* (WP) unterteilt, die sich mit einer Reihe von konkreten Fragen beschäftigen.

In Abschnitt 2.1 werden einige „Empfehlungen“ (*Recommendations*) der ITU-T vorgestellt, die für diese Arbeit interessant sind. Sie stammen aus der H-Serie („Audiovisual and multimedia systems“) und aus der Q-Serie („Switching and signaling“).

1.4.2 IETF

Die *Internet Engineering Task Force* (IETF) ist ein offenes Gremium mit Teilnehmern aus der Forschung, der Industrie und anderen fachlich versierten Personen, die mit der Entwicklung der Architektur des Internets und der reibungsfreien Arbeit der Netze zu tun haben. Jeder, der an der Entwicklung der Internet-Technologien interessiert ist, kann an Veranstaltungen der IETF teilnehmen.

Die IETF unterteilt sich in *Areas*, die wiederum in *Working Groups* zu bestimmten Themen, wie z.B. Routing oder Sicherheit, unterteilt sind. Ein Großteil der Koordination und Zusammenarbeit findet über Mailing-Listen statt.

Die Standards, die von der IETF produziert werden, heißen *Requests for Comments* (RFC). Ein RFC beginnt als ein *Internet Draft*, wird diskutiert (und implementiert) und wird dann — so er Zustimmung gefunden hat — zu einem RFC.

Alle RFCs werden — im Gegensatz zu den Standards der ITU — kostenlos öffentlich zugänglich gemacht.

1.5 Die Arbeitsgruppe Rechnernetze

Diese Diplomarbeit ist entstanden in Zusammenarbeit mit den Mitgliedern der Arbeitsgruppe Rechnernetze (RN) im Studiengang Informatik der Universität Bremen. Das Tätigkeitsfeld der Arbeitsgruppe umfaßt Forschung, Lehre und Entwicklung im Bereich der computergestützten Kommunikation und Kooperation, sowie den Bereich der offenen Dokumentbearbeitung. Die Arbeitsgruppe wird geleitet von Prof. Dr.-Ing. Ute Bormann.

Da gerade im Bereich der Rechnernetze Standards eine große Rolle spielen, findet Forschung und Entwicklung der Arbeitsgruppe — sowie auch dieser Diplomarbeit — im Kontext internationaler Standardisierung statt. Dies schlägt sich nicht nur in der Art der bisher durchgeführten Projekte, sondern auch im Engagement in den entsprechenden internationalen Gremien nieder. Als Beispiel

wäre der *Message Bus* (MBus) zu nennen, der nicht nur eine Anwendung in gegenwärtigen Projekten der Arbeitsgruppe findet, sondern auch ein Entwurf der *Internet Engineering Task Force* (IETF) im Kontext des Working Group MMUSIC ist.

Das Projekt UniTel

Der seit 1978 in Bremen bestehenden Studiengang der Informatik verpflichtet den Studenten im Hauptstudium zur Teilnahme an einem zweijähriges Projekt, das in der Regel einen starken Praxisbezug aufweist.

Das Projektstudium ist in seiner Dimension dazu geeignet, komplexe Aufgabenstellungen zu bearbeiten. Die Studenten organisieren sich im Rahmen dieser Projekte weitgehend selbständig, so daß unter anderem Teamfähigkeit herausgebildet wird, die für eine spätere Arbeit im Berufsleben eine große Rolle spielt.

Gegenwärtig betreut die Arbeitsgruppe Rechnernetze das studentische Projekt UniTel, dessen Ziel es ist, eine Infrastruktur für IP-Telefonie innerhalb der Universität Bremen aufzubauen. Dies geschieht auf der Basis der internationalen Standards, an die die Studenten von den Betreuern, unter anderem auch dem Autor, herangeführt werden. Die Schwerpunkte des Projektes betonen allerdings mehr die Nutzungsseite als die technischen Schichten. So sollen die Studenten technische und organisatorische Lösungen zum Identifizieren und Auffinden eines im Fachbereich an einem Rechner arbeitenden Benutzers (UserLocation), zur Reglementierung der Nutzung von Fachbereichsressourcen (Ressourcenverwaltung), zur selbstbestimmten Wahrung der Privatsphäre jedes einzelnen und nicht zuletzt die Erarbeitung einer fachbereichsweiten Regelung für benutzer- und umweltverträgliche Nutzung dieser Technologien (Policies).

Das Projekt WIPTTEL

Das Projekt WIPTTEL ist ein Projekt der Arbeitsgruppe Rechnernetze. Ziel ist die Entwicklung einer Referenzkonfiguration für die Nutzung von IP-Telefonie-Diensten zum Einsatz innerhalb der an das Wissenschaftsnetz (WiN) angeschlossenen Institutionen einerseits und der Bereitstellung der für institutionenübergreifende Telefonie notwendigen Infrastruktur-Komponenten innerhalb des WiN andererseits.

Neben der Evaluierung, Bestimmung und ggf. Entwicklung von Systemkomponenten spielt in WIPTTEL Betrachtungen der Auswirkungen auf die Netzinfrastruktur des WiN ebenso eine Rolle wie potentielle Mehrwertdienste, die das DFN auf Basis von IP-Telefonie anbieten könnte. Da die Lösungen nicht nur in lokalen Netzen funktionieren sollen, spielen Skalierbarkeit in Bezug auf angebundene Standorte und Zahl der Nutzer pro Standort eine Rolle.

Die im Rahmen von WIPTTEL erarbeitete Referenzkonfiguration soll IP-Telefonie-endpunkte für lokale Standorte, ein lokales Verwaltungssystem für Nutzer und Ressourcen, Realisierung von typischen Funktionen von Nebenstellenanlagen und Gateways zwischen verschiedenen Signalisierungsprotokollen (u.a. ISDN) umfassen. Auf der WiN-globalen Ebene sollen Komponenten und Konzepte für einen Namens- und Adreßauflösungsdienst, die Verwendung und das Auffinden von Gateways, eine Sicherheitsinfrastruktur, Nutzungskonzepte der

Netzinfrastruktur und ein Konzept für Mehrwertdienste im WiN erarbeitet werden.

Bezug zu dieser Arbeit

Beide Projekte benötigen eine standortlokale Verwaltungskomponente, die dem H.323-Gatekeeper entspricht. Diese Komponente muß modular beschaffen sein, damit die im jeweiligen Projektkontext benötigten Erweiterungen leicht hinzugefügt werden können.

Die benötigte Modularität wird durch die Verwendung der ebenfalls in der Arbeitsgruppe in Zusammenarbeit mit dem University College London (UCL) entwickelten Infrastruktur für Telekonferenzsysteme (dem sogenannten *Message Bus* (MBus)) gewährleistet. Genauer über diese Infrastruktur ist dem Abschnitt 2.2.6 zu entnehmen.

Die Implementierung dieser unter Verwendung des MBus modular erweiterbaren Verwaltungskomponente zum Einsatz in den Projekten UniTel und WIP-TEL soll Aufgabe dieser Diplomarbeit sein.

Kapitel 2

Relevante Standards

Um ein tiefergehendes Verständnis für die Funktionsweise von IP-Telefonie zu schaffen, sollen nun zunächst die involvierten Protokolle genauer betrachtet werden. Hierfür sind primär die von der *International Telecommunication Union* (ITU) verabschiedeten Standards wichtig. Sie sind weitgehend Konsens und bilden somit die Basis, auf der die Lösungen erarbeitet werden und zueinander kompatibel sind.

Die Standards sind es auch, die den überwiegenden Teil der Literaturarbeit dieser Arbeit ausmachen, da sie die Details definieren, die eine Verwaltungskomponenten zu einem H.323-Gatekeeper machen.

Seitens der *Internet Engineering Task Force* (IETF) gibt es ebenfalls einige Standards zur IP-Telefonie, die als Alternativen bzw. Ergänzungen zu ITU-Protokollen verwendet werden können.

Die Sicht von ITU und IETF auf die IP-Telefonie unterscheidet sich jedoch, was nicht zuletzt auf den Hintergrund dieser Organisation zurückzuführen ist. Die Standards der ITU kommen aus dem klassischen Telefonie-Umfeld, in dem eine enge Kopplung der Teilnehmer die Regel ist, was zu einem höheren Grad an Administration führt. Die IETF hingegen vertritt eher ein System der losen Kopplung mit verteilten Zuständigkeiten.

Am besten werden die Unterschiede deutlich, wenn man die Ansätze für Konferenzarchitekturen beider Gremien betrachtet: Bei Konferenzen über eine ITU-Architektur ist meist bekannt, wer alles an der Konferenz teilnimmt, d.h. es gibt eine kontrollierende Instanz, die dafür sorgt, daß nur explizit angemeldete Teilnehmer mit den Daten versorgt werden. Bei der IETF hingegen setzt man auf ein offeneres System: Die Verteilung der Daten erfolgt via Multicast und ist somit allen zugänglich, ohne daß man sicher sein kann, wer die Daten alles empfängt.

Diese Diplomarbeit wird, da sie H.323 als Grundlage hat, sich in den wesentlichen Teilen auf die Protokolle der ITU beschränken. Einige Protokollentwürfe der IETF können jedoch als sinnvolle Erweiterung verwendet werden und werden daher ebenfalls behandelt.

Zum besseren Verständnis wird die Verwendung des Glossars und Abkürzungsverzeichnisses empfohlen.

2.1 ITU-T-Standards/Empfehlungen

Die Implementierung des Gatekeepers stützt sich stark auf eine Reihe von Standards der **International Telecommunication Union (ITU)**, namentlich H.323, H.225.0, H.245 und Q.931. Der Inhalt und die Bedeutung dieser Standards sollen im folgenden kurz erläutert werden.

2.1.1 H.323 v2

Titel: „*Packet Based Multimedia Communications System*“

Diese Empfehlung befaßt sich damit, wie generell eng-gekoppelte Multimedia-Kommunikation in paket-orientierten Netzen, wie sie z.B. durch IP zur Verfügung gestellt werden, erfolgen soll. Es wird dabei auf sowohl auf Endpunkte, Server und Dienste eingegangen, wie auch beschrieben wird, wie Audio-, Video- und sonstige Datenströme bzw. Kombinationen davon kodiert, verwendet und übertragen werden können. In diesem Zusammenhang definiert H.323 das Zusammenspiel der unterschiedlichen Protokolleinheiten. So wird z.B. H.225.0 zur Paketierung und Synchronisation verwendet, H.245 [14] zur Call-Control, H.261 und H.263 als Video-Codecs, G.711, G.722, G.728, G.729 und G.723 als Audio-Codecs und die T.120-Serie für Anwendungs Kooperation.

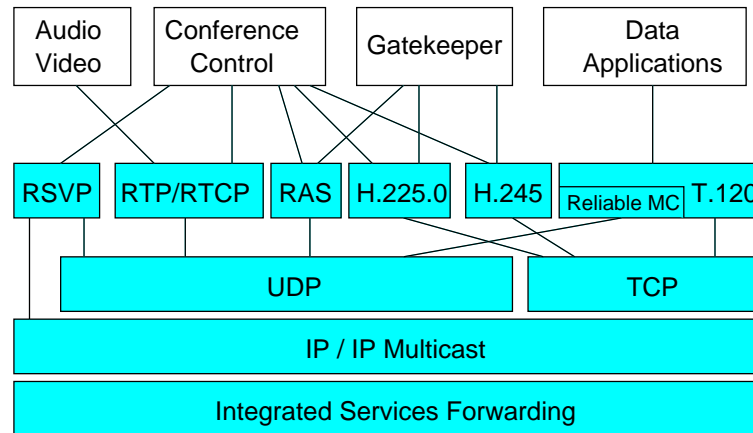


Abbildung 2.1: H323 Protokollarchitektur

Potentielle Anwendungsgebiete für H.323-Endpunkte sind stand-alone Geräte wie z.B. Telefone oder auch Integrationen in PCs und Workstations. Nicht jedes Gerät muß alle Medientypen unterstützen. Einzig Audioübertragung sollte von allen Endpunkten unterstützt werden.

Komponenten eines H.323-Systems

Neben der Verwendung der Standards beschreibt H.323 aber auch Komponenten eines IP-Telefoniesystems (vergl. Abb. 2.2) und deren Zusammenspiel. Wie schon im Abschnitt 1.3.1 erwähnt, wäre da zunächst der **Gatekeeper**, der die

Rolle einer zentralen Verwaltungs- und Vermittlungskomponente spielt. Der Gatekeeper verwaltet alle weiteren Geräte seines Einflusses, der sogenannten *Zone*. Wie groß dieser Einflussesbereich ist, hängt von der konkreten Konfiguration ab.

Die **Terminal**-Komponenten sind Endsysteme, die das Äquivalent eines gängigen Telefons darstellen. Jedes Terminal ist, wie alle anderen Komponenten auch, genau einem Gatekeeper zugeordnet, den es zu Betriebsbeginn ermittelt.

Um auch Konferenzschaltungen zwischen Teilnehmern zu ermöglichen, gibt

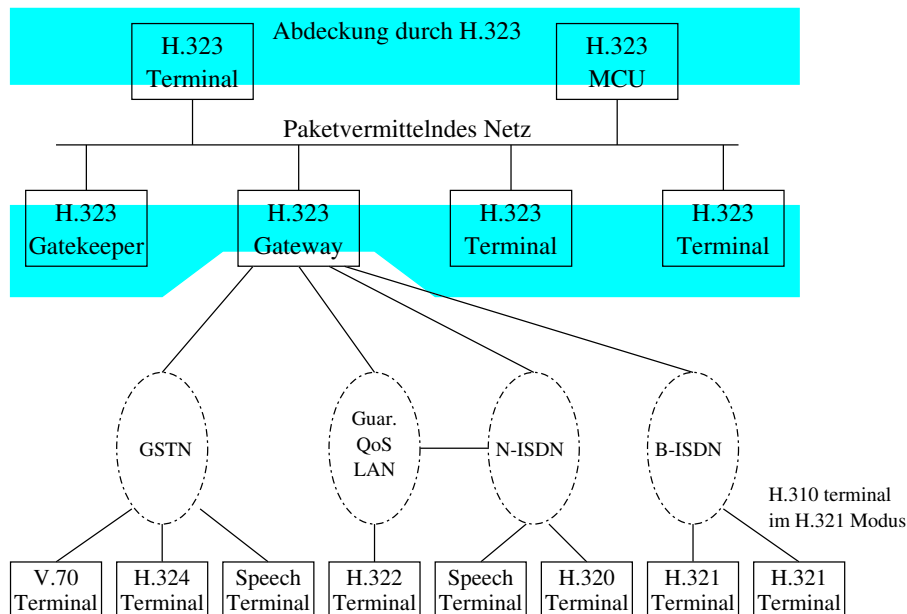


Abbildung 2.2: Interoperabilität von H.323-Endpunkten [16]

es darüber hinaus auch noch **Multipoint Controller (MC)**, deren Aufgabe es ist, die Konferenzsteuerung zu übernehmen und die zu verwendenden Codecs auszuhandeln. Ein **Multipoint Processor (MP)** sorgt, sofern vorhanden, dafür, daß Medienströme gemischt und ggf. von einem Codec auf einen anderen übersetzt werden. Eine **Multipoint Control Unit** besteht aus einem MC und einem optionalem MP.

MCUs sind optional, d.h. ein System ohne MCUs funktioniert, bietet aber keine Konferenzschaltungsfunktionalität.

Mit den bisher angesprochenen Komponenten werden Gespräche innerhalb des gleichen Netztyps ermöglicht. Wenn jedoch Anrufe z.B. zwischen dem IP-Netz und dem ISDN-Netz erfolgen sollen, bedarf es einer Komponente, die die unterschiedlichen Signalisierungen, z.B. der Anzeige eines Verbindungswunsches oder des Verbindungsendes, ineinander umsetzt. Dies ist die Aufgabe des **Gateways**. In einem System kann es durchaus mehrere Gateways geben, die zu unterschiedlichen oder auch gleichen Netzen vermitteln. Vergleiche dazu Abschnitt 1.3.1.

Die Abbildung 2.2 zeigt den Bereich an, der durch H.323 abgedeckt wird.

Adressierung von Teilnehmern

Im herkömmlichen Telefonsystem werden Personen Telefonnummern zugeordnet, über die sie erreichbar sind. Im Prinzip wird diese Nummer aber nicht einer oder mehreren Personen zugeordnet, sondern einem Apparat oder besser einem Anschluß (im Festnetz) oder einer Telefonkarte (bei Mobilfunk).

Bei IP-Telefoniesystemen werden nun, wie z.B. bei eMails, Personen oder Posten bezeichnet. Eine **H.323-Adresse** sieht daher auch ähnlich einer eMail-Adresse aus und wird ähnlich hierarchisch gebildet.

Eine Person, die an einem Rechner eine IP-Telefonieanwendung startet, ist über die der Person zugeordneten H.323-Adresse erreichbar, unabhängig davon, wo sie gerade arbeitet. Soll also jemand angerufen werden, muß zunächst herausgefunden werden, wo sich die Person aufhält, besser gesagt, welche Adresse das Gerät hat, an dem sich die zu rufende Person aufhält. Diese Adresse nennt man **Transportadresse**.

Um jemanden anrufen zu können, muß also zunächst eine Abbildung einer H.323-Adresse auf eine Transportadresse erfolgen. Diese **Adreßauflösung** ist eine der Aufgaben des Gatekeepers der Zone.

Anrufmodelle

Es kann sein, daß der Gatekeeper entscheidet, statt der Transportadresse des gewünschten Teilnehmers, seine eigene Transportadresse beim Verbindungsaufbau zurückzuliefern. Dies hat den Effekt, daß der Anrufer den Gatekeeper anruft, statt den gewünschten Teilnehmer. In diesem Fall vermittelt der Gatekeeper den Anruf weiter an das eigentliche Ziel und verbleibt als Kontrollinstanz in der Mitte.

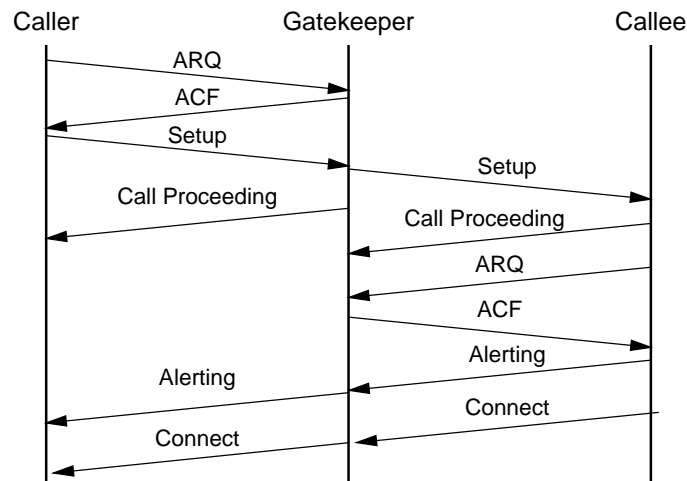


Abbildung 2.3: Beispiel eines Gatekeeper-routed-Calls mit einem Gatekeeper

Dieses **Anrufmodell**, bei dem der Gatekeeper einer Zone als Zwischenstation für die Steuerinformationen an der Verbindung teilnimmt, nennt man

Gatekeeper-routed (Abb. 2.3). Ist der Gatekeeper nicht beteiligt, spricht man von einer **direkten** Kommunikation (Abb. 2.4).

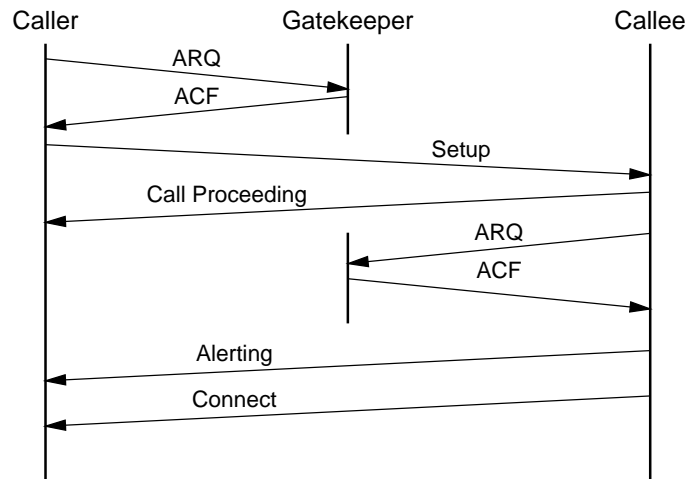


Abbildung 2.4: Beispiel eines Direct-Calls mit einem Gatekeeper

Bei zonenübergreifenden Anrufen sind zwei Gatekeeper involviert, von denen jeder für seine Zone entscheidet, ob die Anrufe direkt oder gatekeeper-routed erfolgen sollen.

Aber selbst bei der direkten Kommunikation liefert der Gatekeeper ggf. eine andere als die Zieladresse zurück, nämlich immer dann, wenn für die Kommunikation ein Gateway verwendet werden muß. In diesem Fall kann der Zielpunkt nicht direkt angerufen werden und es wird stattdessen mit einem Gateway kommuniziert, welches die Daten weiterleitet.

Konferenzen

In H.323 sind nicht nur Zweipunkt-Kommunikationsbeziehungen beschrieben, d.h. Gespräche zwischen zwei Teilnehmern, sondern auch Konferenzen. Dabei ist zu erwähnen, daß der Wechsel von einer Zweipunkt- zu einer Mehrpunkt-Kommunikation und zurück auch während des Gesprächs erfolgen kann.

Die Voraussetzung für Konferenzen ist das Vorhandensein eines MCs in der Zone eines der beteiligten Teilnehmer.

Datenkanäle

Bei H.323 werden verschiedene Arten von Daten auf getrennten Kanälen, ausgetauscht. Zum einen gibt es den UDP-basierten *Request, Admission and Status* (RAS)-Kanal, dem Steuerungskanal zwischen Endpunkten und Gatekeeper. Die hier ausgetauschten Nachrichten (PDUs) sind in H.225.0 (siehe 2.1.2) definiert.

Die Signalisierung von Anrufen erfolgt auf dem auf TCP aufsetzenden Verbindungssteuerungskanal *Call-Signaling* — beim direkten Anrufmodell zwischen zwei Endpunkten oder beim Gatekeeper-routed-Modell zwischen Endpunkten und Gatekeepern und ggf. zwischen Gatekeepern untereinander. Für

Konferenzschaltungen können auch noch MCUs involviert sein. Die hier ausgetauschten PDUs sind ebenfalls in H.225.0 und in Q.931 definiert.

Die Aushandlung der verwendeten Codecs und die Konfiguration der Endpunkte erfolgt auf einer separaten TCP-Verbindung, die erst nach erfolgtem Call-Signaling aufgebaut wird. Hierfür werden PDUs nach H.245 verwendet.

Schlußendlich gibt es noch die logischen Kanäle, auf denen die eigentlichen Daten transportiert werden. In Abhängigkeit von den zu übertragenen Daten (Audio, Video, Anwendungsdaten) wird hier entweder eine TCP-Verbindung oder im Regelfall der Versand von UDP-Datagrammen gewählt.

Mehrwertdienste

Wie schon im Abschnitt 1.1.2 in Bezug auf ISDN erwähnt, bieten moderne Telefonnetze eine Reihe von Diensten, die über das einfache Telefonieren hinausgehen. H.323 sieht daher vor, daß mindestens die typischen Funktionen einer heutigen Telefonanlage von einem Gatekeeper beherrscht werden sollten, bzw. daß Endpunkte nach H.323 die Funktionalität eines gängigen ISDN-Telefons haben sollten.

Protokollphasen

Im folgenden soll ein knapper Überblick über die Funktionsweise eines H.323-Systems gegeben werden. Das Verständnis hierfür ist wichtig, um die Rolle der Verwaltungskomponente, dem Gatekeeper, richtig einzuschätzen. Aus diesem Grund soll nun der Ablauf eines Telefonates aus der Sicht eines Endpunktes, d.h. z.B. eines IP-Telefons, zu verfolgen. Die folgende Phasenübersicht leistet dies und gibt zudem an, welche PDUs des H.225.0-Standards (siehe 2.1.2) der Endpunkt versendet.

Phase 1 — Initialisierung Wenn der Endpunkt seinen Betrieb aufnimmt, weiß er in der Regel nichts über andere Teilnehmer bzw. Gatekeeper. Deshalb wird zunächst ein Gatekeeper gesucht, der den Endpunkt mit in seine Zone aufnimmt. Hierfür sendet der Endpunkt eine Anfrage (*Gatekeeper-Request* (GRQ)) an eine vordefinierte Multicast-Adresse, die sogenannte *Discovery-Address*, auf der jeder Gatekeeper empfangsbereit ist. Alle Gatekeeper, die diese Anfrage erhalten, prüfen, ob sie sich für den Endpunkt zuständig fühlen und teilen ihm dies mit. Aus den eingehenden positiven Antworten sucht sich der Endpunkt einen Gatekeeper aus, mit dem er zusammenarbeiten will.

Bei dem so ausgewählten Gatekeeper registriert sich der Endpunkt jetzt, indem er ihm ein *Registration Request* (RRQ) auf dem RAS-Kanal zusendet. Wenn der Gatekeeper die Registrierung bestätigt hat, ist die Initialisierung abgeschlossen, und der Teilnehmer am Endpunkt kann jetzt Anrufe tätigen bzw. angerufen werden.

Phase 2 — Berechtigung einholen Soll eine Verbindung zu einem anderen Teilnehmer aufgebaut werden - sei es, weil man anrufen will oder einen Anruf entgegennehmen möchte — muß zunächst die Erlaubnis des Gatekeepers eingeholt werden. Da der Gatekeeper die Ressourcen des Netzes verwaltet, kann er z.B. entscheiden, daß momentan wegen fehlender Bandbreite keine weiteren



Abbildung 2.5: Protokollphasen bei H.323

Anrufe mehr zugelassen werden. Des weiteren wird in dieser Phase auch Adreßauflösung betrieben.

Vor einem Anruf sendet der Endpunkt also eine *Admission Request* (ARQ) an den Gatekeeper, in der er angibt, wen er anrufen möchte und welche Bandbreite er wünscht. Der Gatekeeper kann den Wunsch unter Angabe eines Grundes ablehnen oder dem Endpunkt die Adresse nennen, unter der er den Endpunkt erreicht.

An dieser Stelle entscheidet der Gatekeeper auch, welches Anrufmodell (s.o.) er verwenden möchte.

Phase 3 — Anrufsignalisierung Sobald der Endpunkt vom Gatekeeper erfahren hat, welche Adresse er anrufen soll, baut er eine TCP-Verbindung zu der Adresse auf. Je nach Anrufmodell oder Verwendung eines Gateways, kann es sich hierbei um den eigenen bzw. den entfernten Gatekeeper, ein Gateway oder den Zielendpunkt handeln.

Analog zum bisherigen Signalisierungssystem von ISDN, wird eine SETUP-Nachricht versendet, die der angerufene Endpunkt zunächst mit einem CALL-PROCEEDING beantwortet, um mitzuteilen, daß der Anruf angekommen ist. Telefoniert der Angerufene nicht und ist auch keine Rufumleitung eingestellt, so sendet der entfernte Endpunkt nun eine ALERTING-Nachricht, um zu signalisieren, daß es beim entfernten System „klingelt“. Ein CONNECT signalisiert schließlich, daß die Verbindung zustande gekommen ist.

Alle soeben genannten Nachrichten sind in H.225.0 definiert und werden in Q.931-Nachrichten im UserData-Teil ausgeliefert.

Phase 4 — Aushandlung und Konfiguration Haben sich die beiden Endsysteme, zwischen denen eine Kommunikationsbeziehung aufgebaut werden soll, gefunden, muß ausgehandelt werden, wie sie die Daten miteinander austauschen und welche Rolle wer spielt.

Da mehrere Audio- und Videocodecs, d.h. Möglichkeiten Audio- und Videodaten zu kodieren und komprimieren, zur Verfügung stehen, aber in der Regel

nicht jedes Endsystem alle davon unterstützt, müssen sich die Teilnehmer darauf einigen, welches Verfahren verwendet werden soll.

Zu diesem Zweck wird zwischen den Endpunkten eine weitere TCP-Verbindung aufgebaut, auf der, in H.245 kodiert, dem jeweils anderen Endpunkt mitgeteilt, welche Fähigkeiten ein Endpunkt besitzt. Diesen Vorgang des Austauschens von Fähigkeiten nennt man **Capability Exchange**.

Anschließend wird auf der gleichen H.245-Verbindung bestimmt, wer die Kontrolle über die Kommunikation übernimmt, d.h. wer **Master** und wer **Slave** ist. Als Regel gilt, daß immer der die Rolle des Masters übernimmt, der mehr Fähigkeiten besitzt, d.h. der z.B. der in der Lage wäre, aus einer Zweipunkt-Kommunikation eine Konferenz zu machen.

Phase 5 — Medienaustausch Der Master initiiert nun die Datenübertragung. Aus den von allen Teilnehmern unterstützten Codecs sucht er die zu benutzenden aus, legt fest, welche Medien und ggf. welche weiteren Parameter verwendet werden. All dies teilt er den anderen Endpunkten beim Aufbau eines logischen Kanals mit.

Der eigentliche Datentransport findet dann entweder, im Falle von Audio und Video, mittels des *Realtime Transport Protocol* (RTP) auf UDP oder im Falle von Application Sharing nach T.120 auf einer TCP-Verbindung bzw. mit Hilfe von Reliable Multicasting statt.

Phase 6 — Neuaushandlung Falls weitere Teilnehmer dazustoßen, muß erneut ausgehandelt werden, welches die gemeinsamen Fähigkeiten sind. Die hierzu verwendeten Mechanismen entsprechen denen in Phase 4.

Phase 7 — Verbindungsabbau Wenn ein Teilnehmer das Gespräch bzw. seine Konferenzteilnahme beendet, so schließt er die logischen Kanäle, auf denen er die Mediendaten empfangen hat, und teilt dem Partner mit, daß er die Sitzung beendet. Dieser schließt daraufhin seinerseits die logischen Kanäle und beendet die Sitzung. Wenn der Kanal zur Anrufsignalisierung aus Phase 3 noch besteht, wird auch hier das Ende der Verbindung mitgeteilt. Auf diese Weise kann beim Gatekeeper-routed-Anrufmodell der Gatekeeper an dieser Stelle schon erfahren, daß das Gespräch beendet ist.

Anschließend teilt der Endpunkt seinem Gatekeeper mit einer *Disengage Request*-Nachricht (DRQ) mit, daß das Gespräch beendet wurde. Der Gatekeeper gibt daraufhin die vom Endpunkt belegten Ressourcen wieder frei.

Will sich der Endpunkt ganz abmelden, d.h. keine weiteren Gespräche führen oder soll das Gerät abgeschaltet werden, sendet er zum Gatekeeper seine Abmeldung in Form einer *Unregistration Request*-Nachricht (URQ). Der Gatekeeper streicht dann den Endpunkt aus seinen internen Listen.

2.1.2 H.225.0

Titel: „*H.225.0 — Call Signaling Protocols and Media Stream Packetization for Packet Based Multimedia Communications Systems*“

Dieser Standard [15] beschreibt, wie Audio-, Video-, Anwendungs- und Steuerdaten verwendet, kodiert und paketierte werden, um in einem paket-basierten

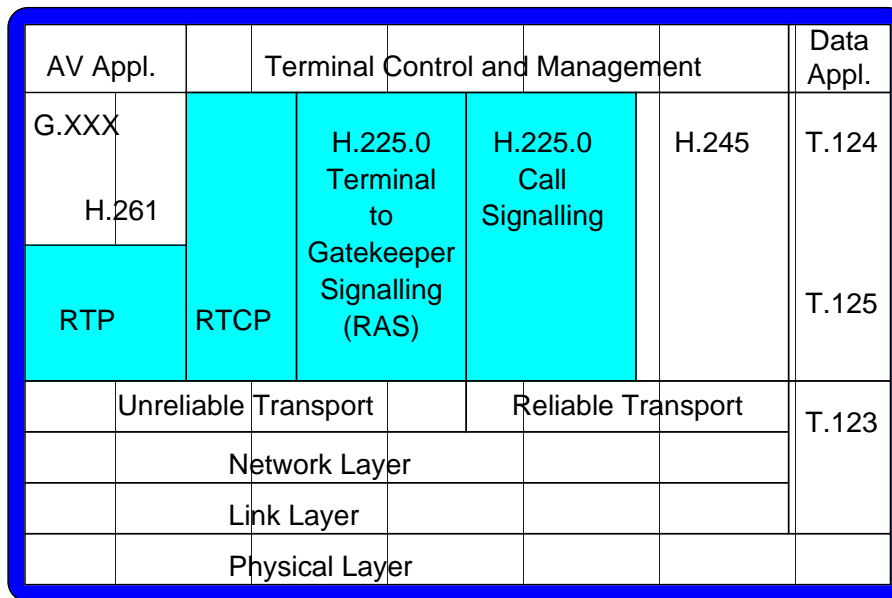
Netz zwischen H.323-Endpunkten übertragen zu werden. Dabei werden die einzelnen Komponenten und Verfahren in H.323 beschrieben, während H.225.0 die Protokolle und Nachrichtenformate in ASN.1[13] behandelt.

H.225.0 ist dazu ausgelegt, in einer Reihe von paket-basierten Netzen zu arbeiten. Dabei wird lediglich die Kommunikation von H.323-Endpunkten in gleichartigen Netztypen betrachtet,. Die Qualität der Übertragungen im gesamten Internet oder auch einfach nur in andersartigen paket-basierten Netzen, ist nicht Teil des Standards.

Obwohl H.225.0 oberhalb der Transportschicht (TCP/UDP) angesiedelt ist, wird doch auf einige Besonderheiten in Zusammenhang mit bestimmten Transportprotokollen eingegangen..

H.225.0 macht Gebrauch von RTP/RTCP zur Paketierung und Synchronisation von Medienströmen für alle unterliegenden Netze.

H.323 Protocol Stack



Reichweite von H.225.0

Abbildung 2.6: H.225.0-Anteil in einem Endpunkt, entnommen aus [15]

Die Verwendung von Q.931

H.225.0 definiert, daß die Nachrichten zur Anrufsignalisierung (*Setup, Call Proceeding, Alerting, ...*) in Q.931 (s. 2.1.4) „eingepackt“ werden müssen. Dafür wird beschrieben, welche Felder der Q.931 Nachrichten wie auszufüllen sind und an welcher Stelle die H.225.0-PDUs in der Q.931-Nachricht plaziert werden.

2.1.3 H.245

Titel: „*H.245 — Control Protocol for Multimedia Communication*“

Dieser Standard beschreibt Syntax und Semantik von Nachrichten und der zugehörigen Verfahren, die zur Aushandlung der Form und Qualität der zu übertragenden Medienströme verwendet werden. Die Nachrichten und Prozeduren umfassen dabei folgende Bereiche:

- **Master-Slave-Bestimmung**

Um zu vermeiden, daß zwei Endpunkte gleichzeitig ein Ereignis auslösen und dies in einen ungewollten Zustand führt, wird unterschieden zwischen Master- und Slave-Endpunkt. Kritische Aktionen dürfen nur von Master-Endpunkten durchgeführt werden.

In H.245 ist definiert, wie bestimmt wird, wer Master und wer Slave ist, was z.B. für die Arbeit mit Multipoint-Controllern (MC) von Bedeutung ist.

- **Capability-Exchange**

Capability Exchange ist ein Verfahren, bei dem ermittelt wird, welche Formen von Multimediaströmen von den beteiligten Endpunkten unterstützt werden. Hierzu teilt jeder Endpunkt dem anderen mit, welche Kodierungen von Medien und welche Transportarten er unterstützt. H.245 definiert Verfahren und Nachrichten, die den Austausch der Fähigkeiten ermöglichen.

- **Signalisierung für logische Kanäle**

Als logischer Kanal wird die Verbindung zwischen zwei Endpunkten bezeichnet, auf der die Multimedia- bzw. Applikationsdaten ausgetauscht werden. H.245 stellt Syntax, Semantik und Verfahren zur Verfügung, mit denen logische Kanäle geöffnet und geschlossen werden können, damit die Empfangsbereitschaft für Daten gesichert werden kann.

- **Anforderung bestimmter Modi**

Nachdem die jeweiligen Fähigkeiten ausgetauscht wurden, muß ein Endpunkt (der Master) dem anderen signalisieren, welcher Modus nun verwendet werden soll. Die hierfür nötigen Nachrichten sind ebenfalls in H.245 definiert.

- **Weitere Kommandos und Meldungen**

In H.245 sind Kommandos bzw. Meldungen zur Verfahrenssteuerung der Applikationen, wie z.B. das An- und Ausschalten von Audio- und/oder Videoströmen, die Wahl einer Verschlüsselung der Ströme oder Flußkontrollkommandos definiert.

2.1.4 Q.931

Titel: „*Q.931 — ISDN User-Network Interface Layer 3 Specification for Basic Call Control*“[12]

Q.931 spezifiziert die Nachrichtensyntax und -semantik für die Anrufsteuerung

auf dem D-Kanal von ISDN-Systemen. Behandelt werden Verbindungsauf- und -abbau sowie Verbindungssteuerung.

Alle H.225.0-Nachrichten zum Verbindungsaufbau werden in Q.931 „eingepackt“ und übertragen. Dabei werden nicht alle Features dieses Standards verwendet — es wird lediglich auf die bewährte Funktionalität von Q.931 bei ISDN zurückgegriffen, um diese auch auf der LAN-Seite zu nutzen.

2.2 Empfehlungen und Entwürfe der IETF

Die im folgenden beschriebenen Entwürfe der *Internet Engineering Task Force* behandeln Fragen, die die ITU-Empfehlungen noch offen lassen. Allen Entwürfen ist zu eigen, daß sie sich noch in der Entwicklung befinden und daher zunächst nur eine begrenzte Lebensdauer von sechs Monaten haben.

Obwohl es sich bei den vorgestellten Entwürfen um noch nicht festgeschriebene Protokolle und Verfahren handelt, sind sie doch in der Regel so weit brauchbar, daß ihre jeweiligen Aspekte für den Entwurf eines Gatekeepers berücksichtigt werden sollten.

2.2.1 IETF-Draft SIP

Titel: „*SIP: Session Initiation Protocol*“ [18]

SIP ist ein Signalisierungsprotokoll zum Erzeugen, Ändern und Beenden von Kommunikationsbeziehungen von ein oder mehr Teilnehmern. Ursprünglich lediglich als Teil der IETF-Architektur für lose gekoppelte Konferenzen entwickelt, um Konferenzteilnehmer direkt einzuladen, anstatt die Einladungen der allgemeinen Öffentlichkeit bekanntzumachen, hat sich seine Funktionalität seit dem erweitert. Heute beabsichtigt SIP, wie auch H.323, Telefonanrufe und Multimediakonferenzen auf IP-basierten Netzen zu ermöglichen.

Ein SIP-System kennt folgende Komponenten:

- **Endpunkte**
Ein SIP-fähiger Endpunkt kann eine Software auf einem Computer oder ein Telefon sein. Jedem Endpunkt sind ein oder mehr Adressen, d.h. SIP URLsm und/oder Telefonnummern zugeordnet.
- **Gateway**
Ein Gateway ist ein spezieller SIP-Endpunkt, der die Kommunikation mit anderen Protokollumgebungen, wie z.B. H.323, oder Netzumgebungen, wie z.B. ISDN, ermöglicht.
- **Proxy Server**
Ein Proxy-Server verwaltet die Endpunkte einer SIP-Umgebung. Dies bedeutet primär, daß sich SIP-Endpunkte beim Proxy Server registrieren und dieser Adreßauflösung und die Wegewahl für Anrufe vornimmt. Er ist somit vergleichbar mit einem H.323-Gatekeeper.
- **Redirect Server**
Ein Redirect-Server ist ein Server, der SIP-Adressen auf eine oder mehr andere SIP-Adressen abbildet.

Es fällt auf, daß eine SIP-Umgebung ähnlich einer H.323-Zone aufgebaut ist. Anders als in H.323 existiert allerdings kein Multipoint-Controller, da davon ausgegangen wird, daß SIP-Endpunkte mit anderen Endpunkten über ein darunterliegendes IP Netz via Multicast kommunizieren können.

Wie auch H.323 ist SIP oberhalb der Internet-Protokolle (IP, IP Multicast, TCP, UDP) angesiedelt (vergl. Abb. 2.7 und 2.1). Auch SIP sieht vor, daß der Medienaustausch über RTP/RTCP und die Ressourcenreservierung über RSVP realisiert wird.

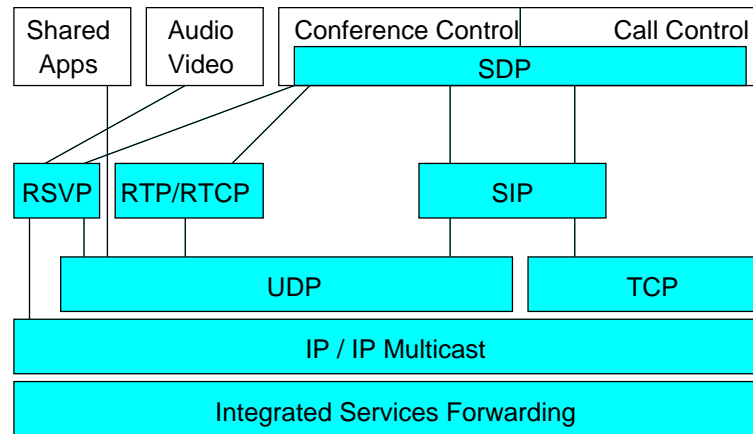


Abbildung 2.7: SIP Protokollarchitektur

Zur Beschreibung der Anrufe bzw. Konferenzen werden SDP-Blöcke als Payload in SIP-Nachrichten übertragen. Das *Session Description Protocol* (SDP) beschreibt Medienströme in Bezug auf verwendete Codecs, Paketisierungsformate, Transportadressen, etc. SDP erledigt also teilweise das, was in einem H.323-System durch H.245 erledigt wird.

2.2.2 IETF-Draft TBGP

Titel: „*The IP Telephony Border Gateway Protocol Architecture*“ [11]

TBGP nimmt sich des Problems des *Call Routings* an, d.h. der Wahl des Weges, den ein Anruf nehmen soll. Immer wenn ein Anruf aus einem IP-Netz heraus zu einem PSTN-Telefon erfolgen soll, muß ein Gateway verwendet werden, das die Umsetzung von einem Netz ins andere vornimmt. Es kann hierfür oft mehrere Gateways geben, die den Anruf zu unterschiedlichen Kosten ausführen können und unterschiedlich ausgelastet sind — die Schwierigkeit liegt darin, zu entscheiden, welches davon verwendet werden soll.

Das Konzept sieht vor, daß in jeder Zone ein TBGP-Repräsentant ist, dessen Aufgabe es ist, TGBP-Sprecher anderer Zonen darüber zu informieren, welche IP-Knoten in der Zone sind und welche PSTN-Ziele über Gateways der Zone erreicht werden können.

TBGP ist dazu ausgelegt, Eigenschaften der Gateways wie z.B. Verbindungskosten, zu verbreiten. Das Auffinden von Gateways und die Verbreitung von Informationen über die Erreichbarkeit von PSTN-Telefonen wird durch das *Gateway Location Protocol* (siehe 2.2.3) ermöglicht.

TBGP basiert auf dem *Border Gateway Protocol* (BGP), d.h. es verwendet ebenfalls TCP als Transportprotokoll und nutzt zur Kommunikation zwischen zwei TBGP-Sprechern den selben Satz an Nachrichten wie BGP.

Ein TBGP-Sprecher sendet Informationen über seine Zone an seine benachbarten TBGP-Sprecher anderer Zonen. In jeden Streckenabschnitt, den eine TBGP-Nachricht dabei nimmt, kann diese modifiziert werden, um z.B. die Eigenschaften des Weges der Nachricht wiederzugeben.

Empfängt ein TBGP-Sprecher eine solche Information aus einer anderen Zone, so nimmt er die Informationen in seine Datenbestände mit auf. Werden unterschiedliche Informationen für identische Ziele empfangen, so wählt der TBGP-Sprecher die „besseren“, d.h. billigeren oder kürzeren Strecken aus und verwirft die anderen.

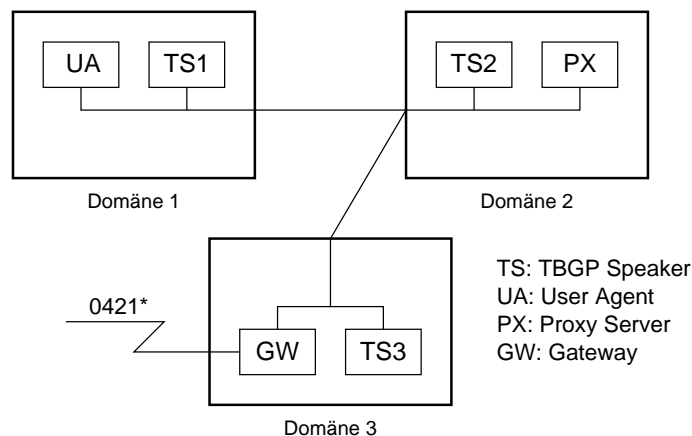


Abbildung 2.8: Interaktion mit den TBGP-Sprechern [11]

Ein TBGP-Sprecher stellt seine Dienste Telefonie-Endpunkten, Gateways oder Proxy-Servern zur Verfügung. Nach dem besten Weg in Richtung Ziel gefragt, teilt er dem Fragesteller immer den nächsten Schritt in Richtung Ziel mit. Ein Beispiel (vgl. Abb. 2.8):

Ein Telefonie-Endpunkt aus der Domäne 1, der eine Telefonnummer 04212182972 anrufen möchte, fragt den TBGP-Sprecher seiner Domäne nach dem besten Weg dorthin. Dieser weiß, daß dazu ein Gateway in Domäne 3 geeignet ist und daß der kürzeste Weg dorthin über Domäne 2 führt. Der TBGP-Sprecher der Domäne 1 antwortet daher mit der Adresse des Proxy-Servers in Domäne 2. Der Endpunkt ruft daraufhin den Proxy-Server an, der wiederum seinem TBGP-Sprecher nach dem besten Weg zum Ziel (Domäne 3) befragt und die Adresse des Gateways zurückerhält. Der Proxy-Server leitet den Anruf also an das Gateway weiter. Das Gateway wiederum akzeptiert ausgehende Anrufe in das 0421-Ortsnetz und ruft den Teilnehmer an.

2.2.3 IETF-Draft GLP

Titel: „*A Framework for a Gateway Location Protocol*“ [32]

Ähnlich wie TBGP beschäftigt sich GLP¹ mit dem Problem des Auffindens von Gateways und Attributen zur Verwendung von Gateways. Es stellt jedoch keine Ergänzung zu TBGP, sondern eine, auf einem anderen Ansatz basierende, Alternative dar. Zwar nutzt auch GLP die Mechanismen von BGP, um Routing-Informationen zwischen den Servern auszutauschen, jedoch findet der Datenverkehr auf der Anwendungsebene und nicht auf der Netzebene statt.

GLP geht davon aus, daß in jeder Zone ein Location Server (LS) existiert, bei dem sich ein Gateway zu Beginn anmeldet. Wie dies geschieht, ist hier nicht definiert.

Im Falle von H.323 wäre der Location Server idealerweise ein Teil des Gatekeepers und die Registrierung erfolgt über eine RRQ-Nachricht auf dem RAS-Kanal.

Die Location Server tauschen untereinander Gateway-Informationen mittels einer TCP-Verbindung aus. Hierzu dient das *Gateway Location Protocol* (GLP) [33]. Das GLP sieht vor, daß Location Server initial ihre gesamten Datenbestände austauschen und später nur noch aktualisieren.

Anhand einer Policy entscheiden Location Server, welche Informationen sie generieren, verbreiten und akzeptieren. Bei den ausgetauschten Informationen sollte es sich aber minimal um Routing-Informationen handeln, d.h. Angaben, welche Telefonnummern von einem Gateway aus erreicht werden können, und eine IP-Adresse, um das Gateway zu erreichen. Weitere Angaben können die Auslastung des Gateways oder die Kosten für Gespräche beinhalten. GLP sieht Definitionen für die Angabe von erreichbaren Telefonnummern und die IP-Adresse des nächsten Schritts auf dem Weg dorthin vor, für Kosten jedoch auf Grund der möglichen Tarifvielfalt bisher nicht. Stattdessen wird die Möglichkeit vorgesehen, Quellen für Tarifinformationen zu verbreiten.

2.2.4 IETF-Draft PGRP

Titel: „*A Framework for a Peer Gatekeeper Routing Protocol*“ [4]

PGRP beschäftigt sich mit dem Problem, daß ein Gatekeeper nach H.323 nur Informationen über Endpunkte und Gateways in seiner eigenen Zone hat. Um jedoch Anrufe zu Endpunkten außerhalb seiner Zone zu machen, muß ein Gatekeeper zunächst Informationen über andere Zonen erhalten.

PGRP ist ein Intra-Domain Protokoll speziell für den Informationsaustausch zwischen Gatekeepern. Angelehnt an den *Domain Name Service* (DNS) sieht PGRP eine Hierarchie von Gatekeepern vor. Ein Gatekeeper tauscht seine Informationen mit einem ihm übergeordneten *Topology Server* (TS) aus.

Zu Beginn versucht ein Gatekeeper seinen zuständigen Topology-Server durch Multicasten einer *Topology Server Request* (TSR) an eine wohldefinierte Adresse zu ermitteln. Das Verfahren ist analog zu dem der Gatekeeper-Discovery bei

¹Nach einer Diskussion auf der H.323-Implementors-Mailingliste im September zeichnet sich TRIP (Telephony-Routing over IP-Protocol) als neuer Name für GLP ab.

H.323-Endpunkten.

Wurde der Topology-Server gefunden, baut der Gatekeeper einen bidirektionalen Kanal für Topologie-Informationen auf.

Damit Skalierbarkeit für größere Netze gewährleistet wird, unterstützt PGRP eine verteilte Architektur von Topology-Servern. Mehrere von Gatekeepern verwaltete Zonen werden zu einer *Area* zusammengefaßt, für die genau ein Topology-Server zuständig ist.

Um nun der Informationsaustausch zwischen Topology-Servern über die Grenzen einer Area hinweg erfolgen kann, erfahren die Topology-Server voneinander initial über den Austausch von Multicast-Nachrichten. Sie bestimmen einen Topology-Server, der die Rolle des *Designated Topology Servers* (DTS) übernimmt und der fortan für die Verteilung von Informationen zwischen Topology-Servern zuständig ist.

Kann ein Gatekeeper eine Adresse nicht selbst auflösen, sendet er an seinen TS eine *Topology State Request*. Der TS wiederum ermittelt aus seinen eigenen Daten in Form eines *Topology State Update* (TSU) eine Antwort, die er sowohl an den Gatekeeper, wie auch dem DTS mitteilt. Der DTS sorgt wiederum für die Verteilung der TSU an alle anderen TS, so daß alle TS eine aktuelle Information erhalten.

PGRP sieht redundante Gatekeeper und Designated Topology Server vor, damit ein ausfallsicherer Betrieb gewährleistet werden kann.

2.2.5 IETF-Draft Call Processing Language (CPL)

Titel: „*CPL: A Language for User Control of Internet Telephony Services*“[22]

Bei CPL handelt es sich um den Entwurf einer Skriptsprache, mit der man festlegen kann, wie mit eingehenden Anrufen verfahren werden soll, d.h. ob sie z.B. abgelehnt oder umgeleitet werden sollen. Als Grundlage wird die *Extensible Markup Language* XML, eine Sonderform von SGML, verwendet, da Skripte dadurch leicht parsierbar und grafisch darstellbar werden.

Die *Document Type Definition* (DTD) von CPL definiert folgende Sprachelemente:

- **Switches**
Switches erlauben die bedingte Behandlung von Anrufen. Bisher ist es möglich, Anrufe in Abhängigkeit von Attributen wie Sender, Empfänger und Betreff oder nach der aktuellen Uhrzeit zu behandeln.
- **Locations**
Locations sind Angaben über Adressen, die mit CPL behandelt werden können. Locations können statisch im Skript angegeben oder aber zur Laufzeit ermittelt werden.
- **Aktionen**
Aktionen beschreiben, was mit einem Anruf geschehen soll. Zu den bisher definierten Aktionen gehören das Durchstellen oder das Umleiten zu

einer anderen Location, das Ablehnen des Anrufes, das Versenden einer Nachricht an eine Adresse oder das Eintragen in ein Log-File.

- **Links**

CPL-Skripte sind als Bäume realisiert. CPL bietet daher die Möglichkeit, auf andere Stellen im Baum zu verweisen — ähnlich einem Sprung in einem iterativen Programm.

Ein kleines Beispiel soll das Aussehen von CPL-Skripten verdeutlichen.

```
<?xml version="1.0" ?>
<!DOCTYPE call SYSTEM "cpl.dtd">

<call>
  <string-switch field="from">
    <string matches="*uni-bremen.de">
      <location url="h323:prelle@informatik.uni-bremen.de">
        <proxy>
          <busy>    <link ref="anrufbeantworter /></busy>
          <noanswer><link ref="anrufbeantworter /></noanswer>
          <failure> <link ref="anrufbeantworter /></failure>
        </proxy>
      </location>
    </string>
    <otherwise>
      <location url="h323:prelle@voicemail.uni-bremen.de"
        merge="clear" id="anrufbeantworter">
        <redirect />
      </location>
    </otherwise>
  </string-switch>
</call>
```

Es wird zunächst geprüft, wer der Anrufer ist. Stammt er aus der Uni Bremen, so wird versucht, den Anruf an die H323-Adresse *prelle@informatik.uni-bremen.de* durchzustellen. Ist dort besetzt oder kommt die Verbindung aus anderen Umständen nicht zustande, wird der Anruf an den Anrufbeantworter weitergeleitet.

Stammt der Anruf von einem Anrufer außerhalb der Uni Bremen (*otherwise*), wird er sofort auf den Anrufbeantworter umgeleitet und evtl. zuvor eingerichtete Adressen gelöscht (*merge='clear'*).

Für den Gatekeeper ist die *Call Processing Language* insofern interessant, als daß sich dadurch eine Möglichkeit bietet, von den Nutzern kontrollierte Anrufbehandlung durchzuführen. Einen entsprechenden CPL-Parser und -Prozessor vorausgesetzt, könnte ein Nutzer sein eigenes CPL-Skript schreiben, welches der Gatekeeper dann ausführen kann, wenn Anrufe für einen Benutzer eintreffen.

Im Rahmen dieser Arbeit wird es keine Unterstützung für CPL geben — lediglich die Möglichkeit zur Ablage von Skripten in den Nutzerdaten der Datenbank wird vorgesehen.

2.2.6 IETF-Drafts zum Message Bus

In der *Arbeitsgruppe Rechnernetze* wird seit einiger Zeit ein besonderes Konzept bei der Erstellung von Konferenz-Software verfolgt: Eine Anwendung setzt sich aus mehreren Komponenten zusammen, die allerdings nicht zu einem großen Programm auf einem System kompiliert und zusammengefaßt werden, sondern als eigenständige Prozesse gleichzeitig laufen.

Diese Architektur führt allerdings dazu, daß ein besonderes Verfahren verwendet werden muß, damit die Module zusammenarbeiten können, da normale Funktionsaufrufe zwischen eigenständigen Programmen, d.h. Programmen mit eigenem Adreßraum, nicht mehr funktionieren. Für dieses Problem wurde in Kooperation mit dem *University College London (UCL)* eine geeignete Schnittstelle entwickelt: der sogenannte *Message Bus (Mbus)*.

Der Bus besteht aus zwei logisch verschiedenen Teilen: einer Adressierungs- und Übertragungsinfrastruktur und einer Menge von gemeinsamen und anwendungsspezifischen Nachrichten. Das Dokument „*A Message Bus for Conferencing Systems*“ [25] beschreibt Adressierungs-, Transport- und Sicherheitsaspekte des Message Bus und „*The Message Bus: Messages and Procedures*“ [26] die Protokollprozeduren für Operationen auf dem Message Bus und Syntax und Semantik der allgemeinen Nachrichten.

Nachrichtenformat

Eine Nachricht auf dem Message Bus besteht aus einem Nachrichtenkopf und dem eigentlichen Inhalt. Im Kopf stehen nach einem festen Schema Informationen, die anzeigen, wie und wohin eine Nachricht gesendet werden soll. Die einzelnen Informationen sind:

- Ein Base64-kodierter **Message Digest**, der einen Hash-Wert der gesamten Nachricht ab dem *ProtocolID*-Feld enthält.
- Ein Feld mit einem Protokollkennzeichner (**ProtocolID**).
- Eine **Laufnummer**. Jede Applikation am Message Bus numeriert ihre Nachricht in aufsteigender Reihenfolge bei 0 beginnend.
- Ein **Zeitstempel**, der die vergangenen Sekunden seit dem 1. Januar 1970 00:00:00 Uhr enthält
- Eine Angabe des **Nachrichtentyps**, der angibt, ob die Nachricht bestätigt werden muß (R), oder nicht (U).
- Die **Adresse des Senders**.
- Die **Adresse der Empfänger**.
- Eine **Liste von Bestätigungen**, d.h. eine Liste von Laufnummern der Nachrichten, die diese Nachricht bestätigt.

Adressierung

Jedem Modul am Message Bus ist mindestens eine Adresse zugeordnet. Eine Adresse besteht aus einer Liste von *Typ/Wert*-Paaren. Gültige Schlüssel sind `conf`, `media`, `module`, `app` und `instance`. Die gültigen Werte hängen vom jeweiligen Schlüssel ab.

Ein Beispiel: Um das Audio-Tool *rat* zu adressieren, könnte `(media:audio module:engine app:rat)` eine gültige Adresse sein.

Transport

Beim MBus werden die Daten über lokales Multicast oder Unicast verteilt. Jede Anwendung, die über den MBus kommunizieren möchte, muß also auf einer wohldefinierten Transportadresse lauschen und Daten entweder direkt an andere lokale Anwendungen, d.h. andere Ports, senden oder aber rechnerlokale Multicast-Pakete verschicken.

Unicast-Pakete kommen nur in Situationen vor, in denen die anderen Einheiten am MBus bereits bekannt sind und eine Nachricht an eines der bekannten Modul gesendet werden soll.

Nachrichten-Syntax

Alle Nachrichten des Message Bus sollten UTF-8 kodiert sein, was bedeutet, daß ASCII-Nachrichten ohne weitere Umwandlung übertragen werden können.

Kommando-Syntax

An den Kopf der Nachricht schließen die Kommandos an — pro Zeile eines. Für die Kommandos gilt folgende Syntax:

```
kommando ( parameter parameter ... )
```

Ein Kommando folgt einer bestimmten Konvention der Namensgebung: die Benennung der Kommandos soll eine Hierarchie bilden, die die Zusammengehörigkeit verwandter Kommandos ausdrückt. Als Trennzeichen wird der „.“ (Punkt) verwendet.

Wie schon erwähnt, gibt es einige Kommandos, die von allen Modulen am Message Bus verstanden werden (sollten), und solche, die modulspezifisch sind. In dieser Arbeit werden für alle entwickelten Module die spezifischen Kommandos im Anhang vorgestellt.

Eine Aufführung der bisher definierten allgemeinen Kommandos kann dem Internet Draft „*A Message Bus for Conferencing Systems*“ [25] entnommen werden.

Verwendung des MBus

Haupteinsatzgebiet des Message Bus sind meist Konferenz-Endsysteme, was sich am Schwerpunkt der bisher definierten Kommandos zur Anrufsteuerung und Mediensteuerung erkennen läßt. Der Message Bus ist aber nicht darauf beschränkt, da er durch die Definition eigener Kommandos zu einem flexiblen Kommunikationsmedium geworden ist.

Die einzelnen Anwendungsprozesse, die über den Message Bus miteinander kommunizieren, bilden gemeinsam die Anwendung. Jeder dieser Prozesse erbringt eine eigene Funktionalität und kommuniziert über wohldefinierte Schnittstellen mit anderen Komponenten - vergleichbar mit Modulen einer Anwendung.

Das typische Bild eines Endsystems sieht z.B. je ein Modul zur Wiedergabe/Aufnahme von Audio- und Videodaten, ein Modul zur Anruf- oder Konferenzsteuerung und ein Steuermodul vor.

Als Mediatools haben sich in der AG Rechnernetze dabei *vat* (*Visual Audio Tool*) [20] bzw. *rat* (*Robust-Audio Tool*) [38] für Audiodaten und *vic* (*Video Conferencing Tool*) [21] für Videodaten bewährt.

Ein oder mehrere Protokollmodule verbinden die Anwendung mit anderen Konferenzteilnehmern, indem sie z.B. Implementierungen der bisher in der Internet-Umgebung verwendeten Protokolle SDP *Session Description Protocol*, SIP (*Session Initiation Protocol*), SAP (*Session Announcement Protocol*) und SCCP (*Simple Conference Control*) *Protocol* beinhalten.

Das Steuermodul sorgt dafür, daß aus den einzelnen Modulen eine Anwendung wird. Es konfiguriert die Module und koordiniert den Datenfluß, so daß nur das Kontrollmodul wissen muß, welche anderen Module eigentlich noch am Message Bus teilnehmen und was für eine Funktionalität die Gesamtheit der Module überhaupt zu erbringen hat. Dies erlaubt es, Module zu schreiben, die für verschiedene Zwecke ohne Umprogrammierung eingesetzt werden können, indem für eine andere Gesamtfunktionalität einfach das Kontrollmodul ausgetauscht wird.

Voting

Gelegentlich kommt es vor, daß ein Modul andere Module befragen muß und in Abhängigkeit von den erhaltenen Antworten reagiert. Ein Beispiel hierfür wäre z.B. ein Steuermodul, das in Erfahrung bringen muß, wie es mit einem eingehenden Anruf verfahren soll. In einem System kann es evtl. mehrere Module geben, die lediglich dafür da sind, derartige Fragen zu beantworten — solche Module werden auch *Policy-Module* genannt.

Eines der allgemeinen Kommandos des Message Bus — `mbus.poll` — erlaubt es, eine Frage mit einer Menge von vorgegebenen Antworten und eventuellen Zusatzinformationen zu definieren. Alle Empfänger, die mit dieser Frage etwas anfangen können, suchen nach den ihnen zugrundeliegenden Regeln unter den vorgegebenen Antworten eine aus und teilen sie dem anfragenden Modul mit dem `mbus.vote`-Kommando mit. Dieses bestimmt anhand geeigneten, situationsbedingten Regeln, die nicht Inhalt des Standards sind, welche der Antworten die wichtigste ist und verfährt dementsprechend. Dieser Vorgang wird *Voting* genannt.

Sicherheit

Durch die Verwendung von lokalem Multicast ist der Message Bus im Prinzip ein recht offenes System. Wenn auf einem Rechner bereits eine Anwendung mit einem Message Bus läuft, können weitere Module gestartet werden, die ebenfalls auf dem Message Bus lauschen und senden und somit womöglich Daten ausspähen oder ggf. das System durcheinanderbringen.

Um Anwendungen vor fremden Einflüssen auf dem Message Bus zu schützen bzw. die Geheimhaltung der Daten zu gewährleisten, werden die Daten auf dem Message Bus in der Regel verschlüsselt. Authentifizierung der Daten wird dadurch erreicht, das alle Module, die vom gleichen Anwender gestartet werden, einen gemeinsamen Schlüssel haben und durch die Kombination von Schlüssel und dem Nachrichteninhalte einen Message Digest bilden, der sowohl Integrität, als auch den Absender der Nachricht verifiziert — letzteres zumindest insofern, als daß sichergestellt wird, daß es sich um irgendeines der zugehörigen MBus-Module handelt. Privatheit durch die Möglichkeit des Einsatzes verschiedener Verschlüsselungsverfahren.

Repräsentation der MBus-Kommandos in diesem Dokument

Für die Darstellung der MBus-Kommandos in diesem Dokument werden drei verschiedene Arten verwendet.

Kurzübersichten In Kapitel 4 werden an einigen Stellen bestimmte `mbus.poll`-Kommandos vorgestellt. Dafür wird folgende Repräsentation gewählt:

Empfänger	<code>mbus.poll</code>	Optionen	Informationen
Policy	<code>isLocalZone</code>	Yes No	IP-Adresse

Die erste Spalte enthält die Empfänger des Kommandos - in der Regel immer nur die Policy-Module. In der zweiten Spalte wird das Schlüsselwort definiert, über das das Poll-Kommando von anderen unterschieden werden kann. Die möglichen Antworten stehen in der dritten Spalte und die vierte listet weitere Informationen auf, die mit dem Kommando versendet werden.

Empfänger	Kommando	Informationen
alle	<code>register</code>	CS-Adresse, RAS-Adresse, Aliasnamen, Endpunkttyp

Tabelle 2.1: Kommandos zur Anmeldung von Endpunkten

Für alle anderen Kommandos wurde eine Darstellung wie in Tabelle 2.1 in Tabellenform gewählt.

Ausführliche Darstellung Im Anhang B werden alle unterstützten MBus-Kommandos aufgelistet. Dabei wird folgende Form der Darstellung gewählt.²

kommandoname

kommandoname - Kurzbeschreibung		
kommando param1 param2 ...		
param1	Typ	Bedeutung
param2	Typ	Bedeutung

²Die Randnotiz ist beabsichtigt und soll im Anhang das Auffinden der Kommandos erleichtern.

2.3 Zusammenfassung

Die vorgestellten Standards berühren diese Arbeit unterschiedlich stark. An wichtigsten sind die ITU-Standards H.323 und H.225.0. Solange der Gatekeeper kein *Gatekeeper-routed*-Anrufmodell unterstützt, bedarf es sogar nur des RAS-Teiles von H.225.0.

Die IETF-Entwürfe finden — abgesehen vom *Message Bus* — keine direkte Anwendung in dieser Arbeit, jedoch ist das System so ausgelegt, daß lediglich ein neues MBus-Modul gestartet werden muß, daß einen neuen Standard unterstützt, um sich ggf. dessen Funktionalität zu bedienen.

Der *Message Bus* ist eine Grundvoraussetzung für das System, da er die einzelnen Programmkomponenten miteinander koppelt.

Die in den folgenden Kapiteln erwähnten Namen von Nachrichten (wie z.B. GRQ, ARQ, etc.) stammen aus H.225.0. Ihre Verwendung wird in H.323 erläutert. Die MBus-Kommandos sind, bis auf die zwei Ausnahmen `mbus.hello` und `mbus.bye`, speziell für diese Diplomarbeit erdacht.

Kapitel 3

Funktionalität des Gatekeepers

Ein Gatekeeper dient als Verwaltungs- und Kontrollinstanz für seine Domäne. Prinzipiell können zwei Endpunkte auch direkt miteinander kommunizieren, ohne daß es eines Gatekeepers bedarf. Dieses Verfahren ist aber nur bedingt praktikabel, da dies Kenntnis über die aktuelle Adresse des anderen Teilnehmers und evtl. nötige Gateways voraussetzt.

Um ohne solche Kenntnisse in den Endpunkten arbeiten zu können, bedarf es des Gatekeepers. Dieser bietet mindestens eine Adreßauflösung von verschiedenen Adreßarten auf IP-Adressen, Zugangskontrolle, Ressourcenverwaltung und Call-Routing an. Optional kann ein Gatekeeper weitere Dienste anbieten, wie z.B. Verzeichnisdienste, Anrufprotokollierung, Kostenberechnung und Mehrwertdienste.

In dieser Diplomarbeit wurden die Grundfunktionalitäten implementiert, die in den nun folgenden Abschnitten erläutert werden sollen.

3.1 Zugangskontrolle

Der Gatekeeper autorisiert Zugriffe auf das LAN. Hierfür werden die ARQ-, ACF- und ARJ-Nachrichten von H.225.0 verwendet. Von H.323 offengelassen werden allerdings die Kriterien, nach denen den Endpunkten der Zugang zum Netz gewährt bzw. verwehrt wird.

Die Gatekeeper-Implementierung dieser Arbeit wird die Entscheidung, ob einem Endpunkt ein Anruf erlaubt wird oder nicht von zwei wesentlichen Faktoren abhängig machen: Bandbreite und Zugangsberechtigung.

Wie soeben im Abschnitt zur Ressourcenverwaltung erwähnt, hat der Gatekeeper den Überblick, wieviel Bandbreite noch frei bzw. in Benutzung ist. Ist nicht mehr genügend Bandbreite vorhanden, so kann der Gatekeeper den Zugang verweigern oder evtl. mit geringerer Bandbreite erlauben.

Neben der Bandbreite hängt die Zugangsberechtigung zum LAN aber auch noch von weiteren Faktoren ab. Denkbar wäre z.B. eine Art Kontostand eines

Nutzers (z.B. eines Kontos mit zuvor eingekauften Einheiten), die angerufene Nummer (so könnte es z.B. immer anrufbare Notfallnummern oder generell gesperrte Nummern geben), die Uhrzeit oder die Herkunft des Anrufes (z.B. keine Telefongespräche aus einem Rechnerpool zur Stoßzeit, damit niemand gestört wird).

Die Entscheidung darüber, ob ein Nutzer Zugang zum IP-Telefonsystem erhält, treffen in der hier vorgestellten Gatekeeper-Implementierung sogenannte Policy-Module. Diese austausch- und erweiterbaren Programmkomponenten werden „befragt“, wie mit einem Anrufwunsch umgegangen werden soll. Folgende Funktionalität wurde dabei vorgesehen:

- **Zulassung in Abhängigkeit von der Herkunft-IP-Adresse¹**
Der Gatekeeper wird nur solchen Endpunkten das Telefonieren gestatten, die sich zuvor bei ihm registriert haben.
- **Zulassung in Abhängigkeit vom Kontostand**
Jeder Nutzer hat einen Kontostand. Dieser kann z.B. wie eine Telefonkarte genutzt werden, d.h. ein Nutzer erhält ein Guthaben, das er abtelefonieren kann und auch durch Bezahlen wieder aufstocken kann. In Kombination mit einer Gruppenzugehörigkeit kann bestimmten Benutzergruppen das Telefonieren auch mit monatlicher Rechnung ermöglicht werden.
- **Zulassung in Abhängigkeit von der Bandbreite**
Sobald ein Endpunkt Zugang zum Netz wünscht, teilt er auch mit, wieviel Bandbreite er verwenden möchte. Dies teilt er dem Gatekeeper mit. Anhand der freien Ressourcen muß der Gatekeeper dann entscheiden, ob er diesen Wunsch so akzeptiert, ihn abweist oder reduziert zuläßt. Der Gatekeeper wird den Zugang gewähren, sofern genug Bandbreite zur Verfügung steht. Bestimmten Benutzern/Benutzergruppen kann eine maximale Bandbreite zugewiesen werden, auf die Anfragen ggf. zurückgestutzt werden.
- **Zulassung in Abhängigkeit von der Uhrzeit**
Der Gatekeeper kann so konfiguriert werden, daß er bestimmten Transportadressen zu vorgegebenen Stunden am Tag den Zugang zum Netz verwehrt.

3.2 Adreßumsetzung

Der H.323-Standard sieht mehrere Arten von Adressen vor, mit denen ein Teilnehmer identifiziert werden kann. Dies können z.B. Telefonnummern nach E.164² sein, aber auch eMail-Adressen, URLs oder H.323-IDs. Bei letzterem handelt es sich um einen 256-Zeichen langen Unicode-String, der in der Regel ähnlich der eMail-Adressen aufgebaut ist. Der Oberbegriff für die verschiedenen Adreßtypen ist *Aliasadresse*.

¹Es gibt explizit keine Zulassung in Abhängigkeit der Ziel-Adresse, da dies entschieden wird, wenn der Angerufene seinerseits eine Zulassung einholt.

²E.164-Adressen bestehen aus den Zeichen „0123456789#*“, d.h. sie lassen sich fast alle mit einem herkömmlichen Tastenwahlfeld von Telefonen erzeugen.

Der in dieser Arbeit entwickelte Gatekeeper beschränkt sich auf die Unterstützung von H.323-IDs und E.164-Adressen, da diese von den meisten Systemen unterstützt werden.

Ein Gatekeeper muß die Umsetzung von Aliasadressen auf tatsächliche Transportadressen implementieren. Dies ist nötig, da netzübergreifende Adressierungen ermöglicht werden müssen und eine H.323-Adresse nicht unbedingt etwas darüber aussagt, ob es sich z.B. um einen Rechner mit einer IP-Adresse oder ein ISDN-Telefon handelt.

Es lassen sich zwei Stufen der Adreßumsetzung, auch User Location genannt, unterscheiden. Zum einen kann ein Gatekeeper selbst über die Information verfügen, unter welcher Transportadresse ein Teilnehmer momentan erreichbar ist. Dies ist in der Regel der Fall, wenn der Endpunkt des Teilnehmers in der Zone des Gatekeepers registriert ist. Zum anderen kann es sein, daß der Gatekeeper die Informationen über die aktuelle Transportadresse des Teilnehmers von anderen Gatekeepern erfragen muß.

Wie im letzten Fall zu verfahren ist, ist bisher im Rahmen von H.323 nur unzureichend geklärt. Zwar kann ein Gatekeeper wie ein normaler Endpunkt auch einen anderen Gatekeeper um eine Adreßauflösung bitten, aber woher ein Gatekeeper von anderen Gatekeepern erfährt und woher er weiß, welchen Gatekeeper er am besten fragt, ist nicht definiert. Mit dem weiter oben vorgestellten Protokoll PGRP (siehe 2.2.4) existiert zwar ein Entwurf eines Verfahrens, jedoch ist dieser Entwurf noch nicht soweit ausgearbeitet, daß er sich implementieren ließe.

Aus diesem Grund wird diese Diplomarbeit Adreßumsetzung nur für zonenlokale Endpunkte unterstützen. Die verwendeten Mechanismen sollten jedoch eine einfache Erweiterung um Adreßumsetzung mittels eines Inter-Gatekeeper-Protokolls erlauben.

Der Gatekeeper wird Aliasadressen, die als H.323-ID vorliegen, in die Transportadressen des Endpunkts im drunterliegenden IP-Netz umsetzen. Etwas komplexer verhält es sich mit E.164-Adressen: Ein Endpunkt kann sich ebenfalls mit einer solchen Adresse anmelden und somit dem Gatekeeper ermöglichen, die E.164-Adresse direkt auf eine Transportadresse abzubilden.

Wenn die E.164-Adressen jedoch einer Menge von normalen Telefonen, z.B. ISDN-Telefonen zugeordnet sind, können sie nur über ein Gateway erreicht werden. Dieses Gateway muß nun dem Gatekeeper bekanntmachen, welche IP-Adressen über das Gateway erreicht werden können. Eine für übersichtliche Szenarios geeignete Methode wäre, daß das Gateway sich mit allen denkbaren Adressen beim Gatekeeper anmeldet. Für ein Gateway in das gesamte restliche Telefonnetz der Welt ist dies natürlich denkbar ungeeignet.

In jedem Fall liefert der Gatekeeper bei Auflösung einer E.164-Adresse hinter einem Gateway die Transportadresse des Gateways zurück.

Der in dieser Arbeit implimentierte Gatekeeper unterstützt, wie bereits erwähnt, lediglich E.164-Adressen und H.323-IDs. Andere Adreßtypen, bzw. nicht auflösbare Adressen werden auf den MBus durchgereicht, so daß Erweiterungsmodule diese Adresse auflösen können.

3.2.1 Adressen, die Personen kennzeichnen

Betrachtet man daß Verhältnis von H.323-Adressen, E.164-Adressen und IP-Adressen, so werden zwei nennenswert unterschiedliche Fälle deutlich.

a) Pro Person eine IP-Adresse

In der Regel sollte ein Mensch innerhalb der Zone eines Gatekeepers durch genau eine H.323-Adresse identifiziert werden können (Abb. 3.1). Wenn eine bestimmte Person mittels der ihr zugeordneten Adresse angerufen werden soll, so erwartet man auch, daß im Prinzip nur diese Person die Anrufe entgegennimmt. Voraussetzung für eine solche Situation ist, daß jeder Benutzer mit einem eigenen IP-Telefon bzw. einem IP-Telefonie-fähigen Computer sitzt, d.h. jedem Menschen eine IP-Adresse zugeordnet werden kann.

<u>H.323-ID</u>	<u>IP</u>	<u>Person</u>
prelle@..	134.102.218.62	Stefan
npollem@..	134.102.218.57	Niels
kollmann@..	134.102.219.29	Ralf

Abbildung 3.1: Abbildung von H.323-Adressen auf Benutzer

Diese Situation ist ideal für den Gatekeeper, um zu ermitteln, ob eine Person erreichbar ist, oder nicht. Jeder Benutzer hat einen eigenen Endpunkt und kann daher einstellen, ob er erreichbar ist, oder nicht.

b) Mehrere Personen hinter einer IP-Adresse

Leider wird eine Situation wie in a) beschrieben nicht die Regel sein. Wie bisher wird — zumindest in der Anfangszeit — es oft vorkommen, daß mehrere Personen in einem Raum sitzen und sich ein Telefon teilen müssen. Dieses Telefon ist über genau eine IP-Adresse erreichbar.(Abb. 3.2).

Eventuell hat das Telefon noch eine Adresse, damit man explizit nur das Tele-

<u>H.323-ID</u>	<u>IP</u>	<u>Person</u>
prelle@..	134.102.228.200	Stefan
npollem@..		Niels
kollmann@..		Ralf

Abbildung 3.2: Mehrere Benutzer teilen sich einen Endpunkt (1)

fon, d.h. irgendeine beliebige Person aus dem Raum, erreichen kann. Eine solche Adresse kann sowohl eine herkömmliche Telefonnummer, wie auch eine H.323-ID sein (Abb. 3.3).

Eine Konfiguration, bei der mehrere Ziele über eine IP-Adresse zu erreichen sind, birgt z.B. das Problem in sich, daß der Gatekeeper nicht sicher sagen kann, ob

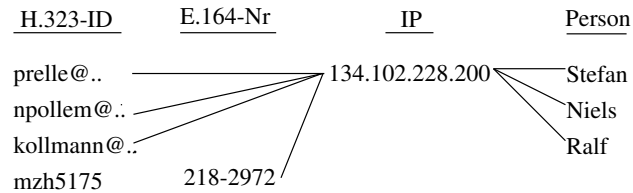


Abbildung 3.3: Mehrere Benutzer teilen sich einen Endpunkt (2)

die Person wirklich gerade erreichbar ist. Damit ist nicht gemeint, ob sich die Person gerade im Raum aufhält, sondern ob sie längerfristig abwesend ist. Im Fall a) war dies gegeben, da jeder Teilnehmer selbst bestimmen kann, ob sein Endpunkt beim Gatekeeper angemeldet sein soll, oder nicht.

Um nun dem Telefon im Raum mitzuteilen, ob eine bestimmte Person aus dem Raum erreichbar ist oder nicht, müßte sich diese jedesmal am gemeinsamen Telefon an- und abmelden. Das Telefon würde daraus eine Nachricht an den Gatekeeper registrieren und diesen über Zu- und Abgang der Person informieren. Dies erzwingt allerdings auch, daß ein Nutzer sich nur an dem Telefon abmelden kann, an dem er sich angemeldet hat.

Eine bessere Lösung ist es, eine Indirektionsstufe hinzuzufügen, in dem man es verbietet, daß mehrere Nutzer einer IP-Adresse direkt zugeordnet werden. Im oben aufgeführten Beispiel mit einem Telefon für alle Personen in einem Raum würde dies bedeuten, daß es Adressen gibt, die explizit nur den Raum, bzw. das Telefon darin identifizieren. Die Personen im Raum definieren — und das ist die Indirektionsstufe — eine Rufumleitung von ihrer eigenen Adresse auf das Telefon des Raumes.

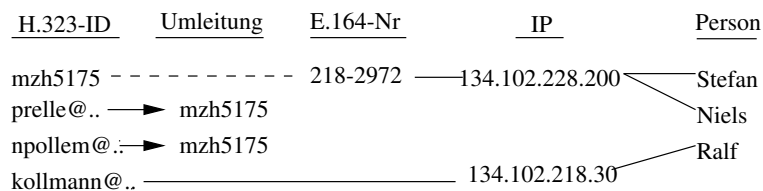


Abbildung 3.4: Mehrere Benutzer mit Rufumleitung

Durch Rufumleitung ist ein Benutzer wesentlich flexibler in der Konfiguration dessen, was mit eingehenden Anrufen geschehen soll. So könnte er z.B. einstellen, daß alle Anrufe auf ein Telefon im gleichen Raum weitergeleitet werden sollen, weil er z.B. lieber damit telefoniert, statt am Computer. Es ist aber auch denkbar, daß er sich die Anrufe nach Hause durchstellen läßt, bzw. auf seinen Anrufbeantworter, da er nicht mehr im Büro erreichbar ist. (Abb. 3.4).

Die Realisierung der nutzerdefinierten Rufumleitung geschieht naheliegenderweise durch ein Anrufbearbeitungs-Modul im Gatekeeper, welches z.B. CPL-Skripte (siehe 2.2.5) ausführt. Ein solches Modul ist jedoch nicht Bestandteil

des im Rahmen dieser Arbeit entstandenen Gatekeepers.

3.2.2 Adressen, die Funktionen kennzeichnen

Bis hierhin wurde nur von den Problemen bei der Umsetzung von Adressen, die Personen oder Räume bezeichnen, auf IP-Adressen besprochen. Es gibt jedoch Fälle, in denen man nicht eine bestimmte Person erreichen möchte, sondern eine Person, die eine bestimmte Funktion, wie z.B. Geschäftsführer, Sekretärin oder Kundensupport, übernimmt. In diesem Fall ist es sinnvoller, Adressen für Funktionen zu haben, da z.B. eine Sekretärin im Laufe der Zeit mal wechseln kann, bzw. mehrere Leute den Kundensupport einer Firma übernehmen.

Adressen, die Funktionen kennzeichnen, können einer oder mehreren Personen zugeordnet sein. Genaugenommen ist eine solche Adresse nur ein Satz von möglichen Rufumleitungen. Es muß also bestimmt werden, zu welcher Adresse der Anruf umgeleitet werden soll. Dazu sind mehrere Verfahren denkbar:

1. *Entscheiden auf Empfängerseite (Broadcast)*
Der Anruf wird an alle Adressen verteilt - der endgültige Empfänger ist der, der den Anruf zuerst entgegennimmt.
2. *Der Reihe nach verteilen*
Der Gatekeeper geht die Liste der möglichen Umleitungen der Reihe nach durch. Für jeden neuen Anruf startet er hinter der Adresse, die den letzten Anruf bekommen hat.
3. *Der Priorität nach verteilen*
Der Gatekeeper führt eine Liste mit Umleitungsadressen, die er versucht, ihrer Priorität nach zu erreichen.

Der in dieser Arbeit entstandene Gatekeeper verwendet die letzte Option, d.h. er verwendet eine priorisierte Liste mit Adressen.

Als nächstes stellt sich die Frage, wo die Konfiguration dieser Rufumleitung vorgenommen werden soll. Folgende Modelle sind möglich:

1. *Konfiguration beim Gatekeeper:*
Der Gatekeeper verwaltet eine Liste, die eine Abbildung von Funktionsadressen auf H.323- oder E.164-Adressen erlaubt. Wird die Funktionsadresse angerufen, wird versucht, eine der zugeordneten Adressen zu erreichen. Wenn keiner der aufgelisteten Adressen erreicht werden kann, wird der Anruf abgelehnt.
Die Liste der Adressen, die einer Funktionsadresse zugeordnet werden, wird vom Systemverwalter konfiguriert — die einzelnen Nutzer haben keinen Einfluß darauf.
2. *Konfiguration durch die Clients:*
Ein Nutzer gibt beim Starten seiner Telefonieanwendung an, daß er in einer bestimmten Funktion aktiv werden möchte, oder auch nicht. Die Anwendung registriert sich dann mit allen Adressen beim Gatekeeper, so daß gewährleistet ist, daß nur dann die Funktionsadresse angerufen werden kann, wenn eine zugehörige Person sich angemeldet hat.

Bei genauerer Betrachtung zeigt sich allerdings, daß auch hier eine zentral gehaltene Konfiguration unabdingbar ist. Irgendwo muß vermerkt werden, wer sich alles einer bestimmten Funktionsadresse zuordnen darf, damit kein Mißbrauch betrieben werden kann. Eine solche zentrale Konfiguration erfordert wieder einen Systemverwalter.

Für den implementierten Gatekeeper wurde das erste Modell gewählt. Im Administrationstool wurde die Möglichkeit geschaffen, Funktionsadressen und deren Abbildung zu verwalten (siehe 5.4.5), wobei die Position in der Liste die Priorität widerspiegelt.

Bei diesem Modell ist es gegenwärtig allerdings nicht möglich, daß eine Person selbst kurzfristig bestimmen kann, ob sie diese Funktion gerade wahrnehmen möchte oder nicht. Dies ist nicht zwingend ein Nachteil, da es schließlich von der Politik der Einsatzumgebung abhängt, ob die Funktionsadressen etwas kennzeichnen, dem man kurzfristig beitreten und sich ebensoschnell wieder abmelden kann. Für den anvisierten Einsatz in einem universitären Umfeld ist die gewählte Lösung ausreichend, da dort die Aufgabenverteilung in der Regel eher langfristig angelegt ist.

3.2.3 Ablauf der Adreßumsetzung

Muß das H323-Modul des Gatekeepers im Rahmen einer *Admission-Request* (ARQ) oder einer *Location-Request* (LRQ) eine Adresse auflösen, so wird zunächst innerhalb des H.323-Moduls in den Listen der registrierten Endpunkte und Teilnehmer gesucht.

Ist die Adresse hier nicht bekannt, so fragt das H323-Modul das Datenbank-Modul, ob die gesuchte Adresse evtl. als Funktionsadresse bekannt ist. Ist dies der Fall, so wird eine Liste von Adressen unterschiedlicher Art (H.323-Adressen, E.164-Adressen und evtl. auch Funktionsadressen) zurückgegeben, die dieser Funktionsadresse zugeordnet sind. Der Vorgang der Adreßumsetzung (vergl. Abb. 3.5 arbeitet rekursiv, d.h. beginnt nun erneut für die einzelnen Elemente dieser Adreßliste und dauert solange, bis entweder eine Aliasadresse in eine IP-Adresse umgesetzt werden konnte oder das Ende der Liste erreicht wurde.

Für den Fall, daß die Adresse nicht zu einem registrierten Benutzer gehört, es sich nicht um eine Funktionsadresse handelt bzw. die Personen, die die durch die Funktionsadresse gekennzeichnete Funktion wahrnehmen, ebenfalls nicht registriert sind, leitet das H.323-Modul die Anfrage zur Adreßauflösung auf den MBus weiter. Falls ein externes Location-Modul existiert, kann dies versuchen, die Adresse aufzulösen, in dem es z.B. mit anderen Gatekeepern redet. Erfolgt auf die Anfrage auf dem MBus innerhalb eines allgemeinen Timeouts³ keine Antwort, bzw. eine negative Antwort, so wird dem Fragesteller bescheinigt, daß die gewünschte Adresse unbekannt ist.

Eine interessante Frage ist, wie es um das Antwortzeitverhalten des Gatekeepers bei der Adreßauflösung bestellt ist. Der H.323-Standard schreibt vor, daß ein Endpunkt innerhalb von fünf Sekunden eine Antwort vom Gatekeeper haben sollte. Dies ist eher unproblematisch, sofern die Adressen registrierter Endpunkte aufgelöst werden sollen. Müssen aber systemexterne Komponenten, wie z.B.

³gegenwärtig 300ms

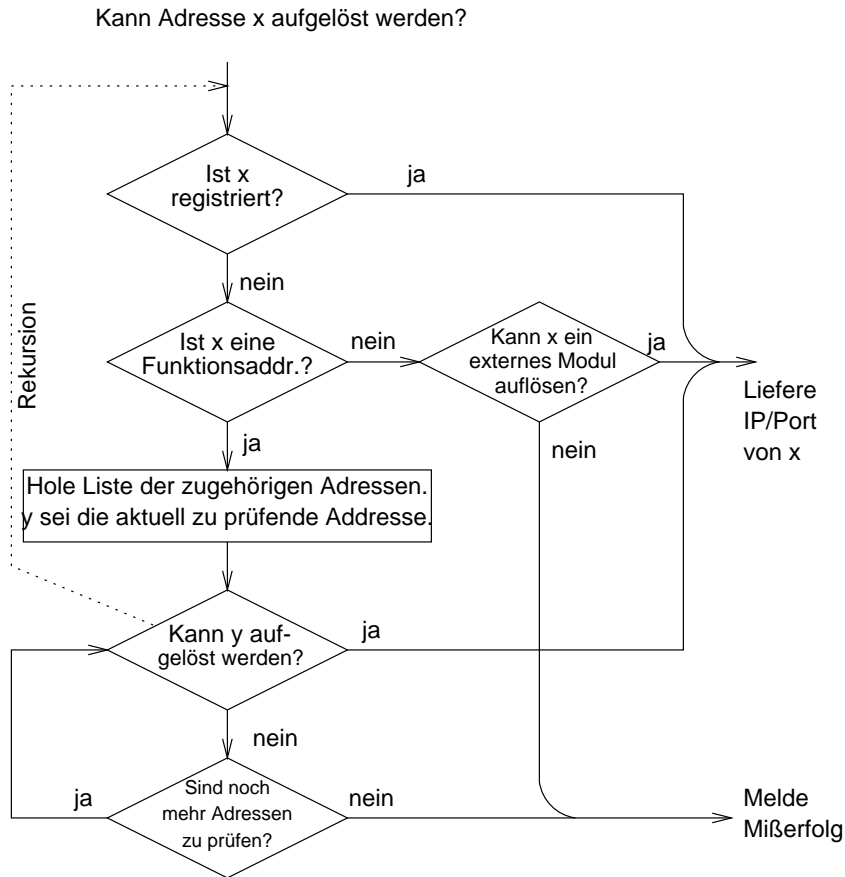


Abbildung 3.5: Ablauf der Adreßumsetzung

weitere Gatekeeper befragt werden, so gilt auch hier wieder ein Timeout von 5 Sekunden — womit es wahrscheinlich wird, daß der erste Timeout nicht eingehalten wird.

Aus diesem Grund sollte bei Beginn der Befragung systemexterner Komponenten dem Fragesteller eine *RequestInProgress* (RIP)-Nachricht gesendet werden, in der der Gatekeeper angibt, wieviel Zeit er maximal noch braucht, um die Anfrage zu beantworten. Der von mir implementierte Gatekeeper tut dies gegenwärtig nicht, da noch keine externen User-Location-Module existieren.

3.3 Call-Routing

Wenn ein Anruf die Zone des Gatekeepers verläßt, so muß der Weg bestimmt werden, den der Anruf nehmen muß, um sein Ziel zu erreichen. Sind beide Teilnehmer über eine IP-Adresse erreichbar, so besteht kein Bedarf, daß der Entscheidungen über Routing trifft, da hier das bekannte IP-Routing greift. Befindet sich der anzurufende Teilnehmer aber im Telefon-Netz, so stellt sich die Frage, wie und wo der Übergang in dieses Netz erfolgen soll.

Den Übergang zwischen zwei Netzen realisiert ein Gateway. Es sorgt für die Umsetzung der Signalisierungen der jeweiligen Netze. Abhängig vom verwendeten Anrufmodell teilt der Gatekeeper dem Endpunkt die Transport-Adresse des Gateways mit (*Direct*) oder er fungiert als Zwischenstation und ruft selbst das Gateway an (*Gatekeeper-routed*). Das Gateway, seinerseits ein H.323-Endpunkt, sorgt für die Umsetzung der Signalisierung zwischen H.225.0 und dem jeweils anderen Netz.

Da aber in anderen Netzen auch andere Tarife für Verbindungen gelten und Gateways in der Anzahl der gleichzeitig umsetzbaren Anrufe meist begrenzt sind, ist das Auffinden eines passenden Gateways nicht trivial. Soll beispielsweise von einem IP-Telefon in Deutschland ein Anruf zu einem normalen Telefonanschluß in Australien abgesetzt werden, stellt sich die Frage, welches Gateway man benutzt. Idealerweise wird natürlich kein Gateway in der Zone des eigenen Gatekeepers genommen, denn dann müßten die vollen Telefongebühren nach Australien bezahlt werden. Am besten ist es also, ein freies Gateway in Australien zu finden, das den australischen Anschluß am günstigsten erreicht. Evtl. ist aber das billigste Gateway nicht immer das beste, da es z.B. eine qualitativ schlechtere Verbindung bietet. Es gilt also zudem noch, einen Trade-off zwischen Kosten und Qualität zu finden.

Wie ein Gatekeeper erfährt, welches Gateway das geeignetste ist und ob dieses Gateway im Moment noch freie Kapazitäten hat, ist erst seit kurzer Zeit in H.323 im Annex G [17] definiert. Die alternativen Entwürfe zu einem *Gateway Location Protocol* (GLP) [32], [33] nehmen sich zwar ebenfalls dieses Problems an, jedoch werden sie, ebensowenig wie H.323 Annex G, im Rahmen dieser Diplomarbeit implementiert.

Die Gateway-Unterstützung des implementierten Gatekeepers ist — nicht zuletzt wegen nicht vorhandener Test-Gateways während der Diplomarbeit — bestenfalls ansatzweise vorhanden, da theoretisch die Möglichkeit besteht, daß ein Gateway sich mit allen unterstützten Adressen explizit beim Gatekeeper anmeldet.

Um trotzdem die Möglichkeit der nachträglichen Ergänzung um diese Funktionalität zu geben, wird die Funktionserweiterung durch weitere Module vorgesehen (siehe Kapitel 4)

3.4 Ressourcenverwaltung

Der Gatekeeper als Verwaltungsinstanz wird naheliegenderweise Kontrolle über die Systemressourcen behalten. Mindestens zwei Arten von Ressourcen müssen dabei bedacht werden: Bandbreite und ggf. freie Leitungen bei Gateways ins herkömmliche Telefonnetz.

Bandbreite ist in der Regel ein knappes Gut. Vor Beginn eines Anrufes fordern Endpunkte ihre gewünschte Bandbreite für die Übertragung an. Auch während der Verbindung können Endpunkte noch andere Bandbreiten anfordern oder aber vom Gatekeeper eine andere Bandbreite zugeteilt bekommen.

Damit die Endpunkte auch tatsächlich die Übertragungsrate bekommen, die

sie angefordert haben, muß der Gatekeeper einen Überblick darüber haben, ob die angeforderte Bandbreite überhaupt noch zur Verfügung steht, bzw. muß bei Bedarf noch Bandbreite freimachen.

An dieser Stelle ist es wichtig, zwischen der vom Gatekeeper zugeteilten und einer z.B. durch *Resource Reservation Protocol* (RSVP) garantierten Bandbreite zu differenzieren. Der Gatekeeper ist weder in der Lage zu kontrollieren, ob die Endpunkte die ihnen zugeteilte Bandbreite einhalten⁴, noch ob tatsächlich soviel Bandbreite zu Verfügung steht, wie er glaubt. Die Reglementierung durch den Gatekeeper funktioniert also nur, wenn sich alle Endpunkte an die Vorgaben des Gatekeepers halten.

Die Gatekeeper-Lösung dieser Arbeit wird in Bezug auf Bandbreite folgende Funktionalität aufweisen:

- Die verwendete Bandbreite einzelner Anrufe und des gesamten Systems wird ständig mitprotokolliert.
- Bandbreite wird nur bis zu einer einstellbaren Obergrenze verteilt.
- Ein einstellbarer Anteil der Bandbreite wird für eingehende Anrufe reserviert, um Erreichbarkeit zu gewährleisten.
- Es können nutzergruppenabhängige Obergrenzen für Bandbreite angegeben werden.
- Bei Bedarf werden Verbindungen mit hoher Bandbreite auf niedrigere Bandbreiten reduziert, um mehr Ressourcen zu erhalten. Dabei wird eine einstellbare untere Grenze nicht unterschritten, damit eine Mindestqualität der Übertragung gewährleistet wird.

Die andere zu verwaltende Ressource, die Kapazität der Gateways, ist etwas komplizierter zu handhaben. Wie aber schon beim Call-Routing erwähnt, ist Gateway Location momentan noch in der Entwurfsphase, weswegen überhaupt nur Gateways der eigenen Zone behandelt werden. Der im Rahmen dieser Arbeit implementierte Gatekeeper wird diese Ressource also nicht selbst verwalten, sondern dem Endpunkt einfach die Adresse des Gateways mitteilen. Der Endpunkt wird seinen Anruf also über das Gateway leiten, das in dem Moment entscheidet, ob es noch Ressourcen frei hat und ggf. den Anruf ablehnt.

⁴Einige H.323-Endpunkte, wie z.B. Microsoft Netmeeting kümmern sich daher überhaupt nicht um die ihnen zugeteilte Bandbreite.

Kapitel 4

Die Architektur des Gatekeepers

Wie bereits mehrfach erwähnt, ist es das Ziel, das System so zu gestalten, daß es leicht erweiterbar ist. Aus diesem Grund wird der Gatekeeper aus einer Anzahl von über den MBus gekoppelten Modulen gebildet, die über eine fest definierte Menge von Kommandos miteinander kommunizieren. Änderungen oder Erweiterungen der Funktionalität des Gatekeepers können somit durch das Austauschen vorhandener oder Hinzufügen neuer Module erfolgen.

4.1 Die Module

Im Rahmen dieser Arbeit sind vier Module vorgesehen, die im folgenden erläutert werden sollen (vgl. Abb. 4.1): 1) Ein Modul, das eine H.323-Protokollimplementierung und das Steuermodul enthält, 2) ein Datenbank-Modul, 3) mindestens ein Policy-Modul, welches für Entscheidungen zur Ressourcenverwaltung und Zugangsregelung herangezogen wird und 4) ein API / GUI, das die Steuerung des Gatekeepers ermöglicht.

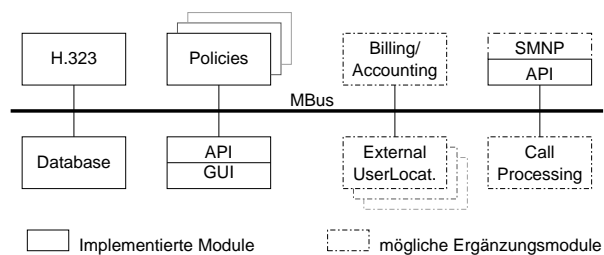


Abbildung 4.1: Komponenten am MBus

In Abschnitt 4.2 wird darauf eingegangen, wie weitere Funktionalität sinnvoll hinzugefügt werden könnte.

4.1.1 Das H323-Modul

Das H.323-Modul (siehe auch 5.3) ist zuständig für die Bearbeitung von eingehenden H.225.0 RAS-Nachrichten, die Verwaltung der registrierten Endpunkte und der UserLocation für die Zone. Durch diese Kombination kann das Modul die meisten Informationen, die es zur Beantwortung von Nachrichten auf dem RAS-Kanal benötigt, intern abfragen und braucht dafür keine anderen Module über den MBus befragen.

Damit die ohnehin schon kritische Antwortzeit auf RAS-Nachrichten nicht unnötig verlängert wird, greift das Modul nur in solchen Fällen auf den MBus zu, in denen es

- a) Daten nicht selbst ermitteln kann (z.B. bei Auflösung externer Adressen),
- b) eine Entscheidung der Policy-Module einholen muß oder
- c) auf Anfragen vom MBus antworten muß.

Alle anderen Fälle werden intern behandelt.

4.1.2 Die Policy-Module

Ein Policy-Modul hat die Aufgabe, darüber zu entscheiden, wie in Situationen verfahren werden soll, in denen die Antwort nicht allein durch das Protokoll vorgegeben ist. Eine solche Situation könnte z.B. sein, daß ein Endpunkt signalisiert, daß er einen Anruf zu plazieren wünscht. Die Entscheidung wird außerhalb des H323-Moduls — nämlich in den Policy-Modulen — getroffen und vom H323-Modul in eine entsprechende Antwort auf dem RAS-Kanal umgesetzt.

Im Gatekeeper können mehrere Policy-Module aktiv sein — hieraus resultiert die Anpaßbarkeit und Erweiterbarkeit des Gatekeepers. Es ist dabei explizit vorgesehen, daß sich mehrere Policy-Module zu einer Entscheidung äußern und die Entscheidungen durchaus voneinander abweichen können. Mittels eines Voting-Mechanismus werden die Antworten dem Fragesteller — in der Regel dem H.323-Modul - übermittelt und von diesem interpretiert.

Die anfallenden Entscheidungen, die die Policy-Module zu treffen haben, lassen sich in drei Kategorien unterteilen: Entscheidungen der Zugangsregelung, der Ressourcen-Verwaltung und der Anrufbearbeitung. Für zwei dieser Kategorien, Zugangsregelung und Ressourcen-Verwaltung, bietet die hier vorgestellte Gatekeeper-Implementierung bereits jeweils ein Policy-Modul. Ein Policy-Modul zur Anrufbearbeitung könnte z.B. durch ein CPL-Modul abgedeckt werden, ist jedoch hier nicht implementiert.

Zugangsregelung

Wie schon im Abschnitt 3.1 erwähnt, beinhaltet die Zugangsregelung Entscheidungen darüber, ob ein Endpunkt senden darf und ob er überhaupt ein Teil jenes Netzes ist, das vom Gatekeeper kontrolliert wird. Betroffen hiervon sind demnach am ehesten die Endpunkte des lokalen Netzes.

Die Entscheidungen können auf folgenden Faktoren basieren:

- **IP-Adresse**
Der Gatekeeper ist nur zuständig für das lokale Netz und wird daher nur Endpunkten aus diesem Netz Zugang zu seinen Diensten gewähren.
- **Kontostand des Anrufes**
Es mag sein, daß ein gewünschter Anruf in einem Abschnitt gebührenpflichtig wird. Dies könnte generell oder nur bis zu einer Obergrenze vertelefonierter Einheiten erlaubt werden. Denkbar ist auch, daß die Nutzer sich ein Kontoguthaben kaufen können, daß sie dann nur abtelefonieren.
- **Gruppenzugehörigkeit**
Unter Umständen kann es sinnvoll sein, die Nutzer in Gruppen aufzuteilen, um somit erweiterte oder eingeschränkte Rechte für bestimmte Personengruppen zu realisieren. Auch Sperrungen mancher Nutzer wären hiermit realisierbar.
- **Uhrzeit**
Dieser Faktor macht wohl nur in Kombination mit den anderen Faktoren Sinn. So könnte man z.B. zu bestimmten Uhrzeiten das Telefonieren von bestimmten Rechnern sperren, weil sie z.B. in einem Rechnerpool stehen und Lärm durch mehrere telefonierende Nutzer ausgeschlossen werden soll.

Der von mir implementierte Gatekeeper berücksichtigt die Zulassung in Abhängigkeit von der IP-Adresse und der Uhrzeit.

Ressourcen-Verwaltung

Eine wichtige Ressource in Bezug auf Telefonie-Anwendungen ist sicherlich die Bandbreite. Steht mehr Bandbreite zur Verfügung, gibt es wahrscheinlich bald darauf eine erhöhte Nutzung der Bandbreite durch die Anwendungen. Es ist sinnvoll, daß in einem Netz, in dem die Kapazitäten nicht nur durch IP-Telefonie sondern auch durch andere Dienste genutzt werden, sichergestellt wird, daß der Anteil der Telefonie an der Bandbreite nicht zu hoch wird.

Folgende Faktoren könnten hier die Entscheidung beeinflussen:

- **Gruppenzugehörigkeit**
Wie schon bei der Zugangsregelung will man Benutzer evtl. unterschiedlich behandeln. Privilegierte Gruppen könnten höhere Bandbreiten zugestanden bekommen, als andere.
- **Zeit**
Zu Stoßzeiten sollte die durch Telefonie verbrauchte Bandbreite geringer sein, als zu Zeiten, in denen wenig anderer Verkehr im Netz ist. In Verbindung mit der Gruppenzugehörigkeit der Nutzer ließe sich konfigurieren, welche Gruppe wann welche Bandbreite nutzen darf.

4.1.3 Das Datenbank-Modul

Im Datenbank-Modul werden all jene Daten verwaltet, die über eine Sitzung hinaus Gültigkeit haben. Zu diesen statischen Daten zählen neben Login, realen Namen und ggf. Adresse auch Accounting-Informationen und CPL-Skripte zur Anrufbearbeitung.

Das Datenbank-Modul besitzt im Prinzip keine eigene Intelligenz. Es dient lediglich zur Aufbewahrung und zum Zugriff auf Daten mit einer hohen Lebensdauer.

User-Daten

In der Regel wird dieses Modul von den Policy-Modulen verwendet. Es stellt sich nun die Frage, welche Daten in der Datenbank aufbewahrt werden sollen. Da theoretisch beliebig viele Policy-Module, darunter auch welche mit noch nicht bedachter Funktionalität, hinzugefügt werden können, ist es unwahrscheinlich, alle möglicherweise relevanten Daten in der Datenbank unterzubringen.

Es bleiben also zwei Alternativen:

1. *Vorabdefinition der zu speichernden Daten*

Es wird einmal definiert, welche Daten, wie in der Datenbank abgelegt werden. Dies sind sinnvollerweise die Daten, die schon vorhandene und bereits geplante Module benötigen. Spätere Änderungen sind nur in Form von Ergänzungen möglich, um die Funktion bestehender Module nicht zu stören.

2. *Grundgerüst plus Erweiterungsmechanismus*

Die Definition der Datensätze wird zunächst auf das absolut notwendigste beschränkt, d.h. das, was ein System ohne Policy-Module zum Laufen benötigt. Jedes Policy-Modul ergänzt dann einen Datensatz um die für den eigenen Betrieb nötigen Felder - losgelöst von den, evtl. identischen, Definitionen anderer Module.

Im Rahmen dieser Diplomarbeit wird der erste Weg gewählt. Die hiermit evtl. selbst auferlegten Einschränkungen werden bei einer gut gewählten Definition an benötigten Daten wahrscheinlich nicht ins Gewicht fallen.

Es wird unterschieden zwischen Daten für jeden einzelnen Anruf, Daten die für jeden Nutzer, anfallen und solchen, die das globale Verhalten des Gatekeepers beeinflussen. Die Anrufrdaten werden von dem von mir entwickelten System gegenwärtig nicht protokolliert - dies wäre die Aufgabe eines *Billing/Accounting-Moduls*, was nicht Teil dieser Arbeit ist. Die anderen beiden Kategorien sollen nun genauer betrachtet werden.

Daten pro Nutzer Als Daten für jeden einzelnen Nutzer werden definiert (vgl. 5.2.5):

- **Bezeichner / H.323-ID**
Dieses Feld identifiziert den Benutzer eindeutig, anhand seiner H.323-ID.
- **E.164-Adresse**
Optionale Angabe einer herkömmlichen Telefonnummer.
- **Realer Name**
- **Rechnungs-Adresse**
Für den Fall, daß man den Nutzern Telefonrechnungen zustellen möchte, dient dieses Feld der Aufnahme der Adresse.

- **eMail-Adresse**
Um ggf. den Benutzern Abrechnungen oder andere Informationen per eMail zukommen zu lassen, muß eine eMail-Adresse zu den statischen Nutzerdaten vermerkt sein.
- **Kontostand**
Hierbei handelt es sich entweder um einen Geldbetrag oder eine Anzahl von Telefoneinheiten. Das Datenbank-Modul interpretiert diese Zahl nicht selbst, insofern bleibt es den anderen verwendeten Modulen überlassen. Wahrscheinlich wird es ein Geldbetrag werden, da die Abrechnung mit Einheiten immer unter Berücksichtigung des aktuellen Tarifs und des Providers geschehen muß.
Gegenwärtig wird eine Interpretation favorisiert, die positive Zahlen als eine Art Guthaben betrachtet, daß z.B. zu Beginn eines Semesters bzw. durch Vorauszahlung angelegt wurde. Negative Zahlen, bzw. ein Betrag von 0, deuten drauf hin, daß alle Gespräche in Rechnung gestellt werden müssen.
- **CPL-Skript**
Das wohl größte Feld dient der Aufnahme eines CPL-Skriptes. Mit einem geeigneten Prozessor für CPL-Skripte steht ein flexibles Werkzeug zur individuellen Konfiguration zur Verfügung. Wie oben bereits erwähnt, wird im Rahmen der Diplomarbeit kein CPL-Prozessor geschrieben werden, sondern wahrscheinlich erst zu einem späteren Zeitpunkt hinzugefügt.
- **Privilegiengruppe**
Verweist auf die Privilegiengruppe, die der Benutzer angehört.

Es gibt einige Daten, die sinnvollerweise auch pro Nutzer gespeichert werden sollten, die in der obigen Liste nicht auftauchen. Dazu gehören z.B. Einzelverbindungsdaten und Authentifizierungsinformationen. Beide hätten die Entwicklung weiterer Module — z.B. eines *Billing/Account-Modul* — und evtl. eine Erweiterung der Architektur um einen Authentifizierungsserver bedeutet und den Rahmen dieser Arbeit gesprengt.

Globale Daten Die folgenden Felder enthalten globale Daten, die teilweise erst die Interpretation der User-Daten bestimmen.

- **Liste aller Endpunkte der Zone inkl. Zeitschema**
Diese Liste gibt an, welche Endpunkte der Gatekeeper als in seiner Zone befindlich versteht (vgl. 5.2.1), inklusive der Information darüber, zu welchen Zeiten von dem Endpunkt telefoniert werden kann.
- **Liste der Service- und Notfalladressen**
Über diese Liste sollen Adressen definiert werden, die jeder zu jederzeit anrufen kann — z.B. um Notfälle zu melden (vgl. 5.2.2). Obwohl gegenwärtig vom implementierten Gatekeeper noch nicht genutzt, ist diese Liste als eine spätere Erweiterung bereits vorgesehen.
- **Privilegiendefinitionen**
Hier wird festgelegt, welche Privilegien für die Personen einer bestimmten Gruppen gelten. Da unterschiedliche Policy-Module diese Information

verwenden können, muß hier leider eine genaue Definition der möglichen Daten erfolgen. (vgl. 5.2.4)

- Initialer Kontostand — sozusagen Freieinheiten
- Kontountergrenze — bei negativen Werten darf überzogen werden.
- maximal zugestandene Bandbreite — -1 = unbegrenzt
- maximale Zahl von Anrufen pro Zeitraum — -1 = unbegrenzt
- maximaler Platz auf internem Anrufbeantworter - in MB

Mit diesen Daten können hoffentlich ausreichend viele Informationen gespeichert werden, die Policy-Module sinnvoll verwenden können.

Die komplette Spezifikation der MBus-Kommandos zur Datenmanipulation und -abfrage findet sich im Anhang B.3.

Lokale oder globale Datenaufbewahrung

Im Rahmen dieser Diplomarbeit wird das Datenbank-Modul als lokale Applikation implementiert, die ihre Daten einem SQL-Server anvertraut. Alternativ könnte aber auch ein LDAP-Server verwendet werden.

Der LDAP-Verzeichnisdienst speichert die Daten als ein hierarchischer Baum von Objekten. Objekte, d.h. die Blätter des Baumes, können durch die Angabe des vollen Pfadnamens aufgefunden werden. Ein Objekt ist eine Sammlung von Attributen.

Die Vorteile von LDAP liegen in der Verteiltheit. Es ist möglich mehrere Verzeichnisdienste, d.h. Bäume zu einem großen Baum zusammenzufassen, in dem die einzelnen Bäume als Teilbäume eines großen Verzeichnis enthalten sind. Jeder einzelne Teilbaum kann von einer lokalen Organisation administriert werden, was es ermöglicht, die Datenpflege dort zu betreiben, wo die Daten herkommen.

Während LDAP eher als ein globales Datenbanksystem geeignet ist, wird jedoch für die Daten des Gatekeepers eher ein lokales System benötigt, was in der Regel lediglich Listen von etwas bedarf und keine Hierarchie benötigt. Hier bieten sich eher relationale Datenbanksysteme an, die eine bequeme Listenverwaltung erlauben und zudem die Möglichkeit haben, den Eintrag einer Liste in Relation zu Einträgen einer anderen Liste zu setzen. Die Wahl fiel daher auf eine SQL-Datenbank.

Es sei jedoch angemerkt, daß die Schnittstelle zum Datenbank-Modul so gehalten ist, daß ein Umstieg auf LDAP lediglich den Austausch des Moduls bedeuten sollte.

4.1.4 Das API/GUI-Modul

Bei diesem Modul muß man eigentlich zwei Schichten betrachten: Zum einen ein API, das als eigenständiges Modul auftritt, aber noch keine Funktion erfüllt; und zum anderen das GUI, das die vom API bereitgestellten Funktionen verwendet.

Das API stellt die folgenden Funktionen bereit:

- Erzeugen, Ermitteln, Ändern und Löschen von Zugangsberechtigungen anhand der IP-Adresse,

- Erzeugen, Ermitteln, Ändern und Löschen von Nutzerdaten,
- Erzeugen, Ermitteln, Ändern und Löschen von Gruppendaten,
- Ermitteln des gegenwärtigen Ressourcenverbrauchs,
- Ermitteln und Ändern der Konfiguration der Ressourcenverwaltung,
- Benachrichtigung bei An- und Abmeldung von Endpunkten und
- Benachrichtigung bei Beginn und Ende von Gesprächen

Diese Funktionen können, außer vom GUI, von beliebigen Modulen als Grundlage verwendet werden. Die konkrete Gestaltung des GUIs läßt sich 5.4 entnehmen.

Anders als bei Endpunkten ist das GUI beim Gatekeeper nicht zwingend notwendig, da dieser für den normalen Betrieb nicht auf Interaktion mit dem Benutzer angewiesen ist. Eine Benutzungsschnittstelle wird erst benötigt, wenn die Datenbestände oder Einstellungen der Policies geändert werden sollen.

4.2 Mögliche Erweiterungen

Da im Rahmen dieser Diplomarbeit nicht alle denkbaren und sinnvollen Erweiterungen implementiert werden können, sei an dieser Stelle knapp darauf hingewiesen, welche Erweiterungen noch sinnvoll zu integrieren wären.

4.2.1 Protokollierung und Abrechnung

Solange durch die IP-Telefonie keine Kosten anfallen, bedarf es nicht unbedingt einer Möglichkeit, Gespräche in Rechnung zu stellen. Wenn aber Gateways ins herkömmliche Telefonnetz ins Spiel kommen, müssen sowohl die Kosten irgendwie abgerechnet werden, wie evtl. auch eine Möglichkeit geschaffen werden, um die Verwendung der begrenzten Gatewayressourcen zeitlich zu begrenzen. Hierfür wäre ein Modul sinnvoll, das Anrufe und verbrauchte Einheiten protokolliert und evtl. die Verbindungsdauer überwacht. Ein solches Modul wäre dann in der Lage Benutzern später eine Rechnung mit Einzelverbindungsübersicht zu erstellen.

In Abhängigkeit von der Privilegiengruppe will man eventuell Freieinheiten verteilen, das Erzeugen von Kosten verbieten oder ähnliche Einstellungen vornehmen.

Um überhaupt die anfallenden Kosten zu erfahren, sei es nach Ende eines Telefonats oder schon währenddessen, ist eine weitere Form des Informationsaustausches zwischen Gatekeeper und Gateway nötig, die nicht durch die RAS-Nachrichten abgedeckt wird.

4.2.2 Anrufbearbeitung

Unter Anrufbearbeitung wird die Reaktion auf eingehende Anrufe verstanden. Im einfachsten Fall bedeutet dies, daß der Gatekeeper prüft, ob das Ziel erreichbar ist, daß Anrufmodell bestimmt und die Verbindung zustandekommt. Wenn

der Zienteilnehmer nicht erreichbar ist, könnte der Anruf abgewiesen werden — er könnte aber auch auf einen Anrufbeantworter umgeleitet, bzw. zu einer anderen Adresse weitergeleitet werden. Unter Umständen sollen solche Aktionen noch in Abhängigkeit von Uhrzeit, dem Anrufer oder ähnlichem ausgelöst werden.

Es ist zudem wünschenswert, daß die Einstellungen zur Anrufbearbeitung vom Nutzer selbst vorgenommen werden können.

Eine bereits erwähnte Möglichkeit der Anrufbearbeitung ist ein Modul, welches CPL-Skripte abarbeitet. Die verwendeten Tabellen für Nutzerdaten in der Datenbank des von mir implementierten Gatekeepers sehen bereits Platz zur Aufnahme solcher Skripte vor. Was fehlt, ist das verarbeitende Modul und eine Möglichkeit, dem Nutzer das Anlegen, Ändern und Löschen solcher Skripte zu erlauben.

4.2.3 Externe User-Location

Bisher implementiert der Gatekeeper nur User-Location für Endpunkte seiner eigenen Zone. Sollen jedoch Adressen außerhalb der eigenen Zone aufgelöst werden, wird es nötig, daß der Gatekeeper Wissen über andere Zonen erhalten kann. H.323 sieht zwar einen Mechanismus vor, wie ein Gatekeeper einen anderen Gatekeeper nach Adressen fragen kann, nicht aber, wie ein Gatekeeper von anderen Gatekeepern erfährt und wie genau in diesem Fall vorgegangen werden soll. Das Problem wird in H.225.0 Annex G angesprochen.

Da das H323-Modul bei unbekanntem Adressen eine Anfrage auf dem MBus sendet, muß lediglich ein Modul hinzugefügt werden, welches auf der einen Seite über den Message Bus kommuniziert und auf der anderen Seite das in H.225.0 Annex G vorgestellte Protokoll spricht.

4.2.4 Gateway Location

In seiner bisherigen Form kann der Gatekeeper nur die Gateways zum Übergang in andere Netze verwenden, die sich bei ihm angemeldet haben, d.h. die in derselben Zone liegen. Nun kann es aber sein, daß in einer Zone überhaupt keine Gateways vorhanden sind, bzw. die Verwendung des lokalen Gateways teurer ist, als die eines externen Gateways.

Damit ein Gatekeeper Informationen über weitere Gateways im Netz erhält und aus diesen Informationen das geeignete Gateway für ein netzübergreifendes Gespräch ermitteln kann (*Call-Routing*), muß hier ein geeignetes Protokoll bereitgestellt werden, das dies ermöglicht. Auch hier bietet sich eine Implementation des in H.225.0 Annex G vorgestellten Protokolls an, da es sich um eine ähnliche Problematik handelt.

4.2.5 Mehrwertdienste

Eine Reihe von Mehrwertdiensten können entweder in Gatekeepern oder Endpunkten zur Verfügung gestellt werden. Einige davon, wie z.B. Anrufweiterleitung, könnten von dem Modul zur Anrufbearbeitung zur Verfügung gestellt, andere, wie z.B. spontane Konferenzen oder Anrufbeantworter, könnten von zentralen Komponenten erbracht werden.

- **Anrufumleitung (Call Transfer)**

Dieser Dienst wird am ehesten vom einem Gatekeeper implementiert, der das *Gatekeeper-routed* Anrufmodell unterstützt. Wird ein Gespräch von A nach B über den Gatekeeper geleitet und soll von B nach C umgeleitet werden, so sollte der Endpunkt B dem Gatekeeper mitteilen, daß das Gespräch nach C umgeleitet werden soll. Der Gatekeeper muß dann das Gespräch parken, evtl. eine Wartemelodie abspielen und eine Verbindung zu C aufbauen. Sobald diese Verbindung steht, behandelt sie der Gatekeeper solange wie ein Gespräch zwischen B und C, bis B den Anruf abgibt — womit dann die Verbindung zwischen B und dem Gatekeeper beendet wäre.

- **Anrufweiterleitung (Call Forwarding)**

Wenn ein Anruf für einen User X vorliegt, soll dieser unter Umständen weitergeleitet werden — z.B. auf einen Anrufbeantworter oder ein anderes Telefon. Geeignet hierfür wäre ein vom Nutzer konfigurierbares CPL-Skript, welches entweder vom Gatekeeper oder vom Endpunkt ausgeführt wird. Da jedoch ein weiterzuleitender Anruf intelligenterweise gar nicht erst bis zum beabsichtigten Endpunkt durchdringen sollte, sollte dies bereits im Gatekeeper gelöst werden.

Im einfachsten Fall, nämlich bei Unterstützung des *Gatekeeper-routed* Anrufmodells, findet der Gatekeeper intern heraus, daß der Anruf weitergeleitet werden soll und baut die Verbindung zum transparent für den Anrufer eine Verbindung zum neuen Ziel auf.

Wenn der Anruf nicht über den Gatekeeper geleitet werden soll, z.B. weil sowohl Anrufer als auch neues Ziel außerhalb der Zone des Gatekeepers liegen, sollte dieser dem anrufendem Endpunkt eine Nachricht senden, die die neue Adresse enthält und sich darauf verlassen, daß der Endpunkt diese Nachricht versteht — was aufgrund der Vielfalt der Hersteller und Protokollversionen eher unwahrscheinlich ist.

- **Anruf abweisen (Call Deflection)**

Ähnlich wie bei der Anrufweiterleitung gilt hier, daß entweder im Gatekeeper oder im Endpunkt konfiguriert werden kann, wenn Anrufe abgewiesen werden sollen. In beiden Fällen könnte jeweils ein Policy-Modul anhand der mitgelieferten Nummer des Anwender bestimmen, ob der Anruf angenommen werden soll oder nicht.

- **Anruf parken (Call park & pickup)**

Hier verhält es sich ähnlich wie bei der Anrufumleitung. Ein Endpunkt teilt dem Gatekeeper mit, daß es den Anruf parken und von einer anderen Adresse wieder fortführen möchte und bricht danach die Verbindung ab. Der Gatekeeper hält den Anruf, während er eine Verbindung zum neuen Endpunkt aufbaut. Dies setzt voraus, daß das *Gatekeeper-routed* Anrufmodell verwendet wird.

- **Message Waiting Indication**

Dies ist eher ein Dienst, der dem Endpunkt vom Anrufbeantworter geboten wird und setzt voraus, daß der Endpunkt in regelmäßigen Abständen beim Anrufbeantworter-Server nachfragt, ob neue Nachrichten vorliegen. Eine andere Variante wäre, daß der Gatekeeper sich merkt, welche Anrufe

wegen Nichtauffindbarkeit des Teilnehmers an den Anrufbeantworter weitergeleitet wurden und aktiv den Anrufbeantworter informiert, sobald sich einer dieser Teilnehmer wieder registriert. Der Anrufbeantworter könnte dann von sich aus den Endpunkt über wartende Nachrichten informieren.

Diese Liste ist bei weitem nicht komplett, noch beschreibt sie die möglichen Verfahren erschöpfend - was aber auch den Rahmen dieses kurzen Vordenkens. Eine genauere Spezifikation der Mehrwertdienste findet sich in H.450.

4.2.6 Andere Administrationsschnittstelle

Mit diesem Gatekeeper kommt eine eigene Konfigurationsschnittstelle für die verwendeten Policy-Module und die Nutzerdatenbank. Vorteilhafter wäre es, auf einem existieren Standard zur Konfiguration von Netzkomponenten aufzusetzen. Ein solcher Standard wird mit dem *Simple Network Management Protocol* (SNMP) definiert.

Es ist denkbar, daß die Konfiguration des Gatekeepers über ein Modul erfolgt, welches SNMP spricht. Mit dem ITU-Standard H.341 steht bereits eine H.323-MIB zur Verfügung.

4.3 Interne Abläufe und Entscheidungen innerhalb des H.323-Moduls

In diesem Abschnitt soll erläutert werden, welche internen Abläufe eingehende Nachrichten beim Gatekeeper hervorrufen und welche Entscheidungen dabei anfallen. Dabei wird bei der Auflistung der Fragen gleich angegeben, wer diese Frage zu beantworten hat. Ein (H) steht dabei für das H323-Modul, ein (P) für die Policy-Module, ein (D) für das Datenbankmodul und (U) für ein evtl. vorhandenes externes User-Location-Modul.

4.3.1 Gatekeeper-Discovery

Ein Endpunkt muß beim Starten in Erfahrung bringen, welcher Gatekeeper für ihn zuständig ist. Hierfür sendet er eine *Gatekeeper-Request-PDU* (GRQ) per Multicast an die wohldefinierte Discovery-Adresse des Gatekeepers (siehe auch 2.1.1). In der GRQ wird neben einigen anderen Informationen auch die Versionsvariante von H.323, die der Endpunkt unterstützt, und evtl. auch explizit ein Gatekeeper angegeben, den der Endpunkt als Gatekeeper bevorzugen würde.

Der empfangende Gatekeeper muß daher folgende Fragen beantworten:

1. *Entspricht der eigene Gatekeeper dem gewünschten? (H)*
Der Endpunkt gibt evtl. einen Gatekeeper an, bei dem er sich gerne anmelden würde. Der Gatekeeper braucht nur solche Endpunkte anzunehmen, die ihn als Wunschgatekeeper angegeben haben.
2. *Wird die Protokollversion des Endpunktes unterstützt? (H)*
H.323 nimmt mit jeder Version (z.Zt. ist Version 3 in Arbeit) an Funktionalität zu. Unterschiedliche Protokollversionen von Endpunkten und Gatekeepern sind daher evtl. ein Entscheidungsgrund.
3. *Fällt der Endpunkt in die Zuständigkeit des Gatekeepers? (P,D)*
Hierbei handelt es sich im Prinzip um die Frage, ob der Endpunkt in unserer Zone liegt oder nicht. Welche Endpunkte in unserer Zone liegen, wird durch die Liste der bekannten Endpunkte in der Datenbank definiert. Diese Liste wird vom Policy-Modul durch das Datenbank-Modul definiert.

Frage 1 wird von dem H323-Modul beantwortet, da hier die Information darüber vermerkt ist, wie sich der Gatekeeper selbst identifiziert. Ebenso verhält es sich mit der zweiten Frage, da das H323-Modul als einziges weiß, mit welchen Versionen von H.323 es arbeiten kann.

Die dritte Frage wird den Policy-Modulen in Form von `mbus.poll.isLocalZone` gestellt (Abb. 4.2 und Tabelle 4.3.1). Anhand der Transportadresse müssen die Policy-Module entscheiden, ob der Gatekeeper für den Endpunkt zuständig ist oder nicht.

4.3.2 Registrierung eines Endpunktes

Wenn ein Endpunkt weiß, welcher Gatekeeper für ihn zuständig ist, ist sein nächster Schritt die Registrierung bei diesem. Hierfür schickt er dem Gatekeeper eine *Registration-Request-PDU* (RRQ), in der unter anderem Daten wie z.B.

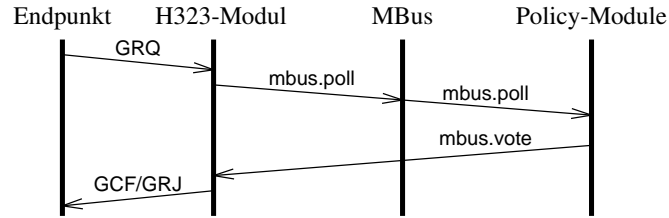


Abbildung 4.2: Nachrichtenaustausch bei Gatekeeper-Discovery

Empfänger	mbus.poll	Optionen	Informationen
Policy	isLocalZone	Yes No	IP-Adresse

Tabelle 4.1: Voting, um Zonenzugehörigkeit von Endpunkten festzustellen

Adressen des Endpunktes, Aliasnamen, Endpunkttyp, die Protokollversion und evtl. der gewünschte Gatekeeper angegeben werden.

An dieser Stelle sei darauf hingewiesen, daß sich ein Endpunkt mit mehreren Aliasnamen für eine Person registrieren kann. Der Gatekeeper verwaltet die Endpunkte anhand ihrer IP-Adressen, um Nachrichten, wie z.B. die Aufforderung zur Auffrischung der Registration, jeweils nur einmal zu einem Endpunkt zu schicken. Im Folgenden tauchen daher oft IP-Adresse und Aliasnamen zusammen auf.

Der Gatekeeper muß jetzt entscheiden:

1. *Entsprechen der Gatekeeper dem gewünschten? (H)*
Der Endpunkt gibt evtl. einen Gatekeeper an, bei dem er sich registrieren möchte. Wir brauchen nur solche Registrationen entgegennehmen, die auch für uns bestimmt sind.
2. *Sind die angegebenen Adressen gültig? (H)*
Evtl. sind die angegebenen Adressen nicht korrekt, d.h. enthalten ungültige Werte. In diesem Fall sollte die Registration abgelehnt werden.
3. *Unterstützen der Gatekeeper die Protokollversion und die Transportarten? (H)*
H.323 nimmt mit jeder Version (z.Zt. ist Version 3 in Arbeit) an Funktionalität zu. Unterschiedliche Protokollversionen von Endpunkten und Gatekeepern sind daher evtl. ein Entscheidungsgrund.
Dazu kommt, daß die verwendete Art von Transportadressen netzabhängig ist, d.h. ein Gatekeeper in einem IP-Netz, mit IPX-Transportadressen nichts anfangen kann.
4. *Wird die Art des angegebenen Aliasadresse unterstützt? (H)*
Es wird geprüft, ob es sich um eine H.323- oder eine E.164-Adresse handelt. Andere Arten von Aliasadressen werden abgewiesen.
5. *Sind bereits Nutzer mit der Aliasadresse registriert? (H)*
Dies soll verhindern, daß Nutzer auf mehreren Endpunkten zugleich regi-

striert sind.

6. *Handelt es sich um einen Benutzer, der sich registrieren darf? (P)*

Da nur bekannte Benutzer telefonieren können, bzw. auch nur solche, die nicht gerade gesperrt sind, werden noch die Policy-Module befragt.

Die meisten dieser Fragen kann das H323-Modul intern beantworten — es müssen also keine anderen Module hinzugezogen werden. Lediglich die letzte Frage geht an die Policy-Module. Erfolgt von diesen eine positive Antwort, so wird die Nachricht `register` auf den MBus gesendet, die es angeschlossenen Modulen ermöglichen soll, registrierte Endpunkte mitzuverfolgen (Abb. 4.3).

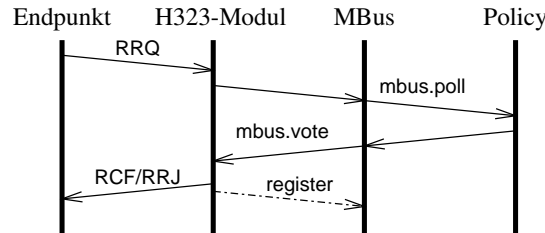


Abbildung 4.3: Nachrichtenaustausch bei der Endpunkt-Registrierung

Empfänger	mbus.poll	Optionen	Informationen
Policy	mayRegister	Yes No	Aliasname

Tabelle 4.2: Voting, um Zonenzugehörigkeit von Endpunkten festzustellen

Empfänger	Kommando	Informationen
alle	register	CS-Adresse, RAS-Adresse, Aliasnamen, Endpunkttyp

Tabelle 4.3: Kommando zur Anmeldung von Endpunkten

Erläuterung zu Tabelle 4.3.2:

- CS-Adresse: Die Call-Signaling-Adresse, d.h. die Transportadresse, an der ein Endpunkt Verbindungswünsche entgegennimmt.
- RAS-Adresse: Die Transportadresse des Endpunkts für eingehende Nachrichten auf dem RAS-Kanal.
- Endpunkttyp: Die Art des sich registrierenden Endpunkts (Gatekeeper, Gateway, Terminal, ...)

4.3.3 Abmelden beim Gatekeeper

Wenn ein Endpunkt sich oder zumindest einige Aliasadressen beim Gatekeeper abmelden möchte, sendet er eine *Unregistration-Request-PDU* (URQ). Dies darf

er jederzeit tun, vorausgesetzt er ist überhaupt registriert und nicht gerade an einem Anruf beteiligt.

Es ergeben sich folgende Fragen:

1. *Ist der Endpunkt bei uns registriert? (H)*
Es können nur solche Endpunkte abgemeldet werden, die auch registriert sind.
2. *Können die angegebenen Aliasadressen abgemeldet werden? (H)*
Ebenfalls in der Liste der registrierten Endpunkte des H323-Moduls wird der Zustand der Endpunkte vermerkt. Ist ein Endpunkt in ein Gespräch verwickelt, dürfen die hierbei verwendeten Aliasadressen nicht abgemeldet werden. Zur Sicherheit sendet der Gatekeeper in diesem Fall jedoch ein *Information Request (IRQ)*, um den Status des Endpunktes abzufragen. Ist dieser nicht mehr erreichbar, wird er aus den internen Listen gestrichen.

Diese Fragen kann sich das H323-Modul wieder selbst beantworten. Nach erfolgter Austragung des Endpunktes aus den internen Listen wird die Nachricht `unregister` auf den MBus gesendet, die die angeschlossenen Module darüber informiert, daß der Endpunkt abgemeldet ist. Der Erhalt dieser Nachricht wird nicht bestätigt, da es kein MBus-Modul gibt, welches auf den Erhalt dieser Information angewiesen ist. Einzig das API/GUI-Modul verwertet diese Information, jedoch ist dies die meiste Zeit nicht am MBus.

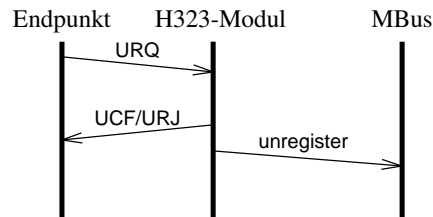


Abbildung 4.4: Nachrichtenaustausch bei der Endpunkt-Abmeldung

Empfänger	Kommando	Informationen
alle	unregister	Aliasnamen, CS-Adresse

Tabelle 4.4: Kommando zur Abmeldung von Endpunkten

4.3.4 Erbitten der Anruferlaubnis

Wenn ein Endpunkt eine Verbindung zu einem anderen Endpunkt aufbauen möchte, egal ob der Anruf von ihm oder vom Partner initiiert wurde, muß er zuvor vom Gatekeeper die Erlaubnis dazu einholen. Dies tut er, indem er eine *Admission-Request-Nachricht (ARQ)* sendet, die neben Informationen über Art und Ziel des Anrufes z.B. auch die gewünschte Bandbreite enthält. Bei eingehenden Anrufen wird nach Erhalt einer *Setup-Nachricht* eine ARQ an den

Gatekeeper gesendet, die weitgehend ähnliche Informationen enthält.

Folgende Fragen resultieren daraus:

1. *Ist der Endpunkt bei uns registriert? (H)*
Das H323-Modul führt die Liste, über die registrierten Endpunkte und kann daher diese Frage selbst beantworten.
2. *Kann die Zieladresse aufgelöst werden? (H,U)*
Diese Frage stellt sich nur bei ausgehenden Anrufen Die Adreßauflösung für die Zieladresse wird zuerst im H323-Modul selbst vorgenommen. Falls dies fehlschlägt, werden evtl. angeschlossene User-Location-Module gefragt.
3. *Darf der Nutzer überhaupt bzw. von der angegebenen Adresse aus telefonieren? (P)*
Evtl. sind die Policies derart, daß zu bestimmten Uhrzeiten und/oder von bestimmten Endpunkten aus das Telefonieren untersagt ist.

Dies wird übrigens bisher nur für die Adresse des anfragenden Endpunktes geprüft. Die Prüfung für die angerufene Adresse findet auf Seiten des Angerufenen statt. Um z.B. zu ermöglichen, daß es einige immer anrufbare Adresse („Notfallnummern“) gibt, müßte das an dieser Stelle eingebaut werden.
4. *Steht die gewünschte Bandbreite zur Verfügung? (P)*
Die Policy-Module zur Ressourcenverwaltung müssen entscheiden, ob die vom Benutzer gewünschte Bandbreite zur Verfügung steht. Gegebenenfalls muß die angeforderte Bandbreite reduziert werden.

Es werden also zwei bis drei Fragen an den MBus weitergeleitet. Zunächst muß evtl. mit `locate` eine Auflösung einer externen Adresse vorgenommen werden. Wenn es absehbar ist, daß die Adreßauflösung länger dauert, kann dem Endpunkt eine *Request-In-Progress-PDU* (RIP) gesendet werden. Kann die Adresse nicht aufgelöst werden, so wird eine *Admission-Reject-PDU* (ARJ) gesendet und der Vorgang abgebrochen.

Im Anschluß wird unter den Policy-Modulen abgestimmt, ob (`mbus.poll`) und mit welcher Bandbreite (`poll.bandwidth` - siehe auch B.5.2) telefoniert werden darf. Ablehnungen der Policy-Module resultieren in ARJ-Nachrichten.

Als letztes wird auf dem MBus ein `beginCall` gesendet, um das Protokollieren des Anrufs zu ermöglichen und dann eine Bestätigung in Form eine *Admission-Confirm-PDU* (ACF) an den Endpunkt gesendet.

4.3.5 Bandbreite eines Gesprächs ändern

Wenn ein Endpunkt wünscht, die Bandbreite für einen Anruf in dessen Verlauf zu ändern, sendet er an den Gatekeeper eine *Bandwidth-Request-PDU* (BRQ), in der er die gewünschte Bandbreite mitteilt. Daraus resultieren folgende Fragen für den Gatekeeper:

1. *Ist der Endpunkt bei uns registriert? (H)*
Das H323-Modul führt die Liste über die registrierten Endpunkte und kann daher diese Frage selbst beantworten.

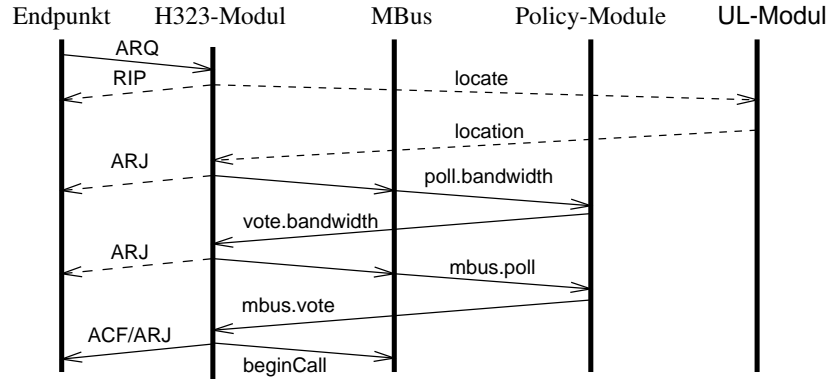


Abbildung 4.5: Nachrichtenaustausch beim Einholen einer Anruferlaubnis

Empfänger	Kommando	Informationen
alle	locate	Aliasnamen
H323	location	Liste von Paaren aus Aliasname und CS-Adresse
Policy	poll.bandwidth	Anruf-ID, gewünschte Bandbreite
Policy	vote.bandwidth	Anruf-ID, gestattete Bandbreite
alle	beginCall	Anruf-ID, lokale ID, Anrufender, Anrufer, Startzeit, Bandbreite

Tabelle 4.5: MBus-Kommandos für Adreßauflösung und Bandbreitenbestimmung

2. *Sind der Endpunkt, von dem die PDU stammt, und der Endpunkt, dessen Bandbreite geändert werden soll, identisch? (H)*
Dies vermeidet, daß ein Endpunkt Bandbreite für fremde Gespräche ändern kann.
3. *Gibt es das betreffende Gespräch überhaupt? (H)*
Natürlich können nur Ressourcen für angemeldete Gespräche geändert werden. Falls das Gespräch nicht existiert, wird das Ansinnen des Endpunktes abgelehnt.
4. *Steht die gewünschte Bandbreite zur Verfügung? (P)*
Die Policy-Module zur Ressourcenverwaltung müssen entscheiden, ob die vom Benutzer gewünschte Bandbreite zur Verfügung steht. Gegebenenfalls muß die angeforderte Bandbreite reduziert werden.

4.3.6 Gesprächsende signalisieren

Hat ein Endpunkt seine Verbindung beendet, so teilt er dies dem Gatekeeper in Form einer *Disengage Request (DRQ)* mit, damit dieser die reservierten Ressourcen wieder freigeben kann. Ein Gatekeeper hat in diesem Zusammenhang drei Sachen zu prüfen:

Empfänger	mbus.poll	Optionen	Informationen
Policy	mayCall	Yes NotThatIP NotNow NotThatUser	Anruf-ID, Alias des Anrufers Quell-Adresse Alias des Angerufenen Ziel-Adresse

Tabelle 4.6: Voting zur Zugangserlaubnis anhand von Nutzer und IP-Adresse

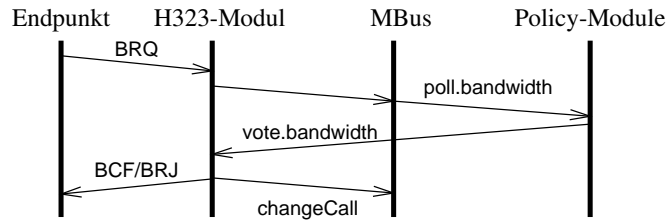


Abbildung 4.6: Nachrichtenaustausch bei einer Bandbreitenänderung

1. *Ist der Endpunkt bei uns registriert? (H)*
Das H323-Modul führt die Liste über die registrierten Endpunkte und kann daher diese Frage selbst beantworten.
2. *Sind der Endpunkt, von dem die PDU stammt, und der Endpunkt, dessen Gespräch beendet werden soll, identisch? (H)*
Dies vermeidet, daß ein Endpunkt die Gespräche anderer Teilnehmer beenden kann.
3. *Gibt es das zu beendende Gespräch überhaupt? (H)*
Natürlich können nur Ressourcen für angemeldete Gespräche freigegeben werden. Sollen also nicht belegte Ressourcen freigegeben werden, wird dies ignoriert.

Die Entscheidungen können alle im H323-Modul fallen. Nach erfolgter Abmeldung muß eine Nachricht auf den MBus gehen, damit die Policy-Module zur Ressourcenverwaltung auch erfahren, daß die Bandbreite nicht mehr benötigt wird.

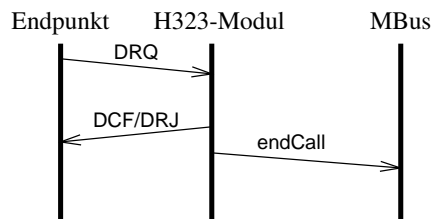


Abbildung 4.7: Nachrichtenaustausch beim Beenden eines Anrufes

Empfänger	Kommando	Informationen
Policy	poll.bandwidth	Anruf-ID, gewünschte Bandbreite
Policy	vote.bandwidth	Anruf-ID, gestattete Bandbreite
alle	changeCall	Anruf-ID, Bandbreite

Tabelle 4.7: MBus-Kommandos für Bandbreitenbestimmung- und -änderung

Empfänger	Kommando	Informationen
alle	endCall	Anruf-ID

Tabelle 4.8: MBus-Nachricht bei Endpunkt-Abmeldung

4.3.7 Adresse auflösen

Es kann vorkommen, daß ein (ggf. nicht registrierter) Endpunkt dem Gatekeeper eine *Location-Request-PDU* (LRQ) sendet, um eine Adresse aufgelöst zu bekommen. Hierbei wird ein etwas anderes Verfahren zur Adreßauflösung verwendet, als beim Erbitten einer Anruferlaubnis: Während im letzten Fall der Gatekeeper auch externe Quellen befragt, um die Adresse aufzulösen, wird bei einer LRQ nur die Liste der registrierten Endpunkte durchgesehen.

Ist der Aliasname bekannt, wird in jedem Fall eine *Location-Confirm-PDU* (LCF) gesendet. Ist der Aliasname nicht bekannt, sollte der Gatekeeper nur dann mit einer *LocationReject* (LRJ) antworten, wenn die Anfrage direkt an ihn gerichtet war und nicht lediglich auf der Discovery-Adresse aufgefangen wurde.

1. *Ist der Endpunkt bei uns registriert? (H)*
Das H323-Modul führt die Liste über die registrierten Endpunkte und kann daher diese Frage selbst beantworten.
2. *Kann irgendein anderes Modul die Adresse auflösen? (alle)*
So die Anfrage direkt an den Gatekeeper gerichtet war, fragt dieser auf dem MBus ein eventuell vorhandenes Location-Modul, das seinerseits wieder andere Gatekeeper fragt.

4.3.8 Status melden

Ein Endpunkt kann vom Gatekeeper aufgefordert werden, seinen gegenwärtigen Status in Form einer *Info-Request-Response-PDU* (IRR) mitzuteilen oder dies in regelmäßigen Abständen unaufgefordert tun. Eine IRR enthält Informationen über den Endpunkt selbst und detaillierte Angaben (Verwendete Codecs, Anrufmodell, Bandbreite ...) über die Anrufe, in die er verwickelt ist.

Im ersten Fall sind keine weiteren Entscheidungen zu treffen, da es sich um vom Gatekeeper explizit angeforderte Informationen handelt. Im zweiten Fall kann der sendende Endpunkt eine Empfangsbestätigung fordern. Der Gatekeeper muß also prüfen:

1. *Ist der Endpunkt bei uns registriert? (H)*
Das H323-Modul führt die Liste über die registrierten Endpunkte und kann daher diese Frage selbst beantworten.

Der Empfang einer IRR führt zu keiner MBus-Aktivität, da das H.323-Modul diese Informationen ausschließlich zu eigenen Zwecken auswertet.

4.4 Interne Abläufe und Entscheidungen in den Policy-Modulen

4.4.1 Zugangsberechtigung für einen Endpunkt prüfen

Wie in 4.3.1 im Kontext einer GRQ beschrieben, muß das H.323-Modul erfahren, ob ein Endpunkt seiner Zone angehört. Grundlage dieser Entscheidung ist die IP-Adresse des betreffenden Endpunktes.

Zunächst wird jedoch geprüft, ob der Gatekeeper evtl. so konfiguriert ist, daß er Endpunkte von jeder beliebigen IP verwaltet (siehe Abb. 4.8). Dies wäre gleichbedeutend mit einer beliebig großen Zone — nicht unbedingt sinnvoll, aber denkbar.

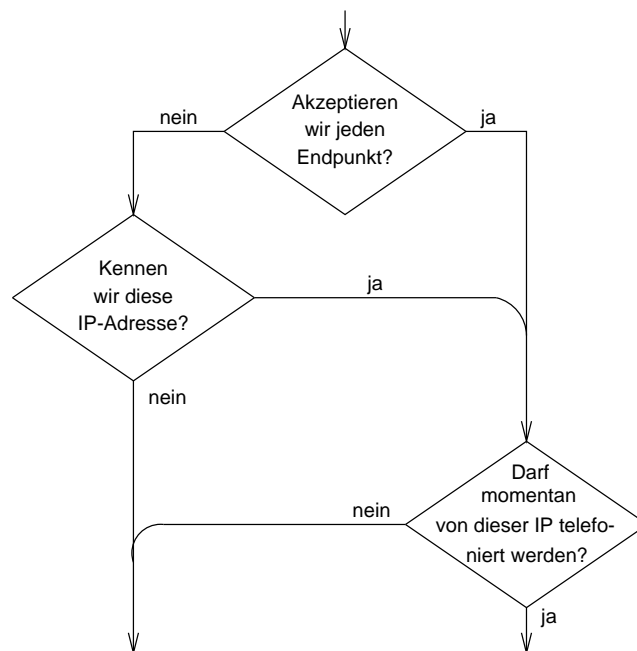


Abbildung 4.8: Policy-Entscheidungsablauf bei `isLocalZone`

Die Entscheidung auf der Basis der IP-Adressen ist übersichtlich: Es werden alle Definitionen aus der Liste der IP-Bereiche aus der Datenbank ermittelt, die auf diesen Endpunkt zutreffen. Die Ergebnisse werden anhand des Grades der Übereinstimmung sortiert.

Ein Beispiel:

Der Endpunkt mit der IP 134.102.218.62 sucht seinen Gatekeeper. Das Policy-Modul findet einen Eintrag für 134.102.218 und einen für 134.102.218.62. Beide

werden zurückgeliefert, aber der genau passende wird als einziger ausgewertet. Hätte es ihn nicht gegeben, wäre der andere Eintrag verwendet worden.

Diese Vorgehensweise erlaubt es, allgemeine Regeln für z.B. ein Subnetz aufzustellen und spezielle Regeln für besondere Endpunkte darin zu setzen. Dies könnte z.B. so aussehen, daß es eine Regel für die Subnetz im Rechnerpool gibt, die das Telefonieren, d.h. den Zugang, zu den Stoßzeiten von 10 - 17 Uhr verbietet aber spezielle Regeln für die Endpunkte des T-Bereichs in diesem Netz dies wiederum zu allen Zeiten erlaubt.

4.4.2 Zugangsberechtigung für einen Benutzer prüfen

Beim Prüfen eines Registrationswunsches eines Benutzers an einem Endpunkt (4.3.2) ist im Prinzip die einzige Entscheidung des Policy-Moduls, ob es den Benutzer kennt (Abb. 4.9).

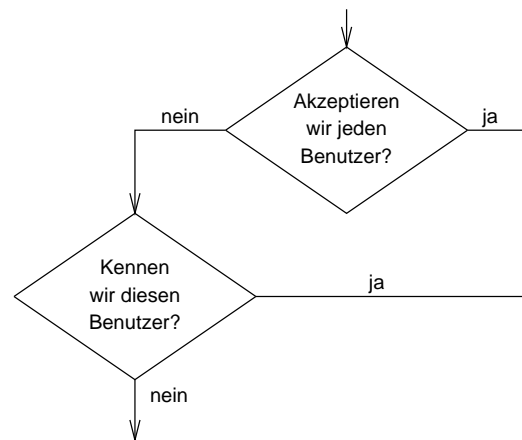
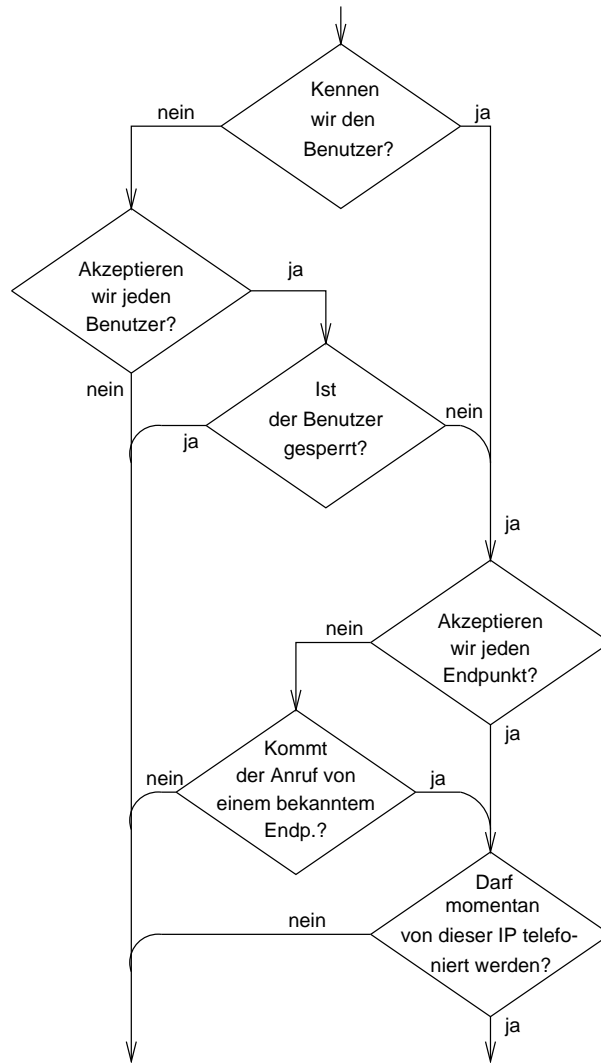


Abbildung 4.9: Policy-Entscheidungsablauf bei mayRegister

Wie schon beim Prüfen auf die Zonenzugehörigkeit (s.o.) kann man jedoch einstellen, daß sich alle Nutzer, die dem Gatekeeper bekannt sind, registrieren können - unabhängig davon, ob sie in der Datenbank vorhanden sind oder nicht. Dies mag in einer Einsatzumgebung sinnvoll sein, in der keine Abrechnungen nötig sind und alle Nutzer gleich behandelt werden sollen.

4.4.3 Prüfen, ob der Anruf erlaubt ist

Um zu entscheiden, ob es einem Nutzer erlaubt ist, einen bestimmten Anruf absetzen darf, wird vom H323-Modul sowohl die Herkunft des Anrufes anhand von IP-Adresse und Aliasadresse, als auch die verwendete Bandbreite geprüft (vergl. 4.3.4).



Wenn das Policy-Modul zur Zugangskontrolle eine mayCall-Anfrage bekommt, wird zunächst der Benutzer geprüft (siehe Abb. 4.4.3). Dies geschieht, indem zunächst geprüft wird, ob generell jeder Benutzer akzeptiert wird und falls nicht, ob der Benutzer in der Liste der Benutzer ist, die dem Gatekeeper bekannt sind, und nicht gesperrt wurde.

Nachdem der Benutzer geprüft wurde, erfolgt die Prüfung in Abhängigkeit der IP-Adresse. Ist das Policy-Modul nicht so konfiguriert, daß es alle IP-Adressen akzeptiert, wird geprüft, ob der Anrufwunsch von einem Endpunkt aus der Zone des Gatekeepers, d.h. einem Endpunkt aus der Datenbank kommt. Sollte dies der Fall sein, bzw. jede IP-Adresse akzeptiert werden, wird geprüft, ob zur gegenwärtigen Stunde von der IP-Adresse telefoniert werden kann. Für den Fall, daß die IP-Adresse des Endpunktes nicht in den lokalen Tabellen auftaucht, das Policy-Modul aber konfiguriert wurde, jede IP-Adresse als Herkunft zuzulassen, ist das telefonieren zu jeder Zeit erlaubt.

4.4.4 Bandbreitenforderung prüfen

Vor dem Beginn eines Gesprächs (4.3.4) oder in dessen Verlauf (4.3.5) gibt ein Endpunkt eine gewünschte Bandbreite an. Über diesen Wunsch nach Bandbreite entscheidet das Policy-Modul zur Ressourcenverwaltung, wobei die gewährte Bandbreite ggf. reduziert werden kann (siehe Abb, 4.10).

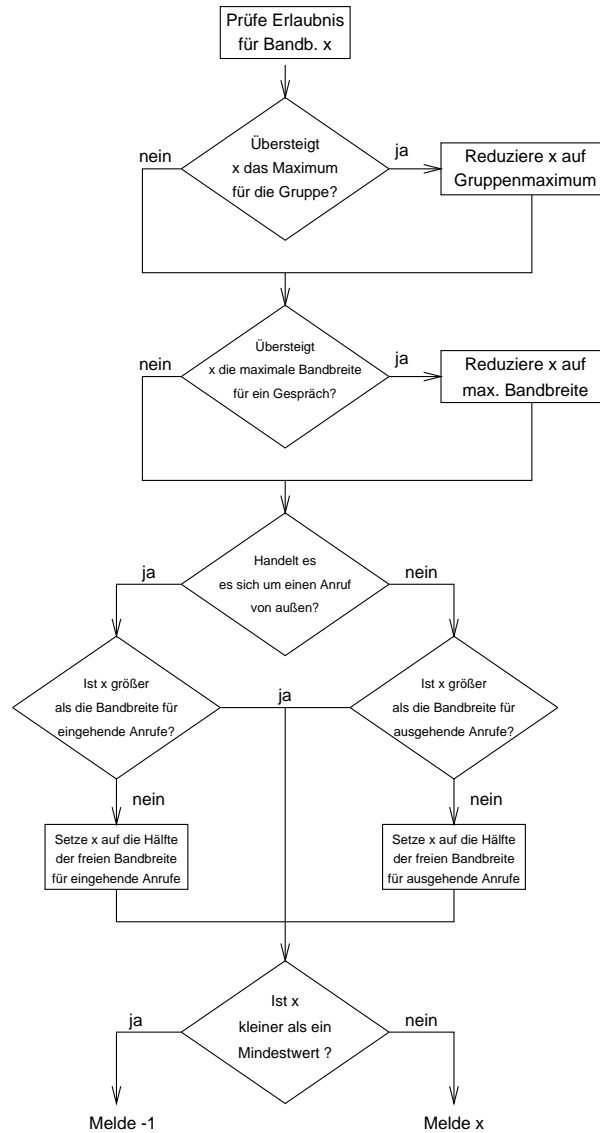


Abbildung 4.10: Entscheidungsablauf bei Ressourcenanfrage

Zunächst wird davon ausgegangen, daß die gewährte Bandbreite identisch mit der angeforderten Bandbreite ist. Dann wird die Gruppe ermittelt, der der Benutzer angehört und die bisher gewährte Bandbreite mit der maximalen Bandbreite verglichen, die der Gruppe zugestanden wird. Bei Bedarf wird

die gewährte Bandbreite auf die maximale Bandbreite für die Gruppe reduziert.

Analog wird anschließend geprüft, ob die gewährte Bandbreite auf die Bandbreite reduziert werden muß, die generell als Obergrenze für ein Gespräch gilt.

Die nächste Prüfung findet in Abhängigkeit von der Herkunft des Anrufes statt. Die Ressourcenverwaltung unterscheidet bei der Zuteilung von Bandbreite, ob der Anruf von außerhalb oder innerhalb der Zone stammt. Diese Aufteilung ermöglicht es, genug Bandbreite freizuhalten, damit auch bei voller Auslastung der Ressourcen für ausgehende Anrufe, d.h. Anrufe, deren Herkunft die Zone des Gatekeepers ist, immer noch Anrufe von außerhalb der Zone entgegengenommen werden können.

Die Ressourcenverwaltung prüft also, ob die gewährte Bandbreite den jeweiligen Anteil für eingehende oder ausgehende Anrufe übersteigt und gewährt ggf. nur die Hälfte der verfügbaren Bandbreite. Dies führt dazu, daß sich die verfügbare Bandbreite bei höher Auslastung asymptotisch Null nähert. Da natürlich ab einer gewissen Untergrenze an Bandbreite nicht mehr zufriedenstellend kommuniziert werden kann, wird die Ressourcenverwaltung zuletzt noch prüfen, ob die gewährte Bandbreite oberhalb eines Minimums liegt. Ist dies der Fall wird die gewährte Bandbreite zurückgeliefert, andernfalls -1.

Kapitel 5

Implementierung

5.1 Programmiersprachen

Der Gatekeeper stellt eine große Anwendung dar, in der viele Aufgaben unabhängig voneinander erledigt werden müssen und die zudem möglichst robust sein sollte. Aus diesem Grund habe ich mich für die Verwendung von JAVA als Programmiersprache entschieden. JAVA bietet durch seine komfortable Unterstützung von Threads und seine Fehlertoleranz die ideale Basis für ein solches Projekt. Ein weiterer Vorteil, der als Seiteneffekt durch die Wahl der Programmiersprache JAVA entstanden ist, ist die Plattformunabhängigkeit des Gatekeepers.

Durch die Verwendung von JAVA fiel die Wahl jedoch zugleich auf eine weniger schnelle, da interpretierte Sprache. Die Auswirkungen sollten sich jedoch erst später herausstellen (siehe 7.1).

5.2 Das Datenbanksystem

Der Gatekeeper verwendet ein Datenbanksystem, um die Zugangskontrolldaten zu speichern. Für diesen Zweck wurde MySQL gewählt, da es sich hierbei um ein ausreichend leistungsfähiges, und darüber hinaus frei verfügbares, System handelt, das auf vielen Betriebssystemplattformen lauffähig ist.

Es folgt nun die Definition der Tabellen der Datenbank.

5.2.1 Bekannte Endpunkte/Netze - endpoints

```
CREATE TABLE endpoints (  
  ip    char(16) DEFAULT '' NOT NULL,  
  name  char(64),  
  mask  tinyint(3) unsigned,  
  plan  mediumint(8) unsigned,  
  PRIMARY KEY (ip)  
);
```

- ip - 16 Zeichen für den String, der die IP-Adresse enthält.

- name - ein Bezeichner für das Netz oder ein Hostname
- mask - Eine Zahl im Bereich 0-3¹, die angibt, wie das Feld *ip* zu interpretieren ist.
- plan - Eine 32-Bit Zahl, die als 24 Bit große Bitmaske interpretiert wird und angibt, ob zu einer bestimmten Stunde das Telefonieren von der definierten IP erlaubt ist oder nicht.

Einträge in dieser Tabelle können eindeutig anhand der IP-Adresse aufgefunden werden.

5.2.2 Kostenlose Adressen - freenumbers

```
CREATE TABLE freenumbers (
  address char(255) DEFAULT '' NOT NULL,
  PRIMARY KEY (address)
);
```

Das Feld address enthält Platz für eine H.323-Adresse, die umsonst bzw. als Notrufadresse immer angerufen werden können. Gegenwärtig wird diese Tabelle nicht genutzt, sondern ist für zukünftige Erweiterungen gedacht.

5.2.3 Funktionsadressen - functions

```
CREATE TABLE functions (
  name varchar(16) DEFAULT '' NOT NULL,
  alias varchar(255) DEFAULT '' NOT NULL,
  PRIMARY KEY (name)
);
```

- name - Bezeichner der Funktionsadresse, wie z.B. „Sekretariat“.
- alias - String, der die Namen der Gruppenmitglieder enthält. Der String enthält durch Komma getrennte Einträge der Form „typ,wert“, so daß ein Beispieleintrag „1,eilert,1,bunti,1,crunchy,1,dmeyer“ lauten könnte.

5.2.4 Gruppendefinitionen - privdef

```
CREATE TABLE privdef (
  id          tinyint(3) unsigned DEFAULT '0' NOT NULL,
  name       varchar(40) DEFAULT '' NOT NULL,
  initCred   smallint(5) unsigned,
  minCred    smallint(6),
  maxBW      int(11),
  maxCalls   smallint(6),
  spaceOnAM  smallint(6),
  PRIMARY KEY (id)
);
```

- id - Eine Identifikationsnummer, die die Gruppe eindeutig identifiziert.

¹tinyint nimmt Zahlen von -128 bis 127 auf.

- name - Bezeichner für die Gruppe
- initCred - Zu Beginn eines Abrechnungszeitraumes gewährte Freieinheiten bzw. gewährter Freibetrag.
- minCred - Betrag an Einheiten, der nicht unterschritten werden darf.
- maxBW - maximale Bandbreite, die Mitgliedern der Gruppe zugestanden wird.
- maxCalls - maximale Anzahl an Anrufen, die pro Abrechnungszeitraum getätigt werden dürfen. Dieses Feld ist gegenwärtig lediglich für zukünftige Erweiterungen definiert.
- spaceOnAM - Platz an Megabytes, der einem Mitglied der Gruppe für den Anrufbeantworter zur Verfügung steht. Auch dies Feld wird gegenwärtig nicht verwendet, da noch kein Anrufbeantworter zur Verfügung steht.

5.2.5 Nutzerdaten - user

```
CREATE TABLE user (
  h323      varchar(255) DEFAULT '' NOT NULL,
  e164      varchar(128) DEFAULT '' NOT NULL,
  name      varchar(40) DEFAULT '' NOT NULL,
  address   varchar(64),
  email     varchar(40) DEFAULT '' NOT NULL,
  account   mediumint(9),
  cpl       text,
  grp       tinyint(3) unsigned,
  state     tinyint(3) unsigned,
  PRIMARY KEY (h323)
);
```

- h323 - H.323-Adresse des Nutzers, anhand derer der Nutzer eindeutig identifiziert werden kann.
- e164 - Angabe einer Telefonnummer, unter der der Nutzer evtl. auch zu erreichen ist.
- name - Voller Name des Nutzers.
- address - Adresse, an die Rechnungen geschickt werden sollen.
- email - EMailadresse des Nutzers.
- account - Gegenwärtiger Stand an Freieinheiten verbrauchten bzw. in Rechnung zu stellenden Einheiten. Der Wert kann im Bereich -32768 bis 32767 liegen.
- cpl - 64k Platz für ein CPL-Skript.
- grp - Gruppe, der der Nutzer angehört.
- state - Gibt an, ob der Nutzer gesperrt ist, oder nicht. Ein Wert von 0 steht für *nicht gesperrt*, eine 1 für *gesperrt*. Andere Werte sind nicht definiert und können bei Bedarf noch definiert werden.

5.3 Das H323-Modul

Das H323-Modul erledigt einen Großteil der Aufgaben des Gatekeepers. Es verwaltet registrierte Benutzer und Endpunkte, sorgt für eine korrekte Kommunikation auf dem RAS-Kanal und zieht ggf. andere Module heran.

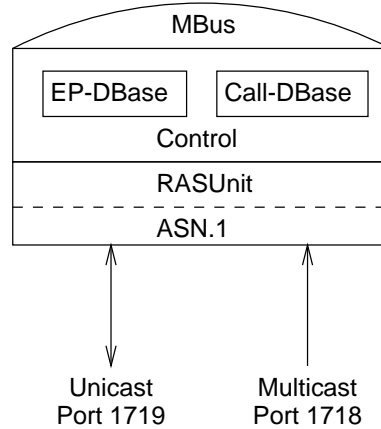


Abbildung 5.1: Interner Aufbau des H323-Moduls

Das Modul ist intern, wie Abbildung 5.1 zu entnehmen ist, in mehrere Schichten unterteilt. Auf der untersten Ebene steht die Umwandlung der ASN.1-kodierten Daten in JAVA-Datenstrukturen und umgekehrt. Hierfür wurde eine Bibliothek verwendet, die von einem, von Boris Nikolaus (*bn@tellique.de*) entwickelten, Konverter aus den ASN.1-Definitionen erzeugt wurde. Diese Bibliothek hält für jede ASN.1-Datenstruktur eine JAVA-Datenstruktur bereit und bietet Mechanismen zum Wandeln zwischen JAVA und ASN.1.

Die unterste Schicht sorgt für die Konvertierung von ASN.1-Datenstrukturen nach Java und zurück. Gemessen an der Menge an Code-Zeilen macht es etwas mehr als 2/3 der Arbeit aus, jedoch ist der größte Teil davon automatisch aus einem ASN.1-File generiert worden. Ich möchte an dieser Stelle Boris Nikolaus für die Bereitstellung der Klassen danken.

Die ASN.1-Schicht arbeitet eng zusammen mit der Schicht, die dafür sorgt, daß die RAS-PDUs mit den richtigen Daten gefüllt und wieder ausgelesen werden. In dieser Schicht ist eine genaue Kenntnis der ASN.1-Strukturen und deren JAVA-Äquivalente vorhanden.

Die RAS-Schicht trifft einige wenige der in Abschnitt 4.3 angesprochenen Entscheidungen, wie z.B. das Überprüfen der Protokollversion, bereits selbst. Andere Entscheidungen, wie z.B. ob ein Benutzer sich anmelden darf oder nicht, werden der darüberliegenden Kontrollschicht überlassen.

Die Kontrollschicht verwaltet den Zustand der registrierten Endpunkte und aktuellen Anrufe und ist somit in der Lage, weitere Dienste anzubieten, wie z.B. den, eine Aliasadresse eines registrierten Endpunktes aufzulösen.

Wenn die Kontrollschicht weiß, daß einige Informationen von dritten benötigt bzw. Entscheidungen von dritten getroffen werden müssen, reicht sie dies an den Mbus weiter. Die Schnittstelle zwischen den beiden Schichten ist so abstrakt gehalten, daß die Kontrollschicht nicht wissen muß, daß sie ihre Informationen

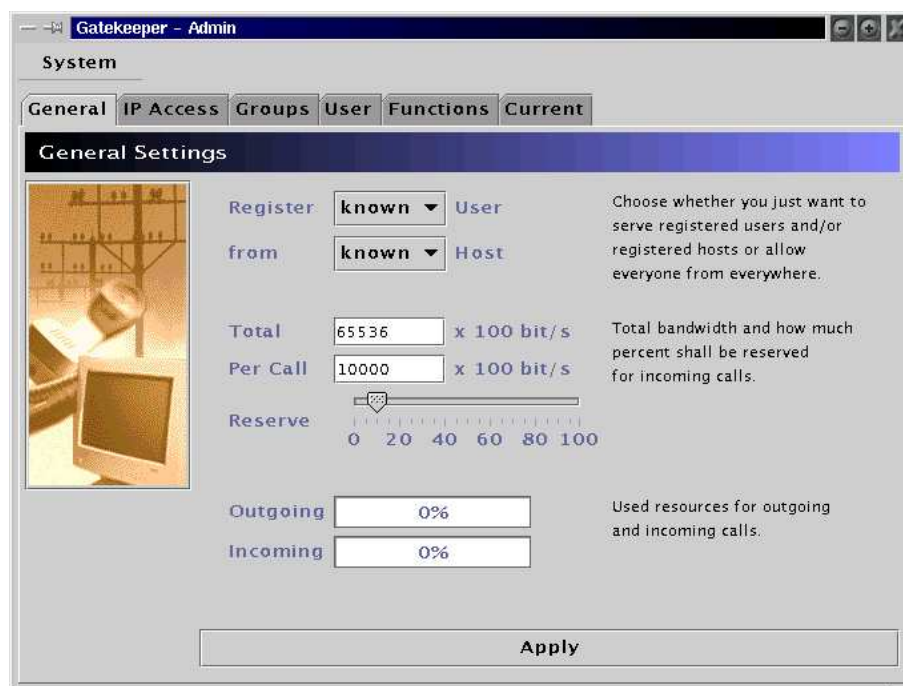
über den MBus bezieht.

5.4 Die Benutzungsschnittstelle

Die Benutzungsschnittstelle erlaubt eine nahezu umfassende Konfiguration des Gatekeepers. Sie wird als eigenständiges Programm auf dem Rechner, auf dem auch der Gatekeeper läuft, gestartet und verbindet sich mit diesem dann über den MBus.

Über verschiedene „Karten“ können Einstellungen zum Verhalten des Gatekeepers, sowie die Verwaltung der Zugangskontrolle und der Benutzer vorgenommen werden.

5.4.1 Allgemeine Einstellungen



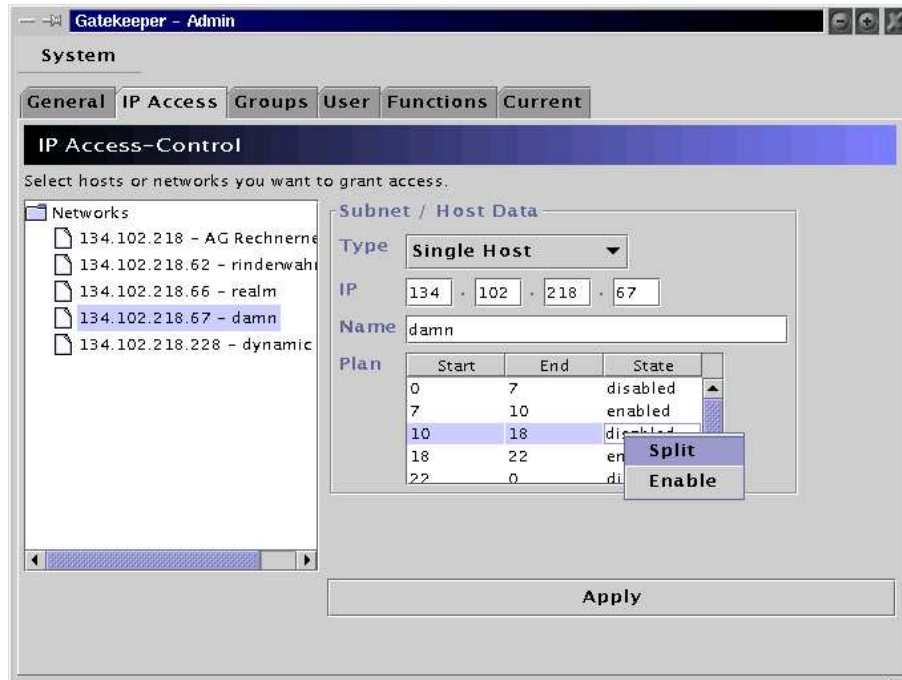
Über diese Karte wird das generelle Verhalten des Gatekeepers in Bezug auf Zugangskontrolle und Ressourcenverwaltung definiert. Es lässt sich einstellen,

- ob der Gatekeeper nur ihm bekannte Nutzer registrieren soll oder ob jeder Nutzer diesen Dienst nutzen kann.
- ob nur Nutzer von bekannten IP-Adressen sich beim Gatekeeper registrieren können, oder dies von überall möglich ist.
- wieviel Bandbreite dem Gatekeeper als Ressource insgesamt zur Verfügung steht.
- was die maximale Bandbreite pro Call sein kann.

- wieviel Prozent der Gesamtbandbreite für eingehende Anrufe reserviert werden sollen.

Daneben wird die gegenwärtige Auslastung der Ressourcen für abgehende und eingehende Anrufe angezeigt.

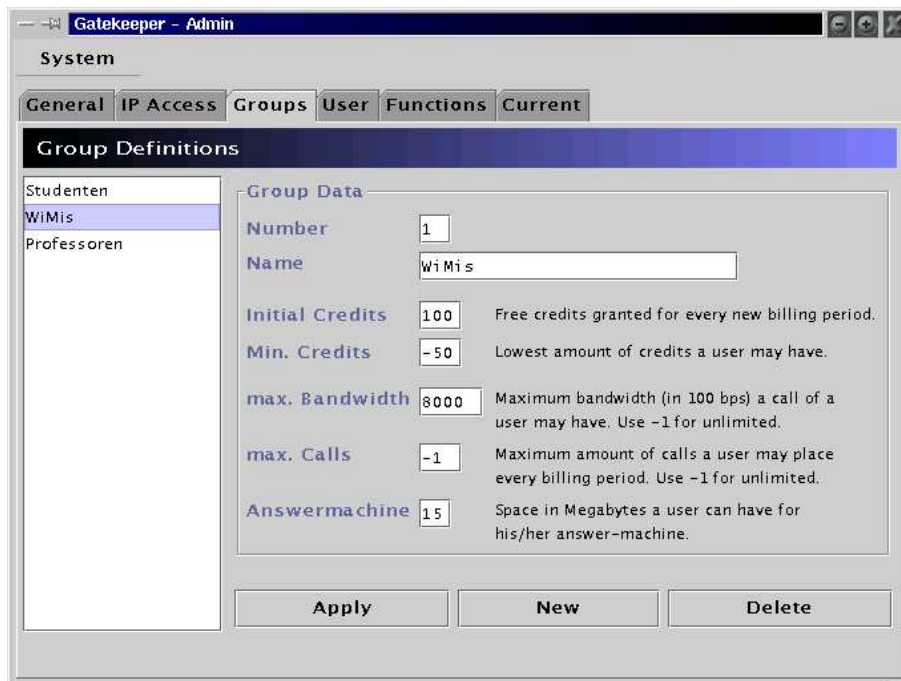
5.4.2 IP-basierte Zugangskontrolle



In dieser Karte kann eingestellt werden, welche Endpunkte zu welcher Tageszeit zugelassen werden. Dabei können Definitionen sowohl für einzelne IP-Adressen, wie auch für Subnetz-ähnliche Bereiche erstellt werden. Die Art dieser Definition ist abhängig davon, ob man „Class A-Network“ bis „Class C-Network“ oder „Single Host“ gewählt hat. Dabei ist es nicht wichtig, ob es sich wirklich um ein Netz von diesem Typ handelt, sondern es ist lediglich eine Aussage darüber, ob nur das erste, die ersten zwei, drei oder alle Bytes des IP-Adresse zur Kontrolle herangezogen werden.

Ein so definierter Endpunkt oder IP-Bereich, „Plan“ genannt, kann benannt und mit einem Zeitschema versehen werden. Ein Zeitschema definiert, ob zu einer Tageszeit (in Stunden), der Zugang erlaubt, oder verboten ist. Die Einstellung gilt für alle Tage, d.h. es sind keine Datums- oder Wochentagsspezifischen Einstellungen möglich.

5.4.3 Gruppenverwaltung

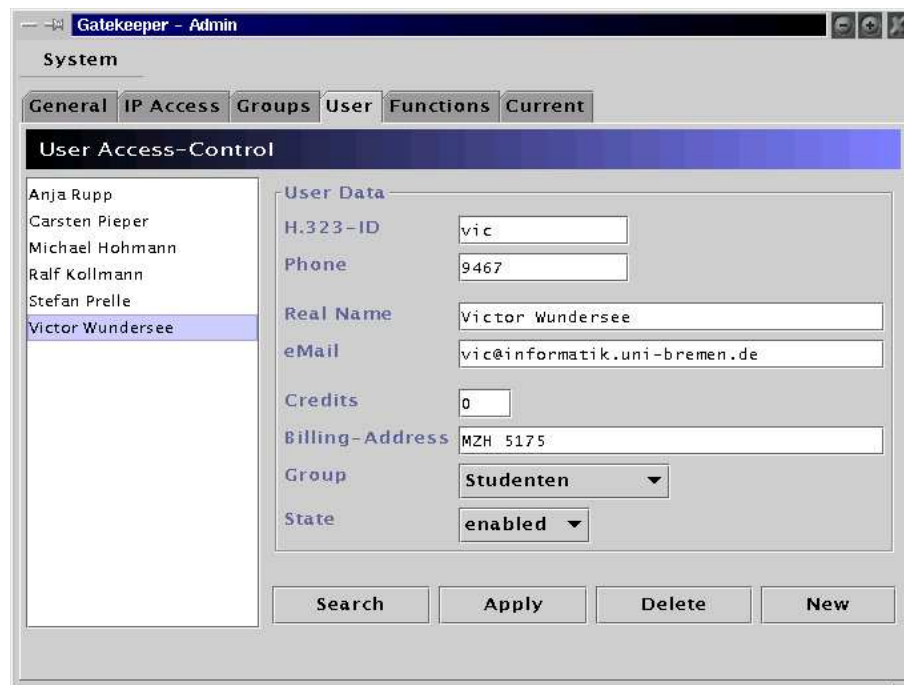


Die Gruppenverwaltung erlaubt das Anlegen, Ändern und Löschen von Gruppendefinitionen. Beim Start des Administrationsprogrammes werden die bekannten Gruppen ermittelt und hier angezeigt. Die Einstellmöglichkeiten pro Gruppe umfassen:

- ID der Gruppe - Ein positiver Integer, der die Gruppe eindeutig kennzeichnet.
- Der Name der Gruppe
- Anzahl der anfänglichen Freieinheiten pro Abrechnungszeitraum für jedes Mitglied.
- Minimum an Einheiten, damit das Telefonieren noch erlaubt ist. Hier sind auch negative Zahlen erlaubt, um z.B. anzudeuten, daß auch nachdem die Freieinheiten verbraucht wurden noch weitertelefoniert werden kann - in dem Fall dann allerdings auf Rechnung. Die Einstellung gilt für jedes Gruppenmitglied extra, d.h. es gibt kein Gruppenkonto. Dies gilt auch für die folgenden Parameter.
- Maximale Bandbreite
- Maximale Anrufe pro Abrechnungszeitraum
- Platz in MB auf dem Anrufbeantworter

Die Größe eines Abrechnungszeitraumes wird übrigens im Rahmen dieser Arbeit nicht definiert, da sie Inhalt eines noch zu schreibenden *Billing&Account*-Moduls wäre.

5.4.4 Benutzerverwaltung



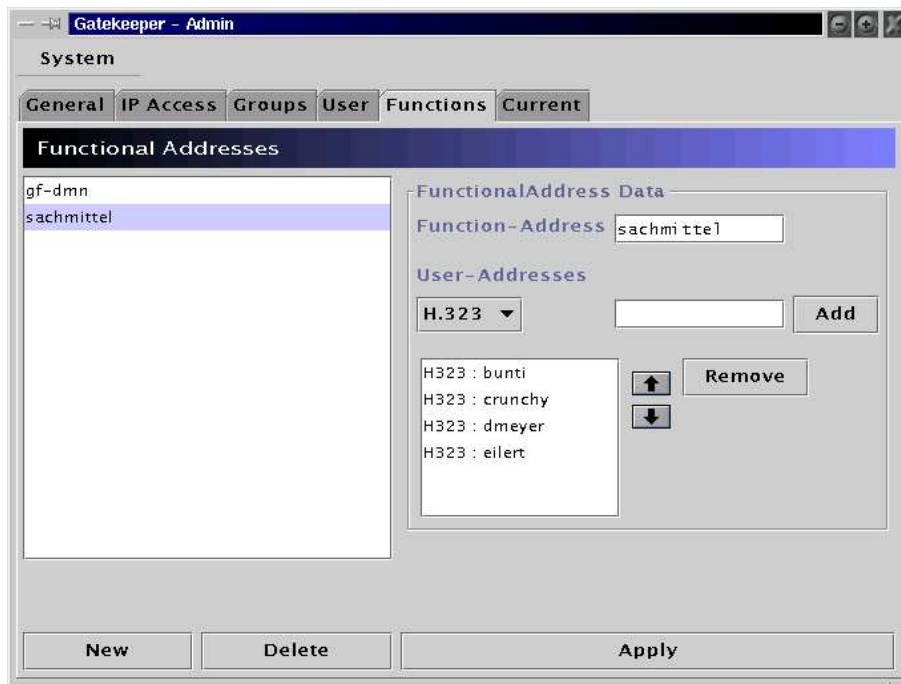
In der Userverwaltung können neue Benutzer angelegt und bestehende gelöscht oder geändert werden. Um die auf dem MBus zu übertragene Datenmenge zu begrenzen, werden nur solche Nutzerdatensätze angezeigt, die explizit angefordert wurden.

Um die Daten eines Nutzers zu ändern, muß zunächst irgendeine Angabe über den Benutzer gemacht werden — am besten die H.323-ID, da diese eindeutig ist — und anschließend mittels des *Search*-Button die Suche angestoßen werden. Auf der linken Seite werden nun die in Frage kommenden Benutzer angezeigt. Wird einer von denen ausgewählt, werden die entsprechenden Daten in der Maske angezeigt, wo sie verändert werden können und mit dem *Apply*-Button übernommen werden.

Ähnlich verhält es sich beim Löschen eines Nutzers. Wieder müssen Suchdaten eingegeben und anschließend der entsprechende Benutzer ausgewählt werden. Hier ist jedoch der *Delete*-Button zu betätigen.

Ein neuer Nutzer wird anlegt, indem mit dem *New*-Button alle Daten der Maske gelöscht, anschließend die neuen Daten eingegen werden. Durch Drücken von *Apply* erzeugt man dann den neuen Datensatz.

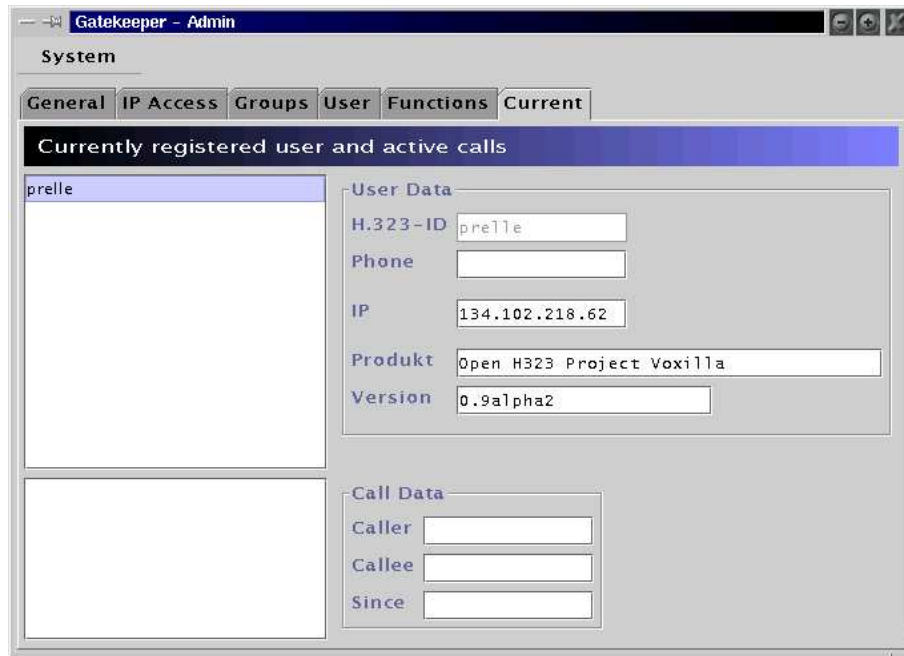
5.4.5 Funktionsadressen-Verwaltung



Bei Start des GUIs stehen alle bekannten Funktionsadressen in der Liste auf der linken Seite der Karte. Wird eine davon ausgewählt, wird sie in der Maske angezeigt und kann modifiziert bzw. gelöscht werden.

Analog zu den übrigen Masken werden Änderungen in der Maske mit *Apply* übernommen, bzw. mit *Delete* die Adressen gelöscht. Um eine neue Adresse einzugeben, wird mit *New* eine leere Maske erzeugt.

5.4.6 Registrierte Benutzer und aktive Gespräche



Diese Karte dient, wie schon die Ressourcen-Anzeige in *General Settings* reinen Informationszwecken. Zu Start des Programmes wird die Liste der gegenwärtig registrierten Benutzer und aktiven Telefonate angezeigt. Diese Listen werden selbständig aktuell gehalten.

Wird ein registrierter Endpunkt ausgewählt, so werden einige Informationen über den Endpunkt auf der rechten Seite der Karte angezeigt. Diese Informationen umfassen eine H.323-ID und eine evtl. Telefonnummer des Endpunktes, die IP-Adresse des Benutzers und Produkt und Version seines Endpunktes. Die Auswahl eines aktiven Gesprächs zeigt Anrufer und Angerufenen des Gesprächs, sowie seinen Startzeitpunkt an.

Es handelt sich hierbei natürlich um sensitive Daten, da sie es erlauben, das Telefonverhalten von Benutzern zu beobachten. Andererseits soll die Benutzungsschnittstelle nur von Systemadministratoren, d.h. einem begrenztem Kreis von Personen, die auch bisher schon Zugriff zu allen sensitiven Informationen hatten, benutzt werden. Zudem ist eine automatische Auswertung der Daten mit dem Benutzungsschnittstelle nicht möglich. Aus diesen Gründen sind Schutzmechanismen zur Wahrung des Datenschutzes bisher nicht vorgesehen.

5.5 Der virtuelle MBus

Bei Betrachtung des Datenflusses zwischen den MBus-Modulen fällt auf, daß Anfragen vom H323-Modul an die Policy-Module stets eine Anfrage der Policy-Module an das Datenbank-Modul mit sich bringen. Da die MBus-Kommunikation über Sockets verläuft, bedeutet dies, daß die Daten im Kernel durch die verschiedenen Schichten geschleust werden müssen, was einen nicht zu vernachlässigenden Aufwand mit sich bringt.

Um diesen durch den MBus herbeigerufenen Nachteil zu beheben, ohne den gleichzeitig errungenen Vorteil der flexiblen Architektur zu verlieren, wurde eine Lösung erarbeitet, die es den bereits fertigen Modulen erlaubt, direkt durch Funktionsaufrufe miteinander zu kommunizieren, und zugleich bei Bedarf mit externen MBus-Modulen redet: Der virtuelle MBus.

Bevor auf das spezielle Konzept des virtuellen MBus eingegangen wird, muß zunächst noch mal auf den normalen Aufbau von MBus-Applikationen, wie er im Rahmen dieser Arbeit verwendet wird, erläutert werden. Die Abbildung 5.2 beschreibt die zwei Methoden, wie eine Applikation auf den MBus zugreifen kann.

Der direkte Weg (a) ist es, mittels des *SimpleTransportLayer* (STL) auf den

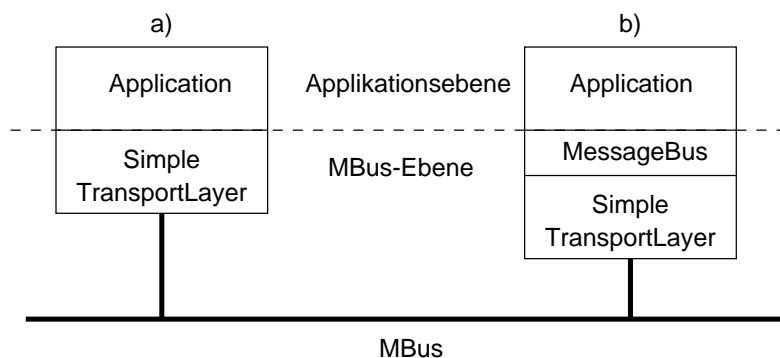


Abbildung 5.2: Generelle Aufbau von MBus-Modulen

MBus zuzugreifen. Der *SimpleTransportLayer* ist eine Implementierung des *Message Bus*, wie er im Draft [25] definiert ist. Dies bedeutet, daß nur asynchrone Kommunikation zur Verfügung steht und die Applikationen selber dafür Sorge tragen müssen, unterstützte Kommandos aus den empfangenen Daten herauszufiltern.

Im zweiten Fall (b) wird noch eine Zwischenschicht verwendet, die synchrone Kommunikation ermöglicht und die man mit den Kommandos konfiguriert, die man gerne senden und empfangen können würde. Diese Konfiguration ist, wie auch das Konzept des virtuellen MBus, eine besondere Eigenschaft der im Rahmen dieser Arbeit entstandenen JAVA-Implementierung des MBus.

Jedes MBus-Modul der Diplomarbeit greift wie im Fall b) auf den MBus zu, da meist die synchrone Kommunikation genutzt wird.

Der virtuelle MBus, oder auch *VirtualTransportLayer*, ist jetzt eine weitere Schicht zwischen dem *SimpleTransportLayer* und dem *MessageBus*, die dafür

Kapitel 6

Verwendung des Gatekeepers

6.1 Systemanforderungen

Um den Gatekeeper laufen zu lassen benötigt man:

- **JAVA 2 (bzw. 1.2) oder höher**
Der Gatekeeper ist dank Java nicht plattformabhängig, d.h. er sollte auf allen Hardware- und Softwareplattformen laufen, die von Java unterstützt werden. Voraussetzung ist, daß die Java-Version 2.0 (auch bekannt als 1.2) installiert ist.
- **MySQL**
Für die Verwaltung der Host-, Gruppen- und Nutzerdaten verwendet der Gatekeeper ein SQL-Datenbanksystem. Es wird *MySQL* empfohlen, da anhand dieses Systems entwickelt wurde. Abschnitt 6.4 befaßt sich mit der Verwendung anderer Systeme.
- **„Hinreichend schneller Rechner“**
Aufgrund der Verwendung einer Interpretersprache und starker Parallelität der Prozesse ist die Rechengeschwindigkeit des verwendeten Systems von Bedeutung. Allerdings haben im Rahmen dieser Diplomarbeit keine Tests auf unterschiedlichen Systemen stattgefunden, so daß in diesem Fall keine Empfehlung ausgesprochen werden kann. Der einzige Anhaltspunkt ist, daß es auf einem *Pentium II* 400 Mhz ausreichend schnell war, sofern der Rechner keine weiteren Aufgaben erfüllte.

6.2 Installation

Die Installation des Gatekeeper erfolgt in wenigen Schritten:

1. Packen Sie die Archiv-Datei in ein Verzeichnis Ihrer Wahl aus.
2. Legen Sie sich eine *.mbus*-Datei in ihrem Homeverzeichnis an. In dem soeben entpackten Archiv findet sich eine Beispieldatei, die Sie verwenden

können.

Wenn Sie nicht wissen, welches Ihr Homeverzeichnis ist, überspringen Sie diesen Punkt. Beim ersten Start des Gatekeepers wird dieser eine Meldung "Could not find a config-file at *xyz*" ausgeben. *xyz* ist dabei der Pfad zu der Stelle, an der sich die *.mbus*-Datei befinden sollte.

3. Wechseln Sie in das Verzeichnis, in das Sie das Archiv ausgepackt haben und führen Sie *startG* aus (auf MS-Windows-Systemen *startG.bat*). Der Gatekeeper wird jetzt gestartet.

6.3 Fehlerbehebung

Dieser Abschnitt soll einige der möglichen Fehlerquellen auflisten und beschreiben, wie diese Fehler zu beheben sind.

- **org.mbus.MBusException: Missing Portnumber in MBus- Configuration**
In Ihrer Konfiguration fehlt die Angabe der Portnummer (bzw. ist falsch geschrieben), oder die Konfigurationsdatei des Message Bus fehlt komplett. Letzteres ist der Fall, wenn kurz zuvor die Meldung **Could not find a config-file at x** angezeigt wurde.
Überprüfen Sie die Konfiguration des Message Bus in der *.mbus*-Datei bzw. legen Sie eine solche Konfiguration an der geforderten Stelle an.
- **Exception in thread „main“ java.lang.IllegalArgumentException: Could not find host 'x':**
Der Gatekeeper kann den Namen des Datenbank-Servers nicht auflösen. Überprüfen Sie den entsprechenden Eintrag in der *gatekeeper.properties*-Datei. Ein Beispiel findet sich im Anhang A.1.
- **java.sql.SQLException: Can't connect to database 'gatekeeper' on server 'damn', port 3306.**
Auf dem angegebenen Rechner läuft kein Datenbank-System, oder die Portnummer für die Datenbank stimmt nicht.
Starten Sie einen Datenbank-Server auf dem Rechner bzw. korrigieren Sie die Portnummer in der *gatekeeper.properties*-Datei.
- **Exception in thread „main“ java.lang.IllegalArgumentException: SQL-Driver not found: java.lang.ClassNotFoundException: xyz**
Der in der *gatekeeper.properties* angegebene Treiber für den Datenbank-Zugriff *xyz* kann nicht instantiiert werden, da sich die entsprechenden Klassen nicht im Java CLASSPATH befinden.

6.4 Andere Datenbanken

Der Gatekeeper wurde entwickelt und getestet mit dem MySQL-Datenbanksystem. Nichtsdestotrotz sollte auch die Verwendung anderer SQL-Datenbanken möglich sein. Hierfür ist lediglich eine Implementierung des JDBC-Treiber für den Zugriff auf das neue System und eine Änderung der Konfigurationsdatei

nötig.

Eine Übersicht über verfügbare JDBC-Treiber gibt es auf den Webseiten von Sun: <http://java.sun.com/products/jdbc/drivers.html>.

Wird z.B. der JDBC-Treiber von *Imaginary* verwendet, so sollte das entsprechende JAR-File, bzw. das Verzeichnis mit den Klassen mit in den *CLASSPATH* aufgenommen werden. Anschließend muß noch in der *gatekeeper.properties*-Datei der Eintrag `JDBCProvider` auf `com.imaginary.sql.mssql.MssqlDriver` gesetzt werden.

Kapitel 7

Zusammenfassung und Ausblick

Die Beschäftigung mit IP-Telefonie macht deutlich, daß es gegenwärtig noch viele offene Fragen gibt. Die meisten dieser Fragen resultieren aus der Absicht, IP-Telefonie zu bestehenden Diensten und Netzen kompatibel zu machen. Der Großteil der Bemühungen geht daher bisher in die Richtung, IP-Telefonie dahin zu bringen, daß es bestehende Telefonie-Dienste ohne Einbußen der Funktionalität ersetzen kann. Bisher existieren nur wenige konkrete Überlegungen zu vollkommen neuen Diensten, wie z.B. ein globaler Kommunikationsbezeichner für eine Person oder das Auffinden von Personen und Durchstellen eines Anrufes zum nächstgelegenen Telefon. Allerdings ist bei solchen Fragen auch die Politik gefordert, internationale Einigungen zu erzielen - was deutlich langwieriger sein dürfte, als die reine Entwicklung der Technik.

Der in dieser Arbeit entstandene Gatekeeper ist lediglich ein Grundgerüst für IP-Telefonie nach H.323. Viele sinnvolle Funktionalität ist noch nicht enthalten, sollte aber aufgrund des sehr offenen Konzeptes leicht integrierbar sein. Abgesehen davon zeigt sich, daß auch die Arbeit entlang bereits gefestigter Standards, nicht ganz unproblematisch ist. Aus diesem Grund sollen kurz einige unerwartete Probleme aufgeführt werden, die im Laufe der Arbeit auftraten. Im Anschluß sollen die naheliegendsten Erweiterungen des Gatekeepers zusammengefaßt werden, bevor ein kurzer Ausblick auf die Entwicklung der IP-Telefonie diese Arbeit abschließt.

7.1 Performance

Die Wahl von JAVA als Programmiersprache führte sicherlich zu einem „saubereren“ Programmierstil, da es das objektorientierte Programmieren viel eher fördert als z.B. C++ und durch das Vorhandensein eines Garbage Collectors Probleme wie Speicherlecks gar nicht erst auftreten ließ. Nicht zu vernachlässigen war auch die hohe Entwicklungsgeschwindigkeit durch kurze Compile-Zeiten und vereinfachte Fehlersuche.

Die Kehrseite der Medaille war jedoch eine geringere Ausführungsgeschwindigkeit, die sich stärker auswirkte, als erwartet. Als Interpreter-Sprache ist JAVA

zwangsweise langsamer als ein plattformabhängiges Compilat — Sun Microsystems nennt einen ungefähren Faktor von 10. Dieser Faktor wird reduziert durch die Verwendung eines Just-In-Time-Compilers, jedoch war das Ergebnis, die Erzeugung von Native-Code zur Laufzeit, nicht unbedingt spürbar.

Die geringe Performance und damit das schlechte Antwortzeitverhalten führten zu ernsteren Problemen als erwartet — jedoch war nicht nur Java die Ursache. Im Laufe der Entwicklung wurden Geschwindigkeitsmessungen für einige Aufgaben vorgenommen und dabei festgestellt, daß das Antwortzeitverhalten auf eingehende PDUs von dem Grad der MBus-Aktivität abhing, die sie erzeugten — weswegen der *virtuelle MBus* (siehe 5.5) ins Leben gerufen wurde. Trotzdem blieb der Effekt, daß manchmal die Antwortzeit statt normaler 400 ms für eine Anfrage 2000 ms betrug — ein Effekt, der in der Regel nicht reproduzierbar war, aber teilweise dazu führte, daß Timeouts nicht eingehalten wurden.

Dieses Problem wurde bisher nicht abschließend gelöst.

7.2 Interoperabilität

Während der Entwicklung war *Microsoft Netmeeting 3.0* der einzige H.323-fähige Endpunkt, der zur Verfügung stand. Leider verfolgt *Netmeeting* eine bestenfalls eigenwillig zu nennende Interpretation des H.323-Standards, was an manchen Stellen nicht ins Gewicht fiel, aber an anderen Stellen doch die Arbeit beeinträchtigte. Auffällig war z.B. das *Netmeeting* einige zwingend erforderliche Felder der versendeten PDUs nicht ausfüllte, das *Netmeeting* keine Gatekeeper-Discovery durchführt oder sich anscheinend nicht um Bandbreitenkontrolle kümmert. Gerade der letzte Punkt führte dazu, daß kein sinnvolles Testen der Ressourcenverwaltung möglich war. Entstanden ist daher wahrscheinlich ein Stück Software, daß H.323-konform mit *Netmeeting*-Zugeständnissen ist.¹

Generell fehlte eine Testphase mit anderen H.323-konformen Produkten. So wäre es interessant gewesen, wie sich andere Gatekeeper, Endpunkte oder Gateways in Zusammenarbeit mit dem entwickelten Gatekeeper verhalten. Nach den Überraschungen mit *Netmeeting* erwarte ich hier noch weitere Komplikationen. Zudem wäre es sicherlich sinnvoll, den Gatekeeper einem Langzeittest zu unterziehen, was aber im Rahmen der sechs Monate der Diplomarbeit absolut unmöglich war.

7.3 Naheliegende Erweiterungen

Mit der Fertigstellung der Diplomarbeit ist die Arbeit am Gatekeeper noch lange nicht beendet. Zwar reicht seine Funktionalität aus, um IP-Telefonie-Unterstützung für eine Institution wie die Universität anzubieten, jedoch fehlt noch einige Funktionalität, die dem Gatekeeper weitere Einsatzfelder erschließen würde, die aber auf Grund der Zeit nicht zu realisieren waren. Einige dieser Funktionalitäten (vergl. 4.2), wie z.B. ein Abrechnungssystem, wurden bereits explizit vorbereitet, andere wiederum sollten sich dank der Struktur des Message Bus leicht hinzufügen lassen.

¹Und ich kann nicht behaupten, daß ich besonders glücklich damit bin.

Billing & Accounting

Zu den explizit vorbereiteten Erweiterungen gehört ein *Billing & Accounting*-Modul, welches die Protokollierung und Abrechnungen für Gespräche ermöglicht. Es wird spätestens dann nötig, sobald die Verwendung von Gateways ins normale Telefonnetz ins Spiel kommt, da hier Kosten anfallen. In den Nutzerdatensätzen und in der Benutzungsschnittstelle des Gatekeepers wurde daher bereits ein Einheiten-Konto vorgesehen.

Weiterhin unterstützt wird auch ein Modul für externe User-Location, da die vom H.323-Modul nicht auflösbaren Adressen an den MBus durchgereicht werden.

Remote Access

Indirekt vorbereitet wurden Methoden der Fernsteuerung des Gatekeepers, wie z.B. über entfernte Prozeduraufrufe oder SNMP. Indirekt bedeutet in diesem Fall, daß solche Module sinnvollerweise das bereits implementierte API verwenden und sich somit keine Gedanken über den internen Aufbau und den MBus machen müssen und sich aus der Menge der vordefinierten Methoden bedienen können.

Load Balancing und Verfügbarkeit

Ein Punkt, auf den bisher überhaupt nicht eingegangen wurde, ist die Verfügbarkeit des Gatekeepers, die z.B. durch einen totalen Ausfall des Systems oder durch eine Überlastung beeinträchtigt wird. Da das Telefon in der Regel aber ein sehr essentielles Kommunikationsmittel ist, sollte es — gerade für Problemfälle — möglichst robust sein.

Das Problem ist nicht neu, jedoch gibt es bezüglich eines H.323-Systems bisher nur eine Diskussion über Lösungen ([27],[24]), weswegen hier ein paar Ideen präsentiert werden sollen.

Um die Last eines einzelnen Gatekeepers zu verringern, bietet es sich an, mehrere Gatekeeper für eine Zone zu verwenden, die die Endpunkte unter sich aufteilen. Dazu wäre ein Inter-Gatekeeper-Protokoll nötig, das dafür sorgt, daß jeder Gatekeeper in einer Gruppe die grundlegenden Daten der anderen Gatekeeper kennt, d.h. daß er weiß, welche Endpunkte mit welchen Nutzern bei welchem Gatekeeper registriert sind und ggf. weitere Informationen über die Auslastung eines anderen zoneninternen Gatekeepers mit übermittelt.

Aus diesen Informationen wären dann die Gatekeeper in der Lage zu bestimmen, bei welchen Gatekeepern der Gruppe ein neuer Endpunkt sich anmelden darf. Für diese Entscheidung ist nicht zwingend ein Master-Slave-System nötig, in der ein Master-Gatekeeper die Entscheidung fällt, da alle Gatekeeper auf der gleichen Datenbasis arbeiten und, ein definiertes Entscheidungsverfahren vorausgesetzt - somit identische Entscheidungen treffen sollten.

Interessant wird es, wenn man den Fall bedenkt, in dem ein oder mehrere Gatekeeper ausfallen. Es läge dann an den übrigen Gatekeepern, die Endpunkte, die vom Ausfall „ihrer“ Gatekeeper betroffen sind, unter sich zu verteilen und dies auch mit den Endpunkten auszuhandeln. Hierbei ist insbesondere die Frage

zu stellen, ob die Gatekeeper der Gruppe alle denselben *Gatekeeper-Identifizier* verwenden, also vorgeben ein einzelner Gatekeeper mit mehreren Transportadressen zu sein, oder ob sie unterschiedliche *Gatekeeper-Identifizier* verwenden. Da H.323 es aber erlaubt, einem Gatekeeper mehrere Transportadressen zuzuordnen, ist der erste Fall (d.h. die Gruppe tritt als ein Gatekeeper mit mehreren Adressen auf) sinnvoller.

Denkt man diesen Gedanken weiter, so kommt man zu dem Schluß, daß es sinnvoller ist, wenn alle Gatekeeper eine exakt identische Datenbasis haben und auch die Zustände der einzelnen Endpunkte untereinander bekanntmachen. Diese Form der *Active Replication* würde dazu führen, daß das Wiederaufsetzen nach dem Ausfall eines Gatekeepers minimal wäre — lediglich Anrufe nach dem *Gatekeeper-routed*-Anrufmodell müßten neu aufgesetzt werden, da anzunehmen ist, daß durch den Ausfall des Gatekeepers auch die über den Gatekeeper gerouteten Verbindungen unterbrochen wurden.

Die Voraussetzung für ein solches System ist jedoch, daß ein Endpunkt in der Lage ist, mehrere RAS-Transportadressen zu nutzen, d.h. daß er auf eine andere ausweicht, falls eine nicht mehr reagiert.

7.4 Ausblick

Die Entwicklung und Verbreitung der IP Telefonie sorgt dafür, daß die Grenzen zwischen Daten- und Sprachkommunikation zunehmend fließend werden. Die Frage, ob es H.323 oder SIP sein wird, welches die Zukunft der IP Telefonie bestimmt, wird sich wahrscheinlich bald nicht mehr stellen, da Produkte in der Entwicklung sind, die sowohl SIP, H.323 und das PSTN-Netz miteinander verbinden können.

Die Verwendung von IP-Telefonie wird zuerst auf den Intranetzen größerer Organisationen erfolgen. Bereits jetzt sind Produkte vorhanden, die dies ermöglichen, doch werden bestehende Telefonanlagen nicht von heute auf morgen, sondern eher nach Ablauf der langfristigen Wartungsverträge, erneuert. Schätzungen von IP-Infrastrukturherstellern zufolge, soll der Anteil des Sprachverkehrsaufkommen über IP im Jahre 2005 bereits 44% betragen.

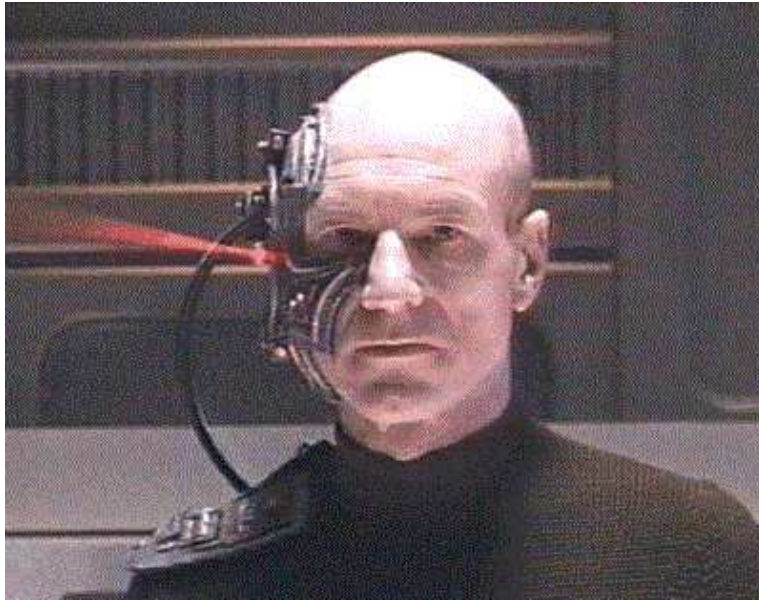
Der Anreiz zur Einführung von IP-Telefonie mag im DFN noch eher auf Grund der zugrundeliegenden Technik erfolgen — um IP-Telefonie aber flächendeckend einzuführen, darf die Funktionalität und Qualität von IP-Telefonie für den Endkunden nicht „schlechter“ als bisherige Lösungen sein, sondern sollte eher Vorteile bieten. Wahrscheinlich werden in Zukunft daher neue Dienste entstehen, mit denen zusätzlich zum Umstieg auf IP-Telefonie gelockt wird.

Etablierte Carrier stellen auch jetzt schon ihre Backbone-Netze auf IP-Technologie um, um für die Zukunft gerüstet zu sein. Diese mit neuester Technik ausgestatteten Netze erlauben es, kontrolliert Bandbreiten zu vergeben und dadurch — anders als bisher im Internet — eine Übertragungsqualität zu garantieren.

Ist diese Umstellung erfolgt, kann man auch Privatkunden in den Genuß von IP-Telefonie kommen lassen. Hier ist damit zu rechnen, daß sich das Gebührenschemata drastisch ändern wird. Da Telefonie über das Internet praktisch entfernungsunabhängig ist, werden auch die Carrier gezwungen werden, ihre Gebühren

dementsprechend umzustrukturieren, wenn sie konkurrenzfähig bleiben wollen. Denkbar wäre, daß es in Zukunft nur noch einen Grundbetrag gibt, der für einen (IP-)Telefonanschluß bezahlt werden muß, und der alle Gesprächskosten unabhängig von der Entfernung des Anrufziels bereits beinhaltet. Eventuell werden in Zukunft auch einfach alle Anrufe nur entfernungsunabhängig abgerechnet. Auf jeden Fall wird die Welt mit Einführung der IP-Telefonie enger zusammenrücken.

Die nötigen Techniken, die IP-Telefonie zum bestehenden Telefonsystem gleichwertig werden lassen, sind in Arbeit. Der Weg ist also frei - der Siegeszug der IP-Telefonie somit nur eine Frage der Zeit: *Resistance is futile!*



Anhang A

Konfigurationsdateien

A.1 System-Konfiguration

Die Konfigurationsdatei heißt `gatekeeper.properties` und sollte sich entweder im Home-Verzeichnis des Benutzers, der den Gatekeeper startet befinden, oder explizit als Option angegeben werden.

In der Datei sind die Konfigurationsdaten im `key=value`-Format angeben. Hier ein Beispiel:

```
bandwidth=65536
incomBand=10
maxSingle=10000
```

```
acceptAnyUser=0
acceptAnyHost=1
```

```
JDBCProvider=org.gjt.mm.mysql.Driver
dbHost=rinderwahnsinn
dbPort=3306
dbName=gatekeeper
```

Die Bedeutung der Schlüssel:

`bandwidth`

Die maximal verfügbare Bandbreite, die zur Verteilung zur Verfügung steht in 100 bit/s.

`incomBand`

Gibt an, wieviel Prozent der maximalen Bandbreite (`bandwidth`) für rein-kommende Anrufe reserviert werden soll.

`maxSingle`

Die maximale Bandbreite, die ein einzelner Anruf beanspruchen darf. Dies kann durch keine Einstellung für eine Gruppe übertroffen werden.

`acceptAnyUser`

Kann 0 oder 1 annehmen und beschreibt, ob nur solche Benutzer akzeptiert werden, für die es einen Nutzerdatensatz gibt (0) oder jeder beliebige Nutzer (1).

acceptAnyHost

Kann 0 oder 1 annehmen und beschreibt, ob der Gatekeeper nur für die Endpunkte da ist, die von einer ihm bekannten IP kommen (0) oder von jeder beliebigen IP (1).

JDBCProvider

Name der Klasse, die den Provider für JDBC enthält, der vom Datenbank-Modul verwendet werden soll. Siehe hierzu auch in der Java-Dokumentation zum JDBC.

dbHost

Name oder IP des Rechners, auf dem der Datenbank-Server läuft

dbPort

Portnummer des Datenbank-Servers. Für MySQL meist 3306.

dbName

Name der Datenbank, die die Tabellen der Gatekeeper-Datenbank enthält. Sollte „gatekeeper\“ lauten.

A.2 MBus-Konfiguration

Wie auch für alle anderen Systeme, die den MessageBus verwenden, muß noch dieser auch noch über eine Konfigurationsdatei eingerichtet werden. Die Datei heißt `.mbus` und liegt im Home-Verzeichnis des Benutzers, der den MBus startet. Ein beispielhafter Aufbau:

```
[MBUS]
ADDRESS=224.255.222.239
PORT=47001
ENCRYPTIONKEY=(NOENCR, )
HASHKEY=(HMAC-MD5-96, T21/U6/ORLxKF/0a)
UPI=prelle@tzi.org
CHECK_DIGEST=no
```

```
[SIP]
MODE=uas
UDP=5160
```

Anhang B

Verwendete Mbus-Nachrichten

Dieser Anhang listet die Mbus-Nachrichten auf, die die einzelnen Module verstehen bzw. selber initiieren. Die folgende Tabelle gibt zunächst eine Übersicht über alle Nachrichten. Die Kürzel F,A und K kennzeichnen dabei die unterschiedlichen Arten von Nachrichten Frage, Antwort und Kommando. Die Kürzel S und R geben an, ob ein Modul die Nachrichten senden (S) und/oder empfangen (R) kann.

Kommando	Art	H323-Modul	Zugangs-Modul	Datenbank-Modul	Bandw.-Modul	GUI-Modul	Seite
add	F	S	R		S		104
beginCall	K	S			R		101
classes	F		R				105
delete	F	S	R		S		105
endCall	K	S			R		101
error	A		R	S		R	104
getAllEPs	F	R				S	102
getConfig	F				R	S	109
getCredits	F		R				106
getEndpoint	F		R			S	106
getFunc	F		R			S	106
getGroup	F		R			S	106
getResources	F				R	S	109
keys	F		R				105
locate	F	SR					102
location	A	SR					102
lookup	F	S	R			S	105
ok	A		R	S		R	104

poll isLocalZone	F	S	R	107	
poll mayRegister	F	S	R	107	
poll mayCall	F	S	R	108	
poll.bandwidth	F	S	R	108	
register	A	S	R	101	
resources	K	S	R	109	
shutdown	K	R	S	102	
select	F	S	R	S	105
setConfig	K	RS	SR	109	
setEndpoint	F	R	S	106	
setFunc	F	R	S	107	
setGroup	F	R	S	106	
setUser	F	R	S	106	
unregister	K	S	R	101	
vote.bandwidth	A	R	S	109	

Tabelle B.1: Übersicht über alle verwendeten MBus-Kommandos

B.1 Generelle Anmerkungen

Die folgenden Datentypen sind im Draft zum Message Bus spezifiziert:

Data Type	Syntax	Description
Integer	„-[0-9]+“	See below for escape characters
Float	„-[0-9]+“.[0-9]+	
String	„“...“““	
List	(Data Type ...)	A predefined protocol value
Symbol	[A-Za-z-][A-Za-z0-9-.-]+	
Data	„<“data“>“	

Um die nachfolgend aufgeführten MBus-Kommandos übersichtlich zu halten, werden häufig verwendete Datenstrukturen, wie z.B. eine Aliasadresse, als neue zusammengesetzte Datentypen definiert. Der neue Typ `Alias` (s.u.) ist demnach eine Liste, die einen Integer und anschließend einen String enthält.

Alias

Alias - Datentyp für eine Aliasadresse		
(typ wert)		
typ	Integer	Art des Aliasnamens (0=E.164, 1=H.323)
wert	String	Aliasname

Tsap

Tsap - Datentyp für IP + Portnummer		
(ip port)		
ip	String	IP-Adresse
port	Integer	Portnummer

Vendor

Vendor - Datentyp für eine Herstellerangabe		
(country ext manu product version)		
country	Integer	T35-Ländercode (0-255)
ext	Integer	T35-Extension (0-255)
manu	Integer	Hersteller-Code (0-65535)
product	String	Produktbezeichner
version	String	Versionsbezeichner

EpType

EpType - Datentyp für einen EndpunktTyp		
(typ [vendor])		
typ	Integer	Bitmaske, die Art des Endpunktes bezeichnet (0=GK,1=GW,2=MCU,3=Terminal,4=MC)
vendor	Vendor	Optionale Angabe eines Herstellers

Endpoint

Endpoint - Datentyp für einen Endpunkt		
(aliases csAddr rasAddr type)		
aliases	List	Liste vom Typ Alias
csAddr	List	Liste vom Typ Tsap
rasAddr	List	Liste vom Typ Tsap
type	EpType	Art des Endpunktes

Call

Call - Datentyp für einen Call		
(callID crv caller callee start end)		
callID	Data	Global eindeutige Anruf-ID
crv	Integer	Lokal eindeutiger Anruf-ID
caller	Endpoint	Anrufender Endpunkt
callee	Endpoint	Angerufener Endpunkt
start	String	Begin des Anrufes in Millisekunden
end	String	Ende des Anrufes in Millisekunden

Function

Function - Datentyp für eine Funktionsadresse		
(name aliases)		
name	String	Name der Funktionsadresse
aliases	List	Liste vom Typ Alias

Group

Group - Datentyp für einen Gruppendefinition		
(id name initC minC maxBW mxCalls space)		
id	Integer	Gruppen-ID
name	String	Name der Gruppendefinition
initC	Integer	Initialer Kontostand
minC	Integer	Minimaler Kontostand zum Tel.
maxBW	Integer	Maximale Bandbreite
mxCalls	Integer	Max. Anrufe pro Zeitraum
space	Integer	Max. Platz auf Anrufbeantworter

Desweiteren wird in den gleich anschließenden Übersichten zu den einzelnen Modulen eine Trennung der Kommandos nach aktiven und reaktiven Kommandos vorgenommen.

Reaktive Kommandos erfolgen als Reaktion auf ein empfangenes Kommando. In den Tabellen ist jeweils angegeben, auf welches Kommando wie reagiert wird. Aktive Kommandos generiert ein Modul selbst — sei es auf Grund Benutzereingabe, in regelmäßigen Intervallen oder externen Gründen.

B.2 Das H323-Modul

B.2.1 Steckbrief

Adresse: app:h323stack module:engine media:h323		
Reaktion auf eingehende Kommandos		
Kommando	Antwort	
PRE1.locate	PRE1.location	
PRE1.shutdown	-	
PRE1.getALLEPs	PRE1.location	
Aktiv gesendete Kommandos		
Kommando	Ziel	erwarte Antwort
mbus.poll isLocalZone	media:policy	mbus.vote isLocalZone
mbus.poll mayRegister	media:policy	mbus.vote mayRegister
mbus.poll mayCall	media:policy	mbus.vote mayCall
PRE1.register	alle	-
PRE1.unregister	alle	-
PRE1.poll.bandwidth	media:policy	PRE1.vote.bandwidth
PRE1.beginCall	alle	-
PRE1.endCall	alle	-
PRE1.locate	alle	PRE1.location
PRE1 = conf.call-control.h323		

B.2.2 An-/Abmelden von Endpunkten

register

register - Registrierung eines Endpunkts		
An	() (alle)	
conf.call-control.h323.register (aliases csAddr rasAddr type)		
aliases	List	Liste vom Typ Alias
csAddr	List	Liste vom Typ Tsap
rasAddr	List	Liste vom Typ Tsap
type	EpType	Art des Endpunktes

Diese Nachricht dient dazu, anderen Modulen am MBus mitzuteilen, daß ein neuer Endpunkt für die Zone registriert wurde.

unregister

unregister - Abmeldung eines Endpunkts		
An	() (alle)	
conf.call-control.h323.unregister (aliases csAddr)		
aliases	List	Liste vom Typ Alias
csAddr	Tsap	Anzumeldende Call-Signaling-Adresse

Diese Nachricht dient dazu, anderen Modulen am MBus mitzuteilen, daß ein Endpunkt sich abgemeldet hat. Sie wird per Broadcast verteilt.

B.2.3 Anrufanfang und -ende signalisieren

beginCall

beginCall - Beginn eines Anrufes signalisieren		
conf.call-control.h323.beginCall (call [bwidth])		
call	Call	Anrufdaten
bwidth	Integer	Optional: Verwendete Bandbreite

Hiermit teilt das H323-Modul mit, daß ein neuer Anruf begonnen hat. Evtl. Accounting-Module sollte dies interessieren, aber auch Module zur Ressourcen-Verwaltung benötigen dies, um die Ressourcen zu belegen.

endCall

endCall - Ende eines Anrufes signalisieren		
conf.call-control.h323.endCall (call-id)		
call-id	Data	Identifikator des Anrufes

Diese Nachricht wird vom H323-Modul gesendet, wenn ein Endpunkt mittels DRQ das Ende seines Telefonats mitteilt.

B.2.4 Adreßauflösung

locate

locate - Adreßauflösung für zonenexterne Adressen		
<code>conf.call-control.h323.locate (id alias)</code>		
id	String	Transactions-ID der Anfrage
alias	Alias	Aufzulösende Aliasadresse

location

location - Resultat einer Adreßauflösung		
<code>conf.call-control.h323.location (id nr loc-list)</code>		
id	String	Transactions-ID der Anfrage, die hiermit beantwortet wird
nr	Integer	Anzahl der möglichen Adressen
loc-list	List	Liste von H323-ID/IP-Adresse-Paaren

Diese Nachricht ist die Antwort auf ein *locate*-Kommando. Eine *nr* von 0 bedeutet, daß die Adresse nicht aufgelöst werden konnte.

Das Modul wird die erste eingehende Anfrage als gültige Antwort nehmen und nachfolgende Anfragen verwerfen.

B.2.5 Sonstiges

getAllEPs

getAllEPs - Alle registrierten Endpunkte erfragen		
<code>conf.call-control.h323.getAllEPs ()</code>		
id	String	Transactions-ID

shutdown

shutdown - H323-Modul runterfahren		
<code>conf.call-control.h323.shutdown ()</code>		

B.3 Datenbank-Modul

Das Datenbankmodul ist eine Implementierung des in der AG Rechnernetze entstandenen Entwurfes für eine Mini-Datenbank (mdb), die generische MBus-Kommandos zum Zugriff auf Datenbanken definiert. Die Kommandos sind so einfach gehalten, daß die verwendete Datenbank sowohl eine SQL-Datenbank, als auch eine einfache Textdatei sein kann.

B.3.1 Steckbrief

Adresse: app:GKDBase module:engine media:h323		
Reaktion auf eingehende Kommandos		
Kommando		Antwort
tools.mdb.add		tools.mdb.ok
tools.mdb.delete		tools.mdb.ok
tools.mdb.lookup		tools.mdb.ok
tools.mdb.select		tools.mdb.ok
tools.mdb.keys		tools.mdb.ok
tools.mdb.classes		tools.mdb.ok
gatekeeper.dbase.getCredits		tools.mdb.ok
gatekeeper.dbase.getEndpoint		tools.mdb.ok
gatekeeper.dbase.getFunc		tools.mdb.ok
gatekeeper.dbase.getGroup		tools.mdb.ok
gatekeeper.dbase.getUser		tools.mdb.ok
gatekeeper.dbase.setEndpoint		tools.mdb.ok
gatekeeper.dbase.setFunc		tools.mdb.ok
gatekeeper.dbase.setGroup		tools.mdb.ok
gatekeeper.dbase.setUser		tools.mdb.ok
Aktiv gesendete Kommandos		
Kommando	Ziel	erwarte Antwort
keine		
PRE1 = gatekeeper.dbase		

Anstelle eines `tools.mdb.ok` kann auch ein `tools.mdb.error` erfolgen, sofern ein Fehler in der Datenbankschnittstelle aufgetreten ist.

B.3.2 mdb-Kommandos

Folgende mdb-Klassen (in diesem Fall SQL-Tables) werden von dem Datenbank-Modul unterstützt:

Klasse	Primärschlüssel	Inhalt
endpoints	ip	Eine Liste von IP-Adressen von Endpunkte, die der Gatekeeper registrieren darf.
freenumbers	address	Eine Liste von Adressen, die frei angerufen werden dürfen.
functions	name	Eine Liste von Adressen, die Funktionen bezeichnen, und Verweise auf H.323-Adressen oder Telefonnummern enthalten.
privdef	id	Definitionen der Privilegiengruppen.
user	h323	Liste aller bekannten Benutzer.

Für die nutzerabhängigen Daten sind folgende Bezeichner erlaubt:

h323	Die H.323-Adresse des Nutzers (eindeutig)
e164	Die Telefonnummer des Nutzers
name	Der volle Name des Nutzers
address	Adresse für Abrechnungszwecke
email	eMail-Adresse
account	Der Kontostand
cpl	Das Call-Processing-Skript
grp	Privilegiengruppenzugehörigkeit
state	Gibt an, ob der Benutzer gesperrt ist (0) oder nicht (1)

tools.mdb.ok

tools.mdb.ok - Aktion bestätigen		
tools.mdb.ok (id param-list)		
id	String	Transactions-ID der Anfrage
param-list	List	Evtl. leere Liste von Key/Value-Paaren

tools.mdb.error

tools.mdb.error - Fehler melden		
tools.mdb.error (id error-id text)		
id	String	Transactions-ID der Anfrage
error-id	Integer	Fehlertyp 0=Interner Fehler, 1=Syntax-Fehler, 2=Unbekannte Klasse, 3=Unbekanntes Feld, 4=Unbekannter Eintrag
text	String	Fehlerbeschreibung

tools.mdb.add

tools.mdb.add - Daten hinzufügen/ändern		
tools.mdb.add (id class param-list)		
id	String	Transactions-ID
class	String	Klasse/Table, in die der Eintrag geschrieben werden soll.
param-list	List	Liste von Key/Value-Paaren für einen Datensatz

Der in der **param-list** übergebene Datensatz wird in die Datenbank geschrieben. Eines der übergebenen Key-Value-Paaren muß einen Schlüssel enthalten,

der einen Datensatz eindeutig identifiziert - vergleichbar mit einem *Primary Key* bei SQL-Datenbanken.

Bereits vorhandene Datensätze werden mit den neuen Daten zusammengeführt, wobei die neuen Daten die alten überschreiben.

Enthält eines der Key/Value-Paare einen Key, der bisher nicht in der Klasse vorkam, so wird diese automatisch um dieses Feld erweitert.

tools.mdb.delete

tools.mdb.delete - Datensatz löschen		
tools.mdb.delete (id class key)		
id	String	Transactions-ID
class	String	Klasse/Table, aus der der Eintrag gelöscht werden soll.
key	String	Wert des Primärschlüssels

Löscht den Datensatz, der durch den Wert des Primärschlüssels eindeutig bezeichnet ist.

tools.mdb.lookup

tools.mdb.lookup - Datensatz suchen		
tools.mdb.lookup (id class key)		
id	String	Transactions-ID
class	String	Klasse/Table, in der der Eintrag gesucht werden soll.
key	String	Wert des Primärschlüssels

Liefert den Datensatz zu dem angegebenen Primärschlüssel. Diese Anfrage erlaubt der verwendeten Datenbank meist eine effektivere Suche.

tools.mdb.select

tools.mdb.select - Datensatz suchen		
tools.mdb.lookup (id class key)		
id	String	Transactions-ID
class	String	Klasse/Table, in der der Eintrag gesucht werden soll.
attr-list	List	Suchmaske

Liefert die Datensätze einer Klasse, für die die Bedingungen gelten, die in der Suchmaske (`attr-list`) spezifiziert wurden.

tools.mdb.keys

tools.mdb.keys - Feldnamen ermitteln		
tools.mdb.keys (id class)		
id	String	Transactions-ID
class	String	Klasse/Table, die betrachtet werden soll.

Liefert die Namen aller Felder einer Klasse.

tools.mdb.classes

tools.mdb.classes - Klassennamen ermitteln		
tools.mdb.keys (id)		
id	String	Transactions-ID

Liefert die Namen aller bekannten Klassen der Datenbank.

B.3.3 Spezielle Kommandos

Neben den generellen, durch die mdb definierten Kommandos, verwendet das Datenbank-Modul noch Kommandos, die zum vereinfachten Zugriff auf die Daten verwendet werden können.

getEndpoint

getEndpoint - Daten eines Endpunktes abfragen		
gatekeeper.dbase.getEndpoint (id ip)		
id	String	Transactions-ID der Anfrage
ip	String	IP-Adresse

setEndpoint

setEndpoint - Daten eines Endpunktes setzen		
gatekeeper.dbase.setEndpoint (id ip)		
id	String	Transactions-ID der Anfrage
ip	String	IP-Adresse

setUser

setUser - Daten zu einem Nutzer schreiben		
dbase.h323.setUser (id user)		
id	String	Transactions-ID
user	User	Userdaten

getGroup

getGroup - Gruppenelemente suchen		
dbase.h323.getGroup (id group)		
id	String	Transactions-ID
group	Group	Daten der Privilegiengruppe

setGroup

Es werden alle Gruppen ermittelt, auf die die in der übergebenen Gruppe gesetzten Daten zutreffen.

setGroup - Daten zu einer Gruppe schreiben		
dbase.h323.setGroup (id user)		
id	String	Transactions-ID
group	Group	Daten der Gruppe

getCredits

getCredits - Kontostand abfragen		
dbase.h323.getCredits (id h323)		
id	String	Transactions-ID
h323	String	H.323-Adresse des Nutzers, dessen Kontostand ermittelt werden soll.

getFunc

getFunc - Funktionsadresse suchen		
dbase.h323.getFunc (id func)		
id	String	Transactions-ID
func	Function	Daten der Funktionsadresse

setFunc - Daten zu einer Gruppe schreiben		
dbase.h323.setFunc (id func)		
id	String	Transactions-ID
func	Function	Daten der Funktionsadresse

setFunc

B.4 Zugangspolicy-Modul (Access-Modul)

B.4.1 Steckbrief

Adresse: app:AccessManager module:engine media:policy		
Reaktion auf eingehende Kommandos		
Kommando	Antwort	
mbus.poll isLocalZone	mbus.vote isLocalZone	
mbus.poll mayRegister	mbus.vote mayRegister	
mbus.poll mayCall	mbus.vote mayCall	
Aktiv gesendete Kommandos		
Kommando	Ziel	erwarte Antwort
keine		

B.4.2 Kommandos

poll isLocalZone - Abfragen, ob der Endpunkt zur Zone gehört		
An	(media:policy)	
mbus.poll (id isLocalZone 2 1 („yes\", „no\") (ip))		
id	String	Transactions-ID der Anfrage
ip	String	IP-Adresse des Endpunkts

poll isLocalZone

Mit dieser Frage kann geprüft werden, ob der Endpunkt einer IP-Adresse zur Zone des Gatekeepers gehört.

poll mayRegister - Abfragen, ob der User sich registrieren darf		
An	(media:policy)	
mbus.poll (id mayRegister 2 1 („yes\", „no\") (ip alias))		
id	String	Transactions-ID der Anfrage
ip	String	IP-Adresse des Endpunkts
alias	Alias	Aliasname

poll mayRegister

Dieses Kommando wird vom H323-Modul an das Zugangspolicy-Modul gestellt, um herauszufinden, ob sich ein Benutzer registrieren darf.

poll mayCall

poll mayCall - Prüfen, ob ein Anruf erlaubt ist		
<code>mbus.poll (id mayCall 4 1 („yes\ „notThatIP\ „notNow\ „notThatUser\) (call-id srcType srcAlias srcIP destType destAlias destIP))</code>		
id	String	Transactions-ID der Anfrage
call-id	String	Identifikator des Anrufes
srcType	Integer	Art des Aliasnamen des Anrufers
srcAlias	String	Aliasname des Anrufers
srcIP	String	IP-Adresse des Anrufers
destType	Integer	Art des Aliasnamen des Angerufenen
destAlias	String	Aliasname des Angerufenen
destIP	String	IP-Adresse des Angerufenen

Diese Anfrage wird vom H323-Modul an alle Policy-Module gesendet, um herauszufinden, ob ein Gespräch zustandekommen darf.

B.5 ResourceManager-Modul

Auch bekannt als *Bandbreiten-Policymodul*.

B.5.1 Steckbrief

Volle Adresse: <code>app:GKResMan module:engine media:policy</code>		
Reaktion auf eingehende Kommandos		
Kommando	Antwort	
<code>PRE1.poll.bandwidth</code>	<code>PRE1.vote.bandwidth</code>	
<code>PRE1.getConfig</code>	<code>PRE1.setConfig</code>	
<code>PRE1.setConfig</code>	-	
<code>PRE1.getResources</code>	<code>PRE1.resources</code>	
Aktiv gesendete Kommandos		
Kommando	Ziel	erwarte Antwort
keine		
PRE1 = <code>conf.call-control.h323</code>		

poll.bandwidth

B.5.2 Kommandos

poll.bandwidth - Prüfen, ob Bandbreite verfügbar		
An	<code>(media:policy)</code>	
<code>conf.call-control.h323.poll.bandwidth (id call-id bwidth)</code>		
id	String	Transactions-ID der Anfrage
call-id	Data	Identifikator des Anrufwunsches
bwidth	Integer	Gewünschte Bandbreite in 100 bps

Das H323-Modul sendet diese Nachricht an alle Policy-Module, um herauszufinden, ob ein Endpunkt die gewünschte Bandbreite verwenden darf. Dieses poll-Kommando unterscheidet sich vom normalen Kommando dadurch, daß es nicht eine vorgegebene Liste von möglichen Entscheidungen mitgibt, sondern eine Zahl, die in der zugehörigen Antwort nach oben oder nach unten verändert werden darf.

vote.bandwidth

vote.bandwidth - Erlaubte Bandbreite mitteilen		
An	Sender des vote.bandwidth-Kommandos	
conf.call-control.h323.vote.bandwidth (id call-id bwidth)		
id	String	Transactions-ID der Anfrage
call-id	Data	Identifikator des Anrufwunsches
bwidth	Integer	Erlaubte Bandbreite in 100 bps

getResources

getResources - Ressourcenauslastung erfragen		
conf.call-control.h323.getResources (id)		
id	String	Transactions-ID

getConfig

getConfig - Aktuelle Konfiguration ermitteln		
conf.call-control.h323.getConfig (id)		
id	String	Transactions-ID

getConfig

getConfig - Aktuelle Konfiguration		
conf.call-control.h323.getConfig (id maxAv maxSi incR)		
id	String	Transactions-ID
maxAv	Integer	Maximal verfügbare Bandbreite
maxSi	Integer	Maximale Bandbreite pro Gespräch
incR	Integer	Reservierter prozentualer Anteil für eingehende Gespräche

resources

resources - Gegenwärtige Ressourcenauslastung		
conf.call-control.h323.resources (id inc out)		
id	String	Transactions-ID
inc	Integer	Prozentuale Auslastung bei eingehenden Gesprächen
out	Integer	Prozentuale Auslastung bei rausgehenden Gesprächen

B.6 API-Modul (GUI-Modul)

B.6.1 Steckbrief

Adresse: app:gatekeeperapi module:engine media:h323		
Reaktion auf eingehende Kommandos		
Kommando	Antwort	
PRE1.register	-	
PRE1.unregister	-	
PRE1.beginCall	-	
PRE1.endCall	-	
Aktiv gesendete Kommandos		
Kommando	Ziel	erwarte Antwort
tools.mdb.add	app:GKDBase	tools.mdb.ok
tools.mdb.delete	app:GKDBase	tools.mdb.ok
tools.mdb.lookup	app:GKDBase	tools.mdb.ok
tools.mdb.select	app:GKDBase	tools.mdb.ok
PRE2.getEndpoint	app:GKDBase	tools.mdb.ok
PRE2.getFunc	app:GKDBase	tools.mdb.ok
PRE2.getGroup	app:GKDBase	tools.mdb.ok
PRE2.setFunc	app:GKDBase	tools.mdb.ok
PRE2.setGroup	app:GKDBase	tools.mdb.ok
PRE2.setUser	app:GKDBase	tools.mdb.ok
PRE1.getAllEPs	app:h323stack	PRE1.location
PRE1.getConfig	app:GKResMan	PRE1.setConfig
PRE1.getRes	app:GKResMan	PRE1.resources
PRE1.setConfig	app:GKResMan	-
PRE1.shutdown	app:h323stack	-
PRE1 = conf.call-control.h323 PRE2 = gatekeeper.dbase		

B.6.2 Kommandos

Glossar und Abkürzungsverzeichnis

Anrufsignalisierung

Mitteilen des Vorliegens eines Anrufes beim Angerufenen. Heute eher der Informationsaustausch über einen vorliegenden Anruf: Telefonklingeln, Besetzt-Zeichen, Anrufannahme bzw. Auflegen.

ACF

Kurz für *Admission Confirm*. Name für die PDU, mit der ein Gatekeeper eine ARQ-PDU positiv beantwortet.

ARJ

Kurz für *Admission Reject*. Name für die PDU, mit der ein Gatekeeper eine ARQ-PDU negativ beantwortet.

ARQ

Kurz für *Admission Request*. Name der PDU, mit der ein Endpunkt bei seinem Gatekeeper die Berechtigung für den Beginn eines Gespräches und evtl. zugleich um Adreßauflösung bittet.

ASN.1

Abkürzung für **Abstract Syntax Notation One**. Eine Typ-(Struktur-) Definitionssprache. ASN.1 definiert einige Grunddatentypen und Mechanismen, diese zu komplexeren Gebilden zusammenzufassen. ASN.1 ermöglicht den Datenaustausch zwischen Rechnern mit unabhängigen lokalen Kodierungen der Daten.

BCF

Kurz für *Bandwidth Confirm*. Name für die PDU, mit der ein Gatekeeper oder Endpunkt eine BRQ-PDU positiv beantwortet.

BRJ

Kurz für *Bandwidth Reject*. Name für die PDU, mit der ein Gatekeeper oder Endpunkt eine BRQ-PDU negativ beantwortet.

BRQ

Kurz für *Bandwidth Request*. Name der PDU, mit der ein Endpunkt seinen Gatekeeper um die Änderung der Bandbreite bittet, bzw. mit der der Gatekeeper einen Endpunkt auffordert, die verwendete Bandbreite zu ändern.

- DCF**
Kurz für *Disengage Confirm*. Name für die PDU, mit der ein Gatekeeper oder Endpunkt eine DRQ-PDU positiv beantwortet.
- DRJ**
Kurz für *Disengage Reject*. Name für die PDU, mit der ein Gatekeeper oder Endpunkt eine DRQ-PDU negativ beantwortet.
- DRQ**
Kurz für *Disengage Request*. Name der PDU, mit der ein Endpunkt seinem Gatekeeper das Ende seines Gespräches mitteilt, bzw. mit der der Gatekeeper einen Endpunkt auffordert, ein laufendes Gespräch zu beenden.
- FTP**
Abkürzung für **File Transfer Protocol**. Ein Mitglied der Internet-Protokollfamilie, das zum Transfer von Dateien zu oder von entfernten Computern dient.
- HTTP**
Abkürzung für **Hypertext Transfer Protocol**. Ein Mitglied der Internet-Protokollfamilie zum Austausch von WWW-Dokumenten.
- Gatekeeper-Discovery**
Unter Gatekeeper-Discovery versteht man den Versuch eines Endgerätes einen zuständigen Gatekeeper für User-Location und Mehrwertdienste zu finden.
Für den Fall, daß ein Endgerät bei Betriebsbeginn noch keinen Gatekeeper kennt, kann das Auffinden eines Gatekeepers durch ein „In die Welt hinausrufen“, d.h. durch Multicasting, erfolgen. Bekannte Gatekeeper können aber auch ohne vorherige Discovery-Phase angesprochen werden.
- IACK**
Kurz für *Info Request Ack*. Name für die PDU, mit der ein Gatekeeper unaufgefordert erhaltene, aber erwünschte IRR-PDUs eines Endpunktes bestätigt, sofern dieser eine Bestätigung angefordert hat.
- INAK**
Kurz für *Info Request Nak*. Name für die PDU, mit der ein Gatekeeper unaufgefordert erhaltene und zudem unerwünschte IRR-PDUs eines Endpunktes bestätigt, sofern dieser eine Bestätigung angefordert hat.
- IP, IPv4**
Kurz für *Internet-Protocol*. Dieses Protokoll ermöglicht die Koppelung von Rechnernetzen und dabei insbesondere eine netzübergreifende Adressierung. Mit dieser „klassischen“ Variante des IP-Protokolls lassen sich 2^{32} , d.h. ca 4 Milliarden Endpunkte adressieren, wovon ziemlich viele Adressen für spezielle Anwendungen reserviert sind.
Es ist absehbar, daß dieses Protokoll aufgrund der geringen Zahl zu adressierender Endpunkte und gestiegenen Anforderungen moderner Anwendungen durch eine neuere Variante (→ IPv6) ersetzt wird.

-
- IPv6**
Die Nachfolge-Variante des Internet-Protokolls (→ IP). Es unterstützt Reservierungen von Bandbreiten, Echtzeitanforderungen und einen größeren Adreßraum.
- IP-Adresse**
Eine Adresse des Internet-Protokolls (→ **IP**).
- IRR**
Kurz für *Info Request Response*. Name für die PDU, mit der ein Endpunkt einem Gatekeeper seinen gegenwärtigen Zustand mitteilt. Diese PDU wird entweder als Antwort auf eine IRQ-PDU versandt oder sie erfolgt unaufgefordert in bestimmten Intervallen.
- IRQ**
Kurz für *Info Request*. Name für die PDU, mit der ein Gatekeeper Informationen über den Zustand eines Endpunkt anfordert.
- LCF**
Kurz für *Location Confirm*. Name für die PDU, mit der ein Gatekeeper eine durch eine LRQ-PDU angestoßene Adreßauflösung positiv bestätigt.
- LRJ**
Kurz für *Location Reject*. Name für die PDU, mit der ein Gatekeeper eine durch eine LRQ-PDU angestoßene Adreßauflösung negativ bestätigt.
- LRQ**
Kurz für *Location Request*. Name für die PDU, mit der ein Endpunkt den Gatekeeper um die Auflösung eines Namens in eine Transportadresse (meist eine IP-Adresse) bittet.
- MC, Multipoint Controller**
Jener Teil einer → **MCU**, der für die Konferenzsteuerung sorgt, d.h. er ermöglicht das Initiieren von und Teilnehmen an Konferenzen und sorgt für die Aushandlung der verwendeten Medien und Medienformate. Das Mischen und Verteilen der eingehenden Medienströme ist Aufgabe des Multipoint Processors (→ **MP**).
- MCU, Multipoint Control Unit**
Eine MCU ist ein Endpunkt, der es drei oder mehr Endpunkten oder Gateways erlaubt, an einer Konferenz teilzunehmen. Die MCU kann auch für 2-Punkt-Gespräche verwendet werden, die später zu einer Konferenz ausgeweitet werden sollen.
Eine MCU besteht mindestens aus einem → **MC** und meist auch einem → **MP**.
- MP, Multipoint Processor**
Jener Teil einer → **MCU**, der das Zentrum einer laufenden Konferenz bildet. Er nimmt alle Daten, die von den Endpunkten gesendet werden, mischt sie und verteilt sie dann an die übrigen Endpunkte. Ohne einen MP müßte jeder Konferenzteilnehmer seine Daten an alle anderen Teilnehmer senden, was erheblich aufwendiger wäre.
-

Multicasting

Das Versenden von Daten an mehrere Empfänger gleichzeitig, ohne daß dem Sender die Empfänger bekannt sein müssen. Die Empfänger müssen zuvor explizit ihren Empfangswillen bekundet haben.

paket-orientiert

In paket-orientierten Protokollen werden die Daten in Paketen, anstatt in kontinuierlichen Datenströmen versendet. Es besteht kein gesicherter gemeinsamer Kommunikationszustand zwischen Sender und Empfänger, d.h. der Sender kann nicht sicher sein, daß der Empfänger die Daten in der richtigen Reihenfolge oder auch nur überhaupt erhalten hat.

PSTN

Abkürzung für *Public Switched Telephone Network*. Das leitungsvermittelte analoge Telefonnetz.

RAC

Kurz für *Resources Available Confirm*. Name für die PDU, mit der ein Gatekeeper den Erhalt einer RAI-PDU bestätigt.

RAI

Kurz für *Resources Available Indicate*. Name für die PDU, mit der ein Gateway seinem Gatekeeper die gegenwärtige Ressourcenauslastung mitteilt.

RCF

Kurz für *Registration Confirm*. Name für die PDU, mit der ein Gatekeeper einen durch eine RRQ-PDU erfolgten Registrierungswunsch positiv bestätigt.

RIP

Kurz für *Request In Progress*. Name für die PDU, mit der ein Endpunkt anzeigt, daß er eine Anfrage bearbeitet. In der Regel wird diese PDU gesendet, um dem Fragesteller anzuzeigen, daß seine Nachricht angekommen ist, aber die Bearbeitung länger dauern wird.

RRJ

Kurz für *Registration Reject*. Name für die PDU, mit der ein Gatekeeper einen durch eine RRQ-PDU erfolgten Registrierungswunsch negativ bestätigt.

RRQ

Kurz für *Registration Request*. Name für die PDU, mit der ein Endpunkt seinen Registrierungswunsch mitteilt.

SIP

Ein im IETF-Umfeld entwickeltes Call-Signaling-Protocol zum Aufsetzen von Konferenzen.

strom-orientiert

In strom-orientierten Protokollen existiert zwischen Sender und Empfänger eine ständige Verbindung. Strom-orientierte Protokolle oberhalb von → IP, wie z.B. TCP, sorgen daher selbstständig dafür, daß die Reihenfolge der Pakete erhalten bleibt und nötigenfalls Pakete erneut gesendet werden.

UCF

Kurz für *Unregistration Confirm*. Name für die PDU, mit der ein Gatekeeper oder Endpunkt eine URQ-PDU positiv beantwortet.

Unicast

Das Versenden von Datenpaketen an einen einzelnen Empfänger via UDP.

URJ

Kurz für *Unregistration Reject*. Name für die PDU, mit der ein Gatekeeper oder Endpunkt eine URQ-PDU negativ beantwortet.

URQ

Kurz für *Unregistration Request*. Name der PDU, mit der ein Endpunkt sich bei seinem Gatekeeper abmeldet, bzw. mit der der Gatekeeper einem Endpunkt mitteilt, daß dieser nicht mehr bei ihm registriert ist.

Literaturverzeichnis

- [1] BORMANN, U., BORMANN, C., AND OTT, J. WIPTTEL - Aufbau einer Infrastruktur für IP-Telefonie-Dienste im Wissenschaftsnetz. Tech. rep., Universität Bremen, Technologiezentrum Informatik, Bereich Digitale Medien und Netze, July 1998.
- [2] CAMARILLO, G. Ip telephony gateways. Master's thesis, Royal Institute of Technology, Stockholm, Feb. 1998.
- [3] CORPORATION, M. *The Professional Companion to NetMeeting 3.0*. Microsoft, June 1999.
- [4] DAVIS, R. A Framework for a Peer Gatekeeper Routing Protocol. Tech. rep., Lucent Technologies, Nov. 1998.
- [5] ELEMEDIA. *Vocaltec Communications*. <http://www.elemedia.com>.
- [6] EQUIVALENCE. *Open H323 Project*. <http://www3.openh323.org>.
- [7] FARLEY, J. *JAVA Distributed Computing*. Addison-Wesley, Sebastopol, Jan. 1998.
- [8] FREUNDLICH, G., KORPI, M., OTT, J., AND SKRAN, D. *H.323: An International Standard for Voice over IP*. International Telecommunication Union, June 1999.
- [9] FUNKE, C. Interfacing between itu and mbone protocols for internet multimedia conference control - design and implementation of a gateway. Master's thesis, Universität Bremen, June 1997.
- [10] HAMPTON, D., ORAN, D., SALAMA, H., AND SHAH, D. The IP Telephony Border Gateway Protocol. Tech. rep., Cisco Systems, June 1999.
- [11] HAMPTON, D., ORAN, D., SALAMA, H., AND SHAH, D. The IP Telephony Border Gateway Protocol Architecture. Tech. rep., Cisco Systems, Feb. 1999.
- [12] INTERNATIONAL TELECOMMUNICATION UNION. *Recommendation Q.931: ISDN User-Network Interface Layer 3 Specification For Basic Call Control*, 1993.
- [13] INTERNATIONAL TELECOMMUNICATION UNION. *Recommendation X.680: Abstract Syntax Notation One (ASN.1): Specification Of Basic Notation*, 1994.

-
- [14] INTERNATIONAL TELECOMMUNICATION UNION. *Recommendation H.245: Control Protocol For Multimedia Communication*, 1996.
- [15] INTERNATIONAL TELECOMMUNICATION UNION. *Recommendation H.225.0 v2: Call Signaling Protocols and Media Stream Packetization for Packet Bases Mutlimedia Communications Systems*, Mar. 1997.
- [16] INTERNATIONAL TELECOMMUNICATION UNION. *Recommendation H.323: Visual Telephone System And Equipment For Local Area Networks Which Provide A Non-Guaranteed Quality Of Service*, 1998.
- [17] INTERNATIONAL TELECOMMUNICATION UNION. *H.225.0 Annex G Draft for Decision*, May 1999.
- [18] INTERNET ENGINEERING TASK FORCE. *RFC 2543 - SIP: Session Invitation Protocol*, Mar. 1999.
- [19] KIMCHI, G. Comments on 13-TD-71. Tech. rep., International Telecommunication Union, May 1999.
- [20] LAWRENCE BERKELEY NATIONAL LABORATORY. *vat Homepage*. <http://www-nrg.ee.lbl.gov/vat/>.
- [21] LAWRENCE BERKELEY NATIONAL LABORATORY. *vic Homepage*. <http://www-nrg.ee.lbl.gov/vic/>.
- [22] LENNOX, J., AND SCHULZRINNE, H. CPL: A Language for User Control of Internet Telephony Services. Tech. rep., Columbia University, Feb. 1999.
- [23] NETSPEAK. *Netspeak Products*. <http://www.netspeak.com>.
- [24] OTT, J., Ed. *Terms of Reference for H.323 Robustness* (Oct. 1999), International Telecommunication Union.
- [25] OTT, J., PERKINS, C., AND KUTSCHER, D. A Message Bus for Conferencing Systems. Tech. rep., Universität Bremen/ University College London, Aug. 1998.
- [26] OTT, J., PERKINS, C., AND KUTSCHER, D. The Message Bus: Messages and Procedures. Tech. rep., Universität Bremen/ University College London, Aug. 1998.
- [27] OTT, J., AND PRELLE, S. Robustness Mechanisms for H.323. Universität Bremen, Oct. 1999.
- [28] QUESCOM. *Radvision H323*. <http://www.quescom.com>.
- [29] RADVISION. *Radvision H323*. <http://www.radvision.com>.
- [30] ROSENBERG, J., SALAMA, H., AND SQUIRE, M. A Gateway Location Protocol. Tech. rep., Bell Labs / Cisco Systems / Nortel Networks, Oct. 1999.
- [31] ROSENBERG, J., SALAMA, H., AND SQUIRE, M. Attributes for a Gateway Location Protocol. Tech. rep., Bell Labs / Cisco Systems / Nortel Networks, Dec. 1999.

-
- [32] ROSENBERG, J., AND SCHULZRINNE, H. A Framework for a Gateway Location Protocol. Tech. rep., Bell Labs / Columbia University, Feb. 1999.
- [33] SQUIRE, M. A Gateway Location Protocol. Tech. rep., Nortel Networks, Feb. 1999.
- [34] SUN MICROSYSTEMS. *Java Telephony Specification - JTAPI version 1.3*, June 1999.
- [35] TANDER, J. Ip telephony for dummies. Master's thesis, Lund Institute of Technology, Lund University, Feb. 1998.
- [36] TANENBAUM, A. S. *Modern Operating Systems*. Prentice Hall, Englewood Cliffs, New Jersey 07632, 1992.
- [37] TOGA, J. Draft H.323 Implementers Guide - Rev 2. Tech. rep., ITU Telecommunication Standardization Sector - TD-14r Study Group, Apr. 1999.
- [38] UNIVERSITY COLLEGE LONDON. *The RAT (Robust-Audio Tool) Home Page*. <http://www-mice.cs.ucl.uk/mice/rat>.
- [39] VIDEOSERVER. *Videoserver*. <http://www.videoserver.com>.
- [40] VOCALTEC COMMUNICATIONS. *Vocaltec Communications*. <http://www.vocaltec.com/products/products.htm>.

Index

- 1869,- DM, 85
- Abrechnung, 55
- AccessPolicy-Modul, 50
- Accounting, 55, 91
- Active Replication, 92
- Adreßauflösung, 40
 - Ablauf, 45
- Adreßumsetzung, 40
- Adressen
 - für Anschlüsse, 42
 - für Funktionen, 44
 - für Personen, 42
- AG Rechnernetze, 14
- Anrufbearbeitung, 55
- Anrufmodelle
 - unter H323, 20
- API-Modul, 54
- Architektur, 49
- ASN.1, 76

- BandwidthPolicy-Modul, 51
- Bell, Alexander Graham, 1
- Benutzungsschnittstelle, 77
- Billing, 55, 91

- Call Processing Language, 31, 55
- Call-Routing, 46, 56
- CPL, 31, 55

- Datenbank, 86
- Datenbank-Modul, 51

- Endpunkte, 8

- Fernsteuerung, 91
- Funktionsadressen, 44

- Gatekeeper, 9
 - Verwendung des, 85
- Gateway, 10
- Gateway Location, 56

- Gateway Location Protocol, 30
- GLP, 30
- Gremien
 - Internationale, 13
- GUI-Modul, 54

- H.225, 24
- H.245, 26
- H.323, 18
 - MBus-Modul, 76
- H.323-Modul, 50
- H.931, 26

- IETF, 14
- IETF-Drafts, 27
- Installation, 85
- Internet, 4
- IP, 5
- IPv6, 5
- ISDN, 3
- ITU-T, 14

- Konferenzen
 - unter H323, 21
- Konfiguration, 95

- Lastenausgleich, 91
- Least Cost Routing, 8
- Load Balancing, 91
- Location Server, 9

- Mbone, 7
- MBus, 33
 - Module, 34
 - Policy-Modul, 35
 - Sicherheit, 35
 - virtueller, 83
 - Voting, 35
- MBus-Kommando
 - Alias, 98
 - beginCall, 101
 - Call, 99

- endCall, 101
- Endpoint, 99
- EpType, 99
- Function, 99
- getAllEPs, 102
- getConfig, 109
- getCredits, 106
- getEndpoint, 106
- getFunc, 106
- getGroup, 106
- getResources, 109
- Group, 99
- kommandoname, 36
- locate, 102
- location, 102
- poll isLocalZone, 107
- poll mayCall, 108
- poll mayRegister, 107
- poll.bandwidth, 108
- register, 101
- resources, 109
- setEndpoint, 106
- setFunc, 107
- setGroup, 106
- setUser, 106
- shutdown, 102
- tools.mdb.add, 104
- tools.mdb.classes, 105
- tools.mdb.delete, 105
- tools.mdb.error, 104
- tools.mdb.keys, 105
- tools.mdb.lookup, 105
- tools.mdb.ok, 104
- tools.mdb.select, 105
- Tsap, 98
- unregister, 101
- Vendor, 99
- vote.bandwidth, 109
- MBus-Modul
 - Datenbank, 51
 - GUI, 54
 - H.323, 50
 - Ressourcenverwaltungs, 51
 - Zugangs, 50
- MCU, 10
- Medienserver, 10
- Mehrwertdienste, 3, 10, 22
 - im Gatekeeper, 56
- Message Bus, 33
- Module, 49
- Multicast Backbone, 7
- Multicasting, 6
- Multipoint Control Unit, 10
- Netzkopplung, 5
- Nikolaus, Boris, 76
- Optimierungen, 83
- paket-orientiert, 4
- Performance, 89
- PGRP, 30
- Policy-Module, 50
- Protokollierung, 55
- Protokollphasen
 - von H323, 22
- Remote Access, 91
- Ressourcenverwaltung, 47
- SCCP, 35
- SDP, 35
- SIP, 27, 35
- statische Daten, 51
- Systemanforderungen, 85
- TBGP, 28
- TCP, 6
- UDP, 6
- UniTel, 15
- User Location
 - externe, 56
- Verfügbarkeit, 91
- Vermittlung
 - automatisch, 2
 - elektronisch, 2
 - manuell, 1
- Vermittlungstechnik, 1
- Voting, 35
- WIPTel, 15
- Zone, 9
- Zugangskontrolle, 39